

منابع متن باز

OpenFlow و SDN از جمله موضوعات پر طرفدار در تحقیقات حوزه شبکه بوده و در هر دو زمینه صنعت و دانشگاه حرفه‌ای برای گفتن دارند. پروژه‌های فعال متعددی در قالب پروژه‌های متن باز پیرامون OpenFlow و SDN وجود دارند که از سوئیچهای نرم افزار محور OpenFlow گرفته تا کنترلرهای OpenFlow، ابزارهای ارکستراسیون^۱، ابزارهای مجازی سازی شبکه، ابزارهای شبیه‌سازی، ابزارهای تست و غیره را شامل می‌شوند. موضوع اصلی این فصل، ارائه خلاصه‌ای کوتاه و مفید از پروژه‌های متن باز پیرامون OpenFlow و SDN است. ما پروژه‌های متن باز زیر را مورد بررسی قرار خواهیم داد:

- سوئیچ‌های XORPlus، LINC، Indigo، Pantou، Open vSwitch
- کنترلرهای Ryu و FlowER، Trema، Maestro، Floodlight، Beacon
- عناصر متفرقه‌ای همچون OSCARS، OFlops / Cbench، RouteFlow، Avior، FlowVisor
- FortNOX و Twister

در این فصل به پروژه‌های مهمی اشاره خواهد شد که مهندسان شبکه می‌توانند در محیط عملیاتی خود، آنها را بکار ببرند.

^۱ ارکستراسیون (Orchestration) به فرایند اتوماتیک‌سازی مدیریت و تنظیم سیستمهای کامپیوتری پیچیده در محیط‌های آبری، گفته می‌شود. این فرایند اغلب در معماری‌های مجازی سازی، معماری‌های مبتنی بر سرویس و مراکز داده آبری مطرح می‌گردد. ابزارهایی که در این راستا این توانایی را ایجاد می‌کنند، ابزارهای ارکستراسیون نام دارند. (متترجم)

سوئیچ‌ها

در این بخش، به پروژه‌های متن بازی می‌پردازیم که تمرکز آنها بر روی عملیات سوئیچینگ نرم افزاری براساس پروتکل OpenFlow است.

Open vSwitch

هایپروایزرها (مانند VMware Player و VirtualBox و Xen) باید بتوانند انتقال ترافیک بین ماشینهای مجازی و دنیای بیرون را میسر سازند. شاید در نگاه اول، استفاده از سوئیچ Bridge گزینه مناسبی باشد. این سوئیچ در سیستم عامل لینوکس قرار داشته و یک سوئیچ لایه ۲ سریع و قابل اطمینان بوده که بصورت از پیش گنجانده شده است. اما سوئیچ Open vSwitch هدفی بزرگتر را دنبال می‌کند و آن، گسترش مجازی‌سازی چند سروری است. هدفی که ابزار Bridge در لینوکس نمی‌تواند به آن دست یابد و گزینه مناسبی برای ایجاد ارتباط بین ماشینهای مجازی در شبکه‌های بزرگ به شمار نمی‌رود.

محیط‌های مجازی‌سازی با طیف گسترده‌ای از سرورهای گوناگون، اغلب با ویژگیهایی از قبیل وجود سرورهای بسیار پویا، محیط انتزاعی، (برخی اوقات) یکپارچه‌سازی بین عناصر مختلف، توانایی انتقال داده به سیستم‌های مختلف و وجود سخت افزار سوئیچینگ با اهداف ویژه مشخص می‌شوند. Open vSwitch از مشخصه‌های کلیدی همچون قابلیت متحرک بودن وضعیت شبکه، پاسخ به تغییرات شبکه، حفظ و نگهداری تگهای منطقی، و یکپارچه‌سازی سخت افزاری، بهره می‌برد که برای برآورده کردن نیازهای محیط‌های مجازی‌سازی با طیف گسترده‌ای از سرورها، کارآمد است.

تمام حالت‌هایی که یک ماشین مجازی برای مهاجرت بین میزبانهای فیزیکی مختلف، در لایه شبکه به آن نیاز دارد به سادگی شناسایی شده و بین **هایپروایزرها** می‌تواند جا به جا شود. این حالت‌ها ممکن است شامل مواردی همچون سوئیچینگ بسته در لایه ۲ به طور مثال یک ورودی در یک جدول ارسال (Forwarding Table)، حالت ارسال بسته در لایه ۲، لیستهای کنترل دسترسی^۷، سیاستهای QoS یا پیکربندی مربوط به مانیتورینگ (برای نمونه پروتکل NetFlow، sFlow) و مواردی از این قبیل باشند. محیط‌های مجازی اغلب با حجم زیادی از تغییرات (به عنوان مثال ایجاد، جا به جایی یا حذف ماشینهای مجازی و تغییرات مربوط به لایه شبکه، که به دنبال آنها مطرح می‌شود)، مواجه هستند. سوئیچ Open vSwitch از مشخصه‌هایی پشتیبانی می‌کند که به یک سیستم کنترل شبکه، اجازه می‌دهند هنگام تغییر محیط، بدان پاسخ داده و شبکه را با آن تغییر منطبق سازد. علاوه بر ویژگیهای همچون Accounting ساده و نمایش آماری ترافیک داده که با استفاده از پروتکلهای NetFlow و sFlow محقق می‌شود، سوئیچ Open vSwitch پایگاه داده‌ای به نام OVSDB را در خود دارد که این پایگاه داده،

Trigger های راه دور^۷ را پشتیبانی می کند. بنابراین بخشی از نرم افزار ارکستراسیون می تواند جنبه های گوناگونی از شبکه را مانیتور کرده و در صورت تغییر این جنبه ها، به آنها پاسخ دهد. از این ویژگیها می توان برای پاسخ و ردیابی مهاجرت ماشینهای مجازی استفاده کرد. سوئیچ Open vSwitch برای کنترل ترافیک از پروتکل OpenFlow به عنوان روشی برای انتقال دسترسیهای راه دور استفاده می کند. سوئیچهای مجازی توزیع شده (برای مثال VMware vDS و Cisco Nexus 1000V) غالباً یک محتوای منطقی از وضعیت شبکه را از طریق افزودن یا مدیریت تگهای مربوط به بسته های شبکه، نگهداری می کنند. این روش بطور منحصر بفردی می تواند برای شناسایی یک ماشین مجازی یا برای نگهداری بخشی از محتواهایی که تنها در دامنه منطقی از یک شبکه مناسب است، بکار رود. قسمت اعظم مشکلات مربوط به ساختن یک سوئیچ مجازی توزیع شده، مدیریت صحیح و کارآمد این تگها است.

Open vSwitch دارای مکانیزم های گوناگونی برای مشخص کردن و حفظ قوانین تگ نگاری (Tagging) روی بسته های شبکه است که تمامی این مکانیزم ها از طریق یک فرایند راه دور جهت هماهنگ سازی، در دسترس هستند.

مسیر ارسال (DataPath) در Open vSwitch (که یک Forwarding Path درون - کرنلی است) برای واگذاری (Offload) پردازش های یک بسته به چیپست سخت افزاری، طراحی شده است، خواه این مسیر ارسال، در شاسی سوئیچ سخت افزاری (آن هم از نوع کلاسیک) واقع شده باشد یا در کارت شبکه یک سرور. این امر باعث می شود Open vSwitch بتواند هم یک پیاده سازی کاملاً نرم افزاری را پشتیبانی کرده و هم یک سوئیچ سخت افزاری را کنترل کند. مزیت پکارچه سازی سخت افزاری، تنها افزایش کارائی سوئیچ درون محیط های مجازی شده نیست. اگر سوئیچ های فیزیکی نیز بتوانند قابلیت های سوئیچ Open vSwitch را در معماری خود پیاده کنند، هر دو محیط فیزیکی (Bare-metal) و مجازی شده می توانند با استفاده از مکانیزم هایی که Open vSwitch برای کنترل شبکه اتوماتیک استفاده می کند، مدیریت شوند.

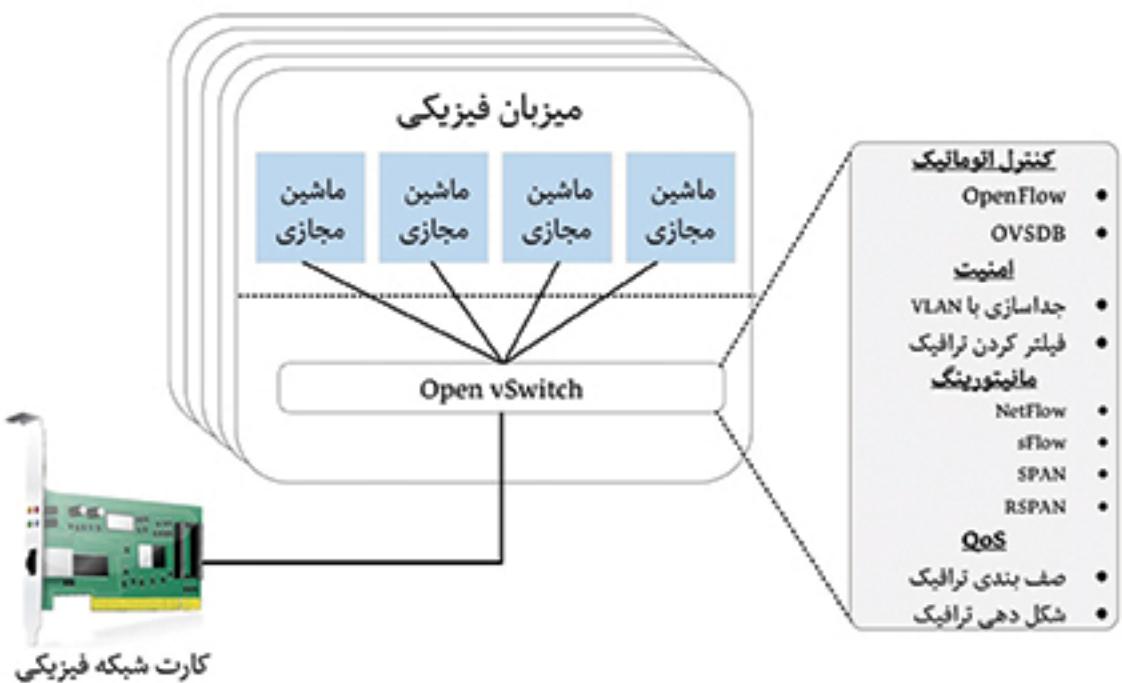
یک سوئیچ مجازی چند لایه تحت لیسانس Apache است. این برنامه برای این منظور طراحی شده که شبکه ای بسیار بزرگ را بصورت اتوماتیک از طریق توسعه در لایه کد نویسی ایجاد و مدیریت کند. در عین حال این سوئیچ از اینترفیسها و پروتکلهای مدیریتی استاندارد (مانند NetFlow، sFlow، RSPAN، SPAN، LACP، 802.1ag، CLI) پشتیبانی می کند. هدف دیگر طراحی این برنامه، عبارت است از پشتیبانی از قابلیت توزیع شدن بر روی چندین سرور فیزیکی که آنرا مشابه سوئیچ توزیع شده مجازی vDS متعلق به شرکت Cisco یا محصول VMware 1000V متعلق به شرکت می کند. (به شکل زیر توجه کنید):

⁷: زمانیکه رویدادی در پایگاه داده رخ دهد، بدنبال آن رویداد دیگری بصورت اتوماتیک اجرا شود. به این فرایند Trigger راه دور گفته می شود. (مترجم)

شکل ۸-۱

سولیج

Open vSwitch



کارت شبکه فیزیکی

Open vSwitch می‌تواند هم به عنوان یک سوئیچ نرم افزاری درون هایپروایزر فعالیت کند و هم به عنوان Control Stack برای سوئیچهای سخت افزاری استفاده شود. این سوئیچ برای پلتفرم‌های مجازی‌ساز و چیپستهای سوتیچینگ، سازگار^۱ (Port) شده است. این سوئیچ بطور پیش فرض در پلتفرم‌های مجازی‌سازی Xen Cloud Platform و XenServer 6.0 مورد استفاده بوده، همچنین از Xen, KVM, Proxmox VE, VirtualBox و OpenNebula پشتیبانی می‌کند. این سوئیچ در تعدادی از سیستمهای مدیریت مجازی مانند openQRM، OpenStack و Virt قرار داده شده است. سوئیچ Open vSwitch با نسخه‌های مختلف سیستم عامل لینوکس توزیع می‌شود و بسته‌های آن برای سیستم عامل‌های Fedora، Ubuntu و Debian موجود است. نسخه منتشر شده (و در حال توسعه) Open vSwitch همچنین از سیستم عامل دانشگاه برکلی یعنی FreeBSD نیز پشتیبانی می‌کند. بخش عمده‌ای از کد این سوئیچ به زبان C مستقل از پلتفرم^۲ نوشته شده است و بسادگی با دیگر محیط‌ها

^۱ همانطور که در این کتاب مطالعه کردید، عبارت پورت، در قسمتهای مختلف معانی متفاوتی را در بر می‌گیرد. یکی از این معنایی‌ها که در این بخش از کتاب استفاده شده، عبارت Port به معنی سازگاری است. در مهندسی نرم‌افزار، Porting به فرایند انتساب نرم‌افزار در یک محیط جدید گفته می‌شود. بر این اساس، یک نرم‌افزار بدون آنکه بطور کامل برای یک پلتفرم جدید دوباره طراحی شود، تنها با برخی تغییرات، با پلتفرم جدید سازگار می‌شود. بنابراین می‌توان به نرم‌افزارهایی اشاره کرد که در ابتدا برای محیط سیستم عامل ویندوز آماده، اما پس از مدتی برای محیط لینوکس نیز سازگار شده‌اند. (متوجه)

^۲ عبارت مستقل از پلتفرم با Platform-independent یا زبان‌هایی گفته می‌شود که می‌توان کدهای آنها را بدون تغییر و با تغییرات اندک، بر روی چندین پلتفرم اجرا کرد. از معروف‌ترین این زبانها می‌توان، به زبان جاوا و C اشاره کرد. (متوجه)

سازگار شده است. همانند لینوکس نسخه ۳.۳، سوچیج Open vSwitch نیز به عنوان بخشی از کرنل و مجموعه ابزارهای فضای کاربر، در سیستم عامل گنجانده شده و در بیشتر نسخه‌های منتشر شده لینوکس موجود است.

برای دریافت اطلاعات بیشتر در خصوص Open vSwitch و دانلود آن، می‌توانید به آدرس زیر مراجعه کنید:

<http://www.openvswitch.org>



Pantou

Pantou ابزاری است که می‌تواند یک مسیریاب بی‌سیم و یا یک Access Point را به یک سوچیج با قابلیت OpenFlow تبدیل کند. پروتکل OpenFlow به عنوان یک اپلیکیشن بر روی سیستم عامل OpenWrt اجرا شده است. OpenWrt یک سیستم عامل است که در ابتدا بر روی دستگاه‌های نهفته (Embedded Devices)^۹ این سیستم عامل عبارتند از Linux Kernel، BusyBox و uClibc. همه کامپوننتهای اصلی بهینه‌سازی شده‌اند تا با فضای محدود ذخیره‌سازی و حافظه موجود در مسیریابهای خانگی سازگار باشند. Pantou مبتنی بر نسخه منتشر شده BackFire OpenWrt است. نسخه‌ای از پروتکل OpenFlow که سوچیج Pantou از آن استفاده کرده است، همان نسخه مرجع دانشگاه استنفورد^{۱۰} است. برای تبدیل مسیریاب و یا Access Point خود به یک سوچیج OpenFlow، باید یک فایل Image مناسب Broadcom چیپست دستگاه خود که پروتکل OpenFlow را پشتیبانی کند داشته باشید (در حال حاضر Atheros و Atheros این فایل را ارائه می‌دهند)، سپس آن را بر روی دستگاه خود بارگذاری کنید و مطمئن شوید همه قسمتهای سخت افزاری و نرم افزاری دستگاه مربوطه با هم کار می‌کنند. همچنین می‌توانید با استفاده از کد مرجع، بجای استفاده از نمونه‌های آماده، فایل Image خود را ایجاد کنید. اکیداً توصیه می‌شود قبل از افزودن هر کدام از قابلیتهای مربوط به OpenFlow، یک نسخه وانیلی^{۱۱} OpenWrt ساخته و بارگذاری کنید. نسخه منتشره کنونی Pantou بر اساس نسخه BackFire OpenWrt است.

^۹ دستگاه نهفته به تجهیزاتی با اهداف ویژه گفته می‌شود. این نوع تجهیزات معمولاً در داخل خود از یک سیستم عامل به همراه یک اپلیکیشن خاص منظوره (Special Purpose) بهره می‌برند. (متترجم)

^{۱۰} منظور، نسخه ۱.۰ پروتکل OpenFlow است که با نام Userspace نیز شناخته می‌شود. (متترجم)

^{۱۱} به کامپیوترها یا نرم افزارهایی که بدون هیچ تغییر نسبت به نسخه اصلی خود، مورد استفاده قرار می‌گیرند، نسخه وانیلی می‌گویند. در این بخش منظور نویسنده این است که قبل از اضافه کردن قابلیت OpenFlow، یک فایل Image بدون هیچگونه تغییری از OpenWrt را بر روی دستگاه خود بارگذاری کنید. (متترجم)

به علاوه، پیاده‌سازی OpenFlow نسخه ۱.۳ استفاده شده در OpenWRT را می‌توانید در آدرس زیر بباید:

<http://github.com/CPqD/openflow-openwrt>



Indigo

Indigo یک پیاده‌سازی متن باز از پروتکل OpenFlow است که بر روی سوئیچهای فیزیکی اجرا می‌شود و از مشخصه‌های سخت افزاری ASIC (که در سوئیچهای اترنت متداول موجود است) استفاده می‌کند تا پروتکل OpenFlow را در فرخهای خط، اجرا کند. این پیاده‌سازی مبتنی بر پیاده‌سازی مرجع OpenFlow از دانشگاه استنفورد است و در حال حاضر تمام مشخصه‌های مورد نیاز را که در نسخه ۱.۰ پروتکل OpenFlow موجود است، پیاده‌سازی می‌کند. پیاده‌سازی نسل اول سوئیچ Indigo دیگر پشتیبانی نمی‌شود. Indigo2 پایه و اساس تکنولوژی Switch Light است که به شرکت Big Switch Network و سوئیچ مجازی Indigo تعلق دارد. Indigo2 دو کامپوننت دارد: Agent مربوط به کتابخانه‌های مرکزی را نمایش می‌دهد و شامل یک لایه انتزاع سخت افزار^۱ (HAL) و لایه انتزاع پیکربندی است. لایه انتزاع سخت افزار باعث ساده شدن یکپارچگی با اینترفیس‌های ارسال کننده و مدیریت پورت مربوط به سوئیچهای فیزیکی یا مجازی می‌گردد. همچنین لایه انتزاع پیکربندی برای پشتیبانی OpenFlow در حال اجرا بر روی یک سوئیچ فیزیکی از نوع هایبریدی مورد استفاده است. LoxiGen یک کامپایلر است که کتابخانه‌های Un-Marshalling و Marshalling مربوط به OpenFlow را در زبانهای گوناگون ایجاد می‌کند. این کامپایلر در حال حاضر از زبان C (که Loci نامیده می‌شود) پشتیبانی می‌کند. ولی برنامه نویسان در حال توسعه آن هستند تا از زبانهای برنامه نویسی / اسکریپت نویسی Java و Python نیز پشتیبانی کند.^۲ IVS یک سوئیچ مجازی شبک با کارائی بالاست که از ابتدا با هدف پشتیبانی از پروتکل OpenFlow طراحی و ساخته شد. این برنامه برای فعل کردن اپلیکیشن‌های مجازی ساز شبکه در مقیاس بزرگ طراحی شده است و مشابه vNetwork متعلق به Cisco، Nexus، VMware متعلق به Open vSwitch باشد و با استفاده از کنترلرهایی با قابلیت OpenFlow، از توزیع در بین سرورهای فیزیکی گوناگون پشتیبانی می‌کند.

اطلاعات بیشتر درباره سوئیچ Indigo را می‌توانید در آدرس زیر مشاهده کنید:

<http://www.projectfloodlight.org/indigo/>

LINC

LINC یک پروژه متن باز است که با تلاش‌های شرکت FlowForwarding هدایت می‌شود. این محصول تحت لیسانس Apache 2.0 پیاده‌سازی شده و مبتنی بر پروتکل OpenFlow نسخه 1.3.1، 1.2 و 1.1 است. FlowForwarding یک مجموعی است که ویژگی‌هایی همچون رایگان بودن، متan باز بودن و تجارت پسند بودن را تحت لیسانس Apache 2.0 و مشخصه‌های OpenFlow و ONF یدک می‌کشد. LINC یک سویچ مبتنی بر زبان Erlang بوده و برای سیستم عامل لینوکس طراحی شده است.

 یک نسخه آلفا از کد اصلی در آدرس زیر در دسترس است:
<https://github.com/FlowForwarding>

XORPlus

با بهبود سریع ASIC‌هایی که وظیفه سوئیچنگ را بر عهده دارند، چیپست‌های این سوئیچها (به عنوان مثال Broadcom) از نظر کارائی و تراکم از چیپ‌های طراحی شده اختصاصی سوئیچهای سنتی (همچون Brocade و Cisco) پیش افتاده‌اند. XORPlus خلاً نبود یک نرم افزار سوئیچینگ متan باز برای ASIC‌های با کارائی بالا را پر می‌کند. Pica8 XORPlus نرم افزار متan باز منحصر بفردي است که در سطح سوئیچ‌های یک مرکز داده اجرا می‌شود تا نه تنها پیاده‌سازی پروتکل را با کیفیتی بالا فراهم کند، بلکه باعث افزایش کارائی سوئیچینگ / مسیریابی نیز گردد. XORPlus یک نرم افزار سوئیچینگ است که توسط یک سازمان غیرانتفاعی پشتیبانی می‌شود. این نرم افزار، از مطرح‌ترین پروتکلهای لایه ۲ / لایه ۳ مورد نیاز کاربران پشتیبانی می‌کند. تمرکز XORPlus در حل معضلات و مشکلات شبکه‌ای مرکز داده در خصوص کارائی، مقیاس‌پذیری و پایداری است. از جمله ویژگی‌های لایه ۲ سوئیچ XORPlus، می‌توان به پروتکلهای STP / RSTP / MSTP، VLAN 802.1q، QoS، LCAP، LLDP، IPv6، IPv4، IGMP، RIP، OSPF / ECMP و ACL اشاره کرد. از ویژگی‌های لایه ۳ این محصول می‌توان پشتیبانی از پروتکلهای این IP برداشت نام برد.

سوئیچ OVS نسخه 1.1، از نسخه 1.0 پروتکل OpenFlow پشتیبانی می‌کند. مهمتر از همه اینکه XORplus مجموعه توسعه دهنده‌گان این حوزه را قادر به نوآوری می‌کند، کاربران می‌توانند پروتکلهای پیشویر و مدیریت ترافیک داده را بدون محدودیتهای سنتی سوئیچهای نهفته (Embedded Switches) توسعه دهند. Pica8 XORPlus مستقل از چیپ‌های سوئیچ اصلی عمل می‌کند و می‌تواند در پلتفرمهای گوناگون با قابلیت توسعه بالا اجرا شود. معماری نرم افزار بدین منظور طراحی می‌شود که به مجموعه‌ای از پروتکلهای اجازه دهد در پلتفرمهای مختلفی از ساخت افزار سوئیچینگ، اجرا شود. این

خصوصیه امکان استفاده از مدلهای بسیار انعطاف‌پذیرتری را نسبت به سوئیچهای سنتی فراهم می‌آورد. با نرم افزار متن باز و پلتفرم باز، کاربران سوئیچهای قدرتمند می‌توانند سرانجام وارد قلمرو سوئیچهای اختصاصی سودآور شوند.

[برای دریافت نرم افزار Pica8 XORPlus به آدرس زیر مراجعه نمایید:

<http://sourceforge.net/projects/xorplus>



OF13SoftSwitch

OF13SoftSwitch یک سوئیچ نرم افزاری فضای - کاربر مبتنی بر پیاده‌سازی سوئیچ نرم افزاری Forwarding Plane Ericsson TrafficLab نسخه 1.1 است که با تغییراتی که در OpenFlow انجام داده، برای پشتیبانی از OpenFlow نسخه 1.3 آماده گردیده است. کد اصلی که در این پروژه مورد استفاده قرار گرفته شده، مبتنی بر پیاده‌سازی نسخه 1.0 پروتکل OpenFlow است. اجزای سازنده در OF13SoftSwitch عبارتند از:

- **ofdatapath**: پیاده‌سازی سوئیچ OpenFlow نسخه 1.3
- **ofprotocol**: کانال امن برای اتصال سوئیچ به کنترلر
- **oflib**: یک کتابخانه نرم افزاری برای تبدیل از / به OpenFlow نسخه 1.3
- **dpctl**: یک برنامه کاربردی مبتنی بر خط فرمان، برای پیکربندی OF13SoftSwitch از طریق کنسول

این پروژه توسط مرکز تحقیقات اریکسون در بروزیل پشتیبانی می‌شود و توسط موسسه CPqD (با همکاری مرکز تحقیقات اریکسون در حوزه فنی) نگهداری می‌شود. دستور العملهای نصب و دانلود سوئیچ نرم افزاری، همراه با راهنمای آموزشی را می‌توانید در صفحه مربوط به این پروژه، در Github ببایدید. می‌توانید نسخه از پیش پیکربندی شده‌ای از OF13SoftSwitch را مورد آزمایش قرار دهید که شامل سوئیچ نرم افزاری OpenFlow نسخه 1.3، نسخه‌ای سازگار (Compatible) از کنترلر NOX، یک پلاگین برای نرم افزار Wireshark و مجموعه‌ای از آزمایشات OpenFlow به نام OF-test می‌شود.

[برای اطلاعات بیشتر از پروژه OF13SoftSwitch، به آدرس زیر رجوع کنید:

<http://cpqd.github.io/ofsoftswitch13/>



کنترل‌ها

ما در فصل ۲، کنترل‌ر POX و کنترل‌های OpenDaylight را بررسی کردیم. در این بخش فهرستی از دیگر کنترل‌های متن باز OpenFlow را ارائه خواهیم کرد.

Beacon

یک کنترل سریع، چند پلتفرمی، ماژولار و مبتنی بر زبان جاوا است که از هر دو عملیات رویداد محور و ریسمانی پشتیبانی می‌کند. توسعه Beacon از اوایل سال ۲۰۱۰ آغاز شده و در چندین پروژه تحقیقاتی، کلاس‌های شبکه و محیط‌های آزمایشگاهی مورد استفاده قرار گرفته است. این کنترل در جاوا نوشته می‌شود و در پلتفرم‌های متعددی از گران قیمت ترین سرورهای لینوکس چند هسته‌ای گرفته تا تلفن‌های اندروید اجرا می‌شود. کنترل‌ر Beacon تحت یک لیسانس ترکیبی از گواهی GPL نسخه 2 و گواهی دانشگاه استنفورد با نام Stanford University FOSS License Exception نسخه 1.0 است. مجموعه کدهای مورد استفاده Beacon در هنگام اجرا، بدون ایجاد وقفه بر روی دیگر مجموعه کدها، قابلیت شروع / توقف / تازه‌سازی^{۱۰} / نصب، را دارد. به عنوان مثال، بدون قطع کردن ارتباط سوئیچها، می‌توانید Net App Learning Switch در حال اجرای Net App با Net App دیگری جایگزین کنید.

می‌توانید اطلاعات بیشتری در خصوص این کنترل را در این آدرس ببایدید:
<https://openflow.stanford.edu/display/Beacon/Home>

Floodlight

کنترل SDN متن باز Floodlight مبتنی بر زبان جاوا، تحت لیسانس Apache بوده و می‌توان آن را مخصوصی در سطح تجاری محسوب کرد. این کنترل‌ر توسعه جامعه‌ای از توسعه دهندهان شامل تعدادی از مهندسان شرکت Big Switch Networks، پشتیبانی می‌شود. همانطور که در ابتداء بیان کردیم، کنترل‌ر Floodlight در جاوا نوشته شده است و بنابراین داخل یک JVM اجرا می‌شود. مخزن کدهای این محصول در Github موجود است. ساده‌ترین راه برای شروع کار با Floodlight، دانلود ماشین مجازی Floodlight در قالب یک Appliance است. Floodlight علاوه بر نقشی که بعنوان یک کنترل‌ر OpenFlow ایفا می‌کند، مجموعه‌های از اپلیکیشنها را شامل می‌شود که بر روی خود قرار داده است. در حالیکه اپلیکیشنها قرار گرفته بر روی کنترل‌ر Floodlight، مشخصه‌های مختلفی را برای برآوردن

نیازهای گوناگون کاربران در شبکه ارائه می‌دهند، خود کنترلر Floodlight، مجموعه‌ای از قابلیتهای متداول را برای کنترل و بررسی یک شبکه OpenFlow محقق می‌سازد. معماری Floodlight در شکل زیر نشان داده است.

ارتباط بین کنترلر Floodlight، اپلیکیشن‌های ساخته شده به عنوان مازولهای جاوا که با کامپایل می‌شوند و اپلیکیشن‌های شبکه‌ای ساخته شده در REST API، در این تصویر نشان داده شده است.



هنگامی که کنترلر Floodlight شروع به کار می‌کند، مجموعه‌ای از اپلیکیشن‌های مازول جاوا، که در فایل Properties مربوط به Floodlight بارگذاری شده‌اند (مانند سوئیچ یادگیرنده، هاب، فایروال و ورودی Static Flow Pusher)، اجرا می‌شوند. REST API ها که با تمام مازولهای در حال اجرا در شکل نشان داده شده‌اند، از طریق پورت مربوط به REST (بصورت پیش فرض ۸۰۸۰) در دسترس هستند. دیگر Net App ها (بعنوان مثال، پلاگین Quantum) مربوط به سیستم عامل آبری OpenStack یا (Circuit Pusher) می‌توانند این REST API را بکار ببرند تا اطلاعات را بازیابی کرده و با ارسال دستورات REST به پورت REST HTTP کنترلر، سرویسها را فراخوانی کنند.

می‌توانید اطلاعات بیشتری را درباره Floodlight در آدرس زیر پیدا کنید:

<http://www.projectfloodlight.org/floodlight/>



Maestro

Maestro یک سیستم عامل شبکه‌ای برای هماهنگ‌سازی اپلیکیشن‌های کنترل شبکه است. اینترفیس‌هایی را برای پیاده‌سازی اپلیکیشن‌های کنترلی فراهم کرده تا به شبکه دسترسی داشته باشند و بتوانند آن را اصلاح کنند و همچنین تعاملات خود را از طریق چندین پروتکل از جمله OpenFlow هماهنگ نمایند. اگرچه Maestro را می‌توان به عنوان یک کنترلر OpenFlow محسوب کرد، اما این سیستم عامل محدود به شبکه‌های مبتنی بر OpenFlow نیست. چهارچوب برنامه نویسی Maestro، اینترفیس‌هایی برای موارد زیر فراهم می‌آورد:

- معرفی عملکردهای سفارشی شده کنترلی جدید با افزودن کامپوننتهای کنترلی که بصورت مازول در آمدند.
- حفظ و نگهداری وضعیت شبکه به نمایندگی از کامپوننتهای کنترلی.
- ساختن کامپوننتهای کنترلی از طریق مشخص کردن توالی اجرا و وضعیت شبکه اشتراکی کامپوننتهای.

در جاوا توسعه پیدا کرده است (هم پلتفرم و هم کامپوننتهای)، که این امر باعث شده کنترلر مذکور برای سیستمهای عامل و معماری‌های گوناگون، بطور فزاینده‌ای قابل حمل (Portable) شود. همچنین با استفاده از تکنیک‌های چند - رسماً (Multi-threading)، از مزایای پردازشگرهای چند هسته‌ای به طور کامل بهره می‌برد. Maestro تحت لیسانس GNU Lesser General Public نسخه 2.1 قرار دارد.

برای جزئیات بیشتر در خصوص دانلود، استفاده و برنامه نویسی Maestro، به آدرس

زیر مراجعه کنید:

<http://code.google.com/p/maestro-platform/>



^{۱۰} عبارت چند - رسماً (Multi-Threading) به توانایی سیستم عامل در اجرای بخش‌های مختلف یک برنامه بصورت همزمان، گفته می‌شود. از مزایای تکنیک چند - رسماً می‌توان به اشتراک گذاری فضای حافظه توسط رسماًها اشاره کرد. چند - رسماً در محیط جاوا همچنین ویژگی‌هایی دارد که از آن جمله می‌توان به عدم بلوکه کردن کاربر بدليل استقلال رسماًها از یکدیگر، عدم تائیرپذیری یک رسماً از رسماًها دیگر و اجرای همزمان رسماًها با یکدیگر را نام برد. (متترجم)

Trema

یک چهارچوب کنترلر OpenFlow است که شامل تمام موارد نیاز برای ایجاد کنترلرهای OpenFlow در زبان Ruby و C است. بسته اصلی Trema شامل کتابخانه‌های پایه و ماژولهای کاربردی‌ای است که مانند یک رابط برای سوئیچهای OpenFlow عمل می‌کند. چندین اپلیکیشن نمونه بر روی Trema توسعه یافته و در دسترس هستند. بنابراین می‌توانید آنها را مانند کنترلرهای OpenFlow اجرا کنید. به علاوه، Trema یک چهار چوب ساده اما قدرتمند است که شبکه‌ای مبتنی بر OpenFlow و میزبانهای نهایی را مدل می‌کند. این چهارچوب برای تست کنترلرهای ارائه می‌شود. برای تشخیص جریانهای داده داخلی، پلاگین Wireshark در این چهارچوب، به عنوان یک ابزار اشکال‌زدایی ارائه شده است. تنها از لینوکس پشتیبانی می‌کند و در پلتفرم‌های زیر آزمایش شده است:

- سیستم عامل Ubuntu نسخه‌های 10.04، 11.10، 12.04، 12.10، 13.04 و 14.04 (نسخه دسکتاپ (i386 / AMD64)
- سیستم عامل Debian GNU / Linux نسخه‌های 6.0 و 7.0 (i386 / AMD64)
- سیستم عامل Fedora نسخه 16 (i386/x86_64)
- زبان برنامه نویسی Ruby نسخه 1.8.7
- RubyGems نسخه 1.3.6 نیز یا بالاتر

ممکن است در آینده روی دیگر نسخه‌های منتشره از پلتفرم‌های گوناگون نیز اجرا شود، ولی تا زمان نوشتن این کتاب هنوز آزمایش نشده و پشتیبانی نمی‌شود.

برای اطلاعات بیشتر درباره Trema به آدرس زیر مراجعه کنید:

<http://github.com/trema>



FlowER

یک کنترلر OpenFlow متن باز مبتنی بر زبان Erlang به شمار می‌رود. هدف آن فراهم کردن یک پلتفرم ساده برای نوشتن نرم افزار کنترل شبکه در Erlang است. این کنترلر اکنون در حال توسعه بوده ولی در حال حاضر شرکت Travelping بعنوان خالق این کنترلر، از آن در محصولات تجاری خود استفاده می‌کند. FlowER برای یک نوع مدل استقراری (Deployment Model) ساخته شده است که هر کدام از اپلیکیشن‌های Erlang را در قالب پسته‌های RPM یا DEB، پسته‌بندی می‌کند.

می‌توانید اطلاعات بیشتر درباره FlowER را در آدرس زیر یافت کنید:

<http://github.com/travelping/flower>



Ryu

Ryu یک چهارچوب SDN مبتنی بر کامپوننت است که با سیستم عامل OpenStack ادغام شده و از پروتکل OpenFlow پشتیبانی می‌کند. این برنامه یک کنترلر از لحاظ منطقی مرکزی و یک API خوش تعریف را ارائه داده که برای اپراتورها، ایجاد مدیریت شبکه‌های جدید و کنترل اپلیکیشنها را ساده می‌سازد. Ryu از پروتکلهای گوناگونی مانند OpenFlow (نسخه‌های 1.0، 1.2، 1.3 و نسخه‌های افزودنی OF-config، Netconf، Nicira) و غیره، برای مدیریت تجهیزات شبکه پشتیبانی می‌کند. هدف Ryu ایجاد سیستم عاملی برای SDN است که چنان سطح کیفی بالایی داشته باشد تا بتوان از آن در یک محیط تجاری بزرگ استفاده کرد. تمام کدها به صورت رایگان بوده و تحت لیسانس Apache نسخه 2.0 در دسترس عموم قرار دارد.

توسط Ryu، اپراتورها می‌توانند بدون استفاده از VLAN، ده‌ها هزار شبکه مجازی مجزا ایجاد کنند. شما می‌توانید شبکه‌های مجازی را ایجاد کرده و مدیریت کنید. این شبکه‌های مجازی با استفاده از پلاگین Ryu و در داخل سیستم عامل OpenStack انتشار می‌یابند. Ryu، توانایی پیکربندی مطلوب سوئیچهای Open vSwitch را دارد. فایل Image از پیش پیکربندی شده ماشین مجازی Ryu، اپراتورها را قادر می‌سازد که به سادگی، یک محیط چند گره‌ای در OpenStack بپا کنند. Ryu در Python پیاده‌سازی می‌شود و توسعه آن برای همگان آزاد است.

شما می‌توانید اطلاعات بیشتر درباره Ryu را در این آدرس زیر ببایدید:

<http://osrg.github.io/ryu/>



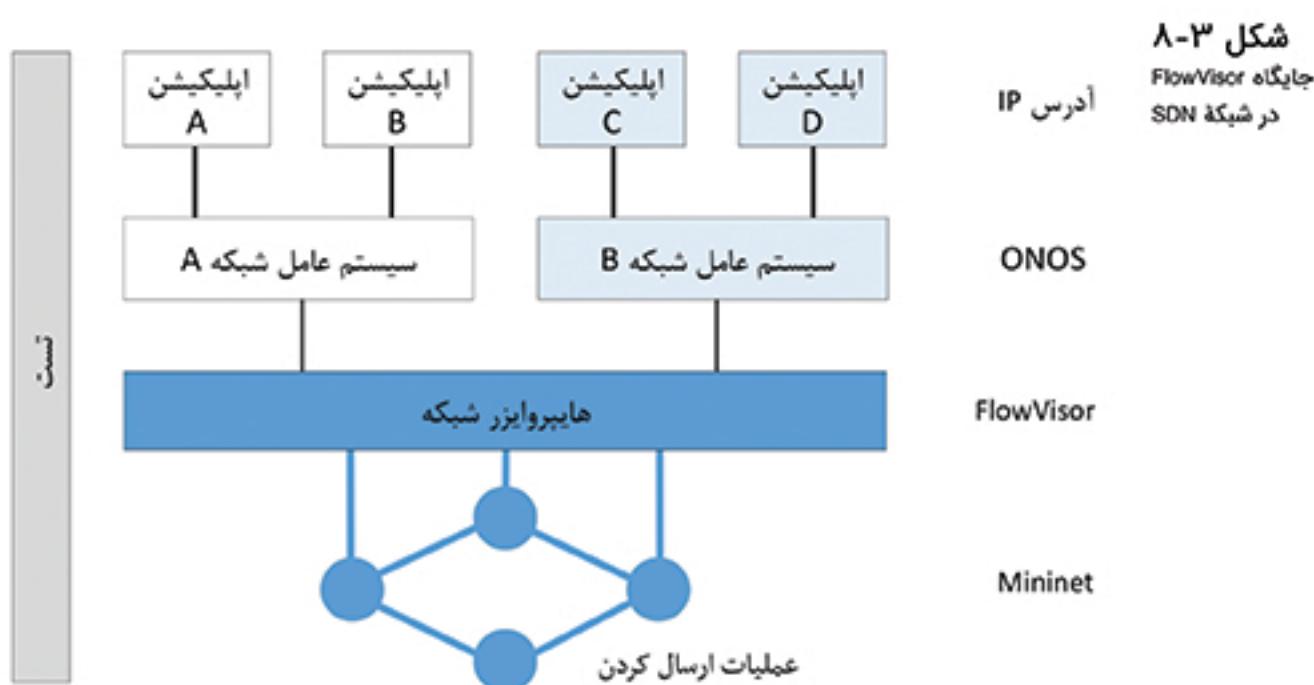
متفرقه‌ها

علاوه بر سوئیچهای نرم افزاری و کنترلرها، تعداد دیگری از پروژه‌های متن باز حول محور OpenFlow و SDN وجود دارند. در این بخش مرا نکات مفیدی در خصوص برخی پروژه‌های مهم متن باز ارائه خواهیم کرد.

FlowVisor

شبکه تعریف شده با نرم افزار با همان SDN، می‌تواند سطحی از مرکزی زدایی منطقی را با کنترلرهای منطقی گوناگون داشته باشد. کنترلر جالبی از نوع کنترلر پراکسی، با نام FlowVisor وجود دارد که می‌تواند برای افزودن سطحی از مجازی‌سازی شبکه در شبکه‌های OpenFlow بکار رفته و به کنترلرهای گوناگون اجازه دهد بطور همزمان تعدادی از سوئیچهای فیزیکی OpenFlow را که با یکدیگر همپوشانی

دارند، کنترل کنند. این محصول در ابتدا به منظور تحقیقات در آزمایشگاه ایجاد شد تا محققین بتوانند بر روی شبکه‌هایی که ترافیک داده واقعی را در خود دارند، آزمایش‌هایی انجام دهند. کنترلر FlowVisor همچنین استقرار سرویس‌های جدید در محیط‌های SDN را تسهیل بخشدیه است. این پروژه، یک کنترلر OpenFlow با اهداف ویژه است که مانند یک پراکسی Transparent بین سوئیچهای OpenFlow و کنترلرهای گوناگون OpenFlow قرار می‌گیرد. تصویر زیر جایگاه FlowVisor در یک شبکه مبتنی بر OpenFlow را نمایش می‌دهد:



همانطور که در فصل ۶ بطور مفصل بیان شد، کنترلر FlowVisor برش‌هایی غنی از منابع شبکه را ایجاد و کنترل هر کدام از برش‌ها را به یک کنترلر واگذار می‌کند و همچنین مجزا بودن برش‌ها را بهبود می‌بخشد. زادگاه کنترلر FlowVisor دانشگاه استنفورد است و بطور گسترده‌ای در شبکه‌های آموزشی و تحقیقاتی به منظور ایجاد برش‌هایی در آن شبکه‌ها، مورد استفاده قرار گرفته است. آزمایشگران گوناگون با استفاده از سیستم عامل شبکه خود و مجموعه‌ای از اپلیکیشن‌های کنترل و مدیریت شبکه، برش مجزای خود را از زیرساخت ایجاد می‌کنند. FlowVisor شما را قادر می‌سازد فعالیت‌های تحقیقاتی در حوزه شبکه را در محیط‌های عملیاتی واقعی هدایت کرده و از ترافیک شبکه واقعی استفاده کنید. FlowVisor یک کنترلر پراکسی متن باز بوده و می‌توان کدهای آن را برای انطباق با نیازها، سفارشی سازی کرد. با یک اینترفیس پیکربندی و مانیتورینگ در JSON برای کاربران (و یک زبان برنامه نویسی

جاوا برای توسعه دهنده‌گان)، هر شخصی توان سفارشی سازی سرویس‌های مختلف با روش "Opt-in" را دارد. شما می‌توانید آزادانه و به سرعت FlowVisor را در محیط SDN، همراه با تمام عملکردهای بنیادی آن محیط آزمایش کنید. این امکان شما را قادر می‌سازد مجازی‌سازی شبکه را فرا گرفته و روش‌های جدیدی را برای استقرار سریع سرویس‌ها، آزمایش کنید.

از آنجاییکه FlowVisor مبتنی بر استانداردهای متن بازی است که در زیر ساختی با محصولات مختلف اجرا می‌شود، شرکتهای سازنده گوناگون مانند NEC، Pronto، HP، OVS و غیره، و همچنین سیستم عاملهای شبکه مختلفی را پشتیبانی می‌کنند.

ما در فصل ۶، FlowVisor را به عنوان بخشی از بحث خود، درباره مجازی‌سازی شبکه، معرفی کردیم. برای کسب اطلاعات بیشتر و دانلود آن به آدرس زیر مراجعه کنید:

<http://onlab.us>



Avior

Applikیشنی است که بطور مجزا از کنترلر Floodlight ساخته شده و بعنوان یک قابلیت، به این کنترلر افزوده می‌شود. این Applikیشن برای مدیران شبکه یک رابط گرافیکی فراهم کرده تا نیازهایشان را پشتیبانی کند. Avior به منظور نظارت یا اداره شبکه و با استفاده از محیط گرافیکی‌ای خود، وابستگی استفاده از اسکرپت‌های Python یا مشاهده REST API را از بین می‌برد. این Applikیشن دیدی کلی از کنترلر، سوئیچ و تجهیزات ارائه می‌دهد و از بخشی به نام Flow Manager تشکیل شده که قابلیت‌های مختلفی را در خود دارد. بخش Controller Overview، اطلاعاتی را درباره کنترلر ارائه می‌دهد؛ از جمله این اطلاعات می‌توان به نام میزبان،^{۴۰} Memory Bloat در JVM، این که آیا کنترلر داده JSON فراهم می‌کند یا خیر و در نهایت مازولهای بارگذاری شده کنونی، اشاره کرد. بخش Switch Overview، اطلاعاتی را درباره پورتها و شمارنده‌های ترافیک مربوط به پورتها و ورودی‌های جدول جریان ارائه

^{۴۰} زمانی که یک شرکت از روش بازاریابی Opt-in استفاده می‌نماید، مصرف کننده با کاربر باید رضایت خود را مبنی بر دریافت اطلاعات محصول، خدمات و اخبار آن شرکت با دادن اطلاعات شخصی و اطلاعات تماس به شخص ثالث (مؤسسات و شرکتهای دیگر همکار با طرف فرازداد با آن شرکت) اعلام نماید. عموماً مصرف کننده با کاربر باید بر روی فرم‌های بخصوصی کلیک کرده و آنها را انتخاب نماید پا به شرکت اینمیل زده و رضایت خود را مبنی بر استفاده از اطلاعات شخصی خود اعلام کند. (متترجم)

^{۴۱} Memory Bloat یا در اصطلاح "پف کردن حافظه" به وضعيتی گفته می‌شود که برنامه حجم زیادی از حافظه را بیش از حد در اختیار می‌گیرد. این مسئله مشکلاتی را به همراه خواهد داشت که از آن جمله می‌توان به کمبود حافظه (Memory Leak) اشاره کرد. بر این اساس برنامه بدليل مشکلاتی که اغلب در کد منبع وجود دارد، حجم زیادی از حافظه را در اختیار می‌گیرد، اما نمی‌تواند آن را آزاد کند. (متترجم)

می‌کند. هر دو جریان ایستا و پویا با مشخصه‌هایی همچون اولویت، انطباق، Action، بسته‌ها، بایتها، مدت زمان و جزئیات وقفه زمانی (Timeout Duration) نشان داده می‌شود. بخش Device Overview، اطلاعاتی را درباره آدرس IP، آدرس MAC، DPID مربوط به سوئیچ متصل به کنترلر، پورت سوئیچ متصل شده و آخرین باری که دستگاه فوق در شبکه دیده شده است را نشان می‌دهد. متوی Flow Manager دیدی کلی و اطلاعاتی دقیق از جریان ایستا برای هر سوئیچ فراهم می‌کند. در این بخش می‌توانید ورودی‌های جریان را مدیریت (همچون حذف یا اضافه) کنید. بطور خلاصه Avior از تعدادی مشخصه مفید پشتیبانی می‌کند. از جمله:

- اضافه، اصلاح و حذف جریانها با استفاده از اینترفیس Static Flow Entry Pusher.
- بررسی خطاهای بصورت کارآمد و تائید جریان
- بروز رسانی دقیق آمارهای مربوط به کنترلر، سوئیچ، تجهیزات، اینترفیس و جریان در زمان واقعی
- مراجعت منطقی با کاربردی آسان Patch Panel

کنترلر Avior در دانشگاه Marist، جهت کار بر روی پروژه OpenFlow ایجاد شده است. برای اطلاعات بیشتر درباره Avior و دانلود آن، می‌توانید به آدرس زیر مراجعه کنید:

<http://github.com/sovietaced/avior>



RouteFlow

یک پروژه متن باز برای ایجاد مسیریابی IP مجازی شده در سخت افزارهای باقابلیت OpenFlow است. این پروژه توسط یک اپلیکیشن کنترلر OpenFlow، یک سرور مستقل و یک محیط شبکه مجازی، ساخته می‌شود که امکان ارتباط یک ژیرساخت فیزیکی را باز تولید کرده و موتورهای مسیریابی IP را اجرا می‌کند. موتورهای مسیریابی، FIB را درون سرویس IP Tables سیستم عامل لینوکس بر اساس پروتکلهای مسیریابی پیکربندی شده (مانند OSPF و BGP)، ایجاد می‌کنند. RouteFlow انتعطاف‌پذیری Stack‌های مسیریابی متن باز مبتنی بر لینوکس (مانند Quagga و XORP) را با کارایی نرخ خط دستگاه‌های OpenFlow ترکیب می‌کند. علاوه بر نوآوری پیرامون مسیریابی IP و انواع مختلف مجازی‌سازی شبکه، اجازه می‌دهد از طریق شبکه IP ترکیبی کنترل محور، مهاجرت به شبکه‌های SDN، فراهم شود. کامپوننتهای اصلی RouteFlow عبارتند از:

- RouteFlow Client (RF-Client)
- RouteFlow Server

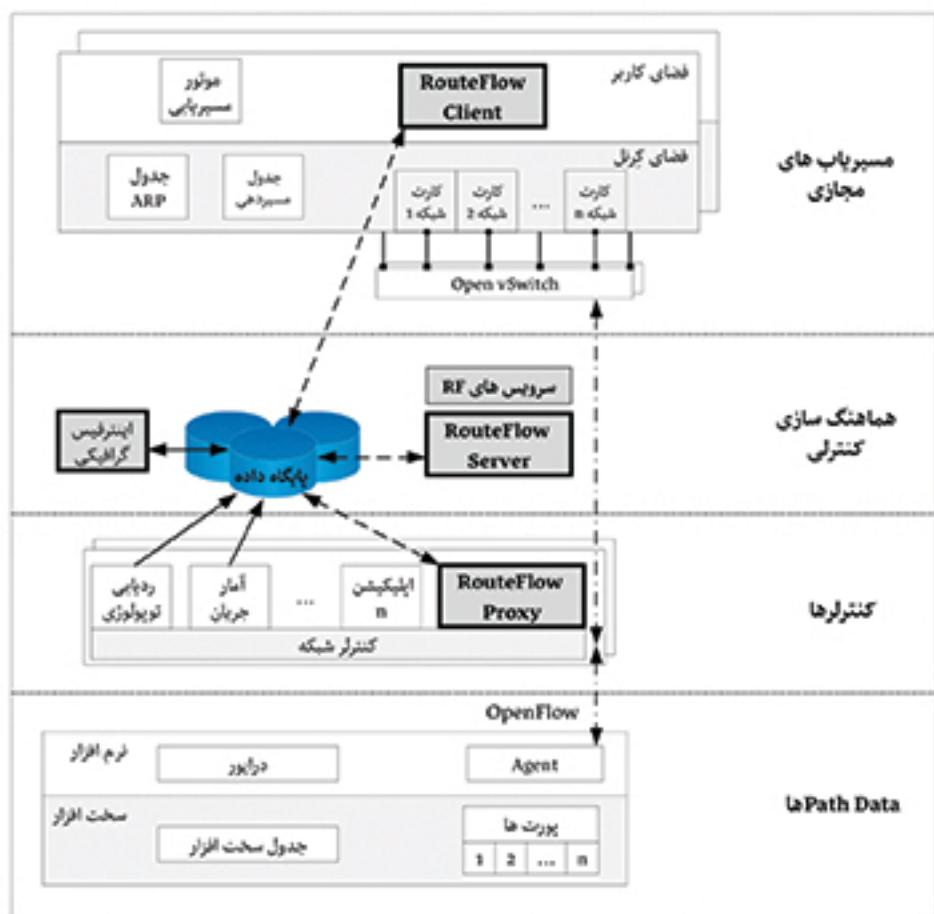
RF-Proxy که در گذشته به نام اپلیکیشن RouteFlow Proxy (RF-Proxy) •
Controller (RF-C) شناخته می‌شد.

هدف اصلی RouteFlow، توسعه یک چهارچوب متن باز برای راه حل‌های مبتنی بر مسیریابی IP مجازی در یک ساخت افزار تجاری است که OpenFlow API را پیاده‌سازی می‌کند. RouteFlow معماری مسیریابی متدالی را هدف گذاری کرده است. این معماری کارائی نوچ خط ساخت افزارهای تجاری را با انعطاف‌پذیری Stack‌های مسیریابی متن باز (اجرا شده بر روی کامپیوتراهای عمومی)، ترکیب می‌کند. نتایج کلیدی بدست آمده از راه حل‌هایی که RouteFlow ارائه می‌دهد، عبارتند از:

- مسیر مهاجرت تکنولوژی از شبکه‌های IP قدیمی به سوی شبکه‌های SDN / OpenFlow
- چهارچوب متن بازی برای پشتیبانی از گونه‌های مختلف مجازی‌سازی شبکه (به طور مثال مسیریابی منطقی، تسهیم سازی^{۱۰} / تجمعی کردن مسیریاب)
- مدل‌های IP Routing-as-a-Service •

شکل ۸-۴

اجزای سازنده
RouteFlow



برای اطلاعات بیشتر درباره RouteFlow می‌توانید به آدرس زیر مراجعه کنید:

<http://sites.google.com/site/routeflow/home>



CBench و OFlops

یک OFlops یک کنترلر مستقل است که جنبه‌های گوناگون یک سوئیچ OpenFlow را محک (Benchmark) می‌زند. یک چهارچوب مازولار را برای افزودن و اجرای تست‌های Implementation-agnostic کانال کنترلی با سوئیچ راه اندازی کرده و چندین پورت شبکه را برای تولید ترافیک در Data Plane (OpenFlow) مورد استفاده قرار می‌دهد. بعلاوه، OFlops به منظور خواندن MIB‌های گوناگون مانند کارکرد CPU، شمارنده‌های بسته و غیره، از پروتکل SNMP پشتیبانی می‌کند. OFlops دو جزء اصلی دارد:

- برنامه قابل اجرا، که قابلیت اصلی پلتفرم را پیاده‌سازی می‌کند.
- مجموعه‌ای از کتابخانه‌های پویای بارگذاری شده که قابلیتهای مورد نیاز را برای ارزیابی کارائی سیستم، پیاده‌سازی می‌کند.

این کامپوننت‌ها با استفاده از مجموعه‌ای غنی از API‌های رویداد محور، با یکدیگر ارتباط برقرار می‌کنند. هر کدام از تست‌های پویا می‌توانند زیر مجموعه‌ای آماده از بخش رسیدگی کننده رویدادها را پیاده‌سازی کرده و رفتار OFlops را تنظیم کند. OFlops به منظور محک زدن کارائی سوئیچ، اندازه‌گیری بسیار دقیقی را در لایه‌های مختلف ارائه می‌کند. این برنامه توازنی چند رشته‌ای (Multi-Threading Parallelism) را مورد استفاده قرار می‌دهد.

CBench برنامه‌ای برای آزمایش کنترلرهای OpenFlow از طریق ایجاد رویدادهای Packet-in برای جریانهای جدید است. CBench وظیفه مدل سازی سوئیچهای شبکه را به عهده دارد و براساس آن، مجموعه‌ای از سوئیچها را که به یک کنترلر متصل هستند مدل می‌کند، پیغامهای Packet-in را ارسال کرده و منتظر تغییر در وضعیت جدول جریان می‌ماند.

می‌توانید اطلاعات بیشتری را درباره OFlops و CBench در آدرس زیر بباید:

<archive.openflow.org/wk/images/3/3e/Manual.pdf>



OSCARs

نرم افزار^{۱۶} OSCARS که درون شبکه پر سرعت ESnet^{۱۷} قرار دارد، مدارهای مجازی با پهنای باند بالا و قابلیت چند - دامنه‌ای را ارائه می‌کند که کارائی مطلوب انتقال داده بین گره‌های مختلف درون یک شبکه را تضمین می‌کند. نرم افزار OSCARS هم به عنوان یک چهارچوب برای نوآوری در محیط‌های آزمایشگاهی و هم به عنوان یک سرویس در سطح تجاری که برای کاربران ESnet قابل اعتماد است، مورد استفاده قرار می‌گیرد. در حالی که ESnet مجموعه‌ای از کامپوننتهای سرویس را در اختیار کاربران مبتدی قرار می‌دهد، از طرف دیگر در جستجوی یک چهارچوب سرویس یکپارچه برای کمک به کاربران با تجربه بوده تا آنها بتوانند سرویسهای تجزیه تا پذیر ماژولار را به گونه دلخواه پیکربندی کنند. همچنین برای محققان شبکه، امکان سفارشی‌سازی را بر اساس پارامترهای مورد استفاده در آزمایشگاه، فراهم می‌کند.

اطلاعات بیشتر را می‌توانید در آدرس زیر یافت کنید:

<http://www.es.net/engineering-services/oscars/>



Twister

Luxoft Twister یک چهارچوب اتوماتیک آزمایشی است که برای مدیریت و هدایت موارد آزمایشی (Test Cases) در زبانهای اسکریپت نویسی سطح Shell، نوشته شده است. Twister از زبانهای TCL، Python و Perl پشتیبانی می‌کند. Twister از یک رابط کاربری شهودی^{۱۸} (Intuitive User Interface) مبتنی بر شبکه، برای پیکربندی، کنترل و گزارش گیری با قابلیت دسترسی از راه دور، استفاده می‌کند. این امر باعث سادگی در تولید مجموعه‌ای از تستها، اجرای آنها و نظارت بر خروجی آنها (از طریق Log‌های خروجی) می‌شود.

برای اطلاعات بیشتر به آدرس زیر مراجعه کنید:

<http://github.com/Luxoft/Twister>



On-Demand Secure Circuits And Advance Reservation System^{۱۹}
Energy Services Network^{۲۰}

این عبارت به رابط کاربری گفته می‌شود که کاربر بتواند بدون آنکه آن را در گذشته تجربه کرده باشد، آن را آموزش دیده باشد با از شخص دیگری کمک گرفته باشد، به آسانی با آن ارتباط برقرار کرده و کار کند. (متترجم)

FortNOX

یک برنامه الحاقی برای کنترلِ متن باز NOX است. FortNOX بطور اتوماتیک نقض یا عدم نقض قوانین جدید مربوط به یک جریان و سیاستهای امنیتی شبکه را بررسی می‌کند. FortNOX با استفاده از مجموعه‌ای از قوانین، می‌تواند تناقضاتی را که در آنها وجود دارد، حتی در صورت وجود تونلی از جریانهای پویا، تشخیص دهد.

[www.openflowsec.org]

برای اطلاعات بیشتر به آدرس زیر مراجعه کنید:



Nettle

برنامه Nettle به شبکه‌های مبتنی بر OpenFlow اجازه می‌دهد که با استفاده از یک زبان سطح بالا، تشریحی و گویا کنترل شوند. این برنامه شامل یک کتابخانه Haskell است که از پروتکل OpenFlow پشتیبانی می‌کند و برای این منظور یک سرور OpenFlow را ارائه می‌دهد.

[haskell.cs.yale.edu/nettle]

می‌توانید اطلاعات بیشتر درباره Nettle را در این آدرس ببایدید:



Frenetic

یک زبان مختص به دامنه^۹ برای برنامه نویسی شبکه‌های OpenFlow است که در زبان Python تعبیه شده است.

[www.frenetic-lang.org]



OESS

NDDI OESS اپلیکیشنی برای پیکربندی و کنترل سوتیچهای OpenFlow است که از طریق یک رابط کاربری بسیار ساده و کاربرپسند اجرا می‌شود. تأمین بستر در کمتر از یک ثانیه، قابلیت Failover

^۹ به زبان برنامه نویسی‌ای گفته می‌شود که برای برآورده ساختن جنبه خاصی از یک نرم‌افزار مورد استفاده است. از جمله این زبانها می‌توان XPath و SQL Query را نام برد. (متترجم)

بصورت اتوماتیک، ارائه مجاز به ازای هر اینترفیس و ارائه آمارهای اتوماتیک به ازای هر VLAN از جمله تواناییهایی است که توسط OESS ارائه می‌شود.

[برای اطلاعات بیشتر در مورد اپلیکیشن NDDI OESS به آدرس زیر مراجعه کنید:
<http://code.google.com/p/nddi/>]



خلاصه

شبکه تعریف شده با نرم افزار (SDN) و OpenFlow از جمله موضوعات بسیار پر طرفدار در حوزه دانشگاه و صنعت هستند. برنامه های توسعه یافته متن باز و تجاری متعددی پیرامون OpenFlow و OpenFlow / SDN وجود دارد. در این فصل شرحی کلی در خصوص پژوهه های مهم متن باز، حول XORPlus، LINC، Indigo، Pantou، Open vSwitch، سوئیچهای OF13SoftSwitch / SDN، OpenFlow / SDN ارائه کردیم. جزو پژوهه های متن باز فعال و مهم در حیطه SDN / OpenFlow، کنترلرهای FlowER، Trema، Maestro، Floodlight، Beacon و Ryu از جمله کنترلرهای FortNOX، Twister، OSCARS، OFLops / Cbench، RouteFlow، Avior، FlowVisor، Frenetic، Nettle، OESS اشاراتی داشتیم.

نمایه

|

۱۰۷.OESS ۱۰۸.Oflops ۱۰۹.OSCARS ۱۱۰.RouteFlow ۱۱۱.Twister ۱۱۲.SANE / Ethane پلاگین RestProxy توضیحات، ۱۱۳ Libuv پلتفرم توضیحات، ۱۱۴ پیغام‌های اصلاح جدول جریان ۱۱۵.ADD ۱۱۶.DELETE ۱۱۷.MODIFY ۱۱۸.DELETE و ۱۱۹.MODIFY پیغام‌های غیر همزمان توضیحات، ۱۱۱ حذف جریان (Flow-Removal) خطأ (Error) وضعیت پورت (Port-Status) ۱۱۲.Packet-in پیغام‌های کنترل - به - سوچ ارسال بسته (Send-Packet) ۱۱۳.Configuration پیکربندی توضیحات، ۱۱۵ حالت خواندن (Read-State) خصوصیات (Features) مانع (Barrier) ۱۱۴.Modify-State وضعیت اصلاح پیغام‌های متقارن توضیحات، ۱۱۲ ۱۱۵.Echo ۱۱۶.Hello ۱۱۷.Vendor پیغام‌های OpenFlow	آزمایشگاه اجزای سازنده، ۷۶ توضیحات، ۶۹ راهاندازی، ۷۱ کنترلرهای ببرونس، ۷۵ میزبان، ۶۹ اپلیکیشن‌های شبکه Floodlight ۱۱۳.Circuit Pusher ۱۱۴.Static Flow Pusher ۱۱۵.Virtual Networking Filter اپلیکیشن POX راهاندازی و اجرا، ۵۱ اجزای سازنده سوچ‌های OpenFlow کنترل، ۹ اصلاح جریان (Flowmod) اینترفیس شمالی توضیحات، ۲۵ اینترفیس REST توضیحات، ۱۲۲ ب بسته‌ها انواع، ۹ توضیحات، ۹ بنیاد شبکه باز، ۸ پ پایگاه اطلاعاتی ارسال (FIB) پروژه‌های متن باز ۱۱۱.Avisor ۱۱۴.Cbench ۱۱۵.FlowVisor ۱۱۶.FortNOX ۱۱۷.Frenetic ۱۱۸.Nettle
--	--

سویچ‌های OpenFlow هایبریدی توضیحات، ۹ سویچ یادگیرنده ازرنت ایجاد، ۹۱ توسعه، ۹۶	پیام‌های غیرهمزن، ۲۲ پیام‌های کنترلر - به - سویچ، ۱۹ پیام‌های متقارن، ۲۲ توضیحات، ۱۷	ت
ش	Net App اپلیکشن ارسال کننده بسته، ۱۰۳ سویچ یادگیرنده ازرنت، ۹۶ فایروال مسدود، ۱۰۰	ج
ف	فیلد شمارنده‌ها توضیحات، ۱۰	جدول جریان توضیحات، ۹ شمارنده‌ها، ۱۰
ک	کنترلرها ۱۴۵، Beacon ۱۴۶، Floodlight ۱۴۸، FlowER ۱۴۹، Maestro ۱۴۱، Ryu ۱۴۸، Trema کنترلرهای OpenFlow پیاده‌سازی، ۵۰ توضیحات، ۴۷ کنترلرهای ویژه، ۶۶ ۶۲، Floodlight ۷۰، NodeFlow ۴۴، Net App ۵۱، NOX ۵۱، POX کنترلر ODL توضیحات، ۸۷ محیط گرافیکی، ۸۷ معماری، ۸۲	۱۱، Ethernet Type ۱۲، IP Protocol ۱۲، IP Destination و IP Source ۱۱، Port FLOW_MATCH انطباق‌های جریان، ۱۱۲ توضیحات، ۱۱۲ سویچ مجازی Indigo توضیحات، ۱۲۲ سویچ مرجع OpenFlow پیام ارسال بسته (Send-Packet)، ۲۱ پیاده‌سازی، ۲۷ پیام پیکربندی (Configuration)، ۲۰ پیام حالت خواندن (Read-State)، ۲۰ پیام خصوصیات (Features)، ۲۹ پیام درخواست صف (Queue Query)، ۲۰ پیام وضعیت اصلاح (Modify-State)، ۲۰ پیام‌های غیرهمزن، ۲۱ پیام‌های متقارن، ۲۲ توضیحات، ۲۷ مانع (Barrier)، ۲۱ سویچ‌های OpenFlow مخصوص توضیحات، ۹
ل	لایه انتزاع سخت‌افزار (HAL) لایه انتزاع سرویس توضیحات، ۸۲	

توضیحات، ۱۴۹، ۱۴۹	لیست کنترل دسترسی
۱۷۷.FLOW_MATCH ساختار	توضیحات، ۶۲
۱۷۸.API	
۱۷۹.ForCES	
FortNOX	
توضیحات، ۱۵۶	
Frenetic	
توضیحات، ۱۵۶	
G	
۱۸۰.GSMP	
H	
Horizon	
توضیحات، ۱۷۶	
I	
۱۸۱.IaaS	
۱۸۲.IETF	
Indigo	
توضیحات، ۱۷۲، ۱۷۴	
K	
Keystone	
توضیحات، ۱۷۶	
L	
LINC	
توضیحات، ۱۷۲، ۱۷۴	
M	
Maestro	
توضیحات، ۱۴۷	
Mininet	
توضیحات، ۲۰	
۲۶، مراپا،	
ویرگی‌ها، ۲۶	
N	
NaaS	
توضیحات، ۱۷۷	
۱۷۸.NETCONF	
P	
معماری شبکه‌سازی OpenStack	
پلاگین ۱۷۹.Agent	
توضیحات، ۱۷۸	
شبکه مرکز داده فیزیکی، ۱۷۰	
۱۷۹.۲ لایپ Agent	
۱۷۹.۳ DHCP Agent	
R	
ورودی‌های جریان	
شمارنده‌ها، ۱۰	
فیلدی‌های هدر، ۱۰	
A	
Avior	
توضیحات، ۱۰۱	
ویرگی‌ها، ۱۵۲	
B	
Beacon	
توضیحات، ۱۶۵	
C	
Cbench	
توضیحات، ۱۵۴	
Cinder	
توضیحات ۱۷۵	
D	
۱۷۷.Dpctl	
F	
Floodlight	
تنظیم و راماندازی، ۱۷۲	
توضیحات، ۱۷۵، ۱۷۷، ۷۲	
N	
FlowER	
توضیحات، ۱۴۸	
Flowvisor	
برش‌بندی، ۱۱۶	

OpenWrt	Nettle
توضیحات، ۱۶۱	توضیحات، ۱۰۷
OSCARs	Neutron
توضیحات، ۱۰۰	توضیحات، ۱۲۸، ۱۲۹
P	
Pantou	NodeFlow
توضیحات، ۱۶۱	توضیحات، ۶۰
POX	NOX
توضیحات، ۵۱	توضیحات، ۵۱
R	
RouteFlow	OESS
توضیحات، ۱۵۱، ۶۶	توضیحات، ۱۰۷
کامپوننت‌ها، ۱۰۱	Of13softswitch
Ryu	توضیحات، ۱۴۴
توضیحات، ۱۴۹	OFlops
S	
SDN	اجزای سازنده، ۱۰۴
توضیحات، ۰	توضیحات، ۱۰۴
سخت‌افزار، ۶	کنترلر ODL
Swift	OpenDaylight
توضیحات، ۱۲۵	آزمایشگاه SDN مبتنی ODL
Switches	۸۱، ۷۵
۱۶۲، Indigo	توضیحات، ۱۰۷
۱۶۲، LINC	اینترفیس شمالی، ۲۵
۱۶۴، OF13SoftSwitch	تاریخچه، ۸
۱۷۸، Open vSwitch	OPENSIG
۱۶۱، Pantou	OpenStack
۱۶۷، XORPlus	توضیحات، ۱۲۲
T	
Trema	کامپوننت‌ها، ۱۲۴
توضیحات، ۱۶۸	معماری، ۱۲۷
Twister	نصب، ۱۲۱
توضیحات، ۱۰۰	۱۷۰، Cinder
X	
X11 Forwarding	۱۷۱، Glance
فعال‌سازی در PuTTY	۱۷۲، Horizon
	۱۷۳، Keystone
	۱۷۴، Neutron
	۱۷۵، Swift
	Open vSwitch
	توضیحات، ۱۲۸، ۲۴

XORPlus

توضیحات، ۱۶۲

ویژگی‌ها، ۱۶۲

واژه‌نامه

Abstraction Layer	لایه انتزاعی	Hypervisor	هاپرولایزر
Asynchronous	غیر همزمان	Incoming Port	پورت دریافت کننده
Benchmark	محک زدن	Learning Table	جدول پادگیری
Built-in	از پیش گنجانده شده	Line Rate	نرخ خط
Cache	ذخیره	Listener	شنونده
Certification	گواهی	Listener port	پورت مشنونده
Cloning	همگن سازی	Match	انطباق / منطبق / تطابق
Cloud Computing	راپاچش ابری	Meta Data	فرا داده
Control Decision	تصمیمات کنترلی	Multi-Threading	چند ریسمانی
Counter	شمارنده	Name space	فضای نام
Customize	سفارشی سازی	Net app	اپلیکیشن شبکه
Debugging	اشکال زدایی	Node	گره
Drop	دور ریخته	Northbound Interface	اینترفیس شمالی
Dump	نمونه برداری	Offload	واآگذاری
Embedded	نهفته	Open Network Foundation	بنیاد شبکه باز
Emergency Mode	حالت اضطراری	Overlap	همبودشانی
Emulate	مدل	Parse	جزیه
Environment Variable	متغیر محیطی	Port Status	وضعیت پورت
Error	خطا	Porting	سازگار
Event	رویداد	Proactive	پیشگیرانه
Event Handler	رسیدگی کننده رویداد	Programmable Network	شبکه برنامه پذیر
Flood	پخش کردن	Pure OpenFlow Switch	سوچیج OpenFlow مخصوص
Flow Entry	وروودی جریان	Raw Block Level	سطح بلاک خام
Flow Modify (Flowmod)	اصلاح جریان	Reactive	واکنشی
Flow Removal	حذف جریان	Redundancy	افزوونگی
Flow Table	جدول جریان	Refresh	تازه‌سازی
Forwarding	ارسال	Replication	همانند سازی
Forwarding App	اپلیکیشن ارسال کننده	Round trip Time	زمان ناخیر پرخاشی
Forwarding Hardware	ساخت افزار ارسال کننده	Rule	قانون
Forwarding Information Base	پایگاه اطلاعاتی ارسال کننده	Scale out	مقیاس افقی
Forwarding Path	مسیر ارسال	Slice	بروش
Framework	چهارچوب	Software Defined Network	شبکه تعریف شده با فرم افزار
Graphical User Interface	رایط کاربری گرافیکی	Source Code	کد منبع
Hardware Abstraction Layer	لایه انتزاع ساخت افزار	Southbound Interface	اینترفیس جنوبی
Header Field	فیلد هدر	Symmetric	متقارن
Hybrid OpenFlow Switch	سوچیج OpenFlow هایبریدی	Table Miss Flow Entry	وروودی جریان جدول نوادرص

Table-miss Entry	ورودی نقص - جدول
Tagging	نگ نگاری
Third Party	راسط
Thread	ریسمان
Topology Discovery	ردیابی توپولوژی
Tunneling	تونل زدن
Userspace	فضای کاربر