وزارت ارتباطات و فناوری شرکت ارتباطات زیرساخت	**عنوان مقاله:** بررسی مزایا و چالش های شبکه های مبتنی بر نرم افزار و لزوم حرکت به سمت آنها			
	تهیه کننده/گان:	رشته تحصیلی	رشته شغلی	اداره کل/دفتر
	شادی نوری	مهندسی کامپیوتر-نرم افزار	رییس اداره طراحی شبکه دیتا	اداره کل طرح و توسعه شبکه
	مریم پسندی	مهندسی کامپیوتر – نرم افزار	کارشناس مسئول طرح و توسعه شبکه دیتا	اداره کل طرح و توسعه شبکه
	فاطمه سادات دشتکی	مهندسی کامپیوتر – معماری	کارشناس طراحی شبکه دیتا	اداره کل طرح و توسعه شبکه
این قسمت توسط دبیرخانه کمیته علمی تکمیل می‌گردد.	عنوان حوزه تحقیقاتی مورد نیاز شرکت: شناسایی و معرفی چالش های شبکه موجود و لزوم حرکت به سمت شبکه های مبتنی بر نرم افزار(SDN)			شماره ردیف عنوان حوزه تحقیقاتی مورد نیاز شرکت: ۶۸

بررسی مزایا و چالش های شبکه های مبتنی بر نرم افزار و لزوم حرکت به سمت آنها

شادی نوری ، مریم پسندی ، فاطمه سادات دشتکی

چکیده

ساختار شبکه های موجود و مشکلاتی که در زمینه پیکربندی و اعمال سیاست ها در این نوع از شبکه ها وجود دارد نه تنها سخت و وقت گیر است بلکه خطاپذیر بوده و مشکلاتی نیز در زمینه توسعه پذیری دارد لذا در نظر گرفتن این مسائل و مشکلاتی که شبکه های فعلی در تامین نیازمندی های شبکه های ابری دارند منجر به تحول در معماری شبکه گردید. شبکه های مبتنی بر نرم افزار(SDN) [۱] یک معماری جدید شبکه است که صفحه کنترل شبکه[۲] را از صفحه دیتا[۳] جدا می کند، با این ایده که استفاده از منابع شبکه بهبود و هزینه های عملیاتی کاهش یابد و باعث نوآوری و تکامل گردد

کلمات کلیدی : شبکه های مبتنی بر نرم افزار(SDN)، صفحه کنترل شبکه ، صفحه دیتا، مهندسی ترافیک، کنترل ازدحام, مدیریت جریان, تحمل پذیری خطا, شبکه محتوا محور, کنترلر, سوئیچ

مقدمه

با توجه به حجم اطلاعات در سطح اینترنت و شبکه های بزرگ، ساختار شبکه‌ها اهمیت فراوانی پیدا کرده است. پیدایش شبکه‌های بدون سیم و همچنین کمبود پهنای باند، نیاز هر چه بیشتر به یک معماری کارآمد و مفید را در شبکه‌های کامپیوتری نمایان می‌سازد[۱]. به همین دلیل ساختار شبکه‌ها باید به گونه‌ای باشد که بتوان در آن ایده‌های جدید را براحتی به کار گرفت و در جهت بهینه سازی این شبکه‌ها گام برداشت.

شبکه های فعلی و محدودیت های آنها

شبکه های مختلف در حال همگرایی به سمت ادغام شدن در یک شبکه واحد می باشند .بدین صورت که بتوان انتقال انواع ترافیک از قبیل داده ,ذخیره سازی و ... را ازبستر یک شبکه تامین نمود که رفع مشکلات و چالش هایی که در تامین نیازمندیهای شبکه های ابری وجود دارد باعث شد که ایده ایجاد تحول در عناصر سازنده شبکه شکل گیرد که یکی از فناوریهایی که در این راستا ارائه شد شبکه مبتنی بر نرم افزار(SDN) می باشد که در این شبکه ها به جای اینکه سخت افزار در مورد بسته ها تصمیم گیری کند نرم افزار

[۱] Software Defined Network
[۲] control plane
[۳] Data plane

می بایست با توجه به شرایط , تعریفی متناسب از نوع رفتاری که باید در قبال هر بسته صورت گیرد به سخت افزار ارائه کند .برای مثال در شبکه های فعلی وقتی یک بسته به مسیر یاب۴ می رسد مسیریاب تصمیم می گیرد که این بسته از چه مسیری ارسال شود ولی در SDN مسیریاب از نرم افزار کنترل کننده، نحوه برخورد با بسته را می پرسد و از این مرحله به بعد هر بسته ای که با مشخصات فوق به مسیر یاب برسد همانند بسته اول ارسال می شود. در شبکه های نرم افزاری تعریف شده صفحه کنترل از صفحه داده در هر مسیر یاب مجزا شده است و به جای چندین کنترل کننده توزیع شده , یک کنترل کننده مرکزی خواهیم داشت.

مزایا و چالش های SDN در یک نگاه

SDN با به کارگیری مجازی سازی در ساختار شبکه و فراهم کردن امکان جداسازی عناصر سازنده تجهیزات(سوئیچ و روتر) مدیران شبکه را قادر می سازد تا بدون نیاز به دسترسی فیزیکی به تجهیزات نسبت به مدیریت بهینه و متمرکز همه آنها اقدام کنند. در واقع این راهکار به گونه ای است که امکان برنامه ریزی متمرکز ترافیک شبکه را به وجود می آورد.

مزایای استفاده از SDN موجب شده تا بسیاری از شرکتهای بزرگ نظیر گوگل مایکروسافت و ....اقدام به همکاری با ONF که شناخته ترین انجمن در زمینه SDN است و استاندارد هایی نظیر openflow را برای این نوع شبکه ها ارائه کرده است، بنمایند لذا با وجود چالش های بسیاری که در این خصوص وجود دارد پروژه های بسیاری را در حوزه آکادمیک به خود اختصاص داده است.

لازم به ذکر است هر ساختاری در کنار مزیت‌هایی که دارد و امکاناتی که در اختیار قرار می‌دهد، با چالش هایی روبه رو است. ساختار شبکه‌های نرم افزار محور نیز از این قاعده مستثنی نیست و معماری معرفی شده، علی رغم برتری که نسبت به شبکه‌های سنتی دارد، مسائل و مشکلاتی نیز دارد که از جمله می توان به مسائلی مانند قابلیت اطمینان، مقیاس پذیری و کارایی شبکه در ساختار جدید اشاره کرد[۶]. بدیهی است که برای به‌کارگیری این شبکه‌ها و استفاده از امکانات آن‌ها در مقیاس وسیع، تضمین این ویژگی‌ها و ارائه‌ی راه‌حلی برای حل این مشکلات ضروری و لازم است.

اهمیت و ضرورت شبکه‌های برنامه پذیر

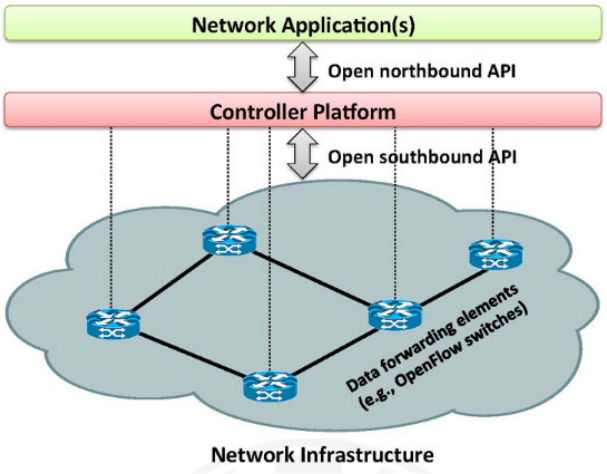
شبکه های IP سنتی که در آن‌ها پروتکل‌های توزیع شده در سوئیچ‌ها و روترها در حال اجرا هستند برخلاف نفوذ گسترده، پیچیده بوده و مدیریت دشواری دارند[۷]. برای توصیف سطح بالای سیاست ها، اپراتورهای شبکه باید هر دستگاه شبکه را جداگانه با استفاده از دستورات سطح پایین و اغلب وابسته به شرکت سازنده تنظیم کنند. علاوه بر پیچیدگی تنظیمات، شبکه ها باید دربرابر خطا مقاوم بوده و با تغییرات بار ترافیکی وفق یابند. مکانیسم های بازتنظیم و پاسخ خودکار در شبکه های IP فعلی مشاهده نمی شود و بنابراین اعمال سیاست ها در چنین محیطی بسیار چالش برانگیز می باشد.از سویی دیگر، صفحات کنترل داده در شبکه های فعلی، در دستگاه های شبکه به یکدیگر بسته شده‌اند که این انعطاف پذیری و نوآوری در شبکه را کاهش می‌دهد. ممکن است ۵ تا ۱۰ سال

[۴] Router

طول بکشد تا یک پروتکل مسیریابی جدید به صورت کامل طراحی و ارزیابی و پیاده سازی شود. به صورت مشابه یک باز طراحی کلی معماری اینترنت(برای مثال جایگزینی IP) در عمل دست یافتنی نیست[۸].

شبکه سازی نرم افزار محور, الگویی است که برای کاستن محدودیت‌های فعلی ظهور کرده و با جداسازی منطق کنترلی از عناصر ارسال داده، به هم پیوستگی صفحات کنترل و داده شکسته می شود که به تبع آن سوئیچها به عناصر ساده ارسال اطلاعات تبدیل شده و لایه کنترلی در یک کنترلر منطقا متمرکز یا به عبارتی سامانه عامل پیاده‌سازی می‌شود. این کار اعمال سیاست ها، بازتنظیم شبکه و تکامل آن را ساده تر می‌کند[۴]. ذکر این نکته ضروری به نظر می رسد که یک کنترلر منطقا متمرکز، الزامی برای تمرکز فیزیکی ایجاد نمی‌کند. در واقع نیاز به تضمین سطوح مناسب کارایی، مقیاس پذیری و قابلیت اطمینان چنین راه حلی را حذف می‌کند.

جدایی صفحات داده و کنترل با استفاده از تعریف مناسب یک واسط بین سوئیچها و کنترلر عملی می‌شود. همانطور که در شکل ۱ دیده می‌شود، کنترلر با استفاده از یک واسط برنامه کاربردی روی وضعیت عناصر صفحه داده کنترل می کند. مهمترین مثال این واسطها، اوپن فلو است. یک سوئیچ اوپن فلو یک یا چند جدول از قوانین برای کار با بسته ها دارد که به هر یک از آن‌ها جدول جریان گفته می شود. هر قانون با بخشی از ترافیک تطبیق پیدا میکند و عملیاتی مانند رها کردن و ارسال وتغییر دادن را روی آن اعمال می‌کند.



شکل ۱. نمایی ساده شده از معماری شبکه نرم افزار محور

تفکیک وظایف در SDN کلید دستیابی به انعطاف‌پذیری، شکستن مساله‌ی کنترل شبکه به قسمت‌های قابل پیگیری و ساده‌سازی مدیریت شبکه و تکامل و نوآوری است. با توجه به اینکه SDN و اوپن فلو به عنوان فعالترین آکادمیک آغاز به کار کرده اند[۸]، در سالهای اخیر توجه زیادی از جانب صنعت کسب کرده‌اند. اکنون اکثر سازندگان سوئیچهای تجاری از اوپن‌فلو در تجهیزات خود پشتیبانی می کنند. حرکت SDN به اندازه‌ای قدرتمند بوده که گوگل، فیس بوک، یاهو، مایکروسافت، ورایزن، و مخابرات آلمان را مجاب به سرمایه‌گذاری در تاسیس او ان اف با هدف اصلی رشد و توسعه SDN با معرفی استانداردهای باز کرد. به عنوان مثال، گوگل SDN را برای اتصال مراکز داده‌اش در سراسر جهان به کار برده است، شبکه‌ای که اکنون بیش از ۴ سال است به این شرکت در بهبود بار عملیاتی و کاهش هزینه‌ها کمک کرده است[۹]. بستر مجازی سازی شبکه‌ی وی‌امور به نام ان اس ایکس نیز نمونه‌ای از یک راه حل تجاری کامل است. بزرگترین شرکت‌های فعال در حوزه‌ی فناوری اطلاعات دنیا از حامل‌های اطلاعات و سازندگان تجهیزات گرفته تا فراهم آورندگان سرویس‌های ابری ومالی اخیر به انجمن های مرتبط با SDN مانند اوان اف و اوپن دی لایت پیوسته‌اند.

در کنار موارد مذکور، خطاهای انسانی در تنظیم شبکه‌های امروزی بسیار رایج‌اند. برای مثال در یک بررسی بیش از ۱۰۰۰ خطا در تنظیمات روترهای BGP مشاهده شده است[۹]. از یک دستگاه که دچار نقص تنظیمات است رفتارهای نامطلوبی مانند از دست رفتن بسته‌ها، حلقه‌های ارسال، شکل‌گیری مسیرهای ناخواسته و نقص قراردادهای خدمات در شبکه اتفاق می افتد. هر چند این اتفاق نادر است، اما یک روتر دچار نقص تنظیمات, ممکن است عملکرد کل اینترنت را مختل نماید[۱۰].

بنابرآنچه گفته شد، SDN با به کارگیری ریزگانی مبتنی بر تجریدصفحه کنترل, پیچیدگی مدیریت شبکه را کاهش و بهره‌وری از منابع شبکه را افزایش داده و هزینه‌های عملیاتی را کاهش می‌دهد و نوآوری و تکامل را برای شبکه‌ها به ارمغان می‌آورد.

شبکه‌های مبتنی بر نرم‌افزار

عبارت SDN در اصل برای نشان دادن کارها و ایده‌های پیرامون اوپن‌فلو در دانشگاه استنفورد شکل گرفت. طبق تعاریف اولیه، SDN به معماری شبکه ای گفته می شود که در آن چگونگی پیشرانی در صفحه‌ی داده توسط یک صفحه تفکیک شده از صفحه قبل کنترل می‌شود. به صورت دقیق‌تر، SDN به عنوان معماری شبکه‌ای با ۴ محور زیر تعریف می‌شود:

۱- صفحات کنترل و داده تفکیک شده‌اند و عملکرد کنترلی از دستگاه‌های شبکه که پس از این به عناصرپیشرانی ساده تبدیل خواهند شد، حذف می‌شود.

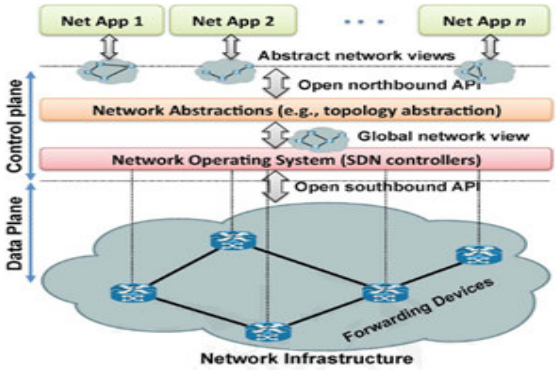
۲- تصمیمات پیشرانی به جای اینکه مبتنی بر مقصد باشند، مبتنی بر جریان هستند. یک جریان به صورت بسته‌هایی با مقادیر یکسان برای یک دسته میدان در سرایند تعریف می‌شود. همه بسته های یک جریان سیاست‌های خدماتی یکسانی در دستگاه‌های پیشرانی دریافت می‌کنند[۱۱].

۳- منطق کنترل به یک موجودیت خارجی منتقل می‌شود که به آن کنترلر SDN یا NOS گفته می‌شود. NOS یک بستر نرم افزاری است که روی یک سرور اجرا شده و براساس یک دید منطقا متمرکز از شبکه، منابع و تجرید لازم را برای تسهیل برنامه نویسی عناصر پیشرانی فراهم می‌کند و از این جهت هدف آن مشابه یک سامانه‌ی عامل سنتی است.

۴- شبکه به واسط برنامه‌هایی که روی NOS در حال اجرا هستند، برنامه پذیر است و NOS با دستگاه های صفحه داده در تعامل است. این یک ویژگی بنیادین SDN است و به عنوان مقصود اصلی آن شمرده می‌شود.

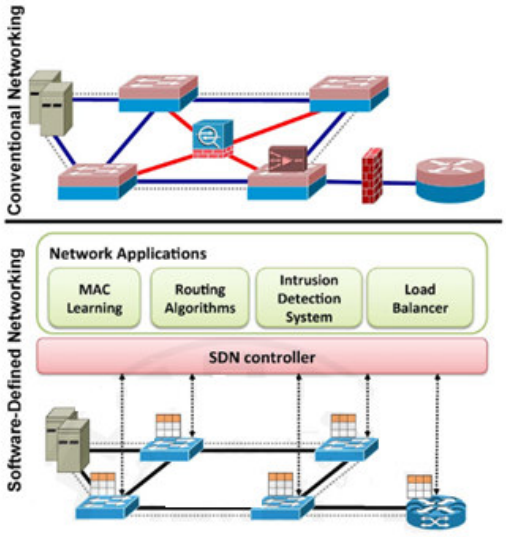
مشاهده می شود که متمرکز کردن منطقی صفحه کنترل به صورت خاص چندین مزیت دیگر نیز ایجاد می‌کند. اولا تغییر سیاست‌های شبکه توسط زبان‌های سطح بالا در مقایسه با تنظیمات وابسته به دستگاه و سطح پایین ساده‌تر است. دوما، یک برنامه کنترلی می‌تواندبه صورت اتوماتیک به تغییرات ناگهانی در وضعیت شبکه پاسخ دهد و از سیاست‌های سطح بالا محافظت نماید. سوما، با داشتن دانش سراسری از وضعیت شبکه، توسعه خدمات و برنامه‌های پیچیده‌تر شبکه آسان می‌شود. بدین ترتیب می‌توان یک شبکه‌ی نرم افزار محور را با سه سطح اساسی تجرید پیشرانی، توزیع و مشخصات تعریف کرد. در واقع تجرید ابزار اساسی پژوهش در علوم کامپیوتر و فناوری اطلاعات است. تجرید پیشرانی باید به برنامه کنترلی مجوز رفتار مطلوبش را بدهد و در عین حال جزئیات سخت افزار زیرین را مخفی کند. اوپن‌فلو نمونه تحقیق چنین تجریدی است. تجرید توزیع باید برنامه های SDN را از دشواری‌های حالت توزیع شده برهاند و مساله‌ی کنترل توزیع شده را به مساله‌ی کنترل منطق متمرکز تبدیل کند. تحقیق چنین تجریدی نیازمند یک لایه‌ی توزیع است که در آن SDN درون NOS قرار دارد و دو عملکرد ضروری دارد: نصب دستورات کنترلی روی دستگاه‌های پیشرانی و جمع‌آوری اطلاعات در مورد وضعیت کنونی لایه‌ی پیشرانی برای اینکه یک دید سراسری از شبکه در اختیار برنامه‌ها بگذارد.

در نهایت تجرید مشخصات باید به برنامه شبکه اجازه دهد رفتار مورد انتظارش از شبکه را بدون اینکه خود مسئول پیاده‌سازی آن باشد، توصیف کند. این کار با راه‌حل های مجازی سازی و زبان‌های برنامه سازی شبکه محقق می‌شود. این روش‌ها تنظیماتی را که برنامه‌ها براساس مدل تجریدی و ساده‌شده از شبکه بیان می‌کند را به تنظیمات فیزیکی برای دید سراسری شبکه که از طریق کنترلر در دسترس است، تبدیل می‌کنند. شکل ۲ معماری، اجزا و مفاهیم SDN را نشان می‌دهد.



شکل ۲. معماری یک شبکه نرم افزار محور و تجریدهای اساسی آن

۵- همانطور که قبلا گفته شد، به هم پیوسته بودن صفحات داده و کنترل در شبکه‌های سنتی، افزون عملکردهای جدید مانند یک الگوریتم مسیریابی جدید را به آن‌ها دشوار می‌کند زیرا صفحه کنترل همه‌ی دستگاه‌های شبکه باید از طریق نصب سیستم عامل و گاه به روزرسانی سخت افزاری تغیر کند. این حقیقت در شکل ۳ نمایش داده شده است.



شکل۳. شبکه سنتی در مقابل SDN

در حال حاضر ویژگی‌های جدید برای شبکه به وسیله تجهیزات گران قیمت، خاص منظوره و با تنظیمات پیچیده‌ای مانندمتوازن‌کننده‌ی بار، سیستم‌های تشخیص نفوذ و فایروال‌ها حاصل می‌شوند که به آن‌ها میان‌افزارهم گفته می‌شود. این ابزارها باید در مکان‌های مناسبی در شبکه قرار داده شوند که این خود تغییر توپولوژی، تنظیمات و عملکرد شبکه را دشوار می‌کند. در مقابل، SDN صفحه کنترل را از دستگاه‌های شبکه جدا می‌کند و به NOS یا کنترلر شبکه تبدیل می‌کند. این کار چند مزیت دارد:

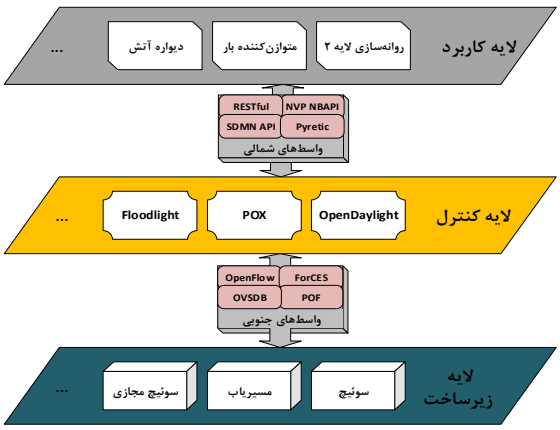
- توسعه‌ی برنامه‌های شبکه آسان‌تر می‌شود زیرا تجرید فراهم شده توسط سکوی کنترل یا زبان‌های برنامه نویسی

می‌تواند به اشتراک گذاشته شود.

- همه برنامه ها می‌توانند از اطلاعات یکسان بهره‌مند شوند (با وجود دیدی سراسری از شبکه)که این منجر به تصمیمات سازگارتر و موثرتر می‌شود.
- این برنامه‌ها می‌توانند از هر نقطه‌ای در شبکه اعمالی مانند بازتنظیم دستگاه ها را انجام دهند. بنابراین برای محل قرارگیری عملکرد جدید در شبکه نیازی به استراتژی پیچیده‌ای نیست.
- یکپارچگی برنامه‌های مختلف آسان‌تر می‌شود، برای مثال توازن بار و مسیریابی می‌توانند به صورت متوالی انجام شوند به صورتی که تصمیمات توازن بار بر سیاست‌های مسیریابی اولویت داشته باشند[۱۲].

معماری و لایه های تشکیل دهنده SDN

در شکل ۴ لایه‌های مختلف تشکیل‌دهنده SDN نشان‌داده شده است. لایه اول، لایه کاربرد نامیده می‌شود که متشکل از برنامه‌هایی است که معمولاً با استفاده از واسط Rest API [9] رفتار شبکه را تعریف می‌کنند. این واسط با استفاده از پروتکل HTTP، برنامه‌های راه دور را قادر به ارسال دستورالعمل‌ها به کنترل‌کننده و یا بازیابی اطلاعات از آن می‌سازد. در لایه کنترل، یک (یا چند) کنترل‌کننده SDN به‌طور منطقی متمرکز وجود دارد که وظیفه رسیدگی به صفحه کنترل و ایجاد یک دید کلی از شبکه را برعهده دارد. وظیفه کنترل‌کننده، ترجمه دستورالعمل‌های برنامه‌های کاربردی برای لایه زیرساخت به‌وسیله یک واسط جنوبی هست. هم‌چنین موظف است برای برنامه‌های کاربردی یک دید



شکل۴. لایه های تشکیل دهنده SDN

بروز از وضعیت شبکه فراهم سازد.

لایه زیرساخت متشکل است از دستگاه‌های شبکه (ازجمله سوئیچ‌ها و مسیریاب‌ها) که وظیفه روانه‌سازی بسته‌ها را برعهده

دارند. ارتباط میان صفحه داده و صفحه کنترل به‌طور متداول از طریق پروتکل اپن‌فلو صورت می‌گیرد.

ویژگی‌های اصلی SDN

شبکه نرم‌افزاری تعریف‌شده یک معماری جدید برای شبکه است که طراحی آن برپایه جداسازی صفحه کنترل از سطح داده و هم‌چنین لایه‌های برنامه پذیر شبکه‌گرا هست. ویژگی‌های مهم SDN در ذیل مطرح شده است:

۱. کنترل مرکزی

با جداسازی صفحه کنترل و صفحه داده SDN، این امکان فراهم می‌شود تا تجهیزات انتقال داده بتوانند بر روی هدایت و هم‌چنین عملکرد خود با سادگی بیشتری تمرکز کنند و صفحه کنترل مرکزی دارای انعطاف‌پذیری و نوآوری بیشتری هست. متمرکزسازی کنترل می‌تواند موجب سادگی در مدیریت اجرای شبکه و ارتقای سرعت پیکربندی آن شده و مزایای دیگری ازجمله ارتقای سریع و نوآوری در شبکه به ارمغان می‌آورد. متمرکزسازی بخش کنترلی شبکه می‌تواند موجب بهینه‌سازی کلی شبکه ازجمله مباحث مربوط به امنیت شود.

۲. باز بودن واسط برنامه‌نویسی نرم‌افزاری

با استفاده از واسط متن‌باز، می‌توان شبکه تجاری اختصاصی شده‌ای برای مشتریان ایجاد نمود. برنامه کاربردی امکان تطبیق شبکه با نیاز مشتریان را فراهم می‌کند.

۳. مجازی‌سازی شبکه

شبکه منطقی از شبکه فیزیکی جداسازی شده تا بتوان از محدود نشدن شبکه منطقی از شبکه فیزیکی اطمینان حاصل نمود. مجازی‌سازی شبکه می‌تواند به سادگی شبکه فیزیکی را به چندین شبکه مجازی تقسیم کند تا امکان تأمین نیازهای مختلف تجاری و ایجاد بستری برای نوآوردی در شبکه فراهم شود.

مولفه‌های اصلی SDN

در یک معماری SDN/Open Flow ، دو مؤلفه اصلی وجود دارد [10]:

تجهیزات روانه‌سازی

کنترل‌کننده‌ها.

یک تجهیز روانه‌سازی، سخت‌افزار یا نرم‌افزاری است که به‌طور اختصاصی وظیفه روانه‌سازی بسته‌ها را بر عهده دارد، درحالیکه کنترل‌کننده، نرم‌افزاری است که بر روی یک پلتفرم سخت‌افزاری مناسب (مانند خدمتگزار یا رایانه شخصی) در حال اجرا هست. در ادامه به تعریف هریک موارد فوق پرداخته شده است.

تجهیزات روانه‌سازی

یک زیرساخت SDN همانند شبکه‌های سنتی دارای مجموعه‌ای از تجهیزات شبکه (ازجمله سوئیچ‌ها، مسیریاب‌ها و جعبه‌های میانی) هست. تنها تفاوتی که در این بین وجود دارد، تبدیل تجهیزات فیزیکی سنتی به عناصر ساده روانه‌سازی هست که این عناصر فاقد بخش کنترلی و یا نرم‌افزاری جهت تصمیم‌گیری‌های خودکار می‌باشند. هوش شبکه از تجهیزات صفحه داده به یک سیستم کنترلی به‌طور منطقی متمرکز انتقال یافته است. این سیستم کنترلی شامل سیستم عامل شبکه و برنامه‌های کاربردی آن هست. به منظور اطمینان از قابلیت همکاری و سازگاری بین انواع مختلف صفحه کنترل و داده، می‌بایست این شبکه‌ها بر روی واسط‌های باز و استانداردی (ازجمله اپن‌فلو) ایجاد شوند. در صورت وجود چنین واسطی، کنترل‌کننده قادر به برنامه‌ریزی تجهیزات روانه‌سازی ناهمگن به‌صورت پویا خواهد بود. این موضوع در شبکه‌های سنتی چالشی اساسی هست که دلیل آن استفاده از تجهیزات شرکت‌های مختلف با واسط‌های بسته و صفحه کنترل توزیع‌شده هست.

یک دستگاه روانه‌سازی مبتنی بر پروتکل اپن‌فلو دارای خط لوله‌ای از جداول جریان است که هر مدخل از این جداول شامل سه بخش هست:

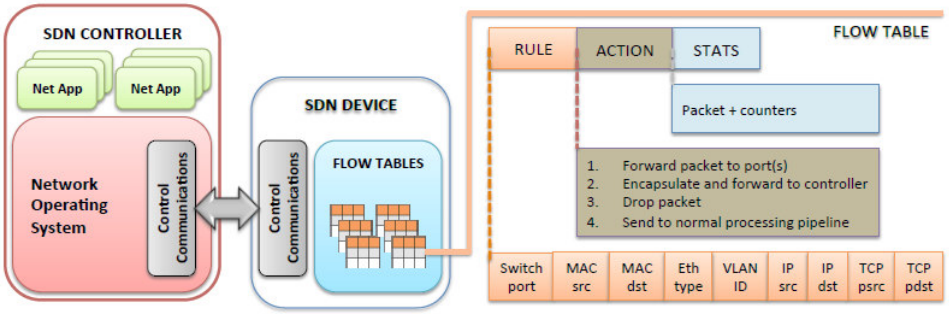
یک قاعده انطباق

یک عملیات که برای بسته‌های انطباق‌یافته صورت می‌پذیرد.

شمارنده‌هایی که آمار بسته‌های انطباق‌یافته را نگهداری می‌کنند.

این مدل سطح بالا از اپن‌فلو در حال حاضر در ساخت و پیاده‌سازی بسیاری از تجهیزات صفحه دادهی SDN بکارگرفته‌شده است.

در یک تجهیز اپن‌فلو، چگونگی رفتار با یک بسته توسط مجموعه‌ای از جداول جریان متوالی مشخص می‌شود. زمانی‌که یک بسته وارد می‌شود، یک فرآیند جستجو از اولین جدول آغاز می‌شود و تا زمانی‌که یک انطباق اتفاق نیفتد (Match) و یا به‌طور قطع قاعده‌ای برای آن بسته یافت نشود (Miss) این روند ادامه می‌یابد.



شکل ۵. یک تجهیز باقابلیت SDN [10].

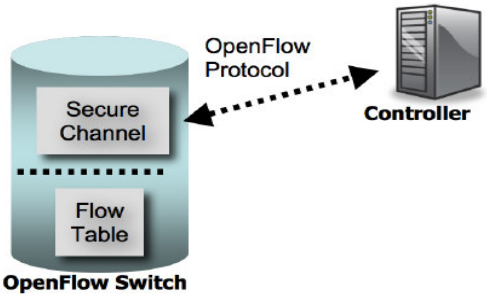
یک قاعده جریان می‌تواند به شکل‌های مختلفی تعریف شود. اگر هیچ قاعده پیش‌فرضی بر روی سوئیچ نصب نشده باشد آن‌گاه بسته دور ریخته خواهد شد. اگرچه به‌طور متداول، یک قاعده پیش‌فرض بر روی سوئیچ نصب خواهد شد که به سوئیچ دستور می‌دهد تمامی بسته‌های دریافتی را به سمت کنترل‌کننده ارسال نماید (و یا به خط لوله معمولی غیر اپن‌فلو موجود در سوئیچ ارسال کند). اولویت‌های این قواعد بر اساس شماره جداول و ترتیب سطرهای جداول جریان هست؛ یعنی ابتدا قواعد موجود در جدول ۰ و سپس قواعد موجود

در جدول ۱ و الی آخر. پس از روی دادن یک انطباق می‌بایست عملیاتی برای آن جریان صورت پذیرد. عملیات (Actions) شامل موارد زیر می‌باشند:

- هدایت بسته به سمت پورت (های) خروجی تعیین شده
- کپسوله و سپس هدایت کردن بسته به سمت کنترل‌کننده
- دور ریختن بسته (Drop)
- ارسال آن به سمت خط لوله عادی (Normal pipeline)
- ارسال آن به جدول جریان بعدی و یا به جداول خاص (مانند جداول گروه و یا جداول اندازه‌گیری) [10].

کنترل‌کننده SDN

کنترل‌کننده همانند یک سیستم عامل شبکه هست که کنترل سخت‌افزار را برعهده گرفته و همچنین مدیریت خودکار شبکه را تسهیل می‌کند. این سیستم عامل، یک واسط قابل‌برنامه‌ریزی متمرکز و یکپارچه را برای تمام شبکه فراهم می‌سازد. همان‌گونه که سیستم عامل موجود بر روی یک رایانه، امکان خواندن و نوشتن را برای برنامه‌های کاربردی فراهم می‌کند، سیستم عامل شبکه نیز قابلیت مشاهده و کنترل شبکه را فراهم می‌سازد؛بنابراین کنترل‌کننده، به تنهایی عمل مدیریت شبکه را انجام نمی‌دهد بلکه صرفاً به عنوان یک واسط قابل‌برنامه‌ریزی هست که امکان مدیریت شبکه را برای نرم‌افزارهای کاربر فراهم می‌کند.



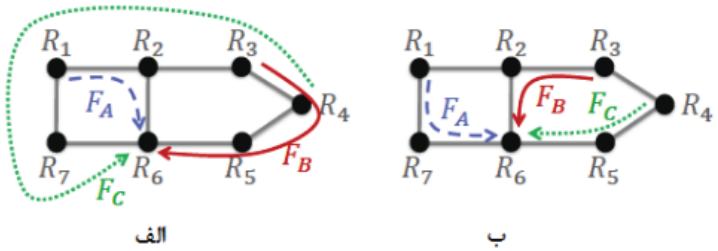
شکل ۶. ارتباط کنترل‌کننده و سوئیچ با استفاده از پروتکل اپن‌فلو [11].

مزایای معماری SDN

برای پرداختن به مزایای معماری مبتنی بر نرم افزار مناسب به نظر می‌رسد سه انتزاع موجود در این معماری معرفی شود[۳]. انتزاع به معنی پنهان‌سازی برخی جزییات یک سیستم از دید سیستم دیگر است. انتزاع جزییات سخت‌افزار، به معنی پنهان‌سازی جزییات سخت‌افزاری سوییچ از دید لایه کنترلی است. اپن‌فلو یک نمونه از تحقق این انتزاع است. انتزاع توپولوژی شبکه، به معنای آن است که کنترلر توزیع شده بودن شبکه را از دید برنامه‌ها پنهان سازد. به بیان دیگر برنامه‌ها گمان می‌کنند با یک موجودیت واحد سروکار دارند. انتزاع ویژگی‌ها ،این گونه توضیح داده می‌شود که برنامه‌ها برای تعیین سیاست‌ها نیاز

نداشته باشند جزییات پیاده سازی را بیان نمایند. برنامه‌ها سیاست‌هایشان را به زبان سطح بالا مطرح ساخته، و کنترلر وظیفه دارد آن‌ها را به دستورات قابل اجرا بر روی سوییچ‌ها ترجمه نماید. معادل موضوع فوق در دنیای کامپیوتر این وضعیت متداول است که برنامه نویسان از یک سری دستورات سطح بالا برای نوشتن برنامه بهره می‌برند، حال آن که مفسر دستورات را به زبان ماشین ترجمه می‌سازد. حال که سه انتزاع موجود در SDN معرفی شد، به مزایای بالقوه این معماری اشاره می‌گردد. لازم به ذکر است ممکن است تمام مزایای زیر در یک شبکه مبتنی بر نرم افزار وجود نداشته باشد، چرا که نحوه پیاده سازی نیز نقش مهمی ایفا می‌نماید.

- وابسته نبودن به تولیدکننده خاص: انتزاع جزییات سخت‌افزار مدیران شبکه را از وابستگی به یک تولیدکننده خاص رها می‌سازد. هر سوییچی که بتواند بر اساس استاندارد موجود (عمدتاً اپن‌فلو) با کنترلر ارتباط برقرار سازد، می‌تواند در شبکه به کار گرفته شود. مزایای اقتصادی چنین رویکردی آشکار است.
- نوآوری در لایه کنترلی بدون نیاز به تغییر سخت‌افزار: کمی قبل اشاره نمودیم که با جداکردن لایه کنترلی از لایه داده، به لایه‌ی کنترلی اجازه می‌دهیم که مستقل از لایه‌ی داده به رشد و پیشرفت خود ادامه دهد. ارایه طرح‌های مختلف با منطق کنترلی متفاوت توسط پژوهشگران، که همگی بر روی یک سخت‌افزار یکسان قابل اجرا هستند، گواهی بر این موضوع است.
- ارزان قیمت شدن تجهیزات شبکه: حذف لایه کنترلی از سوییچ قیمت آن را کاهش می‌دهد. در شبکه‌های مرکز داده که تعداد زیادی سوییچ وجود دارد، این کاهش قیمت ابعاد بزرگی می‌یابد. البته نصب کنترلر نیز هزینه بر است، اما در شبکه‌هایی نظیر شبکه مرکز داده یک کنترلر برای مدیریت صدها و یا هزاران سوییچ کافی است.
- نگاه جامع به شبکه: انتزاع توپولوژی شبکه، که موجب می‌گردد برنامه‌ها کل شبکه را به عنوان یک موجودیت واحد ببینند، منجر به سادگی و بهینه‌تر بودن برنامه‌ها می‌گردد. یک مثال که در [۱۱] آورده شده است، تاثیر نگاه جامع به شبکه را به خوبی نشان می‌دهد. در شکل الف جریان‌های F_A، F_B و F_C به ترتیب حروفشان وارد شبکه شده‌اند و بر اساس کوتاهترین مسیر (در لحظه ورود) به شبکه راه یافته‌اند. این وضعیت غیر بهینه است و اگر نگاه کلانی وجود داشته باشد شکل ب ایجاد می‌شود. در این شکل جریان F_A در هنگام ورود F_B تغییر مسیر داده می‌شود. نگاه کلان بدین معنا است که لایه کنترلی جریان‌ها را بر اساس وضعیت کل شبکه (و نه وضعیت بخشی از آن) مسیریابی می‌نماید.



شکل ۷. دو وضعیت غیربهینه (الف) و بهینه(ب) در مسیریابی جریان‌ها[۱۱]

یکپارچه سازی برنامه‌ها : استقرار تمامی برنامه‌ها بر روی کنترلر، یکپارچه سازی عملکرد آن‌ها را موجب می‌شود. در شبکه‌های فعلی چنین وضعیتی وجود نداشته، برای مثال ممکن است عملکرد ماژول مهندسی ترافیک در عملکرد

فایروال اختلال ایجاد نماید. در SDN این تضاد با تقدم بخشیدن به تصمیمات فایروال به سادگی قابل حل است. به بیان دیگر تنها یک سیاست یکپارچه بر شبکه اعمال می‌شود که ترکیبی مدیریت شده از تصمیمات ماژول‌های مختلف است.

سادگی برنامه‌ها : انتزاع ویژگی‌ها به برنامه نویسان شبکه کمک می‌کند بر روی الگوریتم و منطق کار تمرکز نمایند و درگیر پیچیدگی‌های مربوط به شبکه نشوند. برای مثال برنامه می‌تواند مسیر جریانی را به صورت یک مجموعه از شناسه‌ی چندین سوییچ تعیین نماید، در حالی که کنترلر این تصمیم را با پیغام های جداگانه به سوییچ‌های مورد نظر ابلاغ می‌کند. به بیان کلی برنامه‌ها تنها یک مدل ساده شده از شبکه را می‌بینند، و این کار با مجازی سازی یا زبان های برنامه نویسی شبکه امکانپذیر است.

چالشهای SDN

چالشهای زیادی در ارتباط با معماری SDN وجود دارد که در سالیان گذشته تعداد زیادی مقاله برای حل مسائل یا کاهش ابعاد آن‌ها منتشر شده است. در ادامه به سه مورد از مهمترین چالشها پرداخته می‌شود.

مقیاس پذیری: یکی از چالش های SDN مسئله مقیاس پذیری آن است. در این ساختار هر کنترلر باید تعدادی سوییچ را مدیریت نماید، پس با بزرگتر شدن ابعاد شبکه دچار مشکل می‌شویم. به عبارتی با افزایش تعداد سوییچ‌ها فشار بر روی کنترلرهای موجود افزایش می‌یابد، مگر این که تعداد کنترلرها را زیاد نماییم. افزایش تعداد کنترلرها نیز مشکلات دیگری را پدید می‌آورد، از جمله احتمال ناسازگاری بین تصمیمات گرفته شده توسط آن‌ها.

امنیت: مسئله دیگر در ارتباط با SDN آسیب پذیری آن در برابر حملات می‌باشد. در این معماری چنانچه یک نفوذگر کنترل یک کنترلر یا کانال ارتباطی میان کنترلر و سوییچ را در دست بگیرد، آسیب‌های زیادی را می‌تواند وارد سازد. همچنین برنامه‌هایی که بر روی کنترلر نصب می‌شوند ممکن است قوانین امنیتی را نقض کنند. مسائل مربوط به امنیت SDN در [۱۲]بحث شده است.

اندازه جدول جریان: از آن جا که در SDN مسیریابی بر پایه مفهوم جریان (فلو) انجام می‌شود (و نه آدرس مقصد)، افزایش تعداد جریان‌ها می‌تواند فراتر از ظرفیت سوییچ‌ها باشد. این مسئله به دو شکل کلی حل می‌شود. یکی افزایش ظرفیت سوییچ‌ها، و یا تجمیع سازی جریان‌ها. در روش دوم تعدادی جریان با مشخصات نزدیک به هم توسط یک قاعده مسیریابی می‌شوند.

مروری بر استاندارد اپن‌فلو

در این زیربخش در ابتدا سوئیچ‌ها و سپس پیام‌های پروتکل اپن‌فلو توضیح داده خواهد شد.

سوئیچ اپن‌فلو

یک سوئیچ اپن‌فلو [11] شامل جدول جریانی هست که ارسال بسته‌ها بر طبق آن صورت می‌گیرد و هم‌چنین هر سوئیچ اپن‌فلو دارای یک کانال امن ارتباطی با کنترل‌کننده هست. کنترل‌کننده سوئیچ را از طریق این کانال ارتباطی مدیریت می‌کند. مدیریت سوئیچ‌ها با استفاده از پروتکل اپن‌فلو انجام می‌شود.

هر جدول جریان شامل چندین مدخل جریان، شمارنده‌ها و مجموعه‌ای از عملیات برای اعمال روی بسته‌های ورودی هست. تمامی بسته‌های وارد شده به سوئیچ با مدخل‌های جدول جریان مقایسه می‌شوند، اگر مدخلی با بستهی ورودی مطابقت داشته باشد (مقادیر فیلدهای سرآیند بسته ورودی با مقادیر فیلدهای مدخلی از جدول جریان یکسان باشد) عملیات مشخص شده در این مدخل روی بسته اعمال می‌شوند. برای مثال این عملیات ممکن است ارسال بسته از پورت خاصی باشد. ولی اگر مدخلی با بسته ورودی مطابقت نداشته باشد، بسته به کنترل‌کننده ارسال می‌شود. کنترل‌کننده مسئولیت رسیدگی به بسته‌هایی که سوئیچ نمی‌داند چگونه با آن‌ها رفتار کند را برعهده دارد. کنترل‌کننده این وظیفه را از طریق اضافه و حذف کردن مدخل‌های جدول جریان انجام می‌دهد. هر مدخل جدول جریان دارای ساختاری مشابه شکل زیر هست:

عملیات	شمارنده	فیلدهای سرآیند

شکل ۸. ساختار مدخل جدول جریان [11].

فیلدهای سرآیند با مقادیر فیلدهای سرآیند بسته‌های ورودی تطابق داده می‌شوند.
شمارنده‌ها برای شمارش بسته‌هایی می‌باشند که با هر مدخلی انطباق پیداکرده‌اند.
عملیات، نحوهی رفتار سوئیچ با بسته‌های تطابق یافته را مشخص می‌کند.

فیلدهای سرآیند بسته‌ها برای انطباق با مدخل جدول جریان

جدول ذیل، فیلدهایی از سرآیند یک بسته را نشان می‌دهد که مقادیر آن‌ها با مقادیر همان فیلدها روی سرآیند بسته ورودی مطابقت داده می‌شوند. برای هر بسته ورودی لازم نیست که تمام این فیلدها بررسی شوند بدین معنا که هرکدام از این فیلدها می‌توانند دارای یک مقدار خاصی باشند که مطابقت بسته ورودی با آن مقدار خاص بررسی می‌شود و یا اینکه دارای مقدار ANY باشند که با هر مقداری مطابقت می‌یابد. برای مثال اگر مدخلی روی جدول جریانی نصب شود که مقادیر تمام فیلدهای آن ANY باشند تمامی بسته‌های ورودی با آن مدخل مطابقت خواهند داشت. فیلدهایی که تطابق آن‌ها بررسی می‌شود در جدول ذیل آورده شده‌اند:

جدول ۱. فیلدهایی از سرآیند بسته که برای انطباق با مدخل‌های جدول جریان استفاده می‌شوند [1].

TCP/ UDP dst port	TCP/ UDP src port	IP protocol	IP dst	IP src	VLAN ID	Ether type	Ethr dst	Ethr source	Ingress Port

شمارنده‌های مدخل جدول جریان سوئیچ اپن‌فلو

شمارنده‌ها در سطح جدول، جریان، پورت و صف نگهداری می‌شوند. جدول ۲، مجموعه شمارنده‌ها را نشان می‌دهد. منظور از زمان در این جدول، طول عمر یک جریان یعنی زمانی که مدخل مربوط به جریان روی سوئیچ نصب هست، است.

جدول ۲. شمارنده‌های پیام‌های آماری پروتکل اپن‌فلو [11].

Counter	Bits
Per Table	
Active entries	32
Packet lookups	64
Packet matches	64
Per flow	
Received packets	64
Received bytes	64
Duration	32
Per port	
Receive packets	64
Transmitted packets	64
Received bytes	64
Transmitted bytes	64
Receive drops	64
Transmitted drops	64
Receive errors	64
Transmitted errors	64
Receive frame Alignment errors	64
Receive overrun errors	64
Receive CRC errors	64
Collisions	64

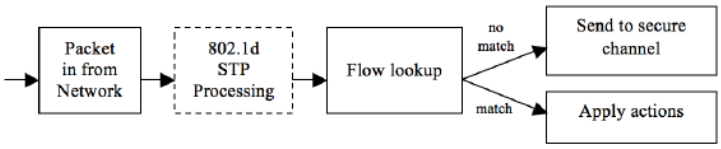
عملیات مدخل جدول جریان سوئیچ اپن‌فلو

هر مدخل جدول جریان شامل صفر یا تعداد بیشتری از عملیات است که نشان می‌دهد سوئیچ با بسته‌های انطباق یافته با آن مدخل چگونه رفتار کند. اگر مدخلی دارای هیچ عملیات ارسال بسته‌ای نباشد بسته رسیده حذف می‌شود. ترتیب اجرای عملیات مربوط به هر مدخلی باید به همان ترتیب نصب شده باشد. اگر سوئیچی نتواند عملیات مشخص شده با ترتیبی خاص را پردازش کند، آن مدخل را رد کرده و فوراً یک پیام خطا را برمی‌گرداند. ترتیب عملیات می‌تواند بین سازندگان مختلف متفاوت باشد. هر سوئیچی نیاز به پشتیبانی از تمام انواع عملیات را ندارد بلکه تمامی سوئیچ‌ها تنها ملزم به پشتیبانی "عملیات ضروری" می‌باشند. زمانی که سوئیچ به

کنترل‌کننده وصل می‌شود مشخص می‌کند که کدام یک از "عملیات اختیاری" را پشتیبانی می‌کند. سوئیچ‌های سازگار با پروتکل اپن‌فلو دو نوع می‌باشند: Open Flow-only و Open Flow-enabled. سوئیچ‌های Open Flow-only تنها "عملیات ضروری" را پشتیبانی می‌کنند ولی سوئیچ‌های Open Flow-enabled علاوه بر عملیات ضروری از عملیات "عادی" و "سیل‌آسا" هم پشتیبانی می‌کنند.

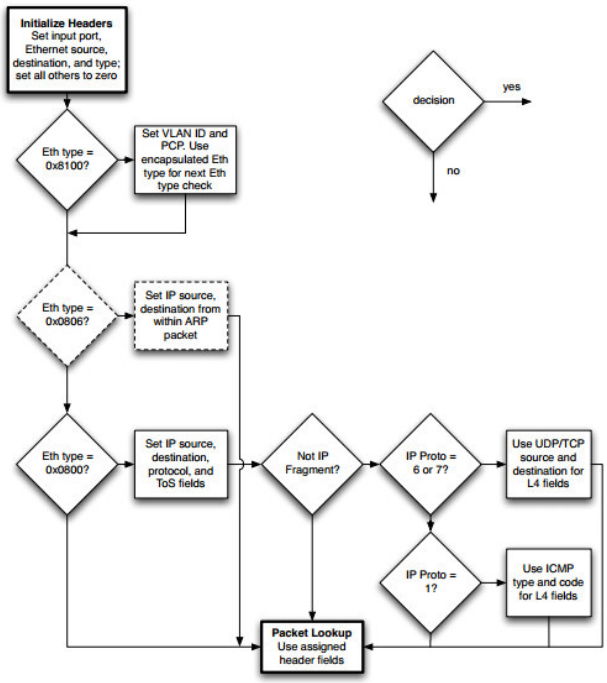
انطباق بسته ورودی

به محض ورود هر بسته‌ای، سوئیچ اپن‌فلو عملیاتی با ساختار نشان داده شده در شکل ۸ را اجرا می‌کند. فیلدهایی از سرآیند بسته که برای جست‌وجو استفاده می‌شوند به نوع بسته بستگی دارند. موارد مربوط به آن در شکل ۹ آمده است:



شکل ۹. عملیات اجرایی روی بسته‌های ورودی سوئیچ‌های اپن‌فلو [11].

اگر مقادیر فیلدهای سرآیند بسته با مقادیر مشخص شده در مدخل جدول جریان یکی باشند بسته با آن مدخل جدول جریان انطباق پیدا می‌کند. اگر فیلدی شامل مقدار ANY باشد با هر مقداری انطباق می‌یابد. هر بسته‌ای که با مدخلی از جدول جریان انطباق می‌یابد، شمارنده مربوط به آن مدخل به‌روزرسانی می‌شود. اگر بسته‌ای با هیچ‌کدام از مدخل‌های موجود انطباق نیابد از طریق کانال امنی به کنترل‌کننده ارسال می‌شود زیرا سوئیچ نمی‌داند چه رفتاری در قبال آن بسته انجام دهد.



شکل ۱۰. نحوه بررسی انطباق بسته ورودی با مدخل‌های جدول جریان [11].

کانال امن رابطی است که هر سوئیچ اپن‌فلو را به کنترل‌کننده وصل می‌کند. کنترل‌کننده از طریق این کانال امـن سـوئیچ‌هـا را مدیریت و پیکربندی می‌کند. رخدادها را از سوئیچ‌ها دریافت می‌کند و بسته‌ها را به سوئیچ‌ها ارسال می‌کند. ارتباط بین کنترل‌کننـده و سوئیچ از طریق یک نشست TLS برقرار می‌شود. به‌صورت پیش‌فرض این نشست روی پورت ۶۶۳۳ TCP برقرار می‌شود.

مراجع

[1] [Online] www.youtube.com/yt/press/statistics.html.

[2] Fundation, O. N. (2012). Software-defined networking: The new norm for networks. ONF White Paper.

[3] Kreutz, D., Ramos, F. M., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. proceedings of the IEEE, 103(1), 14-76.

[4] Celenlioglu, M. R., Alsadi, M., & Mantar, H. A. (2015, July). Design, implementation and evaluation of SDN-based resource management model. In New Technologies, Mobility and Security (NTMS), 2015 7th International Conference on (pp. 1-5). IEEE.

[5] Casado, M., Koponen, T., Shenker, S., & Tootoonchian, A. (2012, August). Fabric: a retrospective on evolving SDN. In Proceedings of the first workshop on Hot topics in software defined networks (pp. 85-90). ACM.

[6] Minei, I., & Lucek, J. (2010). MPLS-enabled applications: emerging developments and new technologies. John Wiley & Sons.

[7] Das, S., Sharafat, A., Parulkar, G., & McKeown, N. (2011, March). MPLS with a simple OPEN control plane. In Optical Fiber Communication Conference (p. OWP2). Optical Society of America.

[8] " OpenFlow Switch. "v1. 3.0"," 2012.

[9] [Online] openvswitch.org.

[10] [Online] http://osrg.github.io/ryu

[11] Hong, C. Y., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M., & Wattenhofer, R. (2013, August). Achieving high utilization with software-driven WAN. In ACM SIGCOMM Computer Communication Review (Vol. 43, No. 4, pp. 15-26). ACM.

[12] Scott-Hayward, S., O'Callaghan, G., & Sezer, S. (2013, November). Sdn security: A survey. In Future Networks and Services (SDN4FNS), 2013 IEEE SDN For (pp. 1-7). IEEE.

[13] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., ... & Turner, J. (2008). OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38(2), 69-74.