علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۱۸، شماره ۲، ۱۳۹۹
صفحات ۵۰-۵۹
مقاله عادی

نشریه علوم رایانش و فناوری اطلاعات

ارائه روشی جهت مدیریت ترافیک شبکه‌های گسترده بر اساس شبکه‌های‌نرم‌افزارمحور

محمد دلخوش[۱*]، رضا جاویدان[۲]

* محمد دلخوش، دریافت: ۱۴۰۰/۰۱/۲۳، بازنگری: ۱۴۰۰/۰۱/۳۱، پذیرش: ۱۴۰۰/۰۲/۰۸

[۱] کارشناسی ارشد مهندسی فناوری اطلاعات، دانشگاه صنعتی شیراز، شیراز، ایران
[۲] دانشیار، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی شیراز، شیراز، ایران

چکیده

شبکه‌های‌نرم‌افزارمحور به دلیل ماهیت متمرکز، دچار محدودیت‌های زیادی در قابلیت عملکرد خود در شبکه‌های‌گسترده می‌شوند و در نتیجه مقیاس‌پذیری این نوع شبکه‌ها با نوعی چالش مواجه است. طراحی و پیاده‌سازی شبکه‌های‌نرم‌افزارمحور با استفاده از چندکنترلر، یک راه‌حل مقیاس‌پذیر ارائه می‌دهد که می‌تواند برای بهبود مدیریت ترافیک و عملکرد در شبکه‌های‌گسترده مورد استفاده قرار گیرد. در این مقاله یک شبکه‌نرم‌افزارمحور چندکنترلر بر اساس مدل افقی پیشنهاد شده است که با توجه به قابلیت توسعه تدریجی و قابلیت سازگاری رو به عقب و با استفاده از پروتکل BGP برای ارتباط بین دامنه‌ای، بهبود مقیاس‌پذیری شبکه و بهبود قابلیت نصب و پیکربندی سریع دامنه‌ها در شبکه را موجب می‌شود. با این حال این روش با چالش محدودیت مدیریت و نظارت سرتاسری و یکپارچه بر دامنه ها مواجه است. بنابراین جهت مدیریت و نظارت بر طرح چندکنترلری در شبکه‌های‌گسترده، روشی بر اساس مدل سلسله مراتبی نیز ارائه شده است. در این روش ارتباط کنترلرهای هر دامنه با رابط کاربری مدیریتی از طریق REST در واسط شمالی هر کنترلر انجام می‌گیرد. همچنین برای افزایش بهره‌وری در هر دامنه شبکه‌های‌نرم‌افزارمحور از روش مهندسی ترافیک استفاده می شود که بر اساس پهنای باند هر لینک بهترین مسیر را انتخاب می‌کند. نتایج پیاده‌سازی بیان‌گر تضمین ارتباط بین دامنه‌ای از طریق BGP، اثربخش بودن روش مهندسی ترافیک و کارآمدی مدیریت و نظارت بر چندکنترلری در شبکه‌های‌گسترده از طریق رابط‌کاربری پیشنهادی می‌باشد.

کلمات کلیدی: مدیریت شبکه، شبکه‌های‌نرم‌افزارمحور چندکنترلر، مهندسی ترافیک، شبکه‌های گسترده، ONOS.

۱- مقدمه

شبکه‌های نرم‌افزار محور معماری جدیدی است که سطح کنترل[۱] را از سطح داده[۲] جدا می‌کند و یک شبکه قابل‌برنامه‌ریزی به وجود می‌آورد. درحالی‌که این اصطلاح نسبتاً جدید است اما تاریخچه شبکه‌های نرم‌افزار محور را می‌توان در ریشه‌ی مهندسی ترافیک و مکانیسم‌های کنترل شبکه‌های مختلف که طی سال‌ها توسعه‌یافته‌اند، پیدا کرد [۱] [۲]. استفاده از شبکه‌های نرم‌افزار محور در مقیاس بزرگ باعث بهبود مدیریت، کنترل مؤثر و افزایش کارایی می‌شود [۳]. دو روش برای توسعه شبکه‌های نرم‌افزار محور در شبکه‌های گسترده[۳] وجود دارد که عبارت‌اند از کنترلر واحد و چندکنترلر. روش کنترلر واحد با توجه به محدودیت ایجاد تنها نقطه شکست و کاهش کارایی برای اجرا در شبکه‌های گسترده مناسب نخواهد بود [۴]؛ بنابراین راه‌حل امیدوارکننده چندکنترلر است [۵]. بااین‌حال طراحی شبکه با کنترلرهای متعدد یا چندکنترلری پیچیدگی شبکه را افزایش داده و چالش‌های بسیاری را در مدیریت کارآمد زیرساخت شبکه ایجاد می‌کند.

در شبکه‌های نرم‌افزار محور چندکنترلر دو معماری ارائه‌شده است که عبارت‌اند از معماری متمرکز و معماری کاملاً توزیع‌شده یا چند دامنه [۵] [۶]. معماری چند دامنه شبکه را به دامنه‌های مختلف تقسیم می‌کند و به هر کنترلر این امکان را می‌دهد تا بتواند دامنه خود را به‌طور مستقل مدیریت کند و با سایر دامنه‌ها ارتباط داشته باشد. در تحقیقات اخیر برای معماری چندکنترلر کاملاً توزیع‌شده یا چند دامنه دو مدل افقی و عمودی پیشنهاد شده است. استفاده از

مدل افقی در معماری چند دامنه باعث نصب و پیکربندی سریع دامنه‌ها و بهبود مقیاس‌پذیری در شبکه می‌شود.

تعداد زیادی از مطالعات این مدل را برای بهبود قابلیت مقیاس‌پذیری و گسترش سطح کنترل در شبکه‌های نرم‌افزار محور پیشنهاد داده و به کار گرفته‌اند [۷]. مدل افقی در معماری شبکه‌های نرم‌افزار محور چند دامنه به پروتکل‌های ارتباطی مورد استفاده در بین دامنه‌ها وابسته است. پروتکل ارتباط بین دامنه‌ای شبکه‌های نرم‌افزار محور با استفاده از واسط شرقی و غربی کنترلر قابل‌اجرا است. وظیفه اصلی این پروتکل ایجاد ارتباط بین کنترلرها در دامنه‌های مختلف و تبادل اطلاعات کنترلی، خدمات و برنامه‌ها است. در حال حاضر API واسط شرقی و غربی استاندارد نشده است که می‌تواند به چالش قابلیت همکاری در توسعه شبکه‌های نرم‌افزار محور چند دامنه منجر شود.

هدف اصلی این مقاله این است که چگونه می‌توان زیرساخت شبکه‌ای متشکل از چندین دامنه شبکه‌های نرم‌افزار محور مجزا را در شبکه‌های گسترده توسعه داد و آن‌ها را مدیریت کرد. بنابراین برای دستیابی به هدف، شبکه توزیع‌شده با چندین دامنه مبتنی بر شبکه‌های نرم‌افزار محور پیاده‌سازی خواهد شد. با توجه به توسعه تدریجی شبکه‌های نرم‌افزار محور و قابلیت سازگاری رو به عقب پروتکل مورداستفاده در ارتباط بین دامنه‌ای شبکه‌های نرم‌افزار محور باید بتواند با شبکه‌های IP سازگار باشد. بنابراین در این مقاله از پروتکل BGP برای ارتباط بین دامنه‌ای استفاده می‌شود. مدل افقی در معماری چند دامنه علی‌رغم وجود مزایای اشاره‌شده، امکان مدیریت و نظارت متمرکز بر دامنه‌ها را ندارد. به‌منظور رفع این مشکل، روشی بر اساس مدل سلسله مراتبی جهت مدیریت و نظارت بر چندکنترلری در شبکه توزیع‌شده ارائه می‌گردد. مدل سلسله مراتبی مناسب‌ترین راه‌حل برای مدیریت شبکه‌های نرم‌افزار محور توزیع‌شده است که باعث مدیریت آسان شبکه می‌شود. روش‌های مهندسی ترافیک برای بهبود عملکرد شبکه، در دسترس بودن، به حداقل رساندن تأخیر و جلوگیری از ازدحام شبکه استفاده می‌شوند، بر همین اساس روشی برای افزایش بهره‌وری در شبکه هر دامنه (مبتنی بر شبکه‌های نرم‌افزار محور) ارائه خواهد شد.

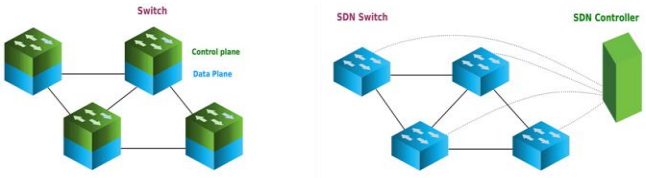
در این مقاله تعاریف پژوهش در بخش دوم ارائه می‌شود. سپس در بخش سوم تحقیقات پیشین مرتبط با روش‌های پیشنهادی ارائه خواهند شد. در بخش چهارم تکنیک‌ها و روش‌های پیشنهادی در این مقاله به‌طور کامل بیان می‌شوند. در فصل پنجم نتایج به‌دست‌آمده از شبیه‌سازی‌ها و پیاده‌سازی‌های مرتبط با روش‌های پیشنهادی مورد ارزیابی قرار خواهند گرفت. در فصل ششم نتیجه‌گیری کلی مقاله ارائه می‌گردد.

۲- شبکه‌های نرم‌افزارمحور

با توجه به اینکه شبکه‌ها ازنظر اندازه و نیازمندی‌ها بسیار بزرگ‌شده‌اند، انتقال ترافیک در سوئیچ‌های سخت‌افزاری به یک کار پیچیده تبدیل‌شده است. حتی تنظیم دستی سوئیچ‌های نرم‌افزاری برای سازمان‌هایی که دارای محیط‌های کاملاً مجازی برای شبکه‌های بزرگ خود هستند نیز به یک کار پیچیده و مستعد خطا تبدیل‌شده است. پیکربندی خودکار و مکانیسم‌های پاسخ به‌طور بی‌درنگ[۴] در شبکه‌های IP فعلی وجود ندارد. بنابراین اجرای سیاست‌های موردنیاز در چنین محیط‌های پویا بسیار چالش‌برانگیز است. علاوه بر این شبکه‌های فعلی به‌صورت عمودی یکپارچه‌شده است و سطح کنترل و سطح داده در داخل دستگاه‌های شبکه قرار می‌گیرند که باعث کاهش انعطاف‌پذیری می‌شود.

شبکه نرم‌افزار محور یک رویکرد جدید برای دنیای فعلی شبکه است که باعث غلبه بر محدودیت‌های زیرساخت در شبکه‌های فعلی می‌شود. شبکه‌های نرم‌افزار محور با جدا کردن منطق کنترل شبکه (سطح کنترل) از ارسال ترافیک (مسیریاب‌های[۵] اصلی و سوئیچ‌ها در سطح داده) مشکل یکپارچه‌سازی عمودی را برطرف می‌کند. این تفکیک سطح کنترل از سطح داده باعث می‌شود، سوئیچ‌های شبکه یک دستگاه انتقال ساده باشند و منطق کنترل در یک کنترلر متمرکز منطقی انجام شود و اجرای سیاست‌ها، تنظیم مجدد شبکه و تکامل شبکه را ساده کند [۸].

رسالت اصلی شبکه‌های نرم‌افزار محور تسهیل در نوآوری، برنامه‌ریزی و کنترل شبکه است. همان‌طور که در شکل ۱ مشاهده می‌شود، با جداسازی منطق کنترلی از سخت‌افزارهای ارسال‌کننده پیاده‌سازی و اجرای پروتکل‌ها و الگوریتم‌های مسیریابی و مدیریت شبکه به‌مراتب ساده‌تر از قبل شده است. چراکه علاوه بر در اختیار داشتن واسط‌هایی با سطح انتزاع بالا، مدیران و اپراتورهای شبکه از مزیت برنامه‌ریزی تعداد زیادی از تجهیزات شبکه به‌صورت متمرکز و کاهش چشمگیر دوباره‌کاری‌ها بهره‌مند خواهند شد [۲].

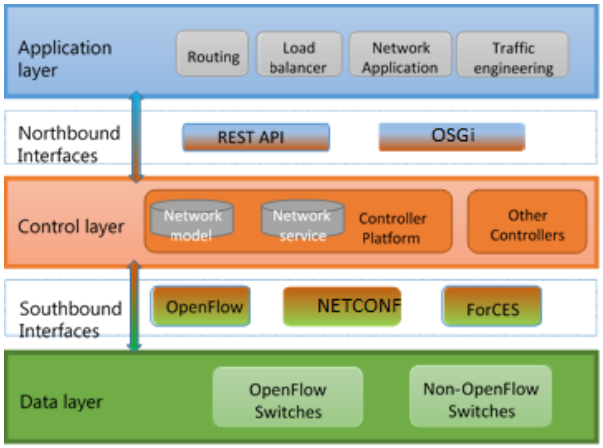


شکل ۱- شبکه‌های نرم‌افزار محور و شبکه‌های سنتی [۹]

معماری شبکه‌های نرم‌افزار محور فرآیندهایی نظیر پیکربندی، تخصیص منابع، اولویت‌بندی ترافیک و حمل‌ونقل ترافیک را به سه سطح اساسی زیر تقسیم می‌کند:

۱) سطح داده
۲) سطح کنترل
۳) سطح کاربرد[۶]

کلیه مؤلفه‌های معماری شبکه‌های نرم‌افزار محور در ادامه شرح داده می‌شود:



شکل ۲- معماری شبکه‌های نرم‌افزار محور [۱۰]

سطح داده: سطح داده مجموعه‌ای از اجزاء شبکه است که می‌تواند سوئیچ‌ها، مسیریاب‌ها، تجهیزات مجازی شبکه، فایروال‌ها و غیره باشد. هدف سطح داده این است که ترافیک شبکه را تا آنجا که ممکن است بر اساس مجموعه مشخصی از قوانین ارسال که توسط سطح کنترل دستور داده‌شده است، هدایت کند. ارتباط بین سطح داده و سطح کنترل توسط APIهای واسط جنوبی حاصل می‌شود. در حال حاضر پروتکل OpenFlow به‌عنوان یک پروتکل برجسته ارتباطی واسط جنوبی موردحمایت بسیاری از توسعه‌دهندگان ازجمله ONF قرارگرفته است [۱۰].

سطح کنترل: سطح کنترل وظیفه تصمیم‌گیری در مورد نحوه عبور و مرور ترافیک شبکه بر اساس نیازمندی‌های کاربر نهایی و ارسال سیاست‌های شبکه

تعیین‌شده به سطح داده را بر عهده دارد. مؤلفه اصلی سطح کنترل، کنترلر است. کنترلر شبکه‌های نرم‌افزار محور نیازمندی‌های برنامه‌ها و اهداف تجاری مانند نیاز به اولویت‌بندی ترافیک، کنترل دسترسی، مدیریت پهنای باند، QoS و غیره را در قوانین ارسال مربوط به اجزای سطح داده در نظر می‌گیرد. با توجه به‌اندازه شبکه می‌تواند در سطح کنترل بیش از یک کنترلر وجود داشته باشد. با معرفی قابلیت برنامه‌نویسی شبکه از طریق سطح کنترل امکان دست‌کاری جداول جریان در عناصر جداگانه بر اساس عملکرد شبکه و نیازهای سرویس در زمان اجرا امکان‌پذیر می‌شود [۱۰] [۸] [۱۱].

سطح کاربرد: سطح کاربرد بالاترین سطح معماری شبکه‌های نرم‌افزار محور است. این لایه از طریق APIهای واسط شمالی (به‌عنوان‌مثال RESTful) با کنترلرهای سطح متوسط شبکه‌های نرم‌افزار محور ارتباط برقرار می‌کند. این سطح شامل برنامه‌های کاربردی و تجاری شبکه است. دید انتزاعی از زیرساخت شبکه از طریق واسط شمالی کنترلر به برنامه‌ها ارائه می‌شود. سطح انتزاع ممکن است شامل پارامترهای شبکه مانند تأخیر، گذردهی و در دسترس بودن باشد که به برنامه‌های کاربردی دید گسترده‌ای از شبکه می‌دهد. سرویس‌ها و برنامه‌های شبکه‌های نرم‌افزار محور می‌توانند از API واسط شمالی برای انتقال قوانین جریان به دستگاه‌های ارسال‌کننده استفاده کنند.

کانال‌های ارتباطی: با استفاده از واسط جنوبی[۷] و واسط شمالی[۸] ارتباط میان سطح‌ها امکان‌پذیر می‌شود. واسط جنوبی بین سطح داده و سطح کنترل و واسط شمالی بین سطح کنترل و سطح کاربرد است. واسط شرقی و غربی ارتباط بین کنترلرهای شبکه‌های نرم‌افزار محور را در یک شبکه توزیع‌شده فراهم می‌کنند. در ادامه به بررسی جزئیات این کانال‌های ارتباطی پرداخته می‌شود.

چارچوب شبکه‌های نرم‌افزار محور با استفاده از پیوند بین سطح داده و سطح کنترل عناصر انتقال ترافیک را مدیریت می‌کند، بنابراین این پیوند باید در دسترس بودن و امنیت زیادی را ارائه دهد. یکی از پروتکل‌های اصلی واسط جنوبی پروتکل OpenFlow است. این پروتکل ارتباط بین سوئیچ‌ها و کنترلر را با استفاده از TLS و تبادل گواهینامه بین سوئیچ‌ها و کنترلر ارائه می‌دهد [۱۲] [۱۳].

علاوه بر ارتباط سوئیچ به کنترلر برنامه‌های کاربردی و سرویس‌های خارجی ممکن است به اطلاعات مربوط به توپولوژی شبکه، قابلیت‌های دستگاه و شبکه برای مدیریت انتقال ترافیک و تعیین سیاست‌های شبکه نیاز داشته باشند. برخلاف پروتکل‌های ارتباطی سوئیچ به کنترلر هیچ رابط برنامه‌نویسی استانداردی برای واسط شمالی وجود ندارد و با توجه به پشتیبانی و سازگاری کنترلر راه‌حل‌هایی به‌صورت موقت انجام می‌شود. معماری و APIهای واسط شمالی در برنامه‌های کاربردی شبکه‌های نرم‌افزار محور بین توسعه‌دهندگان متفاوت است. الگوهای برنامه‌نویسی برجسته مانند RESTful و OSGi برای ارائه سرویس به نیازهای ارتباطی واسط شمالی در بسیاری کنترلرها کاربرد دارد [۱۰] [۸] [۱۱].

واسط شرقی و غربی[۹] ارتباط بین کنترلرهای شبکه‌های نرم‌افزار محور را در یک شبکه توزیع‌شده فراهم می‌کند. واسط شرقی و غربی می‌تواند برای اتصال دامنه‌های IP در شبکه سنتی و شبکه‌های نرم‌افزار محور مورداستفاده قرار گیرد و همچنین دامنه‌های مختلف مبتنی بر شبکه‌های نرم‌افزار محور را به هم متصل می‌کند.

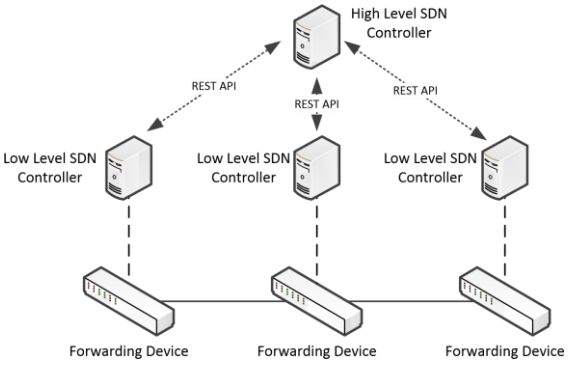
۲-۱- شبکه‌های‌نرم‌افزارمحور در شبکه‌های گسترده

همانطور که گفته شد دو روش برای توسعه شبکه‌های نرم‌افزار محور در شبکه‌های گسترده وجود دارد که عبارت‌اند از کنترلر واحد و چندکنترلر. برای روش چندکنترلر دو معماری ارائه‌شده است که عبارت‌اند از معماری متمرکز و معماری کاملاً توزیع‌شده یا چند دامنه. معماری متمرکز را می‌توان شبیه به طراحی اولیه برای شبکه‌های نرم‌افزار محور در نظر گرفت، درحالی‌که معماری کاملاً توزیع‌شده یا چند دامنه دارای طراحی متفاوتی است که در آن سطح کنترل توزیع‌شده ایجاد می‌شود [۱۴].

۲-۱-۱- معماری شبکه‌های نرم‌افزار محور چند دامنه

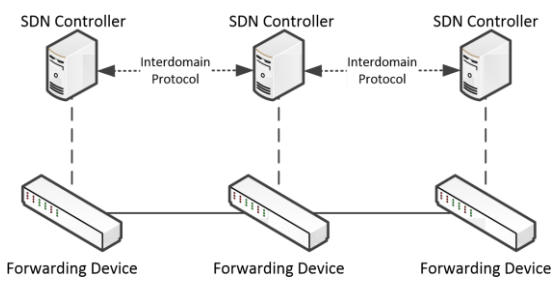
یک معماری شبکه‌های نرم‌افزار محور چند دامنه به یک معماری شبکه اشاره دارد که چندین دامنه شبکه‌های نرم‌افزار محور را به هم متصل می‌کند. دو مدل پیاده‌سازی عمودی و افقی برای شبکه‌های نرم‌افزار محور چند دامنه وجود دارد [۱۵]. در این مدل‌ها توپولوژی شبکه را می‌توان به تعدادی دامنه تقسیم کرد. هر دامنه شامل دستگاه‌های شبکه‌های نرم‌افزار محور است که به‌صورت محلی توسط یک کنترلر واحد مدیریت می‌شوند.

در یک ساختار سلسله مراتبی (عمودی) چندین سطح مدیریت وجود دارد. هر سطح شامل گروهی از کنترلرها است که هرکدام وظیفه مدیریت دامنه خود یا کنترلرهای موجود در سطح زیرین مستقیم خود را بر عهده دارند. کنترلرهای هر سطح دستورالعمل‌های مدیریتی را از کنترلری دریافت می‌کنند که به‌طور منطقی در سطح بالایی قرار دارد. تعداد سطح‌های منطقی در نظر گرفته‌شده مستقیماً به‌اندازه توپولوژی شبکه، میزان درخواست‌های پردازش جریان و اطلاعات وضعیت شبکه جمع‌آوری‌شده بستگی دارد. ارتباط بین کنترلرهای شبکه‌های نرم‌افزار عمدتاً از طریق RESTful انجام می‌شود. در مدل عمودی کنترلرها در یک سطح با یکدیگر ارتباط برقرار نمی‌کنند و در صورت لزوم ارتباط از طریق کنترلر ریشه انجام می‌گیرد [۱۶] [۴].



شکل ۳- مدل عمودی در معماری چند دامنه [۴]

در مدل افقی چندین کنترلر در یک سطح کنترل مسطح قرار دارند که هرکدام زیرمجموعه‌ای از سوئیچ‌های شبکه را مدیریت می‌کنند. در این مدل کنترلرها می‌توانند با استفاده از یک پروتکل استاندارد بین دامنه‌ای با یکدیگر ارتباط برقرار کنند. ارتباط کنترلر دامنه‌های مختلف از طریق واسط‌های شرقی و غربی انجام می‌گیرد. این مدل یک مزیت اساسی نسبت به مدل سلسله مراتبی دارد که آن نصب و پیکربندی سریع دامنه‌های اضافه‌شده است. تعداد زیادی از مطالعات این مدل را برای بهبود قابلیت مقیاس‌پذیری و گسترش سطح کنترل در شبکه‌های نرم‌افزار محور پیشنهاد داده و به کار گرفته‌اند [۱۷].



شکل ۴- مدل افقی در معماری چند دامنه [۴]

مدل افقی برای مقیاس‌پذیری در شبکه‌های گسترده واقع‌بینانه‌تر به نظر می‌رسد، زیرا کنترلرها در این روش می‌توانند با استفاده از یک پروتکل استاندارد بین دامنه‌ای با یکدیگر ارتباط برقرار کنند. این روش اتحاد بین دامنه‌ها را حفظ می‌کند و کنترلر شبکه‌های نرم‌افزار محور در هر دامنه مستقل می‌توانند سیاست‌ها و راه‌اندازی مسیر را تحت کنترل خود داشته باشند.

۳- مروری بر کارهای مرتبط

استفاده از پروتکل جدید برای ارتباط بین دامنه‌ها در شبکه‌های گسترده ممکن است منجر به مشکلات مربوط به سازگاری رو به عقب شود؛ بنابراین استفاده از پروتکل‌های موجود یک روش منطقی است، به‌خصوص زمانی که نیاز به برقراری ارتباط با شبکه‌های سنتی برای مدتی باقی خواهد ماند. برخی از تحقیقات مانند پل شرقی و غربی [۱۸] و سایر کنترلرهای پیشتاز مانند ONOS [۱۹] و ODL [۲۰] از این روش برای ارائه قابلیت‌های ارتباط بین دامنه‌ای کنترلرها استفاده می‌کنند. همگام‌سازی بین ASها یا دامنه‌ها در شبکه‌های گسترده فعلی از طریق پروتکل BGP انجام می‌گیرد، بنابراین BGP یک انتخاب طبیعی در کارهای اولیه انجام‌شده برای توسعه ارتباطات بین دامنه‌ای در کنترلرهایی مانند ODL و ONOS بود.

جرولا و همکاران [۲۱] راه‌حل طراحی‌شده توسط ICONA را پیشنهاد می‌کنند. این راه‌حل یک سناریوی بزرگ شبکه را با کنترلرهای توزیع‌شده ONOS مدیریت می‌کند. پیشنهاد آن‌ها با هدف تقویت تحمل خطا و کاهش تأخیر در پاسخ به رویدادهای ناشی از شبکه‌ها در مقیاس بزرگ است. یکی دیگر از تحقیقات طبق نظر لین و همکاران [۲۲] اجرای تدریجی راه‌حل‌های مبتنی بر شبکه‌های نرم‌افزار محور در دامنه‌های مختلف است که باید با سایر دامنه‌های غیر شبکه‌های نرم‌افزار محور همکاری کنند. آن‌ها یک برنامه همگام‌سازی بین ASهای مختلف بنام SDN-IP را بررسی می‌کنند که در کنترلر شبکه‌های نرم‌افزار محور اجرا می‌شود. برنامه SDN-IP برای رسیدن به هدف ما بسیار مهم خواهد بود.

گوا و همکاران [۲۳] از مدل سلسله مراتبی برای ارتباط کنترلرها در محیط چند دامنه استفاده می‌کنند.کنترلرهای محلی در این شبکه مسئول دامنه‌های خود هستند و کنترلر هماهنگ‌کننده مسئولیت دید سراسری از شبکه و ارتباط بین کنترلرها را بر عهده دارد. کنترلر هماهنگ‌کننده در جاوا پیاده‌سازی شده است. ارتباط بین کنترلر هماهنگ‌کننده و کنترلر دامنه و همچنین ارتباط کنترلر با برنامه‌های کاربردی توسط API واسط شمالی انجام می‌گیرد. واسط شمالی می‌تواند اطلاعات مربوط به کنترلر محلی را به برنامه‌های کاربردی و کنترلر هماهنگ‌کننده ارائه دهد و همچنین برنامه‌های کاربردی و کنترلر هماهنگ‌کننده را قادر می‌سازد تا جداول جریان و ارسال ترافیک را پیکربندی کنند. علاوه بر این دو ماژول با عنوان مدیریت توپولوژی و مدیریت جریان اجرا می‌شود. مدیریت توپولوژی وظیفه دریافت کل توپولوژی را بر عهده دارد و مدیریت جریان مسئول به‌روزرسانی و نصب جریان به ترتیب در کنترلرهای دامنه و سطح داده است.

وانگ و همکاران [۲۴] برای حل مسئله سازگاری در بین کنترلرهای متنوع یک روش با استفاده از کنترلر هماهنگ‌کننده ارائه کرده‌اند که به برقراری ارتباط بین کنترلرهای نامتقارن کمک می‌کند. سطح کنترل در این معماری به دو بخش کنترلر هماهنگ‌کننده و کنترلر دامنه تقسیم می‌شود. کنترلر هماهنگ‌کننده مسئول جمع‌آوری اطلاعات کنترلرهای شبکه و اطلاعات دامنه است. برای حل مسئله تنوع ارائه‌دهندگان از یک رابط یکپارچه واسط شمالی استفاده می‌کند. این API یکپارچه به دو بخش API توپولوژی و API سرویس تقسیم می‌شود. API توپولوژی برای جمع‌آوری اطلاعات شبکه و اتصال عناصر برای ایجاد دید سراسری از شبکه استفاده می‌شود و هدف از API سرویس راه‌اندازی درخواست سرویس و راه‌اندازی ارتباط در شبکه است.

هلبراند و همکاران [۲۵] برای تسهیل گسترش شبکه‌های نرم‌افزار محور در مقیاس بزرگ و انجام مدیریت ترافیک توسط هماهنگ‌کننده‌ها یک معماری برای ارتباط بین دامنه‌های مختلف پیشنهاد کرده‌اند. ارتباط بین دامنه‌ها در معماری پیشنهادی براساس مدل افقی و با استفاده از یک رابط که به آن ماژول INT گفته می‌شود، انجام‌گرفته است. انکوسی و همکاران در [۲۶] از روش جستجوی کوتاه‌ترین مسیر Dijkstra برای پیدا کردن کوتاه‌ترین مسیرها از مبدأ به مقصد و مقایسه بارکاری در این کوتاه‌ترین مسیرها استفاده می‌کنند. در این روش از مسیر با کمترین بارکاری به‌عنوان بهترین مسیر استفاده می‌شود. سرانجام ترافیک به این بهترین مسیر هدایت می‌شود.

تمام کارهای مرتبط ذکر شده سعی کردند با استفاده از یکی از روش‌های زیر مشکل را حل کنند:

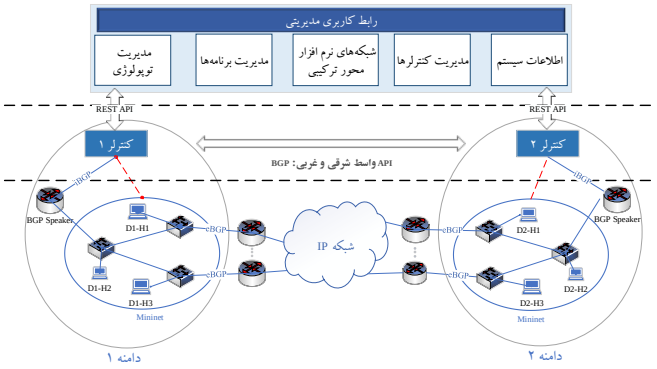
الف) تغییر پروتکل BGP موجود که منجر به هزینه‌های اضافی پروتکل می‌شود و بر روی عملکرد شبکه تأثیر می‌گذارد. همچنین به دلیل عدم توانایی در تغییر محیط در حال اجرا در شبکه‌های گسترده عملاً قابل اجرا نیست.

ب) استفاده از شبکه‌های نرم افزار محور که لازم است همه روترها و سوئیچ‌های شبکه‌های گسترده از شبکه‌های نرم افزار محور پشتیبانی کنند که در محیط واقعی قابل اجرا نیست. همچنین روترها و سوئیچ‌های موجود در مسیر ترافیک باید تحت مدیریت یکسان باشند تا تغییرات جریان ترافیک را از کنترلر شبکه‌های نرم افزار محور یکسان دریافت کنند.

بر این اساس روش پیشنهادی در این مقاله به گونه‌ای طراحی شده است که با توجه به توسعه تدریجی شبکه‌های‌نرم‌افزارمحور و قابلیت سازگاری رو به عقب از پروتکل BGP برای ارتباط بین دامنه‌ای در شبکه‌های گسترده استفاده می‌شود و جهت مدیریت و نظارت متمرکز بر دامنه‌ها در شبکه‌های‌گسترده روشی بر اساس مدل سلسله مراتبی ارائه می‌گردد. همچنین به‌منظور افزایش بهره‌وری در هر دامنه شبکه‌های نرم‌افزار محور از روش مهندسی ترافیک استفاده‌شده است که بر اساس پهنای باند هر لینک بهترین مسیر را انتخاب می‌کند و بهینه‌سازی مسیر را انجام می‌دهد.

۴- روش پیشنهادی

در این مقاله به توسعه یک سیستم توزیع‌شده با چندین کنترلر شبکه‌های نرم‌افزار محور پرداخته می‌شود که از ارتباط انتها به انتها بین دامنه‌ها در شبکه‌های گسترده پشتیبانی می‌کند. مدل افقی علی‌رغم مزایای بسیار با چالش مدیریت و نظارت متمرکز بر دامنه‌ها مواجه است که برای رفع این مشکل یک رابط کاربری برای مدیریت و نظارت بر چندکنترلری در شبکه‌های گسترده ارائه می‌گردد. همچنین برای افزایش بهره‌وری در شبکه هر دامنه (مبتنی بر شبکه‌های نرم‌افزار محور) یک روش مهندسی ترافیک پیاده‌سازی می‌گردد. در این بخش برای رسیدن به اهداف پیشنهادی طراحی سیستم و ایده‌ها ارائه خواهند شد.



شکل ۵- معماری پیشنهادی

شکل ۵ معماری پیشنهادی برای مدیریت ترافیک شبکه‌های‌گسترده بر اساس شبکه‌های نرم‌افزار محور را نشان می‌دهد که از دامنه‌های مبتنی بر شبکه نرم‌افزار محور توزیع‌شده تشکیل‌شده است. هر دامنه دارای یک کنترلر برای مدیریت دامنه خود است و دامنه‌های مختلف از طریق مسیریاب‌های مرزی BGP و شبکه IP به هم وصل می‌شوند. برای سیستم مدیریتی پیشنهادی از مدل سلسله مراتبی استفاده‌شده است که مناسب‌ترین راه‌حل برای مدیریت شبکه‌های نرم‌افزار محور توزیع‌شده است که باعث مدیریت آسان می‌شود. معماری پیشنهادی در ادامه شرح داده می‌شود:

سطح داده: این سطح شامل دستگاه‌های ارسال‌کننده[۱۰] در دامنه‌های مختلف شبکه‌های نرم‌افزار محور، مسیریاب‌های BGP speaker و مسیریاب‌های مرزی BGP است. دستگاه‌های ارسال‌کننده همان سوییچ‌های شبکه‌های نرم‌افزار محور هستند که از پروتکل OpenFlow پشتیبانی می‌کنند و مطابق با سیاست‌های ارسال‌شده از کنترلر مسئولیت هدایت ترافیک را بر عهده دارند. مسیریاب‌های BGP speaker و مسیریاب‌های مرزی BGP برای جمع‌آوری اطلاعات شبکه و تبدیل آن به به‌روزرسانی‌های مسیریابی استفاده می‌شوند. دامنه شبکه نرم‌افزار محور از طریق مسیریاب BGP speaker و با استفاده از BGP با شبکه‌های خارجی همگام می‌شود.

سطح کنترل: هر دامنه توسط کنترلر محلی کنترل می‌شود و چندین کنترلر محلی در شبکه وجود دارد. علاوه بر این کنترلر می‌تواند با استفاده از واسط شمالی استاندارد با سطح مدیریتی و برنامه‌های کاربردی ارتباط برقرار کند. در این مقاله از آخرین نسخه موجود کنترلر ONOS نسخه ۲.۴ استفاده‌شده است که سورس کد این نسخه در تاریخ ۷ فوریه ۲۰۲۰ و نسخه بیلد شده آن در ۵ ژوئن ۲۰۲۰ منتشرشده است.

سطح کاربرد: در سطح کاربرد برنامه‌های مختلف در حال اجرا هستند و مدیران شبکه می‌توانند روش‌هایی را که توسط کنترلر می‌توان فعال و اجرا کرد را تعریف کنند تا رفتار شبکه مورد انتظار آن‌ها باشد. در این مقاله عملیات شبکه مورد انتظار ارتباط بین دامنه‌های مشخص شبکه‌های نرم‌افزار محور در شبکه‌های گسترده است. برنامه SDN-IP برای برقراری ارتباط بین دامنه‌های شبکه‌های نرم‌افزار محور با استفاده از BGP اجرا می‌شود. همچنین برای این کار برخی برنامه‌های ONOS کمکی نیز نیاز به نصب دارند. این برنامه‌ها موردنیاز هستند و برای عملکرد SDN-IP به کنترلر اجازه می‌دهند چندین فایل پیکربندی را بخواند و به درخواست‌های ARP بین مسیریاب‌های مرزی خارجی و BGP Speakers پاسخ دهد.

در این مقاله برنامه‌ای ایجادشده است که اطلاعات مربوط به شبکه توزیع‌شده را برای رابط کاربری مدیریتی فراهم می‌کند و همچنین برنامه SDN-IP و پیکربندی موردنیاز دیگر برای شبکه توزیع‌شده را اجرا می‌کند. همچنین به‌منظور تأمین کیفیت خدمات و افزایش بهره‌وری در هر دامنه شبکه نرم‌افزار محور با استفاده از تکنیک‌های مهندسی ترافیک روشی ارائه‌شده است که با توجه به پهنای باند مسیر، مناسب‌ترین مسیر را انتخاب می‌کند و بهینه‌سازی انجام می‌شود. بر اساس ماهیت شبکه‌های نرم‌افزار محور برای مدیریت و برخورد هوشمندانه با جریان‌های ترافیکی موجود در هر دامنه شبکه نرم‌افزار محور برنامه‌ی ارائه‌شده باید در کنترلر هر دامنه فعال و نصب شود که این برنامه هم در سطح کاربرد قرار می‌گیرد.

سطح مدیریتی[۱۱]: رابط کاربری مدیریتی که در شکل ۵ نشان داده‌شده است، برای مدیریت و نظارت بر دامنه‌های مبتنی بر شبکه‌های نرم‌افزار محور در شبکه‌های گسترده طراحی‌شده است. ارتباط رابط کاربری مدیریتی با دامنه‌های مبتنی بر شبکه‌های نرم‌افزار محور از طریق واسط شمالی کنترلر هر دامنه انجام می‌گیرد. رابط کاربری مدیریتی از قابلیت‌ها و اجزاء زیر تشکیل‌شده است:

• مدیریت کنترلرها: برای مدیریت و نظارت بر دامنه‌ها امکان مدیریت متمرکز کنترلرهای شبکه‌های نرم‌افزار محور را در محیط توزیع‌شده فراهم کند.

• مدیریت توپولوژی: در رابط کاربری مدیریتی امکان نمایش توپولوژی در جهت نظارت متمرکز بر دامنه‌ها در نظر گرفته‌شده است. امکان نمایش توپولوژی باقابلیت‌هایی مانند نمایش به‌صورت پیش‌فرض و Hybrid SDN فراهم‌شده است. همچنین مشاهده وضعیت لینک، وزن لینک بر اساس پهنای باند و بار شبکه در توپولوژی مربوطه امکان‌پذیر است.

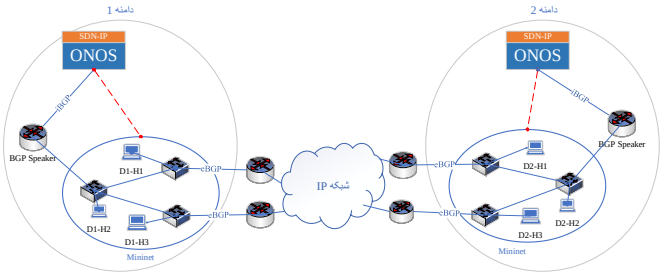
• مدیریت برنامه‌ها: نمایش اطلاعات مربوط به برنامه‌های هر کنترلر و امکان مدیریت برنامه‌ها در این بخش انجام می‌گیرد.

• اطلاعات سیستم: اطلاعات مربوط به شبکه و کنترلر هر دامنه را نمایش می‌دهد.

• شبکه‌های نرم‌افزار محور ترکیبی: اطلاعات مربوط شبکه توزیع‌شده که از پروتکل BGP برای ارتباط بین دامنه‌ای استفاده می‌کند را نمایش می‌دهد.

در ادامه در بخش‌های مختلف به بررسی و شرح روش پیشنهادی با جزئیات بیشتر پرداخته خواهد شد:

۴-۱- شبکه‌های نرم‌افزار محور چندکنترلر



شکل ۶- شبکه‌های نرم‌افزار محور چندکنترلر

با توجه به توسعه تدریجی شبکه‌های نرم‌افزار محور روشی برای ارتباط دامنه‌های مبتنی بر شبکه‌های نرم‌افزار محور از طریق شبکه IP پیشنهادشده است. ارتباط بین ASها در اینترنت امروزی با پروتکل BGPv4 انجام می‌شود [۲۷]. ارتباط بین دامنه‌های مختلف به پروتکل مسیریابی بین دامنه‌ای بستگی دارد و BGP پروتکل پایه برای این ارتباط است. با توجه به این‌که هر دامنه شبکه‌های نرم‌افزار محور توسط کنترلر ONOS کنترل و مدیریت می‌شود، برای ارتباط دامنه‌ها از طریق شبکه سنتی SDN-IP به‌عنوان یک برنامه در ONOS اجرا شده است. هدف از این روش توسعه یک سیستم توزیع‌شده با چندین کنترلر شبکه‌های نرم‌افزار محور است که از ارتباط انتها به انتها که بین دامنه‌های مجزا شبکه انجام می‌گیرد، پشتیبانی می‌کند. در ادامه به بیان این روش پرداخته می‌شود:

در این روش دامنه مبتنی بر شبکه‌های نرم افزار محور از طریق مسیریاب‌های مرزی BGP با دیگر دامنه‌ها ارتباط برقرار می‌کند. کنترلر شبکه‌های نرم‌افزار محور باید برای مقیاس‌پذیری و در دسترس بودن بالا طراحی شود. همچنین باید از برخی خدمات مربوط به شبکه (به‌عنوان‌مثال مسیریابی) را برای دامنه‌های مختلف پشتیبانی کند. برنامه SDN-IP در بالای کنترلر دامنه مبتنی بر شبکه‌های نرم افزار محور اجرا می‌شود که با استفاده از پروتکل مسیریابی BGP امکان برقراری ارتباط بین دامنه‌های شبکه‌های نرم‌افزار محور را فراهم می‌کند. در هر دامنه مبتنی بر شبکه‌های نرم‌افزار محور یک یا چند BGP speaker داخلی وجود دارد.

BGP speaker می‌توانند یک مسیریاب یا نرم‌افزاری باشد که برای اجرای قابلیت‌های BGP بکار گرفته می‌شود. نحوه کار این سیستم به این صورت است که مسیرهای اعلام‌شده توسط BGP speaker دامنه دو توسط BGP speaker در دامنه یک دریافت می‌شود و همگام سازی بین BGP speaker ها انجام می‌گیرد و با توجه به قوانین مسیریابی و فرآیند BGP پردازش انجام می‌گیرد و بهترین مسیر

برای هر مقصد توسط کنترلر شبکه‌های نرم‌افزار محور انتخاب و به یک هدف[۱۲] ترجمه می‌شود. هدف مانند یک کانال است که دو دستگاه شبکه را مستقیماً به هم وصل می‌کند. سپس هر هدف به قوانین ارسال تبدیل می‌شود. پس از آن این قوانین از کنترلر شبکه‌های نرم‌افزار محور به هر دستگاه شبکه (به‌عنوان‌مثال سوئیچ) که در مسیر مسیریابی اولیه BGP وجود دارد، ارسال می‌شود.

SDN-IP دو نوع هدف را نصب می‌کند. اهداف نقطه‌به‌نقطه[۱۳] که ارتباط بین BGP speaker خارجی و گره‌های داخلی BGP را تضمین می‌کند و هدف چند نقطه به تک نقطه[۱۴] امکان ارتباط بین دستگاه‌ها به دامنه‌های مختلف خارجی را فراهم می‌کند.

۴-۲- رابط کاربری مدیریتی

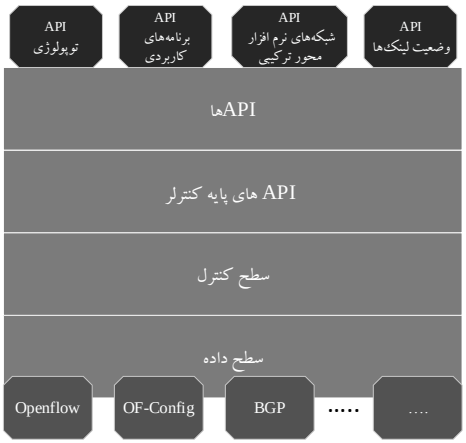
در معماری روش پیشنهادی ارتباط دامنه‌های مبتنی بر شبکه‌های نرم افزار محور با سطح مدیریت از طریق REST API در واسط شمالی کنترلر هر دامنه انجام می‌گیرد. کنترلرها می‌توانند با استفاده از این واسط شمالی یکپارچه اطلاعات و خدمات لازم را برای رابط کاربری مدیریتی در سطح مدیریت فراهم کنند. ساختار کنترلر دامنه شبکه‌های نرم‌افزار محور برای فراهم کردن منابع لازم رابط کاربری مدیریتی در شکل ۷ نشان داده‌شده است که واسط شمالی پیاده‌سازی شده به بخش‌های زیر تقسیم می‌شود:

API توپولوژی: مسئول جمع‌آوری اطلاعات اجزاء شبکه از هر دامنه شبکه‌های نرم‌افزار محور برای ایجاد نمای کلی از شبکه موردنظر است. این API همچنین اطلاعات سیستم شامل خلاصه‌ای از منابع شبکه و اطلاعات کنترلر شبکه‌های نرم‌افزار محور را فراهم می‌کند.

API برنامه‌های کاربردی: اطلاعات مربوط به نمایش برنامه‌ها و ارائه API لازم برای مدیریت برنامه‌ها را فراهم می‌کند.

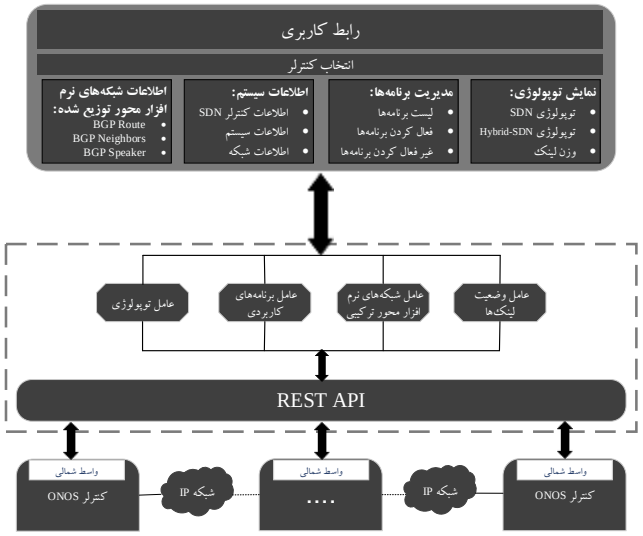
API شبکه‌های نرم‌افزار محور ترکیبی: اطلاعات مربوط به شبکه‌های نرم‌افزار محور توزیع‌شده را فراهم می‌کند که شامل بخش نمایش توپولوژی بر اساس Hybrid SDN و بخش اطلاعات Hybrid SDN است.

API وضعیت لینک: اطلاعات مربوط به نرخ بارکاری و وضعیت لحظه‌ای لینک‌های ارتباطی را ارائه می‌دهد.



شکل ۷- ساختار کنترلر هر دامنه شبکه‌های نرم‌افزار محور

ساختار رابط کاربری مدیریتی در شکل ۸ نشان داده‌شده است که امکان مدیریت کنترلر دامنه‌های مبتنی بر شبکه‌های نرم‌افزار محور را فراهم می‌کند و با انتخاب کنترلر دامنه موردنظر قابلیت‌هایی ازجمله نمایش توپولوژی، اطلاعات سیستم، مدیریت برنامه‌ها و اطلاعات شبکه توزیع‌شده امکان‌پذیر می‌شود. در پیاده‌سازی رابط کاربری مدیریتی عامل[۱۵] توپولوژی، عامل برنامه‌های کاربری و عامل شبکه‌های نرم‌افزار محور ترکیبی وجود دارند که از مجموعه‌ای از توابع برای مقابله با رویدادهای API توپولوژی، API مدیریت برنامه‌های کاربردی، API شبکه‌های نرم‌افزار محور ترکیبی استفاده می‌کنند.



شکل ۸- ساختار رابط کاربری مدیریتی

۴-۳- مهندسی ترافیک

بر اساس ماهیت شبکه‌های نرم‌افزار محور برای مدیریت و برخورد هوشمندانه با جریان‌های ترافیکی موجود در هر دامنه شبکه‌های نرم‌افزار محور برنامه‌ی ارائه‌شده باید در کنترلر فعال و نصب شود. کنترلر ONOS یکی از کاربردی‌ترین و مناسب‌ترین کنترلرهایی است که در زمینه‌ی شبکه‌های نرم‌افزار محور ارائه‌شده است. برنامه‌ها و روش‌ها ارائه‌شده در این مقاله برای استفاده در کنترلر ONOS طراحی و پیاده‌سازی شده‌اند. ONOS از مفهوم Bazel برای سازمان‌دهی فایل‌های سلسله مراتبی که Build نامیده می‌شوند، استفاده می‌کند.

هر برنامه یا پروژه فرعی در ONOS فایل Build مخصوص به خود را دارد. تمام این فایل‌های Build میانی به فایل Build والد در پروژه ONOS مرتبط هستند. فایل Build والد شامل وابستگی‌ها و تنظیمات مشترک در فایل‌های Build پروژه‌های فرعی است. این ادغام فایل‌های Build میانی و والد به ساخت ONOS به روشی کاملاً مستقل کمک می‌کند. برنامه‌ها (بسته‌های نرم‌افزاری یا باندل‌ها) در ONOS با استفاده از زبان برنامه‌نویسی جاوا پیاده‌سازی می‌شوند. روش موردنظر در جهت مهندسی ترافیک با استفاده از زبان برنامه‌نویس جاوا و به‌صورت یک App با رعایت مفاهیم برنامه‌نویسی شیءگرا و به‌صورت چندبخشی پیاده‌سازی شده‌اند.

الگوریتم مهندسی ترافیک در این برنامه عمدتاً پهنای باند مسیر را در نظر می‌گیرد. وزن مسیر بر اساس پهنای باند با وزن بر اساس پهنای باند لینک‌هایی که مسیر را تشکیل می‌دهند، نسبتاً برابر است. وزن لینک بر اساس پهنای باند به‌صورت نسبتی از نرخ بارکاری به‌کل پهنای باند لینک در نظر گرفته‌شده است. بر همین اساس وزن مسیر بر اساس پهنای باند با حداقل مقدار نسبت نرخ بارکاری لینک به‌کل پهنای باند لینک رابطه دارد. با توجه به اینکه در الگوریتم‌های مسیریابی موجود در کنترلر ONOS و ابزارهای محاسبه وزن به‌طور پیش‌فرض لینک‌ها با وزن‌های کم‌تر به‌عنوان لینک مناسب نشان داده می‌شوند، درنتیجه حداقل وزن لینک بهینه است. در این برنامه فرمول نهایی برای وزن به‌صورت زیر است:

$$\text{وزن} = \frac{\text{نرخ بارکاری}}{\text{کل پهنای باند}} \times ۱۰۰$$ فرمول (۱)

در ابتدا الگوریتم از توپولوژی کل شبکه به‌عنوان پارامترهای ورودی استفاده می‌کند. توپولوژی یک ساختار داده‌ای به‌صورت گراف است. در این برنامه از میزبان مبدأ به‌عنوان نقطه شروع و میزبان مقصد به‌عنوان نقطه پایان استفاده می‌شود. در ادامه از الگوریتم جستجوی اول عمق[۱۶] استفاده می‌کند و گراف شبکه را به‌صورت بازگشتی پیمایش می‌کند. هنگام عبور از هر گره، گره و لینکی که از آن عبور کرده را ذخیره می‌کند و هنگام بازگشت به‌صورت بازگشتی آن را حذف می‌کند. وقتی متوجه شد که به میزبان مقصد رسیده است، تمام اطلاعات گره و لینک‌های ثبت‌شده را ادغام می‌کند و آن را به‌عنوان یک مسیر در مجموعه نتیجه ثبت می‌کند. در این روند اگر به گره‌ای تکراری برسد مشخص می‌شود که حلقه تشکیل‌شده است و بازگشت به عقب انجام می‌گیرد. در پایان مجموعه‌ای از مسیرهای بدون حلقه از میزبان مبدأ به میزبان مقصد دریافت می‌شود.

در ادامه وزن بر اساس پهنای باند برای مسیرهای به‌دست‌آمده محاسبه می‌شود. ابتدا وزن هر لینک در مسیر محاسبه می‌شود. بر همین اساس باید اندازه بارکاری و ترافیک فعلی لینک را به دست آورده و با استفاده از فرمول ۱ وزن محاسبه شود. اگر پهنای باند لینک‌ها متفاوت باشد، پهنای باند لینک‌ها به پهنای باند کمترین لینک کاهش پیدا می‌کند. سپس با مقایسه وزن مسیرها و در نظر گرفتن دامنه تحمل، مسیرها با وزن کمتر به‌عنوان مجموعه مسیر بهینه انتخاب می‌شوند. در ادامه اولین مسیر با کمترین تعداد hop به عنوان مسیر بهینه انتخاب می‌شود. در آخر با توجه به اینکه در کنترلر ONOS در گراف توپولوژی لینک بین دستگاه و میزبان درج نشده است بنابراین باید دو لینک دیگر به نتیجه مسیر اضافه شود و سپس نتایج نهایی و مسیر بهینه به شبکه ارسال می‌گردد.

۵- پیاده سازی و ارزیابی نتایج

در این بخش پیاده‌سازی و ارزیابی کارایی روش پیشنهادی مورد بررسی قرار می‌گیرد. برای آزمایش امکان‌سنجی و بررسی عملکرد روش‌های پیشنهادی یک محیط آزمایش چندکنترلر ایجاد شده است. ارتباط رابط کاربری مدیریتی با دامنه‌های مبتنی بر شبکه‌های نرم‌افزار محور از طریق واسط شمالی در هر کنترلر انجام می‌شود و مدیریت دامنه‌ها از طریق این رابط کاربری انجام می‌گیرد. کنترلر ONOS یکی از کاربردی‌ترین و مناسب‌ترین کنترلرها در حوزه‌ی شبکه‌های نرم‌افزار محور است و به همین دلیل از کنترلر ONOS در این مقاله استفاده‌شده است.

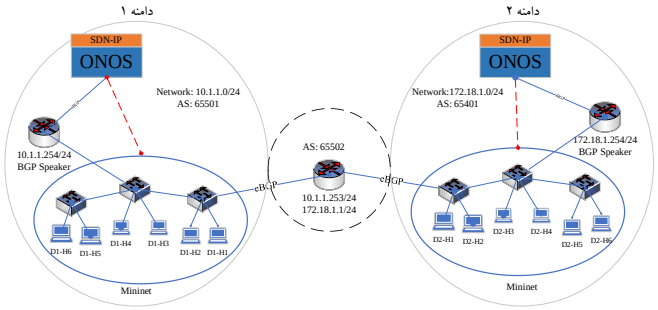
برای پیاده‌سازی و آزمایش سیستم پیشنهادی از نرم‌افزارهای GNS3 و Mininet و VMware Workstation استفاده‌شده است که امکان ایجاد یک شبکه مجازی واقع‌گرایانه و مشابه محیط واقعی را فراهم می‌کنند. برای شبیه‌سازی محیط سیستم از ماشین مجازی در VMware Workstation استفاده‌شده است. ماشین‌های مجازی شامل کنترلر ONOS، مقلد mininet و vm GNS3 است. کنترلرهای شبکه‌های نرم‌افزار محور بر روی سیستم‌عامل Ubuntu 18.0.4 راه‌اندازی شده‌اند. شبیه‌سازی سوییچ‌های OpenFlow در شبکه‌های نرم‌افزار محور با استفاده از مقلد Mininet انجام‌گرفته است. از شبیه‌ساز GNS3 برای شبیه‌سازی مسیریاب‌های شبکه و پیاده‌سازی محیط توزیع‌شده استفاده‌شده است. در این سناریو مسیریاب‌های شبکه IP و BGP speaker را در GNS3 پیاده‌سازی کرده و پیکربندی‌های لازم بر روی آن‌ها انجام‌شده است و با استفاده از قابلیت Cloud به ماشین مجازی کنترلر ONOS و شبیه‌ساز Mininet متصل شده‌اند. ارتباط کنترلرها با رابط کاربری مدیریتی در Vmware از طریق سوییچ‌هایی که در این نرم‌افزار به‌عنوان Vmnet تعریف می‌شوند، انجام‌گرفته است..

در جدول ۱ ابزار و فناوری‌های اصلی مورداستفاده در سیستم پیشنهادی ذکرشده است. در ادامه ارزیابی و پیاده‌سازی روش پیشنهادی در سه بخش بیان می‌شود.

جدول ۱- ابزار و فناوری‌های استفاده‌شده

دسته‌بندی	نرم‌افزار / فناوری
کنترلر شبکه‌های نرم‌افزار محور	ONOS v2.4
شبیه‌ساز شبکه	Mininet, GNS3
Hypervisor مجازی	VMware Workstation 15.5.1
سیستم‌عامل ماشین مجازی	Ubuntu 18.04
اندازه‌گیری ترافیک	Iperf, D-ITG
برنامه‌ها	SDN-IP, M7delkhosh-TE, M7delkhosh-hybrid-sdn

۵-۱- پیاده‌سازی ارتباط بین دامنه‌ای در شبکه‌های گسترده



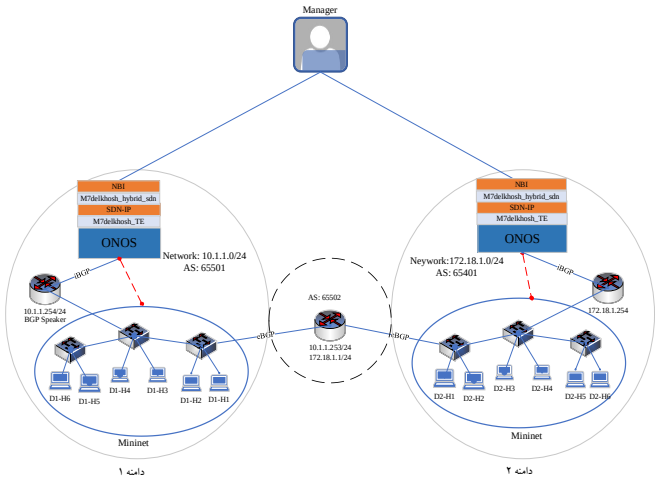
شکل ۹- سناریو پیاده‌سازی شده

برای ارزیابی روش پیشنهادی همان‌طور که در شکل ۹ مشخص است، دو دامنه مبتنی بر شبکه‌های نرم‌افزار محور پیاده‌سازی شده است. در این سناریو دو کنترلر شبکه‌های نرم‌افزار محور وجود دارد که هر کنترلر دامنه مربوطه را که شامل ۳ سوئیچ Openflow و ۶ میزبان است، کنترل می‌کند. برای تأیید موفقیت سیستم از آزمایش عملکردی [۲۲] استفاده‌شده است. در این آزمایش ترافیک ICMP از میزبان h۶ در دامنه یک به میزبان h۶ در دامنه دو ارسال شده است. مدت‌زمان این آزمایش ۵ دقیقه در نظر گرفته‌شده و نتایج زیر به‌دست‌آمده است:

جدول ۲- نتایج به‌دست‌آمده از ارتباط دو دامنه SDN

RTT (ms)	Min	Avg	Max
D1-H6 ⇔ D2-H6	11.208	19.033	151.771

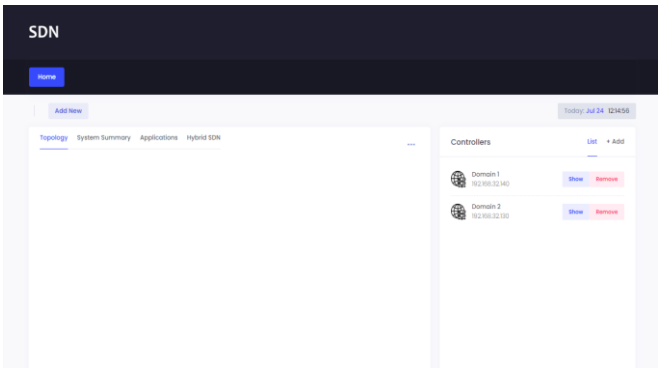
۵-۲- پیاده‌سازی رابط کاربری مدیریتی



شکل ۱۰- سناریو پیاده‌سازی شده

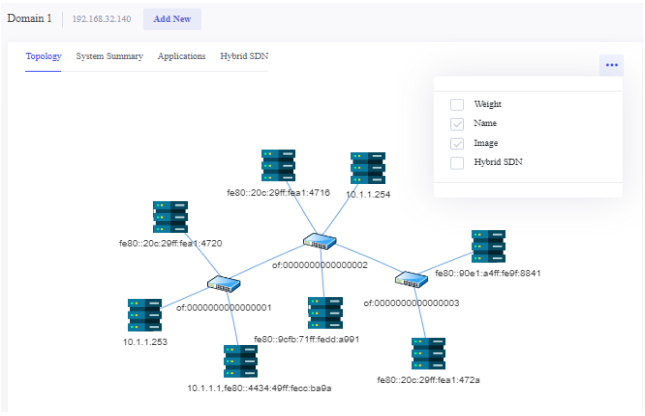
آزمایش به صورت بررسی عملکرد رابط کاربری و آزمایش قابلیت همکاری واسط شمالی کنترلر با رابط کاربری است. در ادامه شرح داده‌شده است که در معماری پیشنهادی اطلاعات مطابق با استاندارد و به‌درستی در رابط کاربری مدیریتی دریافت می‌شود و اطلاعات نمایش داده‌شده بیان‌گر کارآمدی روش پیشنهادی است.

همان‌طور که در شکل ۱۱ نشان داده‌شده است، فهرستی از کنترلرهای موجود در شبکه توزیع‌شده نمایش داده‌شده است که با انتخاب هر یک ویژگی‌ها و اطلاعات مربوط به دامنه موردنظر نمایش داده می‌شود.



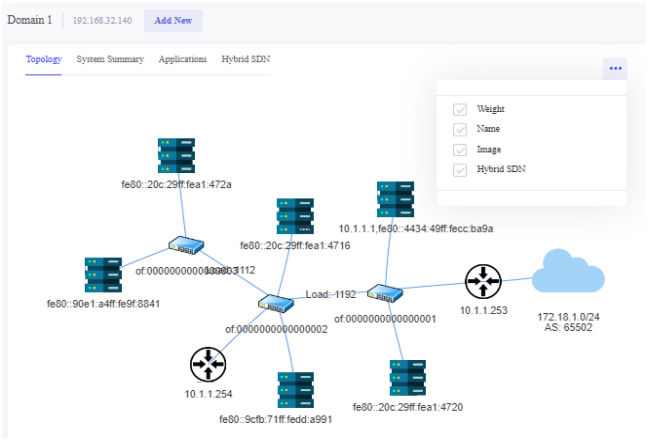
شکل ۱۱- فهرست کنترلرها

یکی از ویژگی‌های مهم نظارت و مدیریت شبکه دید کلی از توپولوژی شبکه است بنابراین ویژگی توپولوژی در رابط کاربری در نظر گرفته شده است. همان‌طور که در شکل ۱۲ مشخص است، توپولوژی شبکه دامنه یک به‌درستی نمایش داده‌شده است. حالت پیش‌فرض در این قسمت شامل دستگاه‌های شبکه نرم‌افزار محور و میزبان‌ها است که مسیریاب‌ها نیز به‌صورت میزبان نمایش داده می‌شوند.



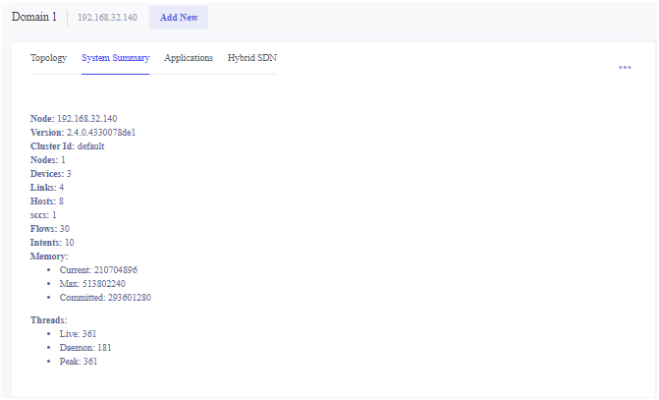
شکل ۱۲- توپولوژی شبکه

با فعال کردن ویژگی name برچسب گره‌ها که شامل مشخصات هر گره است، نمایش داده می‌شود. با فعال کردن قابلیت Hybrid SDN در صورت فعال بودن برنامه مربوطه که در این مقاله ایجاد شده است، توپولوژی شبکه به همراه مسیریاب‌های متصل شده به سوییچ OpenFlow نمایش داده می‌شود و همچنین شبکه‌های متصل به‌صورت ابر با برچسب IP و AS مربوطه مشخص می‌شوند. همان‌طور که قبلاً نیز گفته شد این برنامه اطلاعات لازم در جهت نمایش توپولوژی و همچنین اطلاعات قسمت SDN Hybrid را به‌صورت یک سرور در هر کنترلر برای رابط کاربری فراهم می‌کند و برنامه SDN-IP و پیکربندی موردنیاز دیگر برای شبکه توزیع‌شده را اجرا می‌کند. با فعال کردن قابلیت Weight بارکاری لحظه‌ای هر لینک نمایش داده می‌شود.



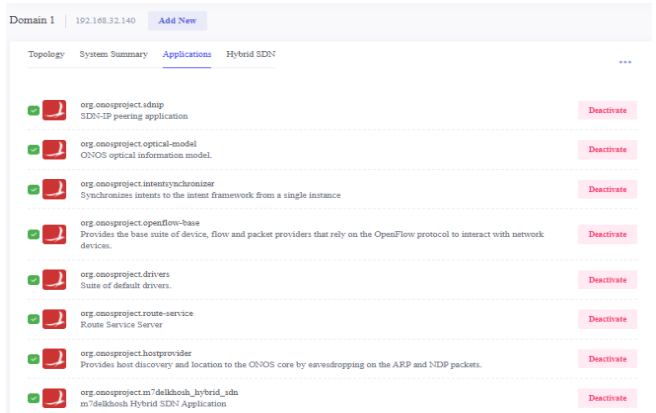
شکل ۱۳- توپولوژی شبکه به‌صورت ترکیبی

در شکل ۱۴ اطلاعاتی از شبکه که شامل اطلاعات سیستم و شبکه هر دامنه است، نمایش داده شده است که با اطلاعات شبکه کاملاً مشابه است.



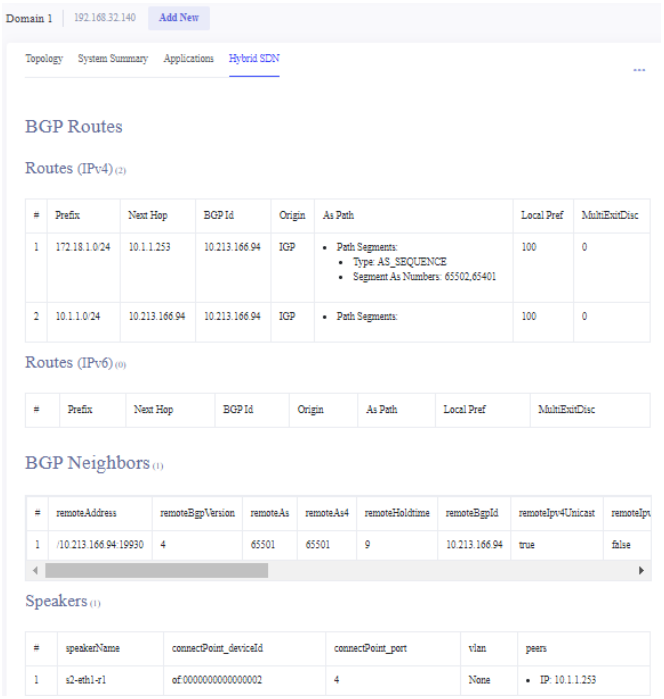
شکل ۱۴- اطلاعات سیستم

در قسمت برنامه‌ها فهرستی از برنامه‌های هر کنترلر شبکه نرم‌افزار محور نمایش داده می‌شود و همچنین اطلاعات کاملی از هر برنامه قابل‌مشاهده است. قابلیت فعال و غیرفعال کردن برنامه‌ها نیز در این قسمت امکان‌پذیر است. در این قسمت به‌راحتی می‌توان وضعیت برنامه‌های هر کنترلر را مطابق با نیاز تغییر داد که این امکان به‌درستی پیاده‌سازی شده است.



شکل ۱۵- مدیریت برنامه‌ها

در بخش Hybrid SDN اطلاعات کاملی از شبکه توزیع‌شده شامل اطلاعات BGP Neighbors، BGP Speakers، BGP Routes به تفکیک IP v4 و IP v6 نمایش داده می‌شود که در شکل ۱۶ قابل مشاهده است.



شکل ۱۶- اطلاعات Hybrid SDN

بررسی انجام‌شده در این بخش بیانگر عملکرد درست و کارآمد رابط کاربری مدیریتی است. رابط کاربری طراحی‌شده اطلاعات کاملی را در برمی‌گیرد و مقیاس‌پذیری بالایی دارد. قابلیت‌های کاربردی در رابط کاربری مدیریتی با در نظر گرفتن ابعاد مختلف پیاده‌سازی شده است.

۵-۳- ارزیابی روش مهندسی ترافیک

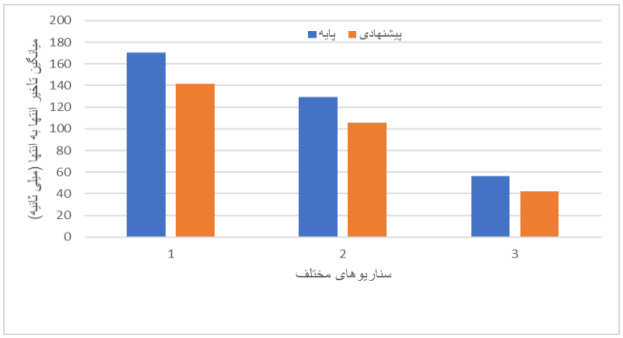
برای ارزیابی کارایی بخش مهندسی ترافیک در شبیه‌ساز Mininet از سه توپولوژی با مقیاس‌های متفاوت استفاده می‌شود تا کارایی روش پیشنهادی به‌درستی ارزیابی شود.

توپولوژی اول NTTNet [۲۸]: از ۵۵ سوییچ و ۱۴۴ لینک تشکیل‌شده است و بزرگ‌ترین شبکه مخابراتی ژاپن است. پهنای باند لینک‌ها به‌صورت تصادفی بین ۱ تا ۱۰ مگابیت بر ثانیه و تأخیر بین لینک‌ها به صورت تصادفی بین ۱ تا ۱۰ میلی‌ثانیه در نظر گرفته‌شده است. در شبکه، ۶ سرور فرستنده و ۶ سرور گیرنده تعریف‌شده است که ترافیک میانی را در شبکه ایجاد می‌کنند.

توپولوژی دوم PalmettoNet [۲۹]: شامل ۴۴ سوییچ و ۶۹ لینک است و در ایالت کارولینا آمریکا ایجادشده است. پهنای باند لینک‌ها به‌صورت تصادفی بین ۱ تا ۳۰ مگابیت بر ثانیه و تأخیر بین لینک‌ها به‌صورت تصادفی بین ۱ تا ۲۰ میلی‌ثانیه در نظر گرفته می‌شود. در شبکه، ۵ سرور فرستنده و ۵ سرور گیرنده تعریف‌شده است که ترافیک میانی را در شبکه ایجاد می‌کنند.

توپولوژی سوم USNET [۳۰]: از ۲۴ سوییچ و ۴۳ لینک تشکیل‌شده و جزء بستر اینترنت ایالت متحده آمریکا است. ابتدا پهنای باند لینک‌ها به‌صورت تصادفی بین ۱ تا ۱۰ مگابیت بر ثانیه و تأخیر بین لینک‌ها به‌صورت تصادفی بین ۱ تا ۲۰ میلی‌ثانیه در نظر گرفته می‌شود. در شبکه، ۴ سرور فرستنده و ۴ سرور گیرنده تعریف‌شده است که ترافیک میانی را در شبکه ایجاد می‌کنند.

برای ارزیابی روش پیشنهادی سه توپولوژی معرفی‌شده در روش پایه و روش پیشنهادی مورد بررسی قرارگرفته است. در جهت ایجاد ترافیک میانی با استفاده از iperf ترافیک UDP با نرخ ۱ مگابیت بر ثانیه از فرستنده‌ها به گیرنده‌ها ارسال می‌گردد و هم‌زمان با شروع این ترافیک با استفاده از ابزار D-ITG از میزبان مبدأ به مقصد ارزیابی انجام می‌گیرد. مدت ارزیابی برای هر روش ۵ دقیقه است که برای هر سه توپولوژی این ارزیابی انجام گرفته است. در شکل ۱۷ نتایج به‌دست‌آمده نمایش داده‌شده است که در آن پارامتر تأخیر مقایسه می‌شود. با توجه به نتایج نمایش داده شده روش پیشنهادی بر اساس میانگین تأخیر انتها به انتها شرایط بهتری در سناریوهای مختلف دارد.



شکل ۱۷- میانگین تأخیر انتها به انتها در سه توپولوژی مختلف

۶- نتیجه گیری

با رشد نمایی ترافیک داده و ناهمگونی شبکه‌های ارتباطی امروزی راه‌حل‌های مدیریت شبکه‌های سنتی به چالش کشیده می‌شود. شبکه‌های نرم‌افزار محور به دلیل قابلیت برنامه‌نویسی و مدیریت متمرکز یک عامل مؤثر برای غلبه بر مشکلات شبکه‌های سنتی است. سناریوهای چندکنترلر در سیستم‌های توزیع‌شده پیچیدگی شبکه را افزایش می‌دهد و چالش‌های زیادی را در مدیریت کارآمد زیرساخت شبکه ایجاد می‌کند. با این حال بهترین راه برای دستیابی به عملکرد خوب شبکه و افزایش کارایی در مدیریت عملکرد و ترافیک در شبکه‌های گسترده، چندکنترلر است، زیرا تقسیم شبکه به دامنه‌های مختلف به هر کنترلر این امکان را می‌دهد تا بتواند دامنه خود را به‌طور مستقل مدیریت کند و به‌طور مداوم با سایر دامنه‌ها همکاری کند؛ همانطور که گفته شد هدف اصلی این مقاله این بود که بدانیم چگونه می‌توان زیرساخت شبکه‌ای متشکل از چندین دامنه اجرائی مجزا را در شبکه‌های گسترده توسعه داد و آن‌ها را مدیریت کرد؛ بنابراین برای دستیابی به هدف پیشنهادی مروری بر ادبیات با هدف درک عملکرد شبکه‌های نرم‌افزار محور چندکنترلر و چگونگی برقراری ارتباط بین دامنه‌های مجزا انجام شد.

در این مقاله یک سیستم شبکه توزیع‌شده با دو دامنه شبکه‌های نرم‌افزار محور طراحی و اجرا شد. در این شبکه توزیع‌شده از اطلاعات مربوط به BGP و برنامه SDN-IP استفاده می‌شود که در آن اهداف به قوانین مسیریابی تبدیل می‌گردد و بر روی سوئیچ‌های OpenFlow نصب می‌شود. سپس برای مدیریت دامنه‌های مبتنی بر شبکه‌های نرم‌افزار به‌صورت متمرکز و در جهت رفع مشکل مدیریت و نظارت بر شبکه‌های توزیع‌شده روشی بر اساس مدل سلسله مراتبی ارائه شد. در این روش برای مدیریت چندکنترلری در شبکه‌های گسترده رابط کاربری مدیریتی پیاده‌سازی شده است. در این مقاله ابعاد مختلف برای مدیریت ترافیک شبکه‌های گسترده در نظر گرفته‌شده است. همچنین برای افزایش بهره‌وری در هر دامنه شبکه نرم‌افزار محور از روش مهندسی ترافیک که بر اساس پهنای باند هر لینک بهترین مسیر را انتخاب می‌کند، استفاده‌شده است.

با توجه به نتایج به‌دست‌آمده در بخش ۵ نشان داده شد که می‌توان از طریق پروتکل BGP و با استفاده از برنامه SDN-IP ارتباط بین دامنه‌ای شبکه‌های نرم‌افزار محور توزیع‌شده را در شبکه‌های گسترده تضمین کرد. بررسی‌ها و آزمایش انجام‌گرفته نشان می‌دهد که رابط کاربری به‌درستی پیاده‌سازی شده است و قابلیت‌های مختلف در جهت مدیریت بهتر در آن در نظر گرفته‌شده است. در روش مهندسی ترافیک پیشنهادی با توجه به نتایج به‌دست‌آمده بر اساس میانگین تأخیر انتها به انتها شرایط بهتری در موقعیت‌ها و سناریوهای مختلف داشته است.

۷- مراجع

[1] R. Mohammadi, R. Javidan, M. Keshtgari, and R. Akbari, "A novel multicast traffic engineering technique in SDN using TLBO algorithm," *Telecommunication Systems,* vol. 68, no. 3, pp. 583-592, 2018.

[2] W. Queiroz, M. A. Capretz, and M. Dantas, "An approach for SDN traffic monitoring based on big data techniques," *Journal of Network and Computer Applications,* vol. 131, pp. 28-39, 2019.

[3] O. Michel and E. Keller, "SDN in wide-area networks: A survey," in *2017 Fourth International Conference on Software Defined Systems (SDS)*, 2017: IEEE, pp. 37-42 .

[4] R. Ahmed and R. Boutaba, "Design considerations for managing wide area software defined networks," *IEEE Communications Magazine,* vol. 52, no. 7, pp. 116-123, 2014.

[5] Y. Zhang, L. Cui, W. Wang, and Y. Zhang, "A survey on software defined networking with multiple controllers," *Journal of Network and Computer Applications,* vol. 103, pp. 101-118, 2018.

[6] F. X. Wibowo, M. A. Gregory, K. Ahmed, and K. M. Gomez, "Multi-domain software defined networking: research status and challenges," *Journal of Network and Computer Applications,* vol. 87, pp. 32-45, 2017.

[7] A. Jalili, M. Keshtgari, R. Akbari, and R. Javidan, "Multi criteria analysis of controller placement problem in software defined networks," *Computer Communications,* vol. 133, pp. 115-128, 2019.

[8] A. Prajapati, A. Sakadasariya, and J. Patel, "Software defined network: Future of networking," in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, 2018: IEEE, pp. 1351-1354 .

[9] F. Bannour, S. Souihi, and A. Mellouk, "Distributed SDN control: Survey, taxonomy, and challenges," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 1, pp. 333-354, 2018.

[10] Z. Latif, K. Sharif, F. Li, M. M. Karim, S. Biswas, and Y. Wang, "A comprehensive survey of interface protocols for software defined networks," *Journal of Network and Computer Applications,* vol. 156, p. 102563, 2020.

[11] G. Pereira, J. Silva, and P. Sousa, "Estudo Comparativo de Controladores Software-Defined Networking (SDN (Comparative Study of Software-Defined Networking (SDN) Traffic Controllers".

[12] A. Mendiola, J. Astorga, E. Jacob, and M. Higuero, "A survey on the contributions of software-defined networking to traffic engineering," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 2, pp. 918-953, 2016.

[13] H. Farhady, H. Lee, and A. Nakao, "Software-defined networking: A survey," *Computer Networks,* vol. 81, pp. 79-95, 2015.

[14] B. Davie, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. Gude, A. Padmanabhan, T. Petty, K. Duda, A. Chanda, "A database approach to sdn control plane design," *ACM SIGCOMM Computer Communication Review,* vol. 47, no. 1, pp. 15-26, 2017.

[15] H. E. Egilmez and A. M. Tekalp, "Distributed QoS architectures for multimedia streaming over software defined networks," *IEEE Transactions on Multimedia,* vol. 16, no. 6, pp. 1597-1609, 201.۴

[16] J.-J. Huang, Y.-Y. Chen, C. Chen, and Y. H. Chu, "Weighted routing in hierarchical multi-domain SDN controllers," in *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2015: IEEE, pp. 356-359 .

[17] Q. Duan, N. Ansari, and M. Toy, "Software-defined network virtualization: An architectural framework for integrating SDN and NFV for service provisioning in future networks," *IEEE Network,* vol. 30, no. 5, pp. 10-16, 2016.

[18] P. Lin, J. Bi, S. Wolff, Y. Wang, A. Xu, Z. Chen, H. Hu, Y. Lin, "A west-east bridge based SDN inter-domain testbed," *IEEE Communications Magazine,* vol. 53, no. 2, pp. 190-197, 2015.

[19] P. Berde *et al.*, "ONOS: towards an open, distributed SDN OS," in *Proceedings of the third workshop on Hot topics in software defined networking*, 2014, pp. 1-6 .

[20] S. Badotra and J. Singh, "Open Daylight as a Controller for Software Defined Networking," *International Journal of Advanced Research in Computer Science,* vol. 8, no. 5, 2017.

[21] M. Gerola, M. Santuari, E. Salvadori, S. Salsano, P. L. Ventre, M. Campanella, F. Lombardo, G. Siracusano, "Icona: Inter cluster onos network application," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, 2015: IEEE, pp. 1-2 .

[22] P. Lin, j. Hart, U. Krishnaswamy, T. Murakami, M. Kobayashi, A. Al-Shabibi, K.C. Wang, and J. Bi, "Seamless interworking of SDN and IP," in *Proceedings of the ACM SIGCOMM 2013 conference*, 2013, pp. 475-476 .

[23] Z. Guo ,G. Shou, Y. Hu, and Z. Guo, "An implementation of multi-domain software defined networking," presented at the 11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015), Shanghai, China, 2015.

[24] J. Wang, G. Shou, Y. Hu, and Z. Guo, "A multi-domain SDN scalability architecture implementation based on the coordinate controller," in *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 2016: IEEE, pp. 494-499 .

[25] P. Helebrandt and I. Kotuliak, "Novel SDN multi-domain architecture," in *12th IEEE International Conference on Emerging eLearning Technologies and Applications*, 2014, pp. 139-143 .

[26] M. C. Nkosi, A. A. Lysko, and S. Dlamini, "Multi-path load balancing for SDN data plane," in *2018 International Conference on Intelligent and Innovative Computing Applications*, 2018: IEEE ,pp. 1-6 .

[27] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (BGP-4)," ed: ISI, USC Information Sciences Institute, 1994.

[28] "NTTNet " http://www.topology-zoo.org/files/Palmetto.gml (accessed 2020/07/10, 2020).

[29] "Palmetto." http//:www.topology-zoo.org/files/Palmetto.gml (accessed 2020/07/12.

[30] C. Cavdar, M. Tornatore, F. Buzluca, and B. Mukherjee, "Dynamic scheduling of survivable connections with delay tolerance in WDM networks," in *IEEE INFOCOM Workshops 2009*, 2009: IEEE, pp. 1-6.

محمد دلخوش کارشناسی مهندسی فناوری اطلاعات و کارشناسی ارشد مهندسی فناوری اطلاعات گرایش شبکه‌های ارتباطی و کامپیوتری را به ترتیب در سال‌های ۱۳۹۷ و ۱۳۹۹ از دانشگاه آزاد اسلامی واحد اراک و دانشگاه صنعتی شیراز اخذ کرده است. زمینه های اصلی تحقیقاتی ایشان شبکه‌های کامپیوتری و شبکه‌های نرم افزار محور می باشد. آدرس پست الکترونیکی ایشان عبارت است از:

m.delkhosh@sutech.ac.ir

رضا جاویدان کارشناسی مهندسی کامپیوتر را در سال ۱۳۷۲ از دانشگاه اصفهان و کارشناسی ارشد و دکتری مهندسی کامپیوتر را به ترتیب در سال‌های ۱۳۷۵ و ۱۳۸۶ از دانشگاه شیراز اخذ کرده است. زمینه‌های اصلی تحقیقاتی ایشان اینترنت اشیاء، امنیت شبکه، شبکه‌های حسگر بی سیم زیر آب، شبکه‌های نرم افزار محور، سیستم‌های سونار و پردازش تصویر می‌باشد. دکتر جاویدان در حال حاضر دانشیار دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی شیراز می‌باشد و از سال ۱۳۹۶ آزمایشگاه اینترنت اشیاء را در این دانشگاه راه اندازی کرده است. آدرس پست الکترونیکی ایشان عبارت است از:

javidan@sutech.ac.ir

---

[1] Control plane
[2] Data plane
[3] Wide Area Networks
[4] Real-time
[5] Routers
[6] Application Plane
[7] Southbound API
[8] Northbound API
[9] East/ Westbound API
[10] Forwarding Devices
[11] Management plane
[12] Intent
[13] Point to Point
[14] Multi-Point to Single-Point
[15] Agent
[16] DFS

The CSI Journal on Computing Science and Information Technology
Vol. 18, No.2, 2021

Proposing a method for WAN traffic management based on Software Defined Networks

Mohammad Delkhosh[1], Reza Javidan[2]

[1, 2] Department of Computer Engineering and Information Technology, Shiraz University of Technology, Shiraz, Iran

Abstract

Software-defined networks (SDNs), due to their centralized nature, face many limitations in their performance when used in wide area networks (WANs). As a result, the scalability of such networks has many challenges. Designing and Implementing SDNs using multi-controllers provides a scalable solution that can be used to improve traffic management and performance of WANs. In this paper, a multi-controller SDN system based on the horizontal model is proposed. In the proposed method, gradual development capability and backward compatibility using BGP protocol for inter-domain communication are used, which improves network scalability and quick installation and configuration on the network. However, this approach faces the challenge of the limitation of global and integrated domain monitoring and management. Therefore, another method based on the hierarchical approach is also proposed to manage and monitor the multi-controller design of WANs. In this method, the controllers of each domain are connected to the management user interface through REST in the northbound interface of each controller. Moreover, to increase performance in each domain of SDNs, a traffic engineering method is used, which selects the best route based on the bandwidth of each link. The implementation results indicate the guarantee of inter-domain communication through BGP, the effectiveness of the traffic engineering method, and the efficiency of management and monitoring on multi-controller through the proposed user interface.

Keywords: Network Management; Multi-Controller SDN; Traffic Engineering; WAN; ONOS