Archive of SID

مقاله پژوهشی

ارائه یک الگوریتم تعادل بار مبتنی بر پیش‌بینی در شبکه‌های نرم‌افزارمحور

حسین محمدی و سید اکبر مصطفوی

چکیده: شبکه‌های نرم‌افزارمحور یک معماری جدید در شبکه است که لایه کنترل را از لایه داده جدا می‌سازد. در این رویکرد مسئولیت لایه کنترل به نرم‌افزار کنترلر واگذار می‌شود تا رفتار کل شبکه را به طور پویا تعیین نماید. نتیجه این امر، ایجاد یک شبکه بسیار منعطف با مدیریت متمرکز است که در آن می‌توان پارامترهای شبکه را به خوبی کنترل کرد. با توجه به افزایش روزافزون کاربران، ظهور فناوری‌های جدید، رشد انفجاری ترافیک در شبکه، برآورده‌سازی الزامات کیفیت خدمات و جلوگیری از کم‌باری یا پرباری منابع، تعادل بار در شبکه‌های نرم‌افزارمحور ضروری می‌باشد. عدم تعادل بار باعث بالارفتن هزینه، کاهش مقیاس‌پذیری، انعطاف‌پذیری، بهره‌وری و تأخیر در سرویس‌دهی شبکه می‌شود. تا کنون الگوریتم‌های مختلفی برای بهبود عملکرد و تعادل بار در شبکه ارائه شده‌اند که معیارهای متفاوتی مانند انرژی مصرفی و زمان پاسخ سرور را مد نظر قرار داده‌اند، اما اغلب آنها از ورود سیستم به حالت عدم تعادل بار جلوگیری نمی‌کنند و خطرات ناشی از عدم تعادل بار را کاهش نمی‌دهند. در این مقاله، یک روش تعادل بار مبتنی بر پیش‌بینی برای جلوگیری از ورود سیستم به حالت عدم تعادل بار با بهره‌گیری از الگوریتم ماشین یادگیری افراطی پیشنهاد می‌شود. نتایج ارزیابی روش پیشنهادی نشان می‌دهد که از نظر تأخیر پردازش کنترل‌کننده، میزان تعادل بار و زمان پاسخ‌گویی به علت تعادل بار بهینه نسبت به روش‌های CDAA و PSOAP عملکرد بهتری دارد.

***کلیدواژه:* شبکه‌های نرم‌افزارمحور، توازن بار، الگوریتم‌های پیش‌بینی، ماشین یادگیری افراطی.**

۱- مقدمه

شبکه‌های نرم‌افزارمحور یک معماری جدید در شبکه است که لایه کنترل را از لایه داده جدا می‌سازد و مدیریت شبکه را به صورت متمرکز و انعطاف‌پذیر تحقق می‌بخشد. گسترش شبکه‌های کامپیوتری و افزایش روزافزون کاربران و همچنین ظهور فناوری‌های جدید مانند رایانش ابری و داده‌های حجیم، مدیریت شبکه‌های سنتی را دشوار ساخته و بنابراین لازم است که معماری شبکه‌های سنتی تغییر یابد. اخیراً برای پرداختن به حل مسئله موجود، رویکرد شبکه‌های نرم‌افزارمحور ارائه شده است که باعث سهولت در مدیریت شبکه و انعطاف‌پذیری و سازگاری بیشتر شبکه می‌شود [۱].

این مقاله در تاریخ ۷ اسفند ماه ۱۳۹۹ دریافت و در تاریخ ۱۹ مهر ماه ۱۴۰۰ بازنگری شد.

حسین محمدی، دانشکده مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران، (email: mohammadi.qezel@gmail.com).

سید اکبر مصطفوی (نویسنده مسئول)، دانشکده مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران، (email: a.mostafavi@yazd.ac.ir).

۱-۱ بیان مسئله

با در نظر گرفتن سربار شبکه، تکنیک‌های تعادل‌سازی بار[1] از اهمیت قابل توجهی برخوردار هستند. تعادل بار مستقیماً بر برنامه و دسترسی به سرویس برای کاربران تلفن همراه تأثیر می‌گذارد [۲] و با هدف بهینه‌سازی استفاده از منبع با به حداکثر رساندن توان عملیاتی، به حداقل رساندن زمان پاسخ و جلوگیری از بارگذاری بیش از حد هر منبع است. برای کاهش بار ترافیکی شبکه و کاهش خطر تبدیل‌شدن یک سرور به عامل اصلی سربار، بسیاری از مراکز داده از روش‌های سخت‌افزاری اختصاصی استفاده می‌کنند تا تعادل بار را برای پشتیبانی از تعداد زیادی از کاربران امکان‌پذیر سازند [۳]. با وجود این، تهیه سیستم‌های سخت‌افزاری معمولاً گران هستند و استقرار آنها از نظر فنی چالش‌برانگیز است و برای کار مداوم نیاز به مداخله انسان دارد [۴]. متعادل‌کننده‌های بار سنتی، پرهزینه هستند و مجموعه سیاست‌های متعادل‌کننده بار باید از قبل در آنها برنامه‌دهی شوند و این عدم انعطاف‌پذیری منجر به ناتوانی در مقابله با شرایط اضطراری می‌شود. رویکرد تعادل بار سنتی نیاز به نگهداری توسط مدیران اختصاصی دارد و به کاربران اجازه نمی‌دهد استراتژی‌های انعطاف‌پذیر را بر اساس شرایط شبکه واقعی خود طراحی کنند. از آنجا که همه درخواست‌ها از طریق یک سخت‌افزار هدایت می‌شوند، هر گونه خرابی در تعادل بار باعث از دست رفتن کل سرویس می‌گردد [۵].

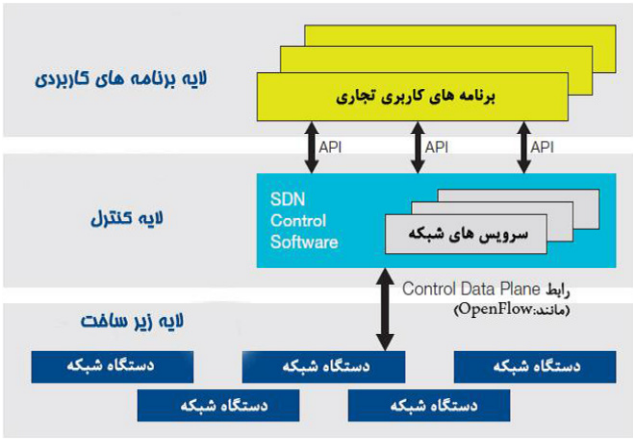
تعادل بار، روشی مطمئن برای افزایش پهنای باند سرورها و سایر دستگاه‌های شبکه و افزایش ظرفیت پردازش بسته داده‌ها و توان شبکه فراهم می‌کند تا قابلیت استفاده و انعطاف‌پذیری شبکه بهبود یابد. سرعت رشد پردازنده سرور و دسترسی به حافظه در مقایسه با توسعه سریع فناوری شبکه نسبتاً کند است. در حال حاضر، پرباری سرورها گلوگاه اصلی توسعه شبکه‌های کامپیوتری است. با توسعه شبکه‌های پرسرعت و افزایش تقاضا برای خدمات، بسیاری از مراکز داده سازمانی و سرورهای پورتال تحت تأثیر رشد انفجاری ترافیک شبکه قرار می‌گیرند. در این حالت، تعادل بار راهکار کلیدی است که برای توزیع تقاضای داده‌ها در بین مجموعه‌ای از سیستم‌های سرور مورد استفاده قرار می‌گیرد [۶] و [۷].

در معماری شبکه سنتی، از تعادل بار بین سرورهای فیزیکی برای برآوردن تقاضای حجم بالای داده استفاده می‌شود. این روش ضمن ارائه مقیاس‌پذیری و انعطاف‌پذیری، به سرمایه‌گذاری بزرگ نیاز دارد و به سختی قادر است از تقاضای بار کاری بسیار پویا از کاربران عظیم موبایل پشتیبانی کند. برای حل این مشکلات، این مقاله به ارائه یک معماری

1. Load Balancing

www.SID.ir

Archive of SID



شکل ۱: معماری SDN [۱۰].

مبتنی بر شبکه SDN[1] و به کارگیری یک روش احتمالاتی جدید برای تعادل بار بر اساس تحلیل واریانس مبتنی بر ماشین یادگیری افراطی می‌پردازد. از این روش می‌توان برای مدیریت پویای جریان‌های ترافیک جهت پشتیبانی از کاربران عظیم سیار در شبکه‌های SDN استفاده کرد. در این تحقیق به جای استفاده از رویکرد سنتی تعویض سخت‌افزار، راه حلی مبتنی بر سوئیچینگ مجازی OpenFlow پیشنهاد می‌گردد. یک کنترل‌کننده SDN با استفاده از تحلیل واریانس مبتنی بر الگوریتم ELM[2]، ترافیک داده‌های هر پورت سوئیچ را رصد می‌کند و یک الگوریتم انتخاب مبتنی بر احتمال برای هدایت ترافیک به صورت پویا با فناوری OpenFlow فراهم می‌نماید. این راه حل در مقایسه با روش‌های موجود متعادل‌سازی بار که برای پشتیبانی از شبکه‌های سنتی طراحی شده‌اند دارای هزینه کمتر، قابلیت اطمینان بالاتر و مقیاس‌پذیری بیشتری است که نیازهای کاربران سیار را برآورده می‌کند.

۱-۲ ضرورت و اهمیت تحقیق

صفحه کنترلی نقش بسزایی در شبکه نرم‌افزارمحور دارد. یک SDN بزرگ معمولاً صفحه کنترلی خود را با چندین کنترل‌کننده توزیع‌شده پیاده‌سازی می‌کند، هر کدام زیرمجموعه‌ای از سوئیچ‌ها را کنترل می‌کنند و با دیگر کنترل‌کننده‌ها همگام‌سازی می‌شوند تا یک دید یکپارچه از شبکه را فراهم کنند. در شرایط نوسان ترافیک شبکه، رابطه نگاشت استاتیک کنترل‌کننده و سوئیچ می‌تواند منجر به عدم تعادل بار کاری شود. کنترل‌کننده‌ها ممکن است دچار پرباری شده و درخواست‌های جدید را رد کنند و در نتیجه توانایی آنها برای پردازش درخواست‌های کنترلی کاهش یابد. اکثر طرح‌های موجود برای مدیریت نگاشت بین کنترل‌کننده‌ها و سوئیچ‌ها متکی به الگوریتم‌های بهینه‌سازی تکرارشونده هستند که از نظر عملکرد، زمان‌بر می‌باشند یا نتیجه رضایت‌بخشی به همراه ندارند. در این مقاله، با به کارگیری الگوریتم ELM، یک طرح پویای تعادل بار کنترل‌کننده بهینه پیشنهاد می‌شود. این مدل در دو مرحله عمل می‌کند: آموزش و تصمیم‌گیری. در مرحله آموزش، هر عامل یاد می‌گیرد که چگونه سوئیچ‌ها را از طریق تعامل با شبکه منتقل کند. در مرحله تصمیم‌گیری، رویکرد پیشنهادی برای تصمیم‌گیری در مورد سوئیچ‌های مهاجر مستقر می‌شود.

1. Software Defined Network
2. Extreme Learning Machine

۱-۳ نوآوری در تحقیق

نوآوری محوری این تحقیق، استفاده از الگوریتم ELM بهینه برای تعادل بار جهت کاهش استفاده از منابع در شبکه‌های SDN است. روش پیشنهادی، یک مکانیسم تعادل بار سرویسگرا جهت مشکل استقرار هماهنگ چند کنترل‌کننده با هدف بهبود تعادل بار است.

۱-٤ ساختار مقاله

ساختار مقاله به صورت زیر سازماندهی شده است: در بخش ۲ مبانی نظری تحقیق به همراه مروری بر مقالات مرتبط ارائه می‌گردد که شامل رویکردهای پیشین در این حوزه می‌شود. در بخش ۳، به حل چالش تعادل بار در کنترلرهای SDN با روش پیشنهادی پیش‌بینی مبتنی بر ELM پرداخته می‌شود. در بخش ۴ شبیه‌سازی رویکرد پیشنهادی و نتایج به دست آمده مورد بحث قرار می‌گیرد و نهایتاً در بخش پنجم، نتیجه‌گیری و کارهای آینده بیان می‌گردد.

۲- مبانی نظری و پیشینه تحقیق

در این بخش به بررسی مبانی نظری در زمینه شبکه‌های نرم‌افزارمحور و الگوریتم ماشین یادگیری افراطی پرداخته می‌شود و سپس کارهای مرتبط در زمینه تعادل بار در SDN مورد بررسی قرار خواهند گرفت.

۲-۱ شبکه‌های نرم‌افزارمحور

SDN یک فناوری متحول‌کننده در حوزه شبکه‌های رایانه‌ای است که در سال‌های اخیر توجه زیادی را به خود جلب کرده است. ایده اولیه این طرح به یک پروژه در سال ۲۰۰۸ در دانشگاه‌های برکلی و استنفورد بازمی‌گردد [۵]. SDN به محققان شبکه امکان می‌دهد تا خدمات شبکه را از طریق انتزاع عملکردهای سطح پایین شبکه به شکل ساده‌تری مدیریت کنند. این رویکرد، کنترل شبکه را که در مورد نحوه ارسال ترافیک تصمیم‌گیری می‌کند (صفحه کنترل) از سیستم‌های انتقال داده که ترافیک را به مقصد انتخاب‌شده هدایت می‌کنند (صفحه داده)، جدا می‌نماید. در نتیجه، شبکه به طور مستقیم توسط مدیر شبکه قابل برنامه‌ریزی است و زیرساخت شبکه از برنامه‌ها و خدمات شبکه تفکیک می‌شود [۶] و [۸]. معماری شبکه‌های موجود، به گونه‌ای طراحی نشده‌اند که نیازهای کنونی شرکت‌ها و کاربران را برطرف کنند. برای رفع نیاز این صنایع، معماری SDN به تدریج به عنوان یک استاندارد در طراحی مسیریاب‌های شبکه مورد استفاده قرار می‌گیرد [۹].

۲-۱-۱ معماری شبکه‌های نرم‌افزارمحور

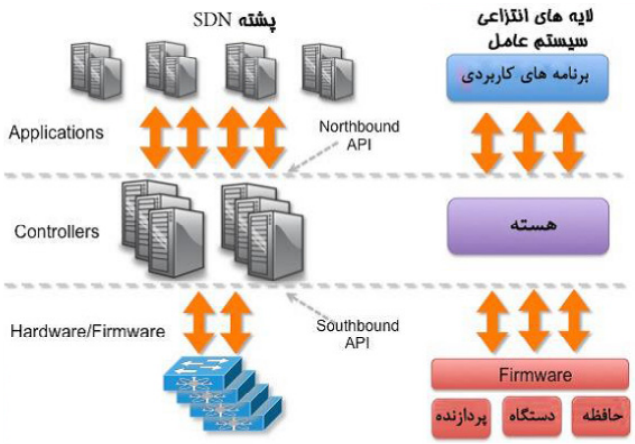
در معماری SDN، کنترل شبکه از هدایت ترافیک مجزا بوده و به طور مستقیم برنامه‌ریزی می‌شود. مجزاسازی کنترل شبکه که قبلاً محدود به سخت‌افزار شبکه بود، ماشین‌های مجازی و زیرساخت‌های شبکه را قادر می‌سازد تا انواع سرویس‌های جدید را تعریف کنند و با طیف جدیدی از برنامه‌های کاربردی ارتباط برقرار نماید. شکل ۱، نمایی از معماری SDN را ارائه کرده است. هوشمندی شبکه به طور منطقی در کنترل‌کننده‌های نرم‌افزاری SDN قرار دارد که ساختار کلی شبکه را کنترل می‌کند و شبکه از دید برنامه‌های کاربردی به صورت یک سوئیچ منطقی و واحد به نظر خواهد رسید [۱۰].

۲-۱-۲ اجزای تشکیل‌دهنده SDN

ایده اصلی SDN این است که یک واحد منطقی به نام کنترل‌کننده با همه دستگاه‌های موجود در یک دامنه شبکه در ارتباط مستقیم باشد، از

www.SID.ir

Archive of SID



شکل ۲: کنترل‌کننده‌ها در SDN [۱۲].

هم‌بندی شبکه آگاه باشد و شبکه را از یک نقطه مرکزی، هدایت، کنترل و برنامه‌ریزی کند. کنترل‌کننده SDN، مدل برنامه‌ریزی شبکه را از حالت توزیع‌شده به حالت متمرکز تبدیل می‌کند. در شکل ۲، کنترل‌کننده SDN به عنوان یک میان‌ابزار جهت برقراری ارتباط به کار گرفته شده است [۱۱] و [۱۲].

۲-۱-۳ پروتکل OpenFlow

پروتکل OpenFlow یک عنصر اساسی برای ساخت راه حل‌های SDN است. این اولین پروتکل ارتباطی استاندارد است که بین لایه کنترل و لایه زیرساخت در معماری SDN تعریف شده است. OpenFlow از مفهوم جریان برای شناسایی ترافیک شبکه بر اساس قوانین تطبیقی استفاده می‌کند که می‌تواند به صورت ایستا یا پویا توسط نرم‌افزار کنترلی SDN برنامه‌ریزی شود و به عنوان پرکاربردترین واسط برنامه‌نویسی برای شبکه‌های SDN، منابع اطلاعاتی برای سیستم عامل‌های شبکه را فراهم می‌کند [۱۳].

۲-۱-۴ تعادل بار در SDN

با در نظر گرفتن سربار شبکه، تکنیک‌های متعادل‌سازی بار از اهمیت قابل توجهی برخوردار هستند. تعادل بار مستقیماً بر برنامه و دسترسی به سرویس برای کاربران تأثیر می‌گذارد. هدف از تعادل بار، بهینه‌سازی استفاده از منبع با به حداکثر رساندن توان عملیاتی، به حداقل رساندن زمان پاسخ و جلوگیری از بارگذاری بیش از حد روی هر منبع واحد است. برای کاهش بار ترافیک و کاهش خطر تبدیل سرور به نقطه حساس به خطا، بسیاری از مراکز داده روش‌های سخت‌افزاری اختصاصی را برای فعال‌کردن تعادل بار برای پشتیبانی از تعداد زیادی از کاربران در نظر می‌گیرند. با وجود این، تهیه سیستم‌های سخت‌افزاری معمولاً گران هستند، استقرار آنها از نظر فنی چالش‌برانگیز است و برای کار مداوم نیاز به مداخله انسان دارند.

شبکه SDN یک روش جایگزین ساده‌تر، آسان‌تر و قابل کنترل برای کنترل جریان شبکه با صرف سرمایه‌گذاری کمتر است. در SDN هنگامی که یک جریان داده به یک سوئیچ می‌رسد، یک جستجوی جدول جریان باید انجام شود. در صورت نیاز به تغییرات جریان یا اعمال اقدامات، سرایندها و شمارنده‌ها برای هر جریان شبکه به‌روز می‌شوند. با ثبت اطلاعات سرایندها در یک پایگاه داده، یک سوئیچ OpenFlow می‌تواند جریان داده را با توجه به سوابق سرآیند پردازش کند. در مدل SDN با یک کنترل‌کننده متمرکز، یک سوئیچ OpenFlow برای قوانین مختلف برای کنترل ترافیک شبکه با استفاده از سوابق سرایندها طراحی شده است. سیستم کنترل جریان، امکان تعریف الگوریتم و برنامه‌نویسی برای تعادل بار شبکه را فراهم می‌کند [۷].

۲-۲ ماشین یادگیری افراطی (ELM)

از جمله مسایل مهم در روش‌های یادگیری، سرعت آموزش و آزمون است. در [۱۴] یک الگوریتم یادگیری به نام ماشین یادگیری افراطی یا ELM ارائه شده که سرعت یادگیری آن بسیار بیشتر از الگوریتم‌های یادگیری مانند مارکارد (LM)[1] و پس‌انتشار خطا[2] است. از ELM برای پردازش‌های متنوعی شامل طبقه‌بندی، رگرسیون[3]، خوشه‌بندی، انتخاب ویژگی، کاهش ابعاد و استخراج ویژگی‌ها استفاده می‌شود. سرعت یادگیری ELM در مقابل شبکه‌های عصبی پس‌خور مبتنی بر گرادیان کاهشی به طور قابل توجهی بهتر است و این الگوریتم عملکرد بسیار بهتری نسبت به بردار ماشین پشتیبان (SVM) در زمینه رده‌بندی و رگرسیون از خود نشان می‌دهد [۱۵]. با توجه به سرعت اجرا و بازدهی بالای الگوریتم ELM، این الگوریتم قادر است تا عملیات یادگیری کم‌باری و پرباری کنترلرهای سوئیچ‌های SDN را بسیار سریع به انجام برساند و پیش از بروز تأخیر قابل توجه، شرایط شبکه را اصلاح نماید.

الگوریتم ELM را می‌توان بر روی داده‌های مصنوعی که دو دسته یا بیشتر باشند و با هم، همپوشانی نداشته باشند، اجرا نمود و به صحت کامل آن رسید. اما این شرایط ایده‌آل برای داده‌های واقعی عموماً برقرار نیست. الگوریتم ELM برای آموزش شبکه‌های عصبی پیش‌خور[4] تک‌لایه پنهان با اجتناب از یک روش آموزش تکراری زمان‌بر با هدف کاهش زمان آموزش و بهبود عملکرد تعمیم‌پذیری داده‌ها ارائه و پیشنهاد شده است. در ELM فقط وزن‌های خروجی بین لایه‌های پنهان و لایه خروجی با روش حداقل مربعات منظم، قابل تنظیم است و پارامترهای لایه پنهان که همان وزن‌های ورودی بین لایه ورودی، لایه پنهان و بایاس‌ها است، هم‌زمان به طور تصادفی تولید می‌شوند و در طول فاز آموزش، ثابت می‌مانند [۱۶] و [۱۷].

۲-۳ مطالعات پیشین

مقالات متعددی بحث تعادل بار در SDN را مورد بررسی قرار داده‌اند. هونگ ژونگ و همکاران به ارائه یک طرح تعادل بار SDN کارامد بر اساس زمان پاسخ سرور پرداخته‌اند که زمان پاسخ مهم‌ترین عامل تعیین‌کننده تجارب کاربر در مدل ارائه خدمات شامل خوشه‌های سرور است. با این حال، طرح تعادل بار خوشه سرور سنتی تحت شرایط سخت‌افزاری محدود است و نمی‌تواند به طور کامل از زمان پاسخ سرور برای تعادل بار استفاده کند. به منظور حل مؤثر طرح‌های متعادل‌سازی بار سنتی، این مقاله با استفاده از مزیت انعطاف‌پذیری SDN به نام LBBSRT یک طرح متعادل‌سازی بار را بر اساس زمان پاسخ سرور پیشنهاد می‌کند. با استفاده از زمان پاسخ به موقع هر سرور که توسط کنترل‌کننده برای تعادل بار اندازه‌گیری می‌شود، با به دست آوردن بارهای سرور به طور مساوی، درخواست‌های کاربر پردازش می‌گردد. آزمایش‌های شبیه‌سازی نشان می‌دهد که طرح این تحقیق با حداقل میانگین زمان پاسخ سرور، اثر متعادل‌کننده بار و درخواست‌های پردازش بهتری را نشان می‌دهد. علاوه بر این، اجرای طرح پیشنهادی این تحقیق آسان است و

1. Levenberg-Marquardt
2. Backpropagation
3. Regression
4. Feed-Forward

www.SID.ir

Archive of SID

دارای مقیاس‌پذیری خوب و ویژگی‌های کم‌هزینه می‌باشد [۱۸].

وی چی چین و همکاران یک ساختار تعادل بار در SDN را مبتنی بر اینترنت اشیا ارائه داده‌اند. در این تحقیق، یک مکانیسم تعادل بار SDN-SFC سرویس‌گرا پیشنهاد شده است. این روش، نوع و اولویت خدمات مورد نیاز هر دستگاه ترمینال را در نظر گرفته و طبقه‌بندی می‌کند. سپس، الگوریتم ابتکاری برای برنامه‌ریزی مسیرهای انتقال بین SFC برای کاهش بار هر SF و بهبود عملکرد کلی شبکه، تصویب و اعمال شده است. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی این مطالعه می‌تواند زمان انتقال داده را کوتاه کرده و به تعادل بار برسد [۱۹].

ال بوئرو و همکاران تعادل بار و مدیریت مهلت صف‌ها را در یک سوئیچ OpenFlow ارائه دادند که BreQoS نام دارد. ویژگی‌های فعلی OpenFlow قادر به تنظیم نرخ خدمات صف در دستگاه‌های OpenFlow نیست. این کمبود اجازه نمی‌دهد که بتوان الگوریتم‌هایی را جهت حفظ الزامات کیفیت خدمات در جریان‌های جدید و موجود پیاده‌سازی نمود. در این مقاله یک راه حل جایگزین ارائه شده که از طریق برخی تغییرات Beacon، یک کنترل‌کننده SDN اجرا گردیده است. این روش به صورت زیر عمل می‌کند: با استفاده از آمار تقریباً آنی OpenFlow Beaconها مسیرهای مختلف را در صف‌های مختلف مسیریابی می‌کند تا رعایت الزامات ضرب‌الاجل را تضمین کند. این کار، یک روش تعادل صف کارامد در یک سوئیچ OpenFlow است که هیچ نوع تغییر ابتکاری یا استاندارد جدیدی نسبت به سوئیچ استاندارد OpenFlow ندارد. سازوکار پیشنهادی این تحقیق که در کنترل‌کننده اجرا می‌شود، با دستگاه‌های OpenFlow معمولی کار می‌کند. تغییرات ایجادشده در کنترل‌کننده SDN، مبنایی برای طراحی دسته‌ای از الگوریتم‌های مسیریابی جدید است که قادر به تضمین محدودیت‌های مهلت و تعادل صف بدون تغییر در مشخصات OpenFlow و همچنین دستگاه‌های OpenFlow هستند [۲۰].

شیرزاد شهریاری و همکاران چارچوبی مبتنی بر SDN برای به حداکثر رساندن توان توزیع و توزیع بار در شبکه ابری ارائه داده‌اند. اگرچه دستگاه‌های سیار در دهه اخیر گسترش قابل ملاحظه‌ای را تجربه کرده‌اند، اما منابع آنها به دلیل اندازه قابل حمل آنها محدود هستند. چنین محدودیت‌هایی را می‌توان با اجرای از راه دور وظایف محاسباتی روی ابر کاهش داد. با ایجاد یک خوشه سرور ابر در لبه شبکه و نزدیک به دستگاه‌های سیار، واگذاری وظیفه با تأخیر قابل قبول‌تری در مقایسه با یک راه حل مبتنی بر ابر انجام می‌شود. به محض این که کاربر، درخواستی را ارسال کند، محدودیت منابع در لبه منجر به کمبود منابع خواهد شد. در این شرایط می‌توان با به کارگیری شبکه ابری جهت به اشتراک‌گذاری منابع، از این چالش جلوگیری کرد. در این مقاله چارچوبی برای مدیریت بهینه منابع و ایجاد تعادل بار عادلانه در شبکه ابری از طریق تکنیک‌های شبکه SDN ارائه شده است. برای دستیابی به این هدف، ابتدا مسئله به عنوان یک مدل بهینه‌سازی برنامه‌ریزی خطی (MILP)[1] مختلط مدل‌سازی شده است تا توزیع وظایف مستقلی را که از دستگاه‌های تلفن همراه واگذار شده‌اند همراه با استفاده بهینه از منابع، متعادل سازد. مدل MILP رعایت ضرب‌الاجل وظایف را تضمین می‌کند و گذردهی سیستم را به حداکثر می‌رساند. در مرحله دوم، با نشان‌دادن این که این مسئله NP- سخت است، یک مدل هموارسازی برنامه‌ریزی خطی برای فعال‌سازی کنترل‌کننده SDN در یک شبکه در مقیاس بزرگ ارائه شده است. جهت ارزیابی، آزمایش‌هایی با الگوبرداری از چارچوب پیشنهادی در Mininet-WiFi، با استفاده از Floodlight به عنوان کنترل‌کننده SDN انجام شده است. نتایج شبیه‌سازی نشان می‌دهد که معماری این روش می‌تواند به طور رضایت‌بخشی تعادل بار را انجام دهد و اثبات کافی را نیز ارائه می‌دهد [۲۱].

در زمینه جایگذاری و ارائه کنترل‌کننده در شبکه‌های SDN، مطالعات مختلفی صورت گرفته است. عبدالحمید آتیا و همکاران یک الگوریتم ازدحامی زنجیره‌وار به صورت آشوبناک برای ارائه چندین کنترل‌کننده در SDN ارائه نموده‌اند. این الگوریتم که اصطلاحاً SSOA یا الگوریتم ازدحامی زنجیره‌ای آشوبناک[2] نام دارد، به ارائه یک نگاشت آشوبناک برای بهینه‌سازی کارایی شبکه SDN پرداخته است. این الگوریتم به ارزیابی پویای تعداد بهینه کنترل‌کننده‌ها و بهبود ارتباطات بین سوئیچ‌ها و کنترل‌کننده‌ها در ابعاد بزرگ SDN می‌پردازد. نتایج حاصل نشان از بهبود رویکرد پیشنهادی نسبت به سایر الگوریتم‌های تکاملی و هوش ازدحامی و همین طور نظریه بازی‌ها از لحاظ زمان اجرا و پایایی دارد [۲۲].

در تحقیق دیگری، مینگ چن و همکاران یک پروتکل با مکانیزم سبک‌وزن همکارانه تحت عنوان LCMSC برای جایگذاری و بهبود کنترل‌کننده‌ها در SDN ارائه نموده‌اند. کاهش پیچیدگی محاسباتی، بهبود زمان اجرا، مدیریت و نظارت بر منابع و بهبود پایداری به عنوان مهم‌ترین مشخصه‌ها و اهداف این تحقیق می‌باشند. اتصال کنترل‌کننده LCMSC به اینترنت و پشتیبانی از خدمات در ساختار شبکه آنلاین به صورت فیزیکی، یکی از دستاوردهای مهم این تحقیق به شمار می‌رود [۲۳].

تیانژو ژانگ و همکاران به نقش اجماع کنترل‌کننده‌های درونی برای استقرار و جایگذاری توزیعی کنترل‌کننده‌های SDN پرداخته‌اند. این عملیات در یک شرکت ارائه‌دهنده خدمات اینترنتی به صورت فیزیکی ارائه و انجام شده که با بهبود کیفیت خدمات به خصوص پهنای باند همراه بوده است. همین طور تخمین زمان واکنش که سوئیچ‌ها تشخیص می‌دهند، با استفاده از الگوریتم‌های تکاملی و هوش ازدحامی برای اعتبارسنجی عملیات SDWAN[3] انجام گرفته است. استفاده از کنترل‌کننده‌های OpenDaylight و ONOS برای استقرار و جایگذاری در شبکه SDN مورد بحث این تحقیق بوده است [۲۴].

پینک سانگ و همکاران یک پروتکل ذره‌ای به صورت موازی برای ارائه کنترل‌کننده در SDN برای ابعاد بزرگ مطرح کردند. بهبود کیفیت خدمات و مسئله مسیریابی به همراه کاهش پیچیدگی محاسباتی و قابلیت مقیاس‌پذیری، از مهم‌ترین اهداف این تحقیق به شمار می‌رود. نام کنترل‌کننده پروتکلی این تحقیق، ParaFlow می‌باشد که دارای قابلیت اجرا در هسته‌های مختلف نرم‌افزاری و همین طور حداکثر بهره‌وری در استفاده از منابع می‌باشد. ایجاد یک محیط رابط کاربری در مقایسه با اکثر پروتکل‌های کنترلی که به صورت رخدادپذیر هستند، یک مسئله مهم می‌باشد که ParaFlow آن را در نظر گرفته است [۲۵].

جی کوی و همکاران به حل مسایل مربوط به مقیاس‌پذیری و قابلیت اطمینان می‌پردازند که کنترل‌کننده متمرکز فاقد آنها است. این امر باعث می‌شود تا ساختار کنترل‌کننده‌های توزیع‌شده پدیدار شوند. یکی از محدودیت‌های اصلی کنترل‌کننده‌های توزیع‌شده، نگاشت سوئیچ و کنترل‌کننده با تنظیمات ثابت است که به راحتی باعث توزیع ناهموار بار در میان کنترل‌کننده‌ها می‌شود. برای رفع این مشکلات، یک طرح تعادل بار برای چندین کنترل‌کننده شبکه‌های نرم‌افزارمحور مبتنی بر زمان پاسخ

1. Mixed-Integer Linear Programming

2. Salp Swarm Optimization Algorithm

3. Operational Software Defined WAN

www.SID.ir

Archive of SID

جدول ۱: مقایسه برخی روش‌های پیشین.

رویکرد پیشنهادی	مزایا	معایب
ارائه یک طرح تعادل بار بر اساس زمان پاسخ سرور	کاهش زمان پاسخ سرور و تعادل بار مؤثر	در نظر نگرفتن صرفه‌جویی انرژی در تعادل بار
ارائه یک مکانیسم تعادل بار SDN-SFC سرویس‌گرا	بهبود عملکرد کلی شبکه و کاهش زمان انتقال داده	اگر همه بسته‌ها، بسته‌های ثابت یا غیر ثابت باشند، منابع کاهش می‌یابد.
تعادل بار و مدیریت مهلت صف‌ها در سوئیچ‌ها (BeaQoS)	تعادل صف بدون تغییر در مشخصات و ایجاد تعادل بار نسبی	عدم مقیاس‌پذیری، عملکرد پایین و تراکم زیاد شبکه
ارائه یک الگوریتم ازدحامی زنجیره‌وار به صورت آشوبناک	بهبود زمان اجرا و پایایی	پیچیدگی محاسباتی بالا و نامناسب‌بودن نتایج تعادل بار
یک پروتکل LCMSC برای جایگذاری و بهبود کنترل‌کننده‌ها در SDN	کاهش پیچیدگی محاسباتی و بهبود زمان اجرا، مدیریت و نظارت بر منابع	عدم مقیاس‌پذیری و نامناسب‌بودن نتایج تعادل بار و عدم بهبود مصرف انرژی
اجماع کنترل‌کننده‌های درونی برای استقرار و جایگذاری توزیعی واپایشگرهای SDN مبتنی بر الگوریتم‌های تکاملی	بهبود کیفیت خدمات به خصوص پهنای باند و تخمین زمان واکنش	پیچیدگی محاسباتی بالا، عدم مقیاس‌پذیری، عدم بهره‌وری از منابع و نامناسب‌بودن نتایج تعادل بار
پروتکل ذره‌ای به صورت موازی برای ارائه کنترل‌کننده در SDN برای ابعاد بزرگ با پروتکل ParaFlow	بهبود کیفیت خدمات، بهبود مسئله مسیریابی، قابلیت مقیاس‌پذیری و حداکثر بهره‌وری در استفاده از منابع	عدم بهبود مصرف انرژی و عدم تخمین زمان واکنش
یک طرح تعادل بار برای چندین کنترل‌کننده مبتنی بر زمان پاسخ که به نام SMCLBRT ارائه دادند.	مقیاس‌پذیری، قابلیت اطمینان و تعادل بار مناسب	این شبیه‌سازی در بیشتر الگوهای توزیع بار اعمال نمی‌شود و هزینه مهاجرت را در نظر نگرفتند.
رویکرد یادگیری تقویتی عمیق مبتنی بر شبکه Q برای جایگذاری کنترلر	ترکیب یادگیری از داده‌های پیشین شبکه و داده‌های آنی	پیچیدگی محاسباتی بالایی دارد.
چارچوبی برای تحلیل رویکردهای مختلف جایگذاری کنترلر	ارزیابی کمی تمامی روش‌های موجود بر اساس معیارهای مختلف	عدم ارائه یک راهکار یادگیرنده و سریع برای جایگذاری کنترلرها
ترکیب شبکه عصبی و یادگیری تقویتی برای جایگذاری کنترلر	استفاده از نسبت بهره‌وری منابع به جای بهینه‌سازی خطی	دقت پایین‌تر نسبت به سایر روش‌های موجود
الگوریتم مکاشفه‌ای مبتنی بر یادگیری اتوماتا برای جایگذاری کنترلر	الگوریتم ساده و سریع	صرفاً تأخیر انتشار بین کنترلر و سوئیچ‌ها را در نظر گرفته است.

با در نظر گرفتن تغییر در ویژگی‌های زمان حقیقی در مقابل بارهای کنترل‌کننده پیشنهاد کردند. این روش با انتخاب حد آستانه زمان پاسخ مناسب و توزیع آنی بار روی کنترل‌کننده‌های پربار کار می‌کند تا مشکل تعادل بار را در سطح کنترل شبکه‌های نرم‌افزارمحور با چندین کنترل‌کننده با بار اضافی حل کند [۲۶]. داساری و همکاران از رویکرد مبتنی بر SDN برای مدیریت شبکه کوانتومی جهت تضمین مبادله اطلاعات بین دستگاه‌ها استفاده نموده‌اند [۲۷]. وو و همکاران [۲۸] یک رویکرد یادگیری تقویتی عمیق مبتنی بر شبکه Q برای مسئله جایگذاری کنترلر را پیشنهاد کرده‌اند. در این روش از داده‌های پیشین شبکه و داده‌های آنی به صورت ترکیبی برای جایگذاری پویای کنترلرهای SDN در شرایط نوسان جریان ترافیکی استفاده می‌شود. تیان و همکاران [۲۹] چارچوبی را برای تحلیل رویکردهای مختلف جایگذاری کنترلر چندگانه در شبکه SDN ارائه و سپس الگوریتم‌های جایگذاری مختلف را به طور گسترده ارزیابی کرده‌اند. یئو و همکاران [۳۰] بر خلاف رویکردهای خطی موجود برای بهینه‌سازی جایگذاری کنترلر، از ترکیب شبکه عصبی و یادگیری تقویتی جهت یافتن پاسخ بهینه استفاده می‌کنند. در این روش، نسبت بهره‌وری منابع مختلف سوئیچ و کنترلر به عنوان ورودی شبکه عصبی مورد استفاده قرار می‌گیرند. مصطفایی و همکاران در [۳۱]، یک الگوریتم مکاشفه‌ای ساده را مبتنی بر اتوماتای یادگیر پیشنهاد می‌کنند که کنترلرها را بر اساس تأخیر انتشار از پیش تعیین شده بین سوئیچ و کنترلر جایگذاری می‌کند تا تأخیر انتشار کلی را کاهش دهد.

در جدول ۱ روش‌های مطرح موجود مورد مقایسه قرار گرفته و مزایا و معایب هر روش قید شده است.

۳- روش پیشنهادی

در این بخش مدل کاملی از مسئله تعادل بار در SDN ارائه می‌شود. این مقاله به ارائه یک ساختار جدید از تعادل بار در SDN مبتنی بر الگوریتم ماشین یادگیری افراطی (ELM) می‌پردازد که مدل‌سازی و فرموله‌سازی آن در این بخش به صورت کامل انجام می‌شود. مسئله مهاجرت سوئیچ‌ها در SDN به عنوان یک پروژه مهم در بخش تعادل بار مطرح است.

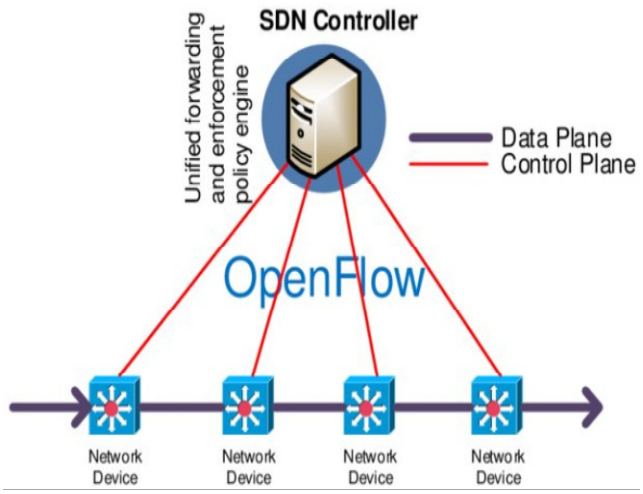
به صورت کلی، ساختار کنترل‌کننده SDN به صورت شکل ۳ است. همان طور که مشاهده می‌شود، بخش پایینی به پشتیبانی از پروتکل‌های چندگانه و توزیع‌شده همراه با تعدادی پلاگین می‌پردازد که شامل OpenFlow در نسخه‌های مختلف و BGP-LS است. این ماژول‌ها به صورت پویا در لایه انتزاعی خدمات (SAL)[1] نصب شده‌اند که می‌توانند پس از کپسوله‌سازی در یک پروتکل مناسب برای دستگاه‌های زیرین شبکه، مورد استفاده قرار گیرند.

کنترل‌کننده نیاز به جمع‌آوری اطلاعات درباره توابع دستگاه‌های زیرین شبکه و نحوه هدایت آنها دارد و این اطلاعات در سیستم مدیریت هم‌بندی، مورد استفاده قرار می‌گیرند. کنترل‌کننده به ارائه یک API سطح بالا برای برنامه‌های کاربردی در SDN می‌پردازد که جهت پشتیبانی برای چارچوب OSGI و رابط کاربری دوطرفه REST مورد استفاده واقع می‌گردد. این چارچوب به ارائه برنامه‌های اجرایی در فضای آدرس کنترل‌کننده عمل می‌کند که REST API به این بخش مهم اشاره

1. Service Abstraction Layer

www.SID.ir

Archive of SID



شکل ۳: کنترل‌کننده SDN [۲۷].

مستقیم دارد و می‌تواند در فضاهای آدرس‌های گوناگون، تنظیم و مورد استفاده واقع گردد. تمامی منطق‌ها و الگوریتم‌ها در این سطح از برنامه‌های کاربردی، اجرا می‌شوند. کنترل‌کننده همراه با یک محیط رابط گرافیکی به استفاده از همان APIها به عنوان برنامه‌های کاربردی می‌پردازد و این APIها می‌توانند توسط سایر برنامه‌های کاربردی، فراخوانی شده و مورد استفاده قرار گیرند.

زمانی که پروتکل کلاسیک OpenFlow برای تطبیق عملیات منطقی طراحی نشده بود، پلاگین‌های OpenFlow در نسخه ۱/۳ به امور مسیریابی، مباحث امنیتی، خوشه‌بندی، اعتمادپذیری، زمان‌بندی، تعادل بار، تحمل خطا و سایر عملیات‌ها می‌پرداختند. پلاگین OpenFlow۱.۰ در یک کنترل‌کننده مجتمع با سایر پلاگین‌های OpenFlow کلاسیک، طراحی شدند که به صورت OpenFlowj شناخته می‌شود. OpenFlowj یک کتابخانه ایستا از نوع جاوا یا OpenFlowJava برای پیغام‌های پروتکل OpenFlow۱.۰ می‌باشد که وابسته به پلاگین OpenFlow۱.۰ است. OpenFlowJava در پلاگین‌های OpenFlow۱.۳ فاقد کتابخانه برای ارسال داده‌ها بود و بنابراین از یک بسته پلاگین دیگر به نام YANG استفاده می‌کرد که جداگانه بر روی آن نصب می‌شد. همین طور دارای استقرار ارتباطی و کدهای پروتکل بود. OpenFlowPlugins بر اساس OpenFlowJava برای پیاده‌سازی فرایندهای ارسال و دریافت داده‌ها در سطح شبکه به کار گرفته شد.

این تحقیق بر اساس کنترل‌کننده‌های مبتنی بر OpenFlow، OpenFlowJava و OpenFlowPlugins است و به بحث کنترل‌پذیری صفحه کنترل می‌پردازد که تعادل بار بر اساس آن ارائه شده و ساختار کنترل‌کننده تعادل بار SDN از چارچوب OSGI برای پیاده‌سازی الگوریتم ELM استفاده می‌کند.

رویکرد ارائه‌شده در این مقاله، به کارگیری روش ELM در SDN جهت تعادل بار بین کنترل‌کننده‌ها است. بار ترافیکی شامل بسته‌های داده‌ای است که به صورت گروهی در محیط تولید می‌شوند و راه حل ارائه‌شده می‌تواند منجر به توزیع ترافیک و افزایش کنترل‌پذیری گردد. به دلیل تنوع کاربران استفاده‌کننده از SDN، سرایندهای IP دربسته‌های داده در هر جریان داده بر اساس توزیع‌های مختلف پروتکل TCP کار می‌کنند که مسئله زمان‌بندی و تعادل بار به این قضیه توجهی ندارند و می‌توان بر اساس صفحه کنترل SDN مبتنی بر OpenFlow آنها را شناسایی نمود. قابل ذکر است که نوع صف بسته‌ها از نوع M/M/۱ می‌باشد. البته می‌توان نوع صف را M/M/C و یا M/M/M در نظر گرفت، اما با توجه به این که دو معیار زمان‌بندی به نام زمان پاسخ و زمان انتظار در سیستم مد نظر می‌باشد، لذا مدل صف به صورت MM۱ در نظر گرفته شده است.

برای به کارگیری الگوریتم ELM، ابتدا ویژگی‌های مختلف جریان‌های ترافیکی شناسایی می‌شوند که این ویژگی‌ها به صورت گروهی از داده‌ها مدل‌سازی می‌گردند. فرض می‌شود که سیستم تحت شرایط نرمال است. اولویت کاربران برای ارسال و دریافت داده‌ها، بر کارایی شبکه تأثیر می‌گذارد که امری عادی است، اما وجود جریان‌های داده نامتعادل و ارسال آنها به صفحه کنترلی، یک مسئله دردسرساز است که علاوه بر کندکردن ارسال و دریافت‌ها، می‌تواند پهنای باند شبکه را اشغال نموده و عملکرد کنترل‌کننده را مختل نماید. اگر مقادیر ویژگی‌ها در مجموعه داده‌ها برابر با فرکانس بالای مقادیر ویژگی‌ها (یعنی بروز داده‌های نامتوازن در سطح شبکه) شود، عملکرد شبکه مختل می‌شود و بالعکس. سپس ترکیب ویژگی‌ها روی جریان داده نرمال سیستم محاسبه می‌گردد. باید توزیع مقادیر این ویژگی‌ها در ویژگی‌های مجموعه داده دارای فرکانس بالا و کارایی کلی شبکه برای ارسال داده‌ها، کاهش یابد که برای این کار باید از روش آموزش محوری مانند ELM استفاده نمود تا بتوان بر اساس اولویت صف در ارسال داده‌ها و استفاده از آزمودن فرضیات، هر گونه داده بار نامتوازن را تشخیص داد.

مدل مسئله مهاجرت سوئیچ‌ها به منظور بهبود تعادل بار به انتخاب شش ویژگی در کل حجم داده ورودی در سطح شبکه می‌پردازد که شامل TTL، نوع پروتکل، آدرس IP، آدرس IP مبدأ و مقصد و همین طور سرایند پرچم‌گذاری در بسته‌های TCP است که مجموعاً دو زوج از کل ۱۵ ویژگی زوج را در این بخش در ساختار SDN شامل می‌شود. منظور از مجموعه دو زوج، یعنی این که هر کدام از این ویژگی‌ها به صورت دوبه‌دو با هم مد نظر قرار می‌گیرند. قابل ذکر است که این ۱۵ ویژگی زوج به عنوان ساختارهای پروتکلی هستند، اما ۴ ویژگی دیگر نیز وجود دارند که برای تجزیه و تحلیل آماری روش استفاده می‌شوند که شامل نرخ جریان‌های معکوس‌پذیر، نرخ بسته‌های TCP، UDP و ICMP که در سرایند پرچم‌گذاری بسته‌های TCP نیز قرار دارد، درصد ارسال جریان داده‌ها در کوتاه‌مدت و همبستگی عناصر سرآمد بسته‌ها است. به صورت کلی، ۱۵ ویژگی شامل موارد زیر می‌باشد:

- نرخ جریان‌های معکوس‌پذیر TCP
- نرخ جریان‌های معکوس‌پذیر UDP
- نرخ جریان‌های معکوس‌پذیر ICMP
- نرخ بسته‌های TCP
- نرخ بسته‌های UDP
- نرخ بسته‌های ICMP
- سرایند پرچم‌گذاری بسته‌های TCP
- سرایند پرچم‌گذاری بسته‌های UDP
- سرایند پرچم‌گذاری بسته‌های ICMP
- درصد ارسال جریان داده‌ها در مدت زمان کوتاه
- درصد ارسال جریان داده‌ها در مدت زمان کوتاه همراه با هر گونه همبستگی عناصر سرآمد بسته‌ها
- همبستگی عناصر سرآمد بسته‌های TCP
- همبستگی عناصر سرآمد بسته‌های UDP
- همبستگی عناصر سرآمد بسته‌های ICMP
- تعیین TTL، نوع پروتکل، آدرس IP، آدرس IP مبدأ و مقصد و همین طور سرایند پرچم‌گذاری در بسته‌های TCP (فقط TCP)

www.SID.ir

Archive of SID

در ساختار SDN بدون حالت تعادل بار و عدم مهاجرت سوئیچ‌ها، نیاز به محاسبه فرکانس وقوع مهاجرت سوئیچ‌ها بر اساس بسته‌های داده نرمال و داده‌های جریانی و به روز رسانی جدول برای تشخیص عملیات تعادل بار می‌باشد. تابعی که به نمایش فرکانس وقوع مقادیر ویژگی‌ها از مجموعه داده‌های در سطح SDN می‌پردازد به فرم (۱) محاسبه می‌شود

$$conf(A_i = a_{i,j}) = \frac{N(A_i = a_{i,j})}{N_n} \quad (۱)$$

که در آن $conf$ یک تابع برای نمایش فرکانس وقوع مقادیر ویژگی‌ها از مجموعه داده‌ها در سطح SDN می‌باشد. $conf$ در واقع به صورت Content $f(x)$ است. طبق (۱)، A_i نشان‌دهنده iامین ویژگی در بسته داده‌ها است که $i = ۱,۲,۳,\ldots,n$ و $j = ۱,۲,۳,\ldots,m_i,n$ تعداد ویژگی‌های جدول را شامل می‌شود. m_i نشان‌دهنده ویژگی‌های A_i از مقادیر است و N_n به نمایش مجموع بسته‌های داده در داده‌های جریانی و $N(A_i = a_{i,j})$ به نمایش مقادیر ویژگی‌های A_i در داده جریانی می‌پردازد. همین طور i تعداد گروه‌های داده‌ای j است. معادله ویژگی‌ها در مقابل مقادیر به صورت (۲) محاسبه می‌شود

$$conf(A_i = a_{i,j_۱}, A_{i_۲} = a_{i_۲,j_۲}) = \frac{N(A_i = a_{i,j_۱}, A_{i_۲} = a_{i_۲,j_۲})}{N_n} \quad (۲)$$

طبق (۲)، $i_۱ = ۱,۲,۳,\ldots,n_۱$، $i_۲ = ۱,۲,۳,\ldots,n_۲$، $j_۱ = ۱,۲,۳,\ldots,m_۱$، $j_۲ = ۱,۲,۳,\ldots,m_۲$ و $(A_i = a_{i,j_۱}, A_{i_۲} = a_{i_۲,j_۲})$ یعنی مقادیر ویژگی‌ها ارزش یک صفت جفت دارند. این بدان معنا است که $a_{i,j_۱}$ یا $a_{i_۲,j_۲}$ که همان $f(x)$ یا تابع اصلی هستند، دارای سرایندهای دوگانه می‌باشند که برای a شامل اندیس‌های i و j و همین طور $i_۲$ و $j_۲$ است. A_i که جزو $a_{i,j_۱}$ و مقادیر ویژگی‌های $a_{i_۲,j_۲}$ است ارزش یک صفت جفت را دارد. $N(A_i = a_{i,j_۱}, A_{i_۲} = a_{i_۲,j_۲})$ گروهی از ویژگی‌ها در داده‌های جریانی با مقادیر $A_{i_۲} = a_{i_۲,j_۲}$ و $A_i = a_{i,j_۱}$ است.

جدول پروفایل در SDN، یک جدول اطلاعاتی برای جمع‌آوری ۱۵ ویژگی از داده‌های سیستمی و جریانی است که دارای مقادیر جفت از ویژگی‌ها است که اشاره به مقادیر Conf دارد. دلیل استفاده از پروفایل این است که اگر ترافیک جدیدی وارد شبکه شد که باید اطلاعات بر آن اساس به روز رسانی شوند، یک پروفایل برای هر ترافیک جدید واردشده ایجاد می‌شود و در این جدول قرار می‌گیرد. آن ۱۵ ویژگی نام‌برده می‌توانند در شرایط مختلف از این پروفایل نیز استفاده کنند و اطلاعات خود را بر اساس همین پروفایل به روز رسانی نمایند. جهت ذخیره‌سازی مقدار حداقل که به صورت minConf است، مجموعه بر اساس توزیع مقادیر ویژگی‌ها در جدول پروفایل SDN می‌باشد. بنابراین عملیات پویش در داده‌های جریان در SDN برای دو مرتبه، امری ضروری است که بار اول به انتخاب مقادیر ویژگی‌ها از مقادیر ویژگی‌های منفرد بزرگ‌تر از minConf در داده‌های جریانی برای تولید مقادیر ویژگی‌های کاندیدا می‌پردازد. برای بار دوم، به انتخاب مقادیر ویژگی کاندیدا از نرخ minConf بزرگ می‌پردازد و سپس به روز رسانی جدول پروفایل SDN در تعادل بار، به روز رسانی می‌شود. قرار است minConf یک کمینه‌سازی را برای تابع Content $f(x)$ که در (۱) ارائه شده بود انجام بدهد. این کمینه‌سازی می‌تواند ویژگی‌های مستقل در Content $f(x)$ را مشخص کند و آن پروفایل را بر اساس ۱۵ ویژگی به‌روزرسانی نماید. در فاز تعادل بار برای هر بسته داده عبوری از سوئیچ، ابتدا به جدول پروفایل SDN برای گرفتن Conf از مقادیر ویژگی‌ها در گروهی از داده‌ها اشاره‌ای می‌کند و آنها را می‌خواند. اگر هیچ مقدار اطلاعاتی از ویژگی‌ها در جدول نباشد، از minConf به جای محاسبات بیشتر استفاده می‌نماید. در ادامه، یک تابع مبتنی بر محتوا (CBF)[1] وارد کار می‌شود. این تابع وظیفه دارد که بسته‌های داده با استفاده از میانگین وزن‌دار و آزمون فرضیه را محاسبه نماید. CBF جهت میانگین وزن‌دار و آزمون فرضیه به صورت (۳) محاسبه می‌گردد

$$Score(p) = \frac{\sum_{k-۱}^{d} W(A_{k_۱}, A_{k_۲}) conf(A_{k_۱} = p(k_۱), A_{k_۲} = p(k_۲))}{\sum_{k-۱}^{d} W(A_{k_۱}, A_{k_۲})} \quad (۳)$$

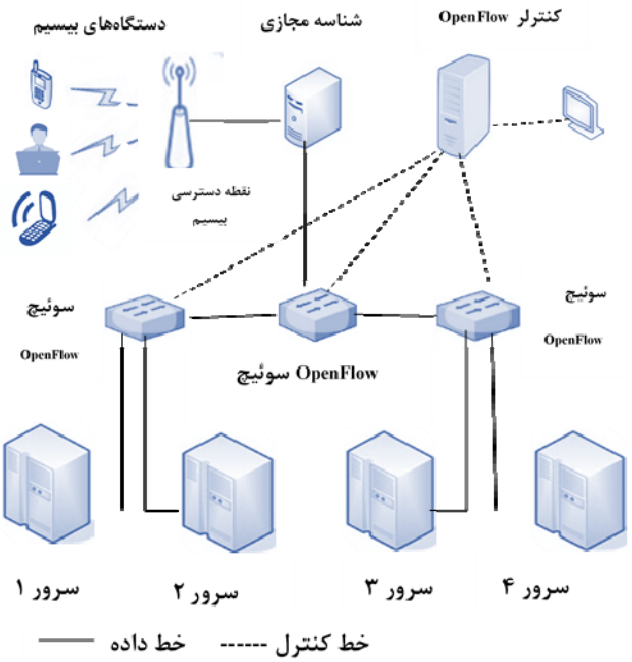
که در (۳)، d تعداد کل ویژگی‌های در نظر گرفته شده، $W(A_{k_۱}, A_{k_۲})$ وزن ویژگی‌های $(A_{k_۱}, A_{k_۲})$، $p(k_۱)$ مقدار ویژگی‌های $A_{k_۱}$ در بسته‌های داده و $conf(A_{k_۱} = p(k_۱), A_{k_۲} = p(k_۲))$ مقدار $conf$ ویژگی‌های $A_{k_۱} = p(k_۱)$ و $A_{k_۲} = p(k_۲)$ است.

در زمانی که بسته‌های داده موجود در شبکه SDN از رویکرد ELM استفاده می‌کنند که قابلیت آموزش و آزمون داده‌ها را دارد، می‌توان برای آموزش کنترل‌کننده از آن و سپس آزمون این روش جهت اجرای شبکه به صورت کامل استفاده نمود. اما از آنجا که تفاوت بزرگی بین مقادیر آدرس IP منبع و شماره درگاه مقصد و همین طور TTL بین بسته‌های داده دارد، یک امر سخت برای مهاجرت سوئیچ‌ها برای شبیه‌سازی این مقادیر ویژگی‌ها در داده‌های بسته نرمال در بسته‌های تعادل بار نشده، ایجاد می‌شود. ویژگی‌های پرچم‌گذاری در داده‌ها در TCP، نوع پروتکل و اندازه کل بسته داده، اندازه‌ای کوچک‌تر از بسته داده ارسالی در سطح شبکه است. با این رویکرد می‌توان نرخ بسته‌های TCP، UDP و ICMP را به دست آورد. بنابراین اگر زوج ویژگی‌ها به دو مورد از سه ویژگی اول، ارسال داده نماید، وزن به صورت بزرگ‌ترین حالت ممکن پیش می‌آید و زوج ویژگی‌ها شامل فقط یکی از سه ویژگی خواهد بود. منظور از این بخش این است که نرخ جریان‌های معکوس‌پذیر، نرخ بسته‌های TCP، UDP و ICMP که در سرایند پرچم‌گذاری بسته‌های TCP نیز قرار دارد، درصد ارسال جریان داده‌ها در مدت زمان کوتاه و همبستگی عناصر سرآمد بسته‌ها ترکیب می‌شوند تا بتوان کارایی را به دست آورد. ویژگی نرخ بسته‌های TCP، UDP و ICMP را نمی‌توان به صورت منفرد یک نمودار کارایی نشان داد، اما سه مورد دیگر را می‌توان به صورت تک‌به‌تک در نمودار کارایی نشان داد و یک حالت ترکیبی نیز از آنها انجام خواهد شد. اگر زوج ویژگی‌ها به دو مورد از آخرین سه ویژگی، ارسال داده را انجام بدهد، وزن به صورت کمترین حالت خواهد بود. این مورد، اصل قضیه آزمون فرضیه را اثبات می‌نماید. مبتنی بر این کار، وزن‌های ویژگی‌های خاص می‌توانند به عنوان مجموعه‌ای مبنی بر شرایط عادی شبکه، تلقی شوند و نهایتاً بسته داده‌های جمع‌آوری شده با رویکرد ELM با یک آستانه مقایسه می‌گردد. اگر این مورد کوچک‌تر از آستانه باشد، تعیین تعادل بار و مهاجرت سوئیچ‌ها در سطح شبکه انجام می‌شود. اگر بزرگ‌تر از سطح آستانه باشد، به عنوان یک مسئله زمان‌بندی در سطح شبکه شناخته می‌شود. در روش ELM که دارای آزمون فرضیه است، جهت ذخیره‌سازی اطلاعات پروفایل جدول SDN، داده‌های جریانی نیاز

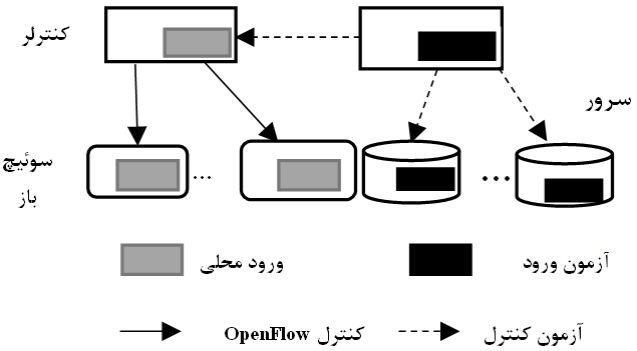
1. Content Based Function

www.SID.ir

Archive of SID



شکل ۴: مدل سیستم مبتنی بر SDN.



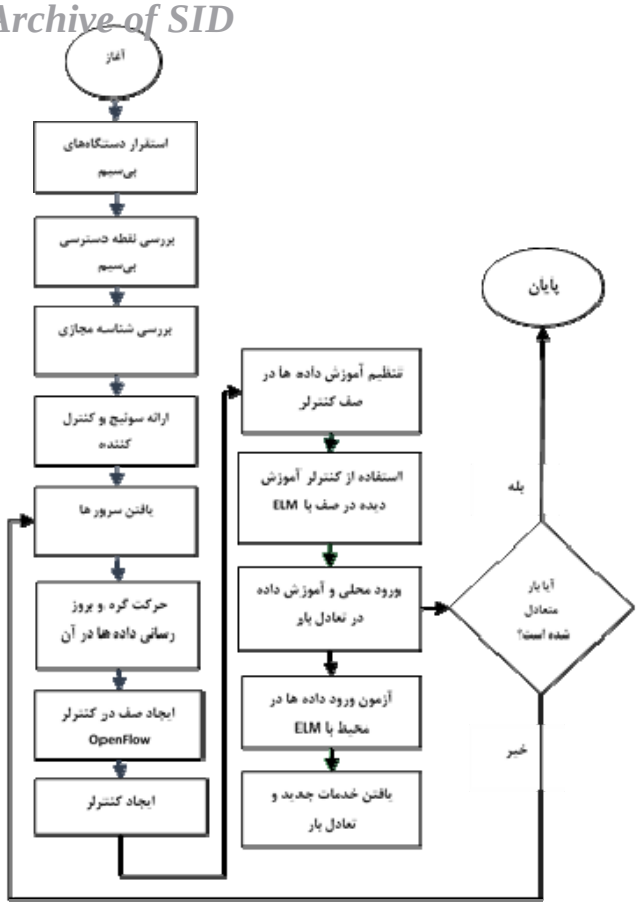
شکل ۵: ساختار کنترلر و سوئیچ.

به پویش در دو مرحله خواهند داشت که یک بار برای تعیین حالت تعادل بار و مهاجرت سوئیچ‌ها و یک بار بدون این دو است که بر اساس صفحه کنترل SDN این کار انجام می‌شود. نیاز است تا نرخ ورود بسته‌ها با ترافیک نرمال نیز محاسبه شود که به صورت (۴) خواهد بود

$$R = \frac{P_a}{P_n} \times ۱۰۰ \qquad (۴)$$

بر اساس (۴)، R نرخ ورود بسته‌ها با ترافیک نرمال در شبکه SDN، P_a تعداد بسته‌های تعادل بار نشده و P_n ترافیک بسته‌های نرمال است. این رابطه جهت شناسایی پایگاه داده کل بارها از ترافیک نرمال مورد استفاده واقع می‌گردد. در تنظیمات ELM، واریانس مشاهده‌شده از یک متغیر خاص به اجزای قابل انتساب به منابع مختلف تغییر، تقسیم می‌گردد. این تحقیق با استفاده از یک روش تحلیل واریانس برای تعیین این که آیا میانگین چندین مجموعه داده با تجزیه و تحلیل آماری داده‌ها برابر است یا خیر، کار اصلی در تعادل بار در SDN را بررسی می‌کند. برای تجزیه و تحلیل ویژگی‌های آماری پورت، این تحقیق روش آزمایش را برای تشخیص این که آیا اختلاف قابل توجهی بین پورت‌ها وجود دارد، تعیین می‌کند تا مشخص شود آیا این عمل برای وضعیت فعلی معتبر است یا خیر. علاوه بر این، از آنجا که جریان داده در شبکه به طور تصادفی انتخاب می‌شود، می‌توان ترافیک از هر پورت را با توزیع نرمال به صورت مستقل مشاهده کرد. اختلافات کلی به ۲ کلاس اساسی تغییر درون‌گروهی

www.SID.ir



شکل ۶: روندنمای رویکرد پیشنهادی.

و تنوع بین گروهی تقسیم می‌شود. تفاوت در کلاس بین گروه برای محاسبه پراکندگی معنی‌دار بین مقادیر متوسط ترافیک درون‌گروهی و میانگین جمعیت محاسبه می‌شود. تفاوت در کلاس درون‌گروهی برای ارزیابی پراکندگی بین یک نمونه بی‌طرف در همان گروه و میانگین جمعیت محاسبه می‌شود. تجزیه و تحلیل آزمون، یک روش آماری است که برای شناسایی مجموعه‌ای از گروه‌ها بر اساس تفاوت‌ها استفاده می‌شود. میانگین مربع از طریق محاسبه اختلافات بین دو قسمت تقسیم بر درجه آزادی آنها به دست می‌آید. با توجه به تجزیه و تحلیل مقایسه‌ای آستانه سطح بازرسی و سطح معناداری، مقدار بازرسی به عنوان نسبت تفاوت‌های "درون" و "بین" تعریف می‌شود. این مشخص خواهد کرد که آیا تفاوت قابل توجهی بین پورت‌ها وجود دارد یا خیر. بر اساس نتایج بالا، فرمول آزمون در ELM در تعادل بار در SDN به شرح (۵) است

$$F = \frac{MS_b}{MS_w} = F(df_b, df_w) \qquad (۵)$$

در این رابطه، MS_b تفاوت بین گروه و MS_w تفاوت درون‌گروهی است. F مقدار باری است که باید در زمان آموزش و آزمون با ELM تعادل یابد و df_b و df_w نسبت تفاوت‌های "درون" و "بین" در عملیات ارسال و دریافت داده در سطح کنترل‌کننده SDN می‌باشند.

۳-۱ مدل سیستم

شکل ۴ نمودار مدل سیستم مبتنی بر SDN، شکل ۵ ساختار کنترلر و سوئیچ و شکل ۶، روندنمای رویکرد پیشنهادی را نشان می‌دهد. همان گونه که در شکل ۴ دیده می‌شود، دستگاه‌های مختلف بی‌سیم از طریق سوئیچ‌های OpenFlow به سرورها متصل می‌شوند. کنترل‌کننده OpenFlow روی یک سرور مجزا وظیفه مدیریت شبکه و ایجاد قواعد

Archive of SID

در سوئیچ‌ها را بر عهده دارد.

بر اساس شکل ۵ قرار است رویکرد پیشنهادی، یک معماری با مسیرهای مختلف بسازد. هر سرور یک ماشین فیزیکی را نشان می‌داد و آدرس IP واقعی خود را دارد. در ادامه یک آدرس IP مجازی ایجاد می‌گردد که از طریق NAT انتشار داده می‌شود و ترافیک ورودی به این آدرس IP مجازی توسط کنترل‌کننده به آدرس‌های IP واقعی مختلف هدایت می‌شود. در مرحله بعد، تجزیه و تحلیل آماری جریان ترافیک به صورت لحظه‌ای از طریق تحلیل واریانس و ماژول‌های الگوریتم زمان‌بندی ترافیک که در کنترل‌کننده ادغام شده‌اند، انجام می‌شود و بر اساس نتیجه تجزیه و تحلیل، مسیر مناسب انتخاب می‌گردد.

به منظور اندازه‌گیری رویکرد پیشنهادی، یک پلتفرم آزمایشی OpenFlow ساخته می‌شود که در بخش بالایی شکل ۴ نشان داده شده است. مدل شبکه SDN شامل چهار گره سرور مستقل، سه سوئیچ OpenFlow و یک کنترل‌کننده است. این سوئیچ‌ها سرعت انتقال را محدود نمی‌کنند و با حداکثر سرعت لینک کار می‌نمایند. برای اجرای تحلیل واریانس و الگوریتم زمان‌بندی الگوریتم ترافیک که در کنترل‌کننده یکپارچه‌شده است، کد نوشته شده و استفاده می‌گردد. کنترل‌کننده می‌تواند اطلاعات صفحه داده و ارتباط متقابل بین کنترل‌کننده و سوئیچ OpenFlow را در یک فایل لاگ محلی ذخیره کند. سرور اندازه‌گیری (آزمون سرور) عملیات همگام‌سازی را در کنترل‌کننده و سرورها انجام می‌دهد و داده‌های ترافیک را به دست می‌آورد.

۳-۲ معیارهای ارزیابی

در این مقاله از چند معیار ارزیابی استفاده شده که شامل میانگین مربعات خطا (MSE)[1] و معیارهای دقت است. در ابتدا هر یک از این معیارها بررسی گردیده و نهایتاً نتایج به دست آمده از پژوهش انجام‌شده با هر معیار ارزیابی ارائه می‌شود.

۳-۲-۱ میانگین مربعات خطا

در این پروژه از میانگین مربعات خطا برای ارزیابی سیستم استفاده شده است. قابل ذکر است که میانگین مربعات خطا، دارای یک مرحله آموزشی هم می‌باشد که در این تحقیق، مد نظر نیست و زمانی از بخش آموزشی آن استفاده می‌شود که از ساختارهای شبکه‌های عصبی استفاده گردد و بعد از آن میانگین مربعات خطا به عنوان یک روش ارزیابی، مورد استفاده واقع گردد. در هر تکرار با روش بازخورد به جلو عمل می‌شود تا وقتی که ماتریس A و خروجی‌ها به روش حداقل مربعات خطا به دست آیند. لازم به ذکر است که همه داده‌های آموزشی باید اعمال گردند و همچنین پارامترهای اولیه ثابت نگه داشته می‌شوند. سپس پارامترهای خالی، ثابت نگه داشته می‌شوند و پارامترهای اولیه توسط گرادیان نزولی تنظیم می‌گردند. زمانی که مقدار کمینه، نشان‌دهنده واریانس فرسایشی کمتری باشد، معمولاً مقدار به صورت درصدی بیان می‌شود. منظور از مقدار به صورت درصد، برای بخشی است که می‌خواهیم دقت، حساسیت و نرخ ویژگی‌ها را برای داده‌ها مورد سنجش قرار بدهیم. زیرا مقدار میانگین مربعات خطا ممکن است به صورت یک عدد واریانسی و یا یک عدد صحیح زیر عدد ۵ باشد که بهترین نرخ آن بین بازه ۰/۱ الی ۰/۹ می‌باشد. بر اساس تغییری که در میانگین مربعات خطا به عنوان ارزیابی انجام می‌شود، ممکن است نتایج روش‌های دیگر ارزیابی، تغییر کند و بر روی آنها تأثیر بالایی بگذارد. در واقع همین میانگین مربعات خطا، بر روی سایر روش‌های ارزیابی، تأثیر می‌گذارد که بر اساس این نرخ حاصل، درصدی برای روش‌های ارزیابی دقت، حساسیت و نرخ ویژگی و یک عدد بر حسب واحد دسی‌بل (dB) برای اوج نسبت سیگنال به نویز و نسبت سیگنال به نویز حاصل می‌شود.

۳-۲-۲ معیارهای دقت

نرخ دقت، معیاری است که بر حسب درصد بیان می‌شود که مهم‌ترین نتیجه کلی از بخش معیارهای ارزیابی، همین دقت است و رابطه آن به صورت (۶) است

$$Accuracy = ۱۰۰ \times \frac{TP + TN}{TN + TP + FN + FP} \qquad (۶)$$

که در (۶)، TP مثبت صحیح[2]، TN منفی کاذب[3]، FP مثبت کاذب[4] و FN منفی کاذب[5] است. رابطه (۷) حساسیت[6] را نشان می‌دهد که بر حسب درصد بیان می‌شود

$$Sensitivity = \frac{TP}{TP + FN} \qquad (۷)$$

رابطه (۸) ویژگی‌های داده[7] را نشان می‌دهد که بر حسب درصد بیان می‌شود. ویژگی‌های داده همان ۱۵ ویژگی مورد نظر در این تحقیق است که با این معیار می‌توان عملکرد کلی آنها را مورد سنجش قرار داد

$$Specificity = \frac{TN}{TN + FP} \qquad (۸)$$

٤- ارزیابی روش پیشنهادی

برای ارزیابی و شبیه‌سازی روش پیشنهادی از پلتفرم شبیه‌سازی OMNET++، سخت‌افزار سیستم پردازنده Core i۷ با حافظه اصلی ۱۶ گیگابایت در محیط سیستم عامل ویندوز ۱۰ استفاده شده است.

برای ارزیابی، یک دیتاست شامل ۱۰،۰۰۰ جریان ترافیکی HTTP، ایمیل و ویدئوی جریانی از شبکه طبق توپولوژی ارائه‌شده با استفاده از Wireshark (کتابخانه pcap) جمع‌آوری گردیده است.

٤-۱ بخش‌ها و اجرای شبیه‌سازی

فرایند شبیه‌سازی به بخش‌های اصلی زیر تقسیم می‌شود:

- طراحی معماری و توپولوژی شبکه
- طراحی سناریوهای روش پیشنهادی جهت محاسبه نتایج

٤-۱-۱ طراحی توپولوژی شبکه

توپولوژی شبکه از سه بخش زیر تشکیل می‌شود:

۱) ساختار گره موبایل

۲) ساختار گره سرویس‌دهنده

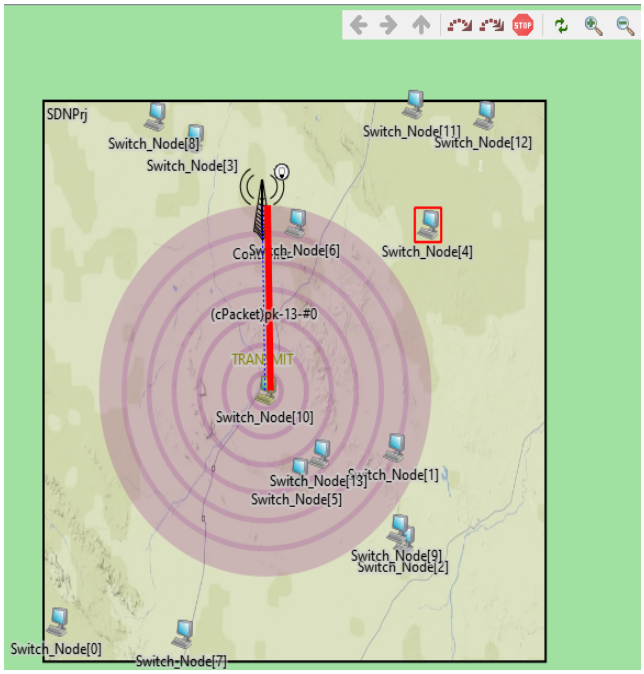
۳) ساختار نهایی شبکه

در جدول ۲ فیلدهای گره موبایل و همچنین در جدول ۳ فیلدهای گره کنترلر نمایش داده شده است.

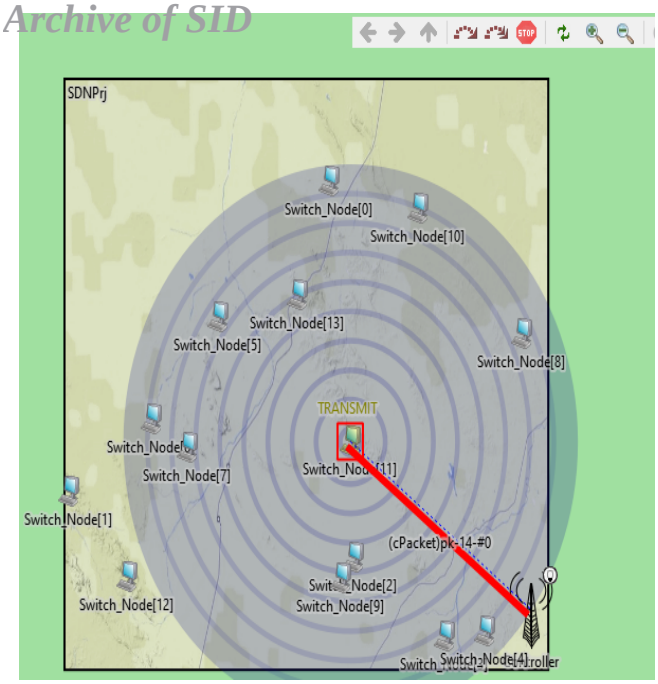
1. Mean Square Error
2. True Positive
3. True Negative
4. False Positive
5. False Negative
6. Sensitivity
7. Specificity

www.SID.ir

Archive of SID



شکل ۷: بار بیش از حد در شبکه با اتصال تک ماشین شبکه تحت عنوان RUN۱.



شکل ۸: اجرای بار بهینه با اتصال تک ماشین شبکه تحت عنوان RUN۲.

جدول ۲: فیلدهای گره سوئیچ.

نام فیلد	توضیحات
txRate	نرخ انتقال سوئیچ
pkLenBits	طول بسته به بیت گره سوئیچ
iaTime	زمان تعامل بسته گره سوئیچ
slotTime	زمان اسلات گره سوئیچ
x,y	طول و عرض جغرافیایی گره سوئیچ
idleAnimationSpeed	هنگامی که هیچ بسته‌ای ارسال نمی‌شود، استفاده می‌شود.
transmissionEdgeAnimationSpeed	هنگامی که انتشار اولین بیت یا آخرین بیت قابل مشاهده است، استفاده می‌شود.
midTransmissionAnimationSpeed	هنگام انتقال استفاده می‌شود.

جدول ۳: فیلدهای گره کنترلر.

نام فیلد	توضیحات
signal [receiveBegin]	سیگنال شروع دریافت: با ورود هر فریم جدید به سرور افزایش می‌یابد و اگر کانال در نهایت بی‌کار شود به ۰ می‌رسد.
signal [receive]	پذیرش‌های موفق (عدم برخورد): ۱ در ابتدای پذیرش و ۰ در پایان پذیرش
@signal [collision]	تعداد فریم‌های برخوردشده در ابتدای دوره برخورد
@signal [collisionLength]	طول آخرین دوره برخورد در پایان دوره برخورد
@signal [channelState]	وضعیت کانال
x,y	طول و عرض گره کنترلر
animationHoldTimeOnCollision	زمان شبیه‌سازی

٤-۱-۲ طراحی سناریوهای روش پیشنهادی

همچنین در زمان اجرا ۵ توپولوژی با عناوین RUN۱ تا RUN۵ به شرح زیر طراحی شده است:

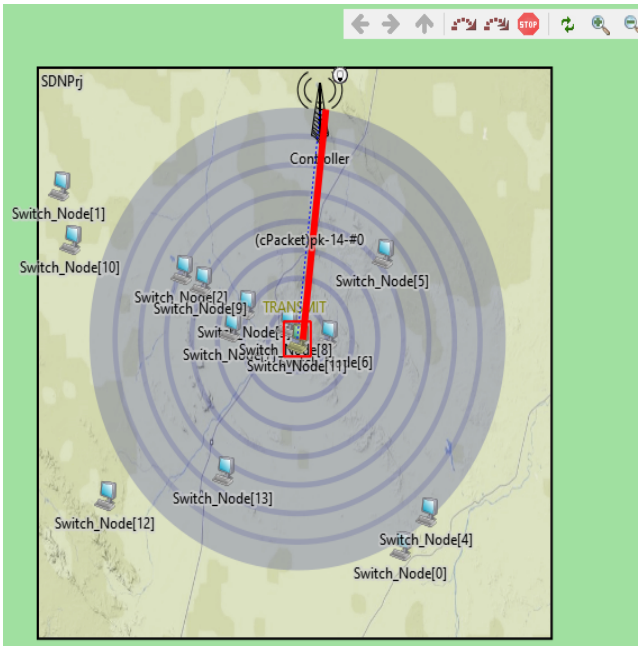
- بار بیش از حد در شبکه با اتصال تک ماشین شبکه تحت عنوان RUN۱
- بار بهینه با اتصال تک ماشین شبکه تحت عنوان RUN۲
- ترافیک سبک با اتصال تک ماشین شبکه تحت عنوان RUN۳
- بار بیش از حد در شبکه با اتصال چند ماشین تحت عنوان RUN۴
- بار بهینه با اتصال چند ماشین تحت عنوان RUN۵

این توپولوژی‌ها با این هدف طراحی شده‌اند که نحوه عملکرد الگوریتم توزیع بار کنترلر در شرایط مختلف بار کاری (بهینه، کم‌باری و بار بیش از حد) و تعداد اتصالات متغیر شبکه ارزیابی شود.

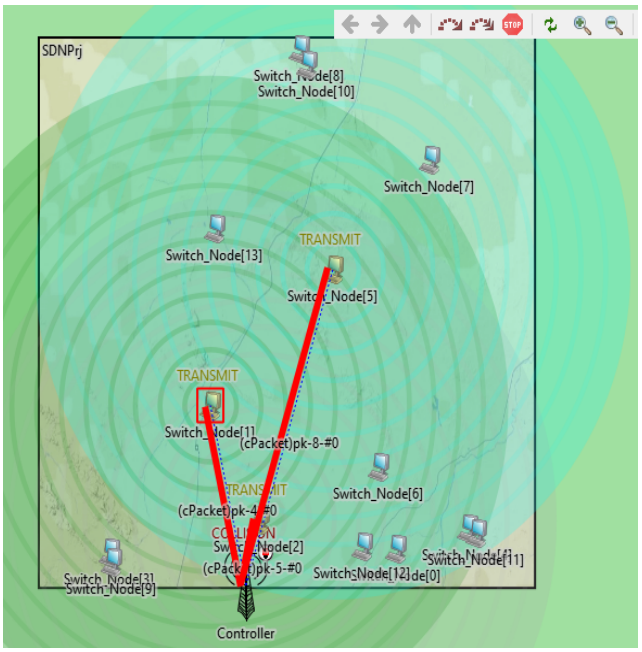
شکل ۷ بار بیش از حد در شبکه با اتصال تک ماشین شبکه تحت عنوان RUN۱ را نمایش می‌دهد. همان گونه که مشخص است در هر مرحله تنها یک ماشین می‌تواند با گره کنترلر ارتباط برقرار نماید. شکل ۸ نیز بار بهینه با اتصال تک ماشین شبکه تحت عنوان RUN۲ را نمایش می‌دهد. شکل ۹ نیز ترافیک سبک با اتصال تک ماشین شبکه تحت عنوان RUN۳ را نمایش می‌دهد. شکل ۱۰ بار بیش از حد در شبکه با اتصال چند ماشین تحت عنوان RUN۴ را نمایش می‌دهد. شکل ۱۱ اجرای توپولوژی بار بهینه در شبکه با اتصال چند ماشین تحت عنوان RUN۵ را نمایش می‌دهد.

www.SID.ir

Archive of SID



شکل ۹: اجرای ترافیک سبک با اتصال تک ماشین شبکه تحت عنوان RUN۳.



شکل ۱۰: اجرای توپولوژی بار بیش از حد در شبکه با اتصال چند ماشین تحت عنوان RUN۴.

در ترافیک سبک، اتصال چند ماشین به شکل هم‌زمان رخ نمی‌دهد و تنها در هر مرحله یک ماشین اجرا می‌شود. به همین دلیل انتخاب توپولوژی ترافیک سبک با اتصال چندین ماشین پیاده‌سازی نشده است.

٤-۲ الگوریتم‌ها و پارامترهای مورد سنجش

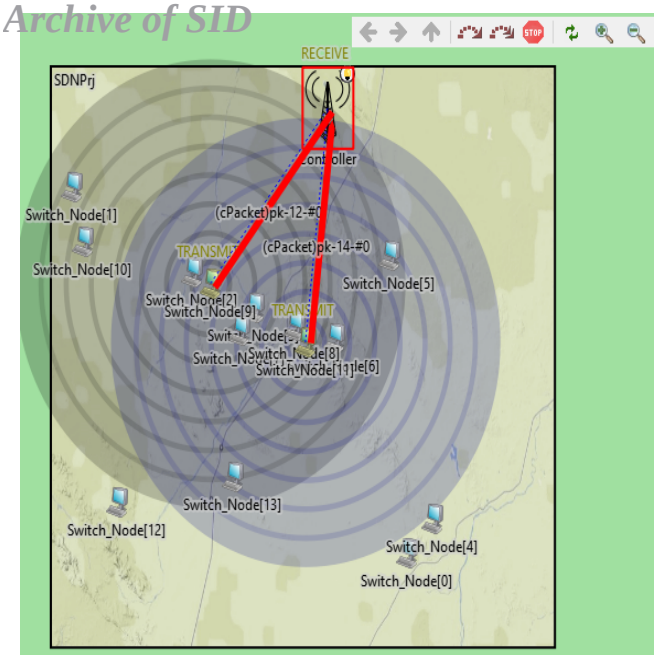
در این تحقیق الگوریتم پیشنهادی با الگوریتم‌های زیر مقایسه می‌شود:

- ژنتیک
- انتشار وابسته
- ترکیب انتشار وابسته و PSO
- الگوریتم تنظیم دامنه کنترل

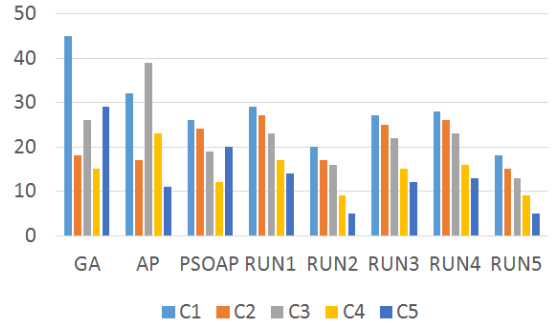
همچنین روش پیشنهادی از نظر پارامترهای زیر با [۱] مقایسه می‌شود:

- تأخیر پردازش کنترل‌کننده
- ترافیک کنترلر

www.SID.ir



شکل ۱۱: اجرای توپولوژی بار بهینه در شبکه با اتصال چند ماشین تحت عنوان RUN۵.



شکل ۱۲: تأخیر پردازش کنترل‌کننده.

- میزان تعادل بار
- زمان پاسخ‌گویی

برای مقایسه عادلانه، الگوریتم‌های فوق با پارامترهایی مشابه الگوریتم پیشنهادی و در محیط شبیه‌سازی OMNET++ پیاده‌سازی شده‌اند.

٤-۳ نتایج شبیه‌سازی

در این بخش نتایج شبیه‌سازی محاسبه و با [۱] مقایسه می‌شود.

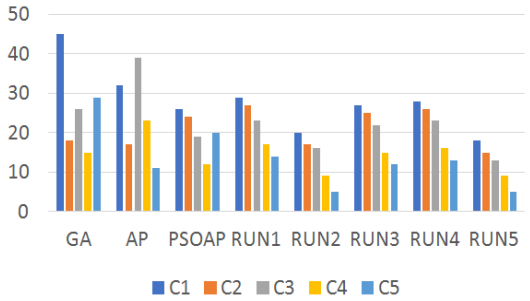
٤-۳-۱ تأخیر پردازش کنترل‌کننده

شکل ۱۲ تأخیر پردازش کنترل‌کننده را در روش پیشنهادی و [۱] مقایسه می‌کند. در RUN۲ و RUN۵ به علت تعادل، بار بهینه نسبت به روش PSOAP و سایر روش‌های [۱] شرایط بهینه‌تری دارد. در واقع RUN۲ نسبت به روش PSOAP ۳۳ درصد و روش RUN۵ نیز نسبت به روش PSOAP ۴۰ درصد تأخیر پردازش کنترل‌کننده را به علت تعادل بار بهینه بهبود بخشیده است.

٤-۳-۲ ترافیک کنترلر

شکل ۱۳ ترافیک کنترلر روش پیشنهادی را با سایر روش‌های [۱] مقایسه می‌کند. در RUN۲ و RUN۵ به علت تعادل، بار بهینه نسبت به روش PSOAP و سایر روش‌های [۱] شرایط بهینه‌تری دارد. در واقع RUN۲ نسبت به روش PSOAP ۳۳ درصد و روش RUN۵ نیز نسبت به روش PSOAP ۱۵ درصد تأخیر پردازش کنترل‌کننده را به علت تعادل بار بهینه بهبود بخشیده است (واحد ترافیک kb بر ثانیه می‌باشد).

Archive of SID



شکل ۱۳: ترافیک کنترل‌کننده.

شکل ۱۴: میزان تعادل بار.

٤-۳-۳ میزان تعادل بار

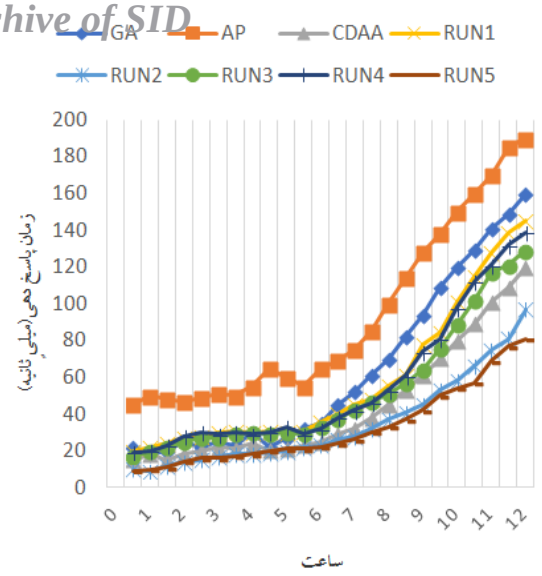
شکل ۱۴ میزان تعادل بار روش پیشنهادی را با سایر روش‌های [۱] مقایسه می‌کند. در RUN۲ و RUN۵ به علت تعادل، بار بهینه نسبت به روش CDAA و سایر روش‌های [۱] شرایط بهینه‌تری دارد. در واقع RUN۲ نسبت به روش CDAA ۵ درصد و روش RUN۵ نیز نسبت به روش CDAA ۳.۵ میزان تعادل بار بهتری را دارا می‌باشد.

٤-۳-٤ زمان پاسخ‌گویی

شکل ۱۵ زمان پاسخ‌گویی روش پیشنهادی را با سایر روش‌های [۱] مقایسه می‌کند. در روش RUN۲ نسبت به CDAA ۲۱ درصد بهبود و در روش RUN۵ نیز ۲۷ درصد بهبود حاصل شده که دلیل آن طراحی الگوریتم تعادل بار بهینه می‌باشد.

۵- نتیجه‌گیری

در این تحقیق، در راستای بهبود چالش‌های پیشین در تعادل بار، رویکرد تعادل بار مبتنی بر پیش‌بینی با استفاده از الگوریتم یادگیری ماشین افراطی پیشنهاد شد. ارزیابی روش پیشنهادی در شبیه‌ساز OMNET++ انجام گردید و نیز به شکل مفصلی به بیان جزئیات روش پیشنهادی، توپولوژی شبکه و شبیه‌سازی پرداخته شد. نتایج حاصل نشان می‌دهد که عملکرد الگوریتم و راه حل ارائه‌شده در تعادل بار نسبت به الگوریتم‌های PSOAP، CDAA و سایر الگوریتم‌های مورد بحث در مقاله پایه برای کاهش ازدحام و جلوگیری از سربار اطلاعاتی، عملکرد بهتری داشته و ضمن بهبود بهره‌وری منابع، توانسته توان عملیاتی سیستم



شکل ۱۵: زمان پاسخ‌گویی.

را بهبود دهد. همچنین این رویکرد تعادل بار توانسته که یک ساختار زمان‌بندی دقیق را برای تعادل بار در SDN ایجاد نماید. برای ادامه روش کار در آینده می‌توان از الگوریتم‌های ترکیبی مثل الگوریتم مارگارتر با رویکرد چندهدفه مبتنی بر منطق فازی، الگوریتم کلونی مورچگان با رویکرد چندهدفه مبتنی بر چند کانال و الگوریتم کرم شب‌تاب چندهدفه فازی استفاده نمود.

مراجع

[1] G. Li, X. Wang, and Z. Zhang, "SDN-based load balancing scheme for multi-controller deployment," *IEEE Access*, vol. 7, pp. 39612-39622, Mar. 2019.

[2] Y. Wu, G. Min, and L. T. Yang, "Performance analysis of hybrid wireless networks under bursty and correlated traffic," *IEEE Trans. Veh. Technol.*, vol. 62, no. 1, pp. 449-454, Jan. 2013.

[3] D. Zhang, H. Huang, J. Zhou, F. Xia, and Z. Chen, "Detecting hot road mobility of vehicular Ad Hoc networks," *Mob. Networks Appl.*, vol. 18, no. 6, pp. 803-813, Dec. 2013.

[4] M. Dong, T. Kimata, K. Sugiura, and K. Zettsu, "Quality-of-Experience (QoE) in emerging mobile social networks," *IEICE Trans. Inf. Syst.*, vol. E97D, no. 10, pp. 2606-2612, Oct. 2014.

[5] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 114-119, Feb. 2013.

[6] S. Sezer, S. Scott-Hayward, P. Kaur, B. Fraser, D. Lake, J. Finnegan, and N. Viljoen, "Are we ready for SDN? implementation challenges for software-defined networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 36-43, Jul. 2013.

[7] H. Zhong, Q. Lin, J. Cui, R. Shi, and L. Liu, "An efficient SDN load balancing scheme based on variance analysis for massive mobile users," *Mob. Inf. Syst.*, vol. 2015, Article ID: 241732, 2015.

[8] D. Wetherall and D. Tennenhouse, "Retrospective on 'towards an active network architecture'," *Comput. Commun. Rev.*, vol. 49, no. 5, pp. 86-89, Nov. 2019.

[9] H. Xu, H. Huang, S. Chen, G. Zhao, and L. Huang, "Achieving high scalability through hybrid switching in software-defined networking," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 618-632, Jan. 2018.

[10] C. Decusatis, M. Haley, T. Bundy, and R. Cunnistra, "Dynamic, software-defined service provider network infrastructure and cloud drivers for SDN adoption," in *Proc. IEEE Int. Conf. Commun. Work*, pp. 235-239, Budapest, Hungary, 9-13 Jun. 2013.

[11] Sandhya, Y. Sinha, and K. Haribabu, "A survey: hybrid SDN," *J. Netw. Comput. Appl.*, vol. 100, pp. 35-55, 15 Dec. 2017.

[12] D. Namiot and M. Sneps-sneppe, *Metadata in SDN API,* arXive, abs/1503.06630, 2015.

[13] A. Shalimov, D. Zimarina, and V. Pashkov, "Advanced study of SDN/OpenFlow controllers," in *Proc. of the 9th Central & Eastern European Software Engineering Conf. in Russia*, 6 pp., Moscow, Russia, 24-25 Oct. 2013.

www.SID.ir

Archive of SID

[14] G. Bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489-501, May 2006.
[15] G. Huang, G. Bin Huang, S. Song, and K. You, "Trends in extreme learning machines: a review," *Neural Networks*, vol. 61, , pp. 32-48, Oct. 2015.
[16] F. K. Inaba, E. O. Teatini Salles, S. Perron, and G. Caporossi, "DGR-ELM-distributed generalized regularized ELM for classification," *Neurocomputing*, vol. 275, pp. 1522-1530, Dec. 2018.
[17] S. Liao and C. Feng, "Meta-ELM: ELM with ELM hidden nodes," *Neurocomputing*, vol. 128, pp. 81-87, Oct. 2014.
[18] H. Zhong, Y. Fang, and J. Cui, "LBBSRT: an efficient SDN load balancing scheme based on server response time," *Futur. Gener. Comput. Syst.*, vol. 68, pp. 183-190, Oct. 2017.
[19] W. C. Chien, C. F. Lai, H. H. Cho, and H. C. Chao, "A SDN-SFC-based service-oriented load balancing for the IoT applications," *J. Netw. Comput. Appl.*, vol. 114, pp. 88-97, Jul. 2018.
[20] L. Boero, M. Cello, C. Garibotto, M. Marchese, and M. Mongelli, "BeaQoS: load balancing and deadline management of queues in an OpenFlow SDN switch," *Comput. Networks*, vol. 106, pp. 161-170, Sept. 2016.
[21] S. Shahryari, S. A. Hosseini-Seno, and F. Tashtarian, "An SDN based framework for maximizing throughput and balanced load distribution in a cloudlet network," *Futur. Gener. Comput. Syst.*, vol. 110, pp. 18-32, Sept. 2020.
[22] A. A. Ateya, A. Muthanna, A. Vybornova, and A. D. Algarni, "Chaotic salp swarm algorithm for SDN multi-controller networks," *Eng. Sci. Technol. an Int. J.*, vol. 22, no. 4, pp. 1001-1012, Aug. 2019.
[23] M. Chen, K. Ding, J. Hao, C. Hu, G. Xie, C. Xing, and B. Chen, "LCMSC: a lightweight collaborative mechanism for SDN controllers," *Comput. Networks*, vol. 121, pp. 65-75, Jul. 2017.
[24] T. Zhang, P. Giaccone, A. Bianco, and S. De Domenico, "The role of the inter-controller consensus in the placement of distributed SDN controllers," *Comput. Commun.*, vol. 113, no. C, pp. 1-13, Nov. 2017.
[25] P. Song, Y. Liu, C. Liu, and D. Qian, "ParaFlow: fine-grained parallel SDN controller for large-scale networks," *J. Netw. Comput. Appl.*, vol. 87, pp. 46-59, Jun. 2017.
[26] J. Cui, Q. Lu, H. Zhong, M. Tian, and L. Liu, "A load-balancing mechanism for distributed SDN control plane using response time," *IEEE Trans. Netw. Serv. Manag.*, vol. 15, no. 4, pp. 1197-1206, Oct. 2018.
[27] V. R. Dasari and T. S. Humble, "OpenFlow arbitrated programmable network channels for managing quantum metadata," *J. Def. Model. Simul.*, vol. 16, no. 1, pp. 67-77, Jan. 2019.
[28] Y. Wu, S. Zhou, Y. Wei, and S. Leng, "Deep reinforcement learning for controller placement in software defined network," in *Proc. IEEE Conf. on Computer Communications Workshops* pp. 1254-1259, Toronto, Canada, 6-9 Jul. 2020.
[29] X. Tian, H. Shen, X. Chen, and C. Li, "Analysis of SDN multi-controller placement," in *Proc. Int. Conf. on Data Processing Techniques and Applications for Cyber-Physical Systems, Advances in Intelligent Systems and Computing*, vol. 1379, pp. 677-683, Jun. 2021.
[30] S. Yeo, Y. Naing, T. Kim, and S. Oh, "Achieving balanced load distribution with reinforcement learning-based switch migration in distributed SDN controllers," *Electronics*, vol. 10, no. 2, pp. 1-16, Jan. 2021.
[31] H. Mostafaei, M. Menth, and M. S. Obaidat, "A learning automaton-based controller placement algorithm for software-defined networks," in *Proc. IEEE Global Communications Conf.,* 6 pp., Abu Dhabi, United Arab Emirates, 9-13 Dec. 2018.

حسین محمدی مدرک کارشناسی خود را در رشته مهندسی تکنولوژی اطلاعات و ارتباطات را در سال ۱۳۹۴ از دانشگاه کابل و مدرک کارشناسی ارشد خود را در رشته مهندسی کامپیوتر شبکه های کامپیوتری و ارتباطی در سال ۱۴۰۰ از دانشگاه یزد دریافت کرد. زمینه‌های تحقیقاتی مورد علاقه وی شامل اینترنت اشیا و شبکه‌های نرم‌افزارمحور است..

سید اکبر مصطفوی مدرک کارشناسی مهندسی فناوری اطلاعات را در سال ۱۳۸۶ از دانشگاه صنعتی شریف و کارشناسی ارشد و دکتری مهندسی کامپیوتر شبکه‌های کامپیوتری را به‌ترتیب در سال‌های ۱۳۸۹ و ۱۳۹۴ از دانشگاه صنعتی امیرکبیر دریافت نمود. دکتر مصطفوی از سال ۱۳۹۴ در دانشکده مهندسی کامپیوتر دانشگاه یزد مشغول به فعالیت گردید و اینک نیز عضو هیآت علمی این دانشکده می_باشد. زمینه_های تحقیقاتی مورد علاقه نام‌برده شامل مباحث نوین در شبکه‌های کامپیوتری و سیستم‌های توزیع شده شامل رایانش ابری و مه، شبکه‌های نرم افزارمحور، اینترنت اشیا، زنجیره بلوکی و شبکه‌های بی‌سیم و سیار است.

www.SID.ir