

علمی - پژوهشی

ارائه الگوریتم بهبود مقیاس پذیری کنترل کننده های توزیع شده شبکه نرم افزار محور

علی ناصری^{۱*}، محمدرضا مهرپویا^۲، سید مرتضی بنی هاشمی^۳

۱- دانشیار، ۲- دانشجوی کارشناسی ارشد، ۳- کارشناس ارشد، مهندسی کامپیوتر دانشگاه جامع امام حسین (ع)

(دریافت: ۱۳۹۹/۰۶/۳۰، پذیرش: ۱۳۹۹/۰۸/۰۵)

چکیده

در سال های اخیر، نسل جدیدی از شبکه ها به نام شبکه های نرم افزار محور (SDN) معرفی شده است که توجه اصلی آن بر جداسازی منطق کنترل از سخت افزار و تمرکز آن در یک نرم افزار مرکزی به نام "کنترل کننده" است. SDN با مدیریت یکپارچه شبکه انعطاف پذیری و کارایی شبکه را بهبود داده و هزینه ها را کاهش می دهد. علی رغم مزایای بسیار، فناوری SDN با چالش های بسیاری از جمله مقیاس پذیری و قابلیت اطمینان سامانه متمرکز مواجه است که می توان با تمرکززدایی فیزیکی سطح کنترل و ارائه کنترل کننده های توزیع شده به رفع آن پرداخت. با این حال، کنترل کننده های توزیع شده نیز با چالش هایی مانند مقیاس پذیری، پایداری و استراتژی هماهنگی مواجه است که در این تحقیق به بهبود مقیاس پذیری کنترل کننده های توزیع شده با استفاده از مفهوم توازن بار پرداخته شده است. برای این منظور، پیشنهاد شده یک تابع تشخیص بار کنترل کننده (CLDF) بر روی هر یک از کنترل کننده های مرتبط با هم، قرار داده شده و در صورت افزایش بار بیش از سطح آستانه، بار جدید به کنترل کننده دیگری با کمترین بار منتقل شود. روش پیشنهادی به صورت توزیع شده بر روی کنترل کننده Floodlight و با استفاده از بستر شبیه سازی mininet و روی سیستم عامل اوبونتو ۱۴/۰۴ اجرا شده و نتایج شبیه سازی نشان می دهد روش پیشنهادی به طور میانگین سبب رشد ۳۱/۶ درصدی نرخ انتقال می گردد.

کلیدواژه: شبکه نرم افزار محور، کنترل کننده، توزیع شدگی، مقیاس پذیری، توازن بار.

۱- مقدمه

مختص یک دستگاه (سوئیچ یا مسیریاب) است و در ساختار شبکه ای و به طور کلی نمی توان یک پیکربندی را به همه سوئیچ ها یا مسیریاب ها اعمال کرد.

- سطح کنترل فاقد انعطاف بوده و ویژگی های محدودی را پشتیبانی می کند: به عنوان مثال، اگر قرار باشد یک پروتکل مسیریابی جدید نوشته شود، باید از صفر شروع شود. چراکه یک کتابخانه برنامه نویسی منعطف وجود ندارد که بتوان بر اساس آن پروتکل جدیدی را نوشت.

ابزارهای فعال و میانی شبکه ها دارای دو واحد اساسی می باشند، «سطح کنترل آ» که تصمیم می گیرد چگونه با ترافیک شبکه رفتار شود و «سطح داده آ» که ترافیک را با توجه به تصمیمات گرفته شده توسط واحد کنترل، ارسال می کند. شبکه های امروزی به صورت عمودی مجتمع می باشند، بدین معنی که سطح کنترل و سطح داده در تجهیزات شبکه یکپارچه بوده، در نتیجه باعث کاهش انعطاف پذیری آن شده و مانع تحول و نوآوری در زیرساخت شبکه می شود. در معماری شبکه های SDN

مباوجود مقبولیت گسترده اینترنت، شبکه های IP سنتی پیچیده بوده و مدیریت آن ها بسیار سخت است، زیرا برای اعمال سیاست های سطح بالای دلخواه در شبکه، اپراتورهای شبکه مجبورند هر دستگاه شبکه را با استفاده از دستورات سطح پایین اختصاصی آن دستگاه و شرکت تولیدکننده، جداگانه پیکربندی کنند. علاوه بر پیچیدگی های پیکربندی، شبکه می بایست خطاهای پویا را نیز به دوش کشیده و با توجه به تغییرات حجم بار در شبکه، خود را با آن وفق دهند. بازپیکربندی خودکار و فرایندهای پاسخ گویی در شبکه های امروزی به طور کامل قابل اجرا نبوده و از طرفی اجرای سیاست های مورد نیاز در محیط های پویا بسیار چالش برانگیز است. دو چالش عمده شبکه های سنتی که قرار است در شبکه های جدید تحت عنوان SDN بهبود یابد عبارتند از:

- نبود دید انتزاعی در سطح کنترل: بدین معنا که پیکربندی و الگوریتم های مربوطه در سطح کنترل، به صورت محلی و

² Control Plane³ Data Plane

بخش‌های مختلف این شکل عبارت‌اند از:

- ❖ لایه هدایت روبه‌جلوی بسته: این بخش به‌طور مستقیم مسئول ارسال داده‌ها بر اساس جدول‌های برنامه‌ریزی شده توسط کنترل‌کننده‌ها است.
 - رابط جنوبی: توسط ONF و بانام OpenFlow استاندارد شده است. پروتکل OpenFlow امکان دسترسی مستقیم و ایجاد تغییر در برنامه ارسال تجهیزات شبکه نظیر سوئیچ‌ها را هم به‌صورت فیزیکی و هم مجازی فراهم می‌کند.
 - ❖ کنترل‌کننده: با همه دستگاه‌های موجود در یک دامنه شبکه ارتباط مستقیم داشته، از ساختار شبکه آگاه است و شبکه را از یک نقطه مرکزی برنامه‌ریزی می‌کند.
 - ❖ رابط شمالی: برای برقراری ارتباط میان کنترل‌کننده و سرویس‌ها و برنامه‌های کاربردی در حال اجرای شبکه، مورد استفاده قرار می‌گیرد. از این واسط برای تسهیل نوآوری و ارائه سرویس‌های جدید با روشی آسان استفاده می‌شود.
 - ❖ کاربرد: این برنامه‌ها از طریق واسط شمالی مابین رفتارها و منابع موردنیاز کنترل‌کننده ارتباط برقرار می‌کند. علاوه بر این، برنامه‌های کاربردی می‌توانند از طریق جمع‌آوری اطلاعات از کنترل‌کننده، یک دید انتزاعی از شبکه ایجاد نمایند که برای تصمیم‌گیری مورد استفاده قرار می‌گیرد.
- به‌طور کلی SDN را به‌عنوان معماری شبکه‌ای با چهار رکن زیر تعریف می‌کنند:

۱. سطوح داده و کنترل از یکدیگر جدا هستند. عملکردهای کنترلی از تجهیزات شبکه حذف شده‌اند و در نتیجه تجهیزات شبکه به مؤلفه‌های هدایت‌گر ساده مبدل شده‌اند.

تصمیمات هدایت‌گری به جای اینکه بر طبق آدرس مقصد انجام شوند، مبتنی بر جریان انجام می‌شوند. در SDN، یک جریان دنباله‌ای از بسته‌ها بین یک مبدأ و مقصد است که دارای مقادیر مشترک در بعضی فیلدهای سرآیندشان هستند. همه بسته‌های یک جریان، به سرویس مشابهی در تجهیزات شبکه نیاز دارند.

منطق کنترل به یک موجودیت خارجی انتقال داده شده که به آن کنترل‌کننده SDN و یا «سیستم‌عامل شبکه» می‌گویند.

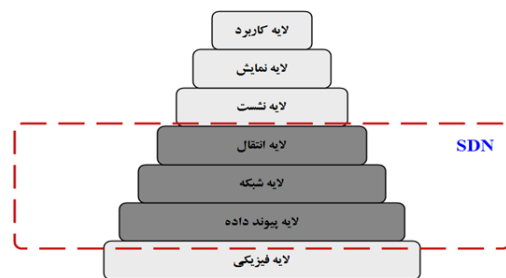
۲. شبکه را می‌توان به‌وسیله برنامه‌های کاربردی که در بالای سیستم‌عامل شبکه در حال اجرا بوده و با دستگاه‌های سطح داده زیرین تعامل دارند، برنامه‌ریزی کرد.

در کنار مزایای SDN، این فناوری همچنان با چالش‌هایی همچون متمرکز بودن کنترل‌کننده، مواجهه است. چالش فعلی پیاده‌سازی و استفاده عملی از SDN، به‌کارگیری کنترل‌کننده

سعی شده توانایی و هوشمندی شبکه‌های کامپیوتری بیشتر شود. بر اساس تعریف بنیاد شبکه‌های باز (ONF)^۱، SDN یک نوع از معماری شبکه‌ای است که سطوح داده و کنترل را از یکدیگر جدا می‌کند. این امر با قرار دادن سطح کنترل (فراهم‌کننده سیاست‌ها و هوشمندی شبکه) در سرور یا برنامه‌ای که کنترل‌کننده نام دارد امکان‌پذیر است. برنامه‌های کاربردی هم در بالای سطح کنترل در سطح کاربرد^۲ قرار می‌گیرند.

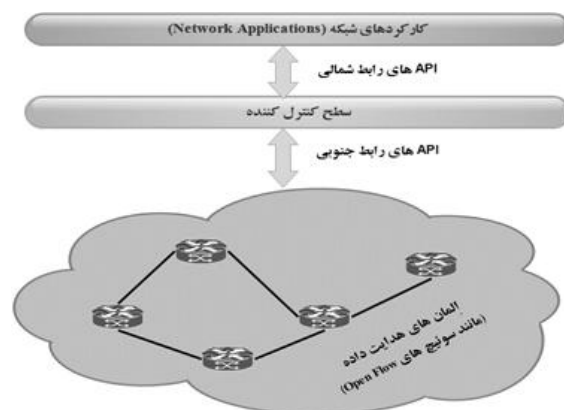
یک کنترل‌کننده نرم‌افزاری متمرکز که دیدی از کل شبکه را در اختیار دارد، بسیار مؤثرتر و هوشمندتر از بخش‌های کنترلی پراکنده مسیریاب‌ها و سوئیچ‌ها عمل خواهد کرد. با یک کنترل‌کننده متمرکز، قابلیت‌هایی مانند برنامه‌ریزی، مقیاس‌پذیری، انعطاف‌پذیری، خودکارسازی و توسعه نرم‌افزاری به شبکه اضافه می‌شود.

چنانچه قرار باشد جایگاه فناوری SDN را در لایه‌های مدل مرجع OSI^۳ بیان نمود، مطابق شکل (۱) می‌توان گفت عمده فعالیت SDN در لایه‌های «انتقال»، «شبکه» و «پیوند داده» است.



شکل (۱): جایگاه فناوری SDN در لایه‌های مدل مرجع OSI

با توجه به توضیحات فوق، یک نمای ساده از معماری شبکه‌های نرم‌افزارمحور در شکل (۲) نشان داده شده است.



شکل (۲): نمای ساده از معماری شبکه‌های نرم‌افزارمحور [۱]

^۱ Open Networking Foundation (ONF)
^۲ Controller
^۳ Application Layer
^۴ Open Systems Interconnection

ب- مسئله قدرت: کنترل کننده‌های متمرکز SDN، دارای یک مشکل واحد با عنوان تک نقطه خرابی^۱ هستند. وقتی یک کنترل کننده متمرکز خراب می‌شود، سوئیچ‌ها توانایی خود را برای ارسال بسته‌های جدید از دست خواهند داد و در نهایت، کل شبکه دچار خرابی و شکست خواهد شد. برای حل این مسئله، محققان می‌توانند سوئیچ را به صورت ترکیبی (OpenFlow- OpenFlow hybrid) پیکربندی کنند [۵]. سوئیچ وقتی که نمی‌تواند به کنترل کننده SDN متصل شود، از سوئیچ مدرن (به عنوان مثال از سوئیچ نرم افزار پایه) به سوئیچ سنتی تبدیل خواهد شد.

کنترل کننده توزیع شده شبکه نرم افزارمحور: کنترل کننده توزیع شده SDN تضمین می‌دهد هر دو مشکل مطرح شده در کنترل کننده متمرکز را حل کند. ایده کلی تشکیل چندین کنترل کننده در کنار یکدیگر است که می‌تواند بار را در شبکه به طور مساوی به اشتراک بگذارد. علاوه بر این، یک کنترل کننده می‌تواند کنترل کننده دیگری را در هنگام خراب شدن، کنترل کند. جزئیات این رویه‌ها پیچیده است که ممکن است نیاز به برخی از فناوری‌های مورد استفاده در دستگاه‌های توزیع شده داشته باشد.

براندن هلر و همکاران [۶] در مورد مشکل قرار دادن کنترل کننده بحث کرده‌اند تا تعیین کنند چه تعداد کنترل کننده مورد نیاز است و در کجا باید قرار گیرند. به عنوان مثال، در همان مطالعه، براندن هلر و همکاران، با سه کنترل کننده زمان پاسخگویی را بررسی کردند و متوجه شدند زمان پاسخگویی به نصف زمان حالت قبل می‌رسد.

Jimenez و سایرین [۷] نشان داده‌اند که تعداد کنترل کننده‌های انتخاب شده و محل قرارگیری آن‌ها باعث تغییر در عملکرد شبکه می‌شود. وقتی تعداد کنترل کننده‌ها به طور افزایشی زیاد شود، به نظر می‌رسد که بیش از آنکه مؤثر باشد، هزینه‌بر است. بنابراین بهترین نتیجه زمانی رخ می‌دهد که شبکه بتواند بهره‌برداری نزدیک ۱۰۰٪ با تعداد کمی از کنترل کننده‌ها را اجرا کند. Elasticon [۸] سعی در دستیابی به ۱۰۰٪ استفاده از ایده جدید دارد که این خوشه از کنترل کننده‌ها را قادر می‌سازد تا بر اساس بار شبکه، کنترل کننده‌ها را اضافه یا حذف کنند. خوشه مقدار آستانه تراکم را حفظ می‌کند. هنگامی که ترافیک از سطح آستانه تراکم عبور می‌کند، خوشه می‌تواند یک کنترل کننده جدید برای پیوستن به خوشه و کمک به مدیریت شبکه به آن اضافه کند. بعد از اتصال کنترل کننده جدید، کنترل کننده باید به خوشه اختصاص داده شود. علاوه بر این، خوشه می‌تواند زمانی که ترافیک تا آستانه پایین کاهش می‌یابد، کنترل کننده‌ها را از خوشه حذف کند.

به صورت متمرکز است. نقطه مقابل تمرکز، توزیع شدگی است که بنا دارد وابستگی کل شبکه به یک نقطه واحد را مرتفع سازد. با رفع مسئله تمرکز در کنترل کننده‌ها و ورود به بحث کنترل کننده‌های توزیع شده، می‌توان به این چالش‌ها پاسخ گفت. همچنین هر چند کنترل کننده‌ها سبب بهبود کارایی مدیریت شبکه می‌شوند، اما همچنان نمی‌توانند پاسخگوی تقاضاهای جریان‌های حجیم و آمارهای مانیتورینگ شبکه باشند. به عنوان مثال یکی از کنترل کننده‌های اولیه SDN به نام NOX، در هر ثانیه فقط 30k تقاضا را می‌تواند در کمتر از 10ms پاسخ دهد. بدیهی است این میزان برای شبکه‌های بزرگ، کافی نیست. این چالش که به عنوان «مسئله مقیاس پذیری» نامیده می‌شود به ذات متمرکز معماری این شبکه برمی‌گردد. لذا فهم مسئله مقیاس پذیری در کنترل کننده‌های SDN و بهبود آن از طریق مفاهیم توزیع شدگی یک چالش اساسی در سازگاری این معماری با شبکه‌های مقیاس بزرگ و شبکه‌های با جریان بالا است که حل این چالش، هدف ارائه این مقاله می‌باشد.

ادامه این مقاله بدین صورت است که در بخش ۲، مفاهیم کنترل کننده‌های توزیع شده ارائه شده و در بخش ۳، روش پیشنهادی بهبود مقیاس پذیری کنترل کننده‌های توزیع شده SDN شرح داده شده است. در بخش ۴ شبیه‌سازی و ارزیابی روش پیشنهادی انجام شده و در بخش ۵ از مباحث مطرح شده، نتیجه‌گیری شده است.

۲- کنترل کننده‌های توزیع شده

در معماری SDN یک کنترل کننده متمرکز منطقی شبکه به کار گرفته شده است [۲]. برای دستیابی به تمرکز منطقی در شبکه‌های نرم افزارمحور، دو نوع پیاده‌سازی وجود دارد: کنترل کننده متمرکز و کنترل کننده توزیع شده.

کنترل کننده متمرکز: در این مدل تنها یک ماشین کنترل کننده در شبکه وجود دارد. این دستگاه بدون هیچ زحمتی همه سوئیچ‌ها را به صورت متمرکز هدایت می‌کند، زیرا همه سوئیچ‌ها به همان کنترل کننده واحد متصل می‌شوند. بسیاری از تجهیزات اولیه SDN، از این ساختار پیروی می‌کنند. متأسفانه، این مدل دارای دو مشکل اساسی است:

الف- مسئله مقیاس پذیری: کنترل کننده SDN برای رسیدگی به حجم بسیار بالای درخواست‌ها، یک منبع محدود دارد. برای حل این مسئله، محققان تلاش می‌کنند تعداد درخواست انتقال مسیر ارسال به کنترل کننده متمرکز را محدود کنند [۳ و ۴]. با این حال این استراتژی، هوش را به سوئیچ اضافه می‌کند و بدین ترتیب مفهوم SDN نقض می‌گردد.

¹Single Point of Failure (SPF)

۳- روش پیشنهادی

طی این فرآیند یک یا چند دستگاه میانی که به کنترل کننده با بار زیاد متصل بودند، به کنترل کننده با بار کم متصل می شود و این روند تا زمانی که اختلاف بار از مقدار آستانه پایین (LLT^3) کمتر نشود، ادامه پیدا می کند. هر زمان بار کنترل کننده ای که به مرز اشباع (HLT) رسیده بود، پردازش شده و از سطح LLT کمتر شود، آن کنترل کننده مجدداً خود را به خوشه کنترل کننده های معرفی می کند که آماده دریافت ترافیک جدید است. بدیهی است در این حالت، این کنترل کننده اولین گزینه برای دریافت بار ترافیکی از کنترل کننده ای است که به مرز اشباع پردازشی (HLT) رسیده؛ به شرط اینکه کنترل کننده دیگری قبل از آن به سطح LLT نرسیده باشد. سطح آستانه پایین زمانی اتفاق می افتد که حجم بار به کمتر از ۲۰ درصد توان کنترل کننده برسد.

برای کنترل بهتر بار ترافیک، چند عامل دیگر نیز باید در نظر گرفته شود. اول از همه، بازه زمانی گزارش دهی اطلاعات بار کنترل کننده ها به CLDF است. این بازه باید طوری انتخاب شود که تعاملی بین دقت توازن بار و سر بار پیام دهی باشد. مثلاً اگر بازه زمانی خیلی کوتاه باشد، هر کنترل کننده با گزارش دهی مکرر، بار قابل توجهی را به شبکه تحمیل می کند. با توجه به این که بار ترافیکی هر کنترل کننده با مقدار بار ورودی به شبکه رابطه مستقیم دارد، زمان گزارش دهی متناسب با مقدار نرخ ارتباط جدید در شبکه در نظر گرفته می شود، بدین صورت که هر چه نرخ تماس (ارتباط) جدید بیشتر باشد، بازه گزارش دهی کوتاه تر می شود. این شیوه باعث می شود که کنترل کننده در برابر بار ناگهانی به خوبی پاسخ دهد و به سرعت با انتقال اطلاعات بار، روال توازن بار را فعال نماید.

مورد دوم این است که CLDF باید فرایند مناسبی برای ارزیابی بار سرورها و آغاز روال توازن بار داشته باشد. فرایند پیشنهادی بدین گونه است که بعد از تجزیه و تحلیل اطلاعات بار ارسالی از کنترل کننده ها، CLDF درصد مشغولیت کنترل کننده ها را با حد آستانه اشباع آن کنترل کننده (HLT) مقایسه می کند. به محض اینکه سطح بار یک کنترل کننده از مقدار آستانه بالا (HLT) بیشتر شود، با تقریب اینکه هر دستگاه میانی چه کسری از بار کنترل کننده ها را ایجاد می کند، ابتدا آن کنترل کننده را (در صورتی که کنترل کننده ای در حالت غیر اشباع وجود داشته باشد) موقتاً از خوشه خارج کرده و اقدام به اتصال مجدد تعداد مناسب تجهیزات میانی از کنترل کننده با بار زیاد به کنترل کننده با بار کم می کند. این روند تا زمانی که میزان بار پردازش نشده کنترل کننده به سطح LLT نرسد، ادامه می یابد. هنگامی که سطح بار پردازش نشده بر روی کنترل کننده به سطح LLT رسید، آن کنترل کننده مجدداً به خوشه معرفی شده و آماده دریافت ترافیک

همان طور که پیش از این بیان شد، یکی از چالش های اصلی کنترل کننده های توزیع شده، مسئله مقیاس پذیری آن ها است. در واقع از آنجایی که کنترل کننده متمرکز و منفرد خودش می توان محل تک نقطه خرابی باشد، پیشنهاد داده شد تا با ایجاد افزونگی در تعداد کنترل کننده ها، این مشکل برطرف شود. پس از آن، بیان شد که می توان با استفاده از فناوری توزیع شدگی در کنترل کننده های افزونه، حداکثر بازدهی را از منابع موجود اخذ کرد تا بدین ترتیب مقیاس پذیری کنترل کننده های شبکه بهبود یابد. از آنجایی که در این حالت، ممکن است منابع برخی از کنترل کننده ها به حالت اشباع برسد، باعث می شود کارایی این کنترل کننده ها به شدت و به طور نمایی کاهش یابد. لذا در پژوهش جاری مفهوم کنترل کننده «توزیع شده، اما با منطق متمرکز» ارائه شده و مورد بررسی قرار گرفته است. راه حل پیشنهادی متکی بر مجازی سازی کنترل کننده شبکه از طریق چندین نمونه کنترل کننده سازمان یافته در خوشه است. مجازی سازی کنترل کننده می تواند بر چالش مقیاس پذیری غلبه کرده و بر کارایی کنترل کننده SDN در میزبانی مرکز داده ی بزرگ، بیفزاید.

الف) طرح پیشنهادی لایه کنترل کننده SDN در معماری توزیع شده

توازن بار مبتنی بر اطلاعات جدول توازن، می تواند به عنوان روشی از سوی اپراتورها جهت عملکرد پایدار شبکه مورد استفاده قرار گیرد که این روش ضمن توازن بار، به کمینه کردن تأخیر ارتباط نیز اهمیت می دهد.

تابع متوازن کننده بار می تواند در اجزاء مختلف شبکه پیاده سازی شود که در این مقاله به عنوان جزئی مجزا به نام تابع تشخیص بار کنترل کننده ($CLDF^1$) تعریف و طراحی شده است. وظایف CLDF شامل جمع آوری دوره ای اطلاعات بار کنترل کننده ها و به کار انداختن روش توازن بار است. در این روش، n به عنوان تعداد کنترل کننده ها و m به عنوان تعداد تجهیزات میانی در ارتباط با کنترل کننده ها تعریف می شود. وقتی تفاضل بار بین دو کنترل کننده از مقدار آستانه بالا (HLT^2) بیشتر شد، روالی برای انتقال ترافیک های جدید از تجهیزات میانی از یک کنترل کننده با بار زیاد به کنترل کننده ای با بار کم آغاز می شود. سطح آستانه بالا زمانی اتفاق می افتد که حجم بار به بیش از ۸۰ درصد توان کنترل کننده برسد.

¹ Controller Load Detection Function (CLDF)

² High Level Threshold (HLT)

³ Low Level Threshold (LLT)

یک کنترل کننده به کنترل کننده بعدی را کمینه سازد که می توان از آن به عنوان تابع هدف بهره برد. دو معیار زمان اجرا و زمان انتقال از طریق روابط زیر قابل محاسبه می باشند:

$$T_{exe} = \frac{\text{Number of Traffic Tasks}}{\text{Processor Speed}} \text{ (sec)} \quad (2)$$

که در آن، Processor Speed، تعداد ترافیکی است که پردازنده می تواند در یک ثانیه پردازش کند.

$$T_{trans} = \sum_{i=1}^n \sum_{k=1}^m \sum_{z=1}^m (1 - x_{ik} x_{iz}) \frac{DT_{kz}}{B_{kz}} \quad (3)$$

که در آن، DT_{kz} حجم داده ای است که بین k امین و z امین کنترل کننده تبادل می شود. B_{kz} پهنای باند بین دو کنترل کننده k و z است. x_{iz} : اگر i امین ترافیک از کنترل k ام به کنترل کننده z ام منتقل شود، برابر ۱ و در غیر این صورت برابر صفر خواهد بود. حال مسئله اصلی، پیدا کردن روشی برای انتخاب معیارهای فوق است. در مسئله جاری و در ادامه فرض شده است که وزن معیارهای فوق در تعیین حد آستانه توزیع بار، برابرند. برای آنکه سطح اهمیت معیارهای فوق در فرایند پردازش تعیین سطح آستانه مشخص گردد، باید معیارهای فوق با ضریب مناسبی، مقداردهی گردند. رابطه وزن ضرایب به صورت زیر بیان می گردد:

$$U = \alpha T_{exe} + (1 - \alpha) T_{trans} \rightarrow \min \quad (4)$$

که در آن، α ضریب پردازش ترافیک ($\alpha \in (0, 1)$) و $(1 - \alpha)$ ضریب زمان انتقال ترافیک است. به ازای سطوح منابع در دسترس، سطح منابع استفاده شده و سرعت پردازش ترافیک کنترلی، باید سطوح مختلف HLT مورد بررسی قرار گرفته تا در نقاطی که سطح کمینه این تابع تأمین می گردد، HLT هر کنترل کننده انتخاب شود.

همان طور که بیان شد، کمینه سازی رابطه فوق، یک مسئله NP-HARD است که برای حل آن نیاز به روش های فرا ابتکاری است. بامطالعه در روش های فرا ابتکاری، الگوریتم ازدحام ذرات (PSO)³ به دلیل سرعت بالای اجرا انتخاب شده است.

ب-۲) حل مسئله D-HLT با استفاده از PSO

با توجه به توضیحات فوق، برای حل مسئله D-HLT با استفاده از PSO باید گام های زیر طی گردد:

گام ۱: جمع آوری اطلاعات راجع به کنترل کننده ها به عنوان ورودی الگوریتم PSO شامل تعداد و سرعت پردازش پردازنده هر کنترل کننده، میزان حافظه آزاد، تعداد وظایف ترافیکی در حال پردازش، زمان پردازش هر ترافیک ورودی، نرخ ترافیک ورودی و ...

کنترل کننده خواهد بود. بدین طریق با توجه به مقادیر آستانه، توازن بار می تواند با دقت مطلوب انجام شود.

ب) بهبود کارایی تابع CLDF با استفاده از سطح آستانه بالایی پویا (Dynamic HLT)

همان طور که بیان شد، اساس الگوریتم CLDF پیشنهادی، بر بررسی سطوح آستانه بالا و پایین کنترل کننده ها نسبت به بار ترافیکی پردازش نشده و موجود در کنترل کننده ها استوار است. نکته بسیار مهم در اینجا این است که در یک سخت افزار ممکن است چند کنترل کننده با هم به صورت ماشین مجازی فعالیت کنند (چه بسا بسیاری از کنترل کننده های تجاری موجود مانند Onus، Floodlight و OpenDeyLight قابلیت نصب بر روی مجازی ساز ESXi را دارند)، همچنین حتی اگر به عنوان تنها سیستم عامل موجود در سخت افزار باشند نیز پردازش های پس زمینه سیستم عامل سبب می شود توان پردازشی متغیری در اختیار کنترل کننده قرار بگیرد. این امر سبب خواهد شد تا نتوان حد آستانه ثابتی را برای کنترل کننده ها تعریف کرد (خصوصاً حد آستانه بالا).

با توجه به توضیحات فوق، جهت بهبود کارایی الگوریتم پیشنهادی، سازوکاری برای تعریف حد آستانه بالایی پویا (D-HLT¹) طراحی شده است. از آنجایی که اطلاعات پیرامون به کارگیری منابع مانند عملکرد CPU، مصرف حافظه، سرعت ورود پیام های کنترل، سرعت پردازش ترافیک کنترل کننده و نوع ترافیک ورودی کنترلی از الگوی خاصی پیروی نمی کنند، همواره در مسائل تخصیص منابع، تعیین حد آستانه جزو مسائل NP²-Hard (مسائلی که پیچیدگی چند جمله ای غیرقطعی دارند) دسته بندی می شود. لذا برای حل این مسائل نمی توان از روش های الگوریتمی عادی بهره برد.

ب-۱) شرایط مسئله D-HLT

فرض کنیم که m تعداد کنترل کننده های موجود در خوشه شبکه کنترل کننده ها باشد:

$$\text{CTRL} = \{\text{CTRL1}, \text{CTRL2}, \dots, \text{CTRLm}\} \quad (1)$$

جهت تعیین حد آستانه برای تابع توزیع بار CLDF، باید بتوان نقطه بهینه ای که تناسب مناسبی بین ترافیک کنترل کننده های خوشه داشته باشد، برقرار کرد. این تناسب را می توان با استفاده از توابع بهینه سازی ایجاد نمود. در واقع این تابع بهینه سازی سعی دارد زمان پردازش ترافیک (T_{exe}) و زمان انتقال ترافیک (T_{trans}) از

¹ Dynamic HLT

² Non-deterministic Polynomial-time HARDness (NP-HARD)

³ Particle Swarm Optimization (PSO)

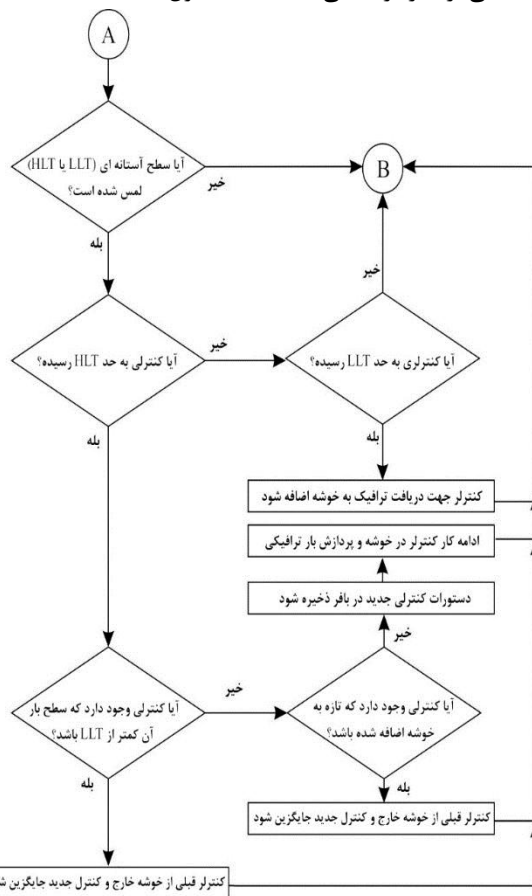
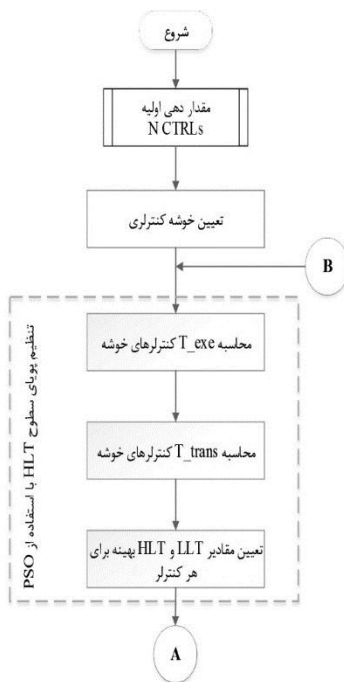
اختصاص داده می‌شود، بر اساس خروجی الگوریتم PSO، ترافیک جدید به کنترل‌کننده‌های دیگر منتقل خواهد شد. برای این منظور کافی است تا طبق جدول (۱)، پارامترهای محیطی کنترل‌کننده را به پارامترهای پیش‌فرض PSO نگاشت نمود.

جدول (۱): نگاشت پارامترهای محیطی مسئله به پارامترهای پیش‌فرض PSO

پارامتر PSO	پارامتر محیطی کنترل‌کننده
Particle	ترافیک کنترل‌کننده
Particle Position	مکان (شماره) کنترل‌کننده
Optimal Solution	راه‌حل بهینه یک بردار n-بعدی در N تعداد ترافیک ورودی است
Particle Migration	هدایت ترافیک جدید به کنترل‌کننده ممکن بعدی

ب-۳) ساختار نهایی الگوریتم پیشنهادی

با توجه به توضیحات داده‌شده، نمای بلوکی الگوریتم پیشنهادی به صورت شکل (۳) است که در آن اثر D-HLT بر روی الگوریتم پیشنهادی قابل مشاهده است.



شکل (۳): الگوریتم پیشنهادی

گام ۲: پایش داده‌های کنترل‌کننده‌ها اعم از وضعیت جریان پردازشی هر کنترل‌کننده، میزان بار هر کنترل‌کننده، کنترل‌کننده‌های اشباع‌شده، ترافیکی که باید به کنترل‌کننده جدید منتقل شود و زمان انتقال.

گام ۳: پیدا کردن کنترل‌کننده بهینه بر اساس حدود آستانه مبتنی بر الگوریتم PSO به‌عنوان کنترل‌کننده مقصد برای انجام پردازش‌های ترافیکی.

گام ۴: پردازش ترافیک کنترل‌کننده و به‌روزرسانی داده‌های زمان اجرای بهینه، زمان انتقال بهینه و مشخصه‌های کنترل‌کننده جاری به‌عنوان خروجی الگوریتم تعیین حدود آستانه مبتنی بر الگوریتم PSO.

گام ۵: انتقال ترافیک جدید به کنترل‌کننده انتخاب‌شده.

در روش پیشنهادی به‌کارگیری الگوریتم PSO، تطبیق پارامترها بدین‌صورت است که: ذرات، همان ترافیک کنترلی هستند که باید در کنترل‌کننده مورد پردازش قرار گیرند. هنگامی که سطح خاصی از بار ترافیکی به یک کنترل‌کننده

۴- شبیه سازی و ارزیابی

بستر سخت افزاری شبیه سازی: تمام شبیه سازی های صورت گرفته در این مقاله، بر روی یک سیستم خانگی با 16GB حافظه اصلی و پردازنده 2.6GHz Intel Core I7 core two doul انجام شده است.

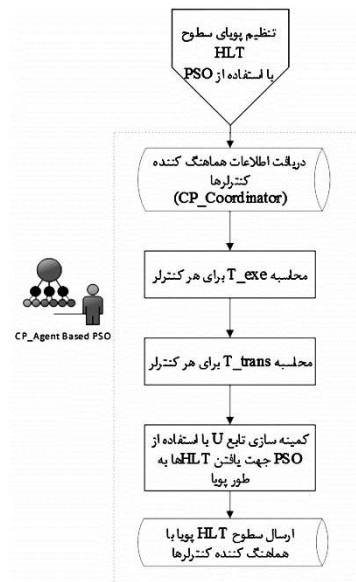
کنترل کننده مورد استفاده: برای انتخاب کنترل کننده، گزینه های زیادی وجود دارد. کنترل کننده Floodlight با جامعه کاربری و پشتیبانی بسیار فعال خود و واسط گرافیکی کاربرپسند و همچنین مستندات کاملی که دارد، می تواند انتخاب مناسبی باشد. در نسخه جدید این کنترل کننده، امکان تنظیم نقش کنترل کننده و استفاده از چندین کنترل کننده توزیعی نیز فراهم شده است.

بستر نرم افزاری شبیه سازی: شبیه سازی های صورت گرفته، در سیستم عامل اوبونتو ۱۴.۰۴ و با استفاده از کنترل کننده Floodlight نسخه ۱/۲ و شبیه ساز Mininet که یک شبیه ساز اختصاصی کنترل کننده های SDN است انجام شده است. همچنین به منظور ارزیابی شبیه سازی ها، بررسی و رصد عملکرد و سرعت پورت ها و آدرس ها از ابزار WireShark استفاده شده است.

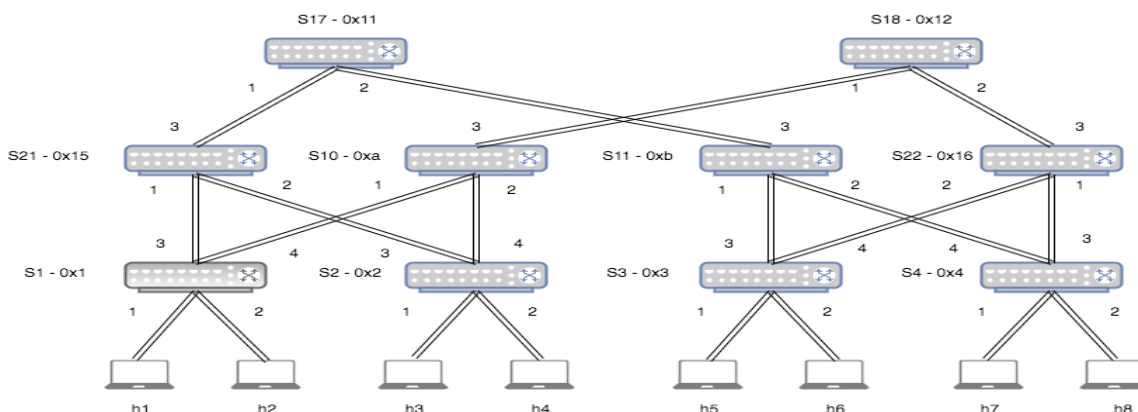
ساختار مورد استفاده: جهت سهولت در نشان دادن عملکرد روش پیشنهادی، از ساختار FatTree بهره گرفته شده است. پس از شناخت ساختار FatTree، نیاز است تا ساختار مدنظر از این ساختار که بتواند مسئله توزیع بار را نشان دهد، طراحی کرد. ساختار طراحی شده به صورت شکل (۵) است.

درواقع D-HLT با بررسی منابع در دسترس و مصرفی همه کنترل کننده ها، نسبت به تنظیم سطوح آستانه بالای کنترل کننده ها، اقدام می کند. همچنین ساختار بلوکی فرایند تنظیم پویای سطوح HLT برای همه کنترل کننده ها با استفاده از PSO در شکل (۴) آمده است.

با توجه به توضیحات فوق می توان گفت با استفاده از کنترل های توزیع شده می توان به حل چندین چالش اساسی SDN، مانند تک نقطه خرابی، مقیاس پذیری، تأخیر سراسری، قابلیت اطمینان و قابلیت دسترس پذیری پرداخت. هر چند تاکنون روش های مختلفی برای توزیع بار بین تجهیزات شبکه با استفاده از کنترل کننده مرکزی ارائه شده است، اما در این فصل یک الگوریتم جدید برای توزیع بار کنترلی بین کنترل کننده های توزیع شده شبکه که به صورت یک خوشه عمل می کند، پیشنهاد شده است.



شکل (۴): ساختار فرایند تنظیم پویای سطوح HLT با استفاده از PSO



شکل (۵): ساختار مد نظر شبکه در شبیه سازی های صورت گرفته

۴-۱- نتایج ارزیابی

که در تمام شبیه‌سازی‌های صورت گرفته، هزینه لینک ارتباطی برابر صفر در نظر گرفته شده است.

بررسی ارتباط H1 به H3: نتیجه بررسی این ارتباط پیش از اجرای CDLF (Before-CLDF) و پس از اجرای (After-CLDF) در جدول (۲) آمده است.

از آنجایی که هدف الگوریتم پیشنهادی در فرایند توزیع شدگی، بهبود مقیاس پذیری بوده است؛ ترافیکی حداکثری بین تجهیزات (سوئیچ‌ها) برقرار شده تا ترافیک کنترل‌کننده نیز به حداکثر برسد، سپس کیفیت ارتباطات قبل و بعد از اجرای الگوریتم پیشنهادی CLDF مورد بررسی قرار گرفته است. لازم به ذکر است

جدول (۲): بررسی اثر اجرای CLDF بر ارتباط H1-H3

Transfer (Gbytes) Before-CLDF	B/W(Gbits) Before-CLDF	Transfer (Gbytes) After-CLDF	B/W(Gbits) After-CLDF
۱۵,۷	۱۳,۵	۳۸,۲	۳۲,۸
۲۱,۹	۱۸,۸	۲۷,۶	۳۲,۳
۲۴,۶	۲۱,۱	۴۰,۵	۳۴,۸
۲۲,۳	۱۹,۱	۴۰,۸	۳۵,۱
۳۹,۸	۳۴,۲	۱۶,۵	۱۴,۲
Average = ۲۴,۸۶	Average = ۲۱,۳۴	Average = ۳۲,۷۲	Average = ۲۹,۸۴

بررسی ارتباط H1 به H7: نتیجه بررسی این ارتباط پیش از اجرای CDLF (Before-CLDF) و پس از اجرای (After-CLDF) در جدول (۳) آمده است.

نتایج جدول فوق نشان می‌دهد، اجرای الگوریتم پیشنهادی CLDF سبب بهبود ۳۱/۶ درصدی در نرخ انتقال و ۳۹/۸ درصدی در به‌کارگیری پهنای باند موجود، شده است.

جدول (۳): بررسی اثر اجرای CLDF بر ارتباط H1-H7

Transfer (Gbytes) Before-CLDF	B/W(Gbits) Before-CLDF	Transfer (Gbytes) After-CLDF	B/W(Gbits) After-CLDF
۱۸,۵	۱۵,۹	۳۷,۲	۳۱,۹
۱۸,۱	۱۵,۵	۳۹,۹	۳۴,۳
۲۳,۸	۲۰,۲	۴۰,۲	۳۴,۵
۱۷,۸	۱۵,۳	۴۰,۳	۳۴,۶
۳۸,۴	۳۲,۹	۱۸,۴	۱۵,۸
Average = ۲۳,۳۲	Average = ۱۹,۹۶	Average = ۳۵,۲	Average = ۳۰,۲۲

برای این منظور یک تابع تشخیص بار کنترل‌کننده (CLDF) پیشنهاد و ارائه شده که با بررسی سطح بار کنترل‌کننده فعلی و دیگر کنترل‌کننده‌ها، در صورتی که سطح بار کنترل‌کننده فعلی از سطح آستانه تعریف شده بالاتر رود، بار ورودی، به کنترل‌کننده دیگری که در شبکه کمترین بار را دارد، منتقل می‌شود. نتایج شبیه‌سازی‌های انجام شده نشان می‌دهد روش پیشنهادی سبب بهبود توان عملیاتی شبکه می‌گردد.

نتایج جدول (۳) نشان می‌دهد، اجرای الگوریتم پیشنهادی CLDF سبب بهبود ۵۰/۹ درصدی در نرخ انتقال و ۵۱/۴ درصدی در به‌کارگیری پهنای باند موجود، شده است. از آنجایی که پردازش‌های پس‌زمینه سیستم‌عاملی، سبب تغییر منابع پردازشی می‌شود، در جداول فوق هر آزمایش ۵ بار تکرار شده و میانگین آن‌ها مورد محاسبه قرار گرفته است.

۵- نتیجه‌گیری

با توجه به ویژگی‌های کنترل‌کننده‌های SDN، در این مقاله یک کنترل‌کننده توزیع‌شده ارائه شده است که با استفاده از مفهوم توزیع بار پردازشی کنترل‌کنندگی، به بهبود مقیاس‌پذیری و توان عملکردی کنترل‌کننده‌های توزیع‌شده می‌پردازد.

۶- مراجع

- [1] A. T. Oliveira, Bruno José C. A. Martins, M. F. Moreno, A. B. Vieira, A. Tadeu A. Gomesy, and A. Ziviani, "SDN-Based Architecture for Providing QoS to High Performance Distributed Applications" Conference Paper, June 2018.

- WAN,” *Sigcomm Computer Communication Review*, vol. 43, pp. 3–14, 2013.
- [6] N. Katta, H. Zhang, M. Freedman, and J. Rexford, “Ravana: Controller Fault-Tolerance in Software-Defined Networking, In: Proceedings of the ACM SIGCOMM Symposium on SDN Research,” SOSR’15, (Santa Clara, CA, USA), June 2015.
- [7] A. Shalimov, “The Runos OpenFlow Controller. In: Software Defined Networks (EWSDN),” 2015 Fourth European Workshop on. IEEE, 2015.
- [8] V. Gramoli, G. Jourjon, and O. Mehani, “Disaster-tolerant storage with SDN,” In: International Conference on Networked Systems, Springer, pp. 293–307, 2015.
- [2] Y. Zhang, “A survey on software defined networking with multiple controllers,” *Journal of Network and Computer Applications*, vol. 103, pp. 101–108, 2017.
- [3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, and J. Rexford, “OpenFlow: enabling innovation in campus networks,” *ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 69–82, 2008.
- [4] G. Lakhani and A. Kothari, “Coordinator controller election algorithm to provide failsafe through load balancing in Distributed SDN control plane,” In: Proceedings of the 1st Springer CCIS series conference, COMS2, March 2020.
- [5] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, and A. Singh, “Experience with a globally-deployed software defined

Presentation and Performance Evaluation of an Efficient Algorithm for a Distributed Controller in SDN

A. Naseri*, M. R. MehrPouya, S. M. Banihashemi

*Imam Hossein Comprehensive University

(Received: 17/02/2020, Accepted: 05/08/2020)

ABSTRACT

In recent years, a new generation of networks called software defined networks (SDN) has been introduced whose main focus is on separating control logic from hardware and concentrating it on a central software called the "controller." SDN improves the network efficiency and reduces the expenses. Despite numerous advantages, SDN faces a lot of challenges such as scalability and reliability that can be fixed through physical decentralization of the control level and introduction of distributed controllers. However, the distributed controllers also face challenges including scalability, stability, and coordination strategy. This research deals with the improvement of distributed controllers' scalability using the concept of load balancing. For this purpose, we have suggested that a controller load detection function (CLDF) be placed on each of the related controllers, and if the load exceeds the threshold level, the new load be transferred to a controller with the least load. The suggested method is implemented on the Floodlight controller in a distributed manner, and implemented on Ubuntu 14.04 operating system using the mininet simulation platform. The simulation results show that suggested method causes an average growth of 31.6 percent on the transition rate.

Keywords: Software Defined Network, Controller, Distribution, Scalability, Load Balancing

* Corresponding Author Email: anaseri@ihu.ac.ir