Journal of Electronical & Cyber Defence
Vol. 9, No. 1, 2021, Serial No. 33

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

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Publisher: Imam Hussein University © Authors

نشریه علمی "پدافند الکترونیکی و سایبری"
سال نهم، شماره ۱، بهار ۱۴۰۰، ص ۱۸۳-۱۷۵

علمی – پژوهشی

ارائه الگوریتم بهبود مقیاس‌پذیری کنترل‌کننده‌های توزیع‌شده شبکه نرم‌افزارمحور

علی ناصری*[۱]، محمدرضا مهرپویا[۲]، سید مرتضی بنی‌هاشمی[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 سال‌های اخیر، نسل جدیدی از شبکه‌ها به نام شبکه‌های نرم‌افزارمحور (SDN) معرفی شده است که توجه اصلی آن بر جداسازی منطق کنترل از سخت‌افزار و تمرکز آن در یک نرم‌افزار مرکزی به نام "کنترل‌کننده" است. SDN با مدیریت یکپارچه شبکه انعطاف‌پذیری و کارایی شبکه را بهبود داده و هزینه‌ها را کاهش می‌دهد. علی‌رغم مزایای بسیار، فناوری SDN با چالش‌های بسیاری ازجمله مقیاس‌پذیری و قابلیت اطمینان سامانه متمرکز مواجه است که می‌توان با تمرکززدایی فیزیکی سطح کنترل و ارائه کنترل‌کننده‌های توزیع‌شده به رفع آن پرداخت. بااین‌حال، کنترل‌کننده‌های توزیع‌شده نیز با چالش‌هایی مانند مقیاس‌پذیری، پایداری و استراتژی هماهنگی مواجه است که در این تحقیق به بهبود مقیاس‌پذیری کنترل‌کننده‌های توزیع‌شده با استفاده از مفهوم توازن بار پرداخته شده است. برای این منظور، پیشنهاد شده یک تابع تشخیص بار کنترل‌کننده (CLDF) بر روی هر یک از کنترل‌کننده‌های مرتبط باهم، قرار داده شده و در صورت افزایش بار بیش از سطح آستانه، بار جدید به کنترل‌کننده دیگری با کمترین بار منتقل شود. روش پیشنهادی به‌صورت توزیع‌شده بر روی کنترل‌کننده Floodlight و با استفاده از بستر شبیه‌سازی mininet و روی سیستم‌عامل اوبونتو ۱۴/۰۴ اجرا شده و نتایج شبیه‌سازی نشان می‌دهد روش پیشنهادی به‌طور میانگین سبب رشد ۳۱/۶ درصدی نرخ انتقال می‌گردد.

کلیدواژه: شبکه نرم‌افزارمحور، کنترل‌کننده، توزیع‌شدگی، مقیاس‌پذیری، توازن بار.

۱- مقدمه

مباوجود مقبولیت گسترده اینترنت، شبکه‌های IP سنتی پیچیده بوده و مدیریت آن‌ها 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 پیچیدگی‌های پیکربندی، شبکه می‌بایست 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 پاسخ‌گویی در شبکه‌های امروزی به‌طور کامل قابل اجرا نبوده و از طرفی اجرای سیاست‌های موردنیاز در محیط‌های پویا بسیار چالش‌برانگیز است.

دو چالش عمده شبکه‌های سنتی که قرار است در شبکه‌های جدید تحت عنوان SDN بهبود یابد عبارت‌اند از:

- نبود دید انتزاعی در سطح کنترل: بدین معنا که پیکربندی و الگوریتم‌های مربوطه در سطح کنترل، به‌صورت محلی و مختص یک دستگاه (سوئیچ یا مسیریاب) است و در ساختار شبکه‌ای و به‌طورکلی نمی‌توان یک پیکربندی را به همه سوئیچ‌ها یا مسیریاب‌ها اعمال کرد.
- سطح کنترل فاقد انعطاف بوده و ویژگی‌های محدودی را پشتیبانی می‌کند: به‌عنوان‌مثال، اگر قرار باشد یک پروتکل مسیریابی جدید نوشته شود، باید از صفر شروع شود. چراکه یک کتابخانه برنامه‌نویسی منعطف وجود ندارد که بتوان بر اساس آن پروتکل جدیدی را نوشت.

ابزارهای فعال و میانی شبکه‌ها دارای دو واحد اساسی می‌باشند، «سطح کنترل[۲]» که تصمیم می‌گیرد چگونه با ترافیک شبکه رفتار شود و «سطح داده[۳]» که ترافیک را با توجه به تصمیمات گرفته‌شده توسط واحد کنترل، ارسال می‌کند. شبکه‌های امروزی به‌صورت عمودی مجتمع می‌باشند، بدین معنی که سطح کنترل و سطح داده در تجهیزات شبکه یکپارچه بوده، درنتیجه باعث کاهش انعطاف‌پذیری آن شده و مانع تحول و نوآوری در زیرساخت شبکه می‌شود. در معماری شبکه‌های SDN

[2] Control Plane
[3] Data Plane

*رایانامه نویسنده پاسخگو: anaseri@ihu.ac.ir

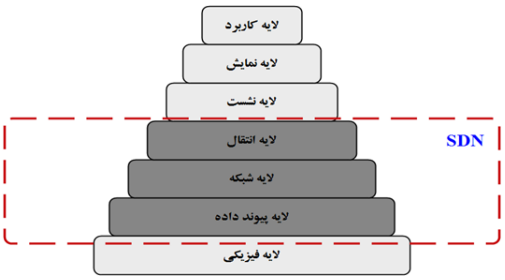
* این مقاله یک مقاله با دسترسی آزاد است که تحت شرایط و ضوابط مجوز Creative Commons Attribution (CC BY) توزیع شده است.
© نویسندگان

ناشر: دانشگاه جامع امام حسین (ع)

سعی شده توانایی و هوشمندی شبکه‌های کامپیوتری بیشتر شود. بر اساس تعریف بنیاد شبکه‌های باز (ONF[1])، 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[2] نام دارد امکان‌پذیر است. برنامه‌های کاربردی هم در بالای سطح کنترل در سطح کاربرد[3] قرار می‌گیرند.

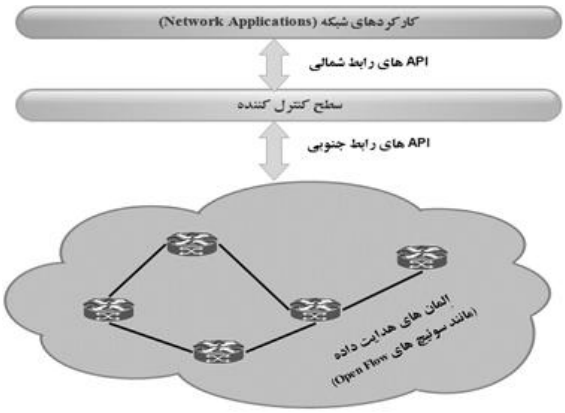
یک کنترل‌کننده نرم‌افزاری متمرکز که دیدی از کل شبکه را در اختیار دارد، بسیار مؤثرتر و هوشمندتر از بخش‌های کنترلی پراکنده مسیریاب‌ها و سوئیچ‌ها عمل خواهد کرد. با یک کنترل‌کننده متمرکز، قابلیت‌هایی مانند برنامه‌ریزی، مقیاس‌پذیری، انعطاف‌پذیری، خودکارسازی و توسعه نرم‌افزاری به شبکه اضافه می‌شود.

چنانچه قرار باشد جایگاه فناوری SDN را در لایه‌های مدل مرجع OSI[4] 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): نمایی ساده از معماری شبکه‌های نرم‌افزارمحور [۱]

[1] Open Networking Foundation (ONF)
[2] Controller
[3] Application Layer
[4] Open Systems Interconnection

بخش‌های مختلف این شکل عبارت‌اند از:

- لایه هدایت روبه‌جلوی بسته: این بخش به‌طور مستقیم مسئول ارسال داده‌ها بر اساس جدول‌های برنامه‌ریزی‌شده توسط کنترل‌کننده‌ها است.

رابط جنوبی: توسط ONF و با نام OpenFlow استانداردشده 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، مورداستفاده 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 مورداستفاده قرار می‌گیرد.

به‌طورکلی 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 درنتیجه تجهیزات شبکه به مؤلفه‌های هدایت‌گر ساده مبدل شده‌اند.

تصمیمات هدایت‌گری به‌جای 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 داده‌شده 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، مواجه است. چالش فعلی پیاده‌سازی و استفاده عملی از SDN، به‌کارگیری کنترل‌کننده

به‌صورت متمرکز است. نقطه مقابل تمرکز، توزیع‌شدگی است که بنا دارد وابستگی کل شبکه به یک نقطه واحد را مرتفع سازد. با رفع مسئله تمرکز در کنترل‌کننده‌ها و ورود به بحث کنترل‌کننده‌های توزیع‌شده، می‌توان به این چالش‌ها پاسخ گفت. همچنین هرچند کنترل‌کننده‌ها سبب بهبود کارایی مدیریت شبکه می‌شوند، اما همچنان نمی‌توانند پاسخگوی تقاضاهای جریان‌های حجیم و آمارهای مانیتورینگ شبکه باشند. به‌عنوان‌مثال یکی از کنترل‌کننده‌های 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 به‌عنوان «مسئله مقیاس‌پذیری» نامیده می‌شود به ذات متمرکز معماری این شبکه برمی‌گردد. لذا فهم مسئله مقیاس‌پذیری در کنترل‌کننده‌های SDN و بهبود آن از طریق مفاهیم توزیع‌شدگی یک چالش اساسی در سازگاری این معماری با شبکه‌های مقیاس‌بزرگ و شبکه‌های با جریان بالا است که حل این چالش، هدف ارائه این مقاله می‌باشد.

ادامه این مقاله بدین‌صورت است که در بخش۲، مفاهیم کنترل‌کننده‌های توزیع‌شده ارائه شده و در بخش ۳، روش پیشنهادی بهبود مقیاس‌پذیری کنترل‌کننده‌های توزیع‌شده SDN شرح داده شده است. در بخش ۴ شبیه‌سازی و ارزیابی روش پیشنهادی انجام شده و در بخش ۵ از مباحث مطرح‌شده، نتیجه‌گیری شده است.

۲- کنترل‌کننده‌های توزیع‌شده

در معماری SDN یک کنترل‌کننده متمرکز منطقی شبکه به کار گرفته شده است [۲]. برای دست‌یابی به تمرکز منطقی در شبکه‌های نرم‌افزارمحور، دو نوع پیاده‌سازی وجود دارد: کنترل‌کننده متمرکز و کنترل‌کننده توزیع‌شده.

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

الف- مسئله مقیاس‌پذیری: کنترل‌کننده 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 کنترل‌کننده متمرکز را محدود کنند [۳ و ۴]. بااین‌حال این استراتژی، هوش را به سوئیچ اضافه می‌کند و بدین ترتیب مفهوم SDN نقض می‌گردد.

ب- مسئله قدرت: کنترل‌کننده‌های متمرکز SDN، دارای یک مشکل واحد با عنوان تک نقطه خرابی[1] هستند. وقتی یک کنترل‌کننده متمرکز خراب می‌شود، سوئیچ‌ها توانایی خود را برای ارسال بسته‌های جدید از دست خواهند داد و درنهایت، کل شبکه دچار خرابی و شکست خواهد شد. برای حل این مسئله، محققان می‌توانند سوئیچ را به‌صورت ترکیبی (OpenFlow-OpenFlow hybrid) پیکربندی کنند [۵]. سوئیچ وقتی‌که نمی‌تواند به کنترل‌کننده SDN متصل شود، از سوئیچ مدرن (به‌عنوان مثال از سوئیچ نرم‌افزار پایه) به سوئیچ سنتی تبدیل خواهد شد.

کنترل‌کننده توزیع‌شده شبکه نرم‌افزارمحور: کنترل‌کننده توزیع‌شده SDN تضمین می‌دهد هر دو مشکل مطرح‌شده در کنترل‌کننده متمرکز را حل کند. ایده کلی تشکیل چندین کنترل‌کننده در کنار یکدیگر است که می‌تواند بار را در شبکه به‌طور مساوی به اشتراک بگذارد. علاوه بر این، یک کنترل‌کننده می‌تواند کنترل‌کننده دیگری را در هنگام خراب شدن، کنترل کند. جزئیات این رویه‌ها پیچیده است که ممکن است نیاز به برخی از فناوری‌های مورداستفاده در دستگاه‌های توزیع‌شده داشته باشد.

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

Jimenez و سایرین [۷] نشان داده‌اند که تعداد کنترل‌کننده‌های انتخاب‌شده و محل قرارگیری آن‌ها باعث تغییر در عملکرد شبکه می‌شود. وقتی تعداد کنترل‌کننده‌ها به‌طور 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 کنترل‌کننده‌ها را اجرا کند. Elasticon [۸] سعی در دست‌یابی به ۱۰۰٪ استفاده از ایده جدید دارد که این خوشه از کنترل‌کننده‌ها را قادر می‌سازد تا بر اساس بار شبکه، کنترل‌کننده‌ها را اضافه یا حذف کنند. خوشه مقدار آستانه تراکم را حفظ می‌کند. هنگامی‌که ترافیک از سطح آستانه تراکم عبور می‌کند، خوشه می‌تواند یک کنترل‌کننده جدید برای پیوستن به خوشه و کمک به مدیریت شبکه به آن اضافه کند. بعد از اتصال کنترل‌کننده جدید، کنترل‌کننده باید به خوشه اختصاص داده شود. علاوه‌بر این، خوشه می‌تواند زمانی که ترافیک تا آستانه پایین کاهش می‌یابد، کنترل‌کننده‌ها را از خوشه حذف کند.

[1] Single Point of Failure (SPF)

۳- روش پیشنهادی

همان‌طور که پیش‌ازاین بیان شد، یکی از چالش‌های اصلی کنترل‌کننده‌های توزیع‌شده، مسئله مقیاس‌پذیری آن‌ها است. درواقع ازآنجایی‌که کنترل‌کننده متمرکز و منفرد خودش می‌توان محل تک نقطه خرابی باشد، پیشنهاد داده شد تا با ایجاد افزونگی در تعداد کنترل‌کننده‌ها، این مشکل برطرف شود. پس‌ازآن، بیان شد که می‌توان با استفاده از فناوری توزیع‌شدگی در کنترل‌کننده‌های افزونه، حداکثر بازدهی را از منابع موجود اخذ کرد تا بدین ترتیب مقیاس‌پذیری کنترل‌کننده‌های شبکه بهبود یابد. ازآنجایی‌که در این حالت، ممکن است منابع برخی از کنترل‌کننده‌ها به حالت اشباع برسد، باعث می‌شود کارایی این کنترل‌کننده‌ها به‌شدت و به‌طور نمایی کاهش یابد. لذا در پژوهش جاری مفهوم کنترل‌کننده «توزیع‌شده، اما با منطق متمرکز» ارائه‌شده و موردبررسی قرارگرفته است. راه‌حل پیشنهادی متکی بر مجازی‌سازی کنترل‌کننده شبکه از طریق چندین نمونه کنترل‌کننده سازمان‌یافته در خوشه است. مجازی‌سازی کنترل‌کننده می‌تواند بر چالش مقیاس‌پذیری غلبه کرده و بر کارایی کنترل‌کننده SDN در میزبانی مرکز داده‌ی بزرگ، بیفزاید.

الف) طرح پیشنهادی لایه کنترل‌کننده SDN در معماری توزیع‌شده

توازن بار مبتنی بر اطلاعات جدول توازن، می‌تواند به‌عنوان روشی از سوی اپراتورها جهت عملکرد پایدار شبکه مورداستفاده قرار گیرد که این روش ضمن توازن بار، به کمینه کردن تأخیر ارتباط نیز اهمیت می‌دهد.

تابع متوازن کننده بار می‌تواند در اجزاء مختلف شبکه پیاده‌سازی شود که در این مقاله به‌عنوان جزئی مجزا به نام تابع تشخیص بار کنترل‌کننده (CLDF[1]) تعریف و طراحی شده است. وظایف CLDF شامل جمع‌آوری دوره‌ای اطلاعات بار کنترل‌کننده‌ها و به کار انداختن روش توازن بار است .در این روش، n به‌عنوان تعداد کنترل‌کننده‌ها و m به‌عنوان تعداد تجهیزات میانی در ارتباط با کنترل‌کننده‌ها تعریف می‌شود. وقتی تفاضل بار بین دو کنترل‌کننده از مقدار آستانه بالا (HLT[2]) بیشتر شد، روالی برای انتقال ترافیک‌های جدید از تجهیزات میانی از یک کنترل‌کننده با بار زیاد به کنترل‌کننده‌ی با بار کم آغاز می‌شود. سطح آستانه بالا زمانی اتفاق می‌افتد که حجم بار به بیش از ۸۰ درصد توان کنترل‌کننده برسد.

طی این فرآیند یک یا چند دستگاه میانی که به کنترل‌کننده با بار زیاد متصل بودند، به کنترل‌کننده با بار کم متصل می‌شود و این روند تا زمانی که اختلاف بار از مقدار آستانه پایین (LLT[3]) کمتر نشود، ادامه پیدا می‌کند. هر زمان بار کنترل‌کننده‌ای که به مرز اشباع (HLT) رسیده بود، پردازش‌شده و از سطح LLT کمتر شود، آن کنترل‌کننده مجدداً خود را به خوشه کنترل‌کننده‌هایی معرفی می‌کند که آماده دریافت ترافیک جدید است. بدیهی است در این حالت، این کنترل‌کننده اولین گزینه برای دریافت بار ترافیکی از کنترل‌کننده‌ای است که به مرز اشباع پردازشی (HLT) رسیده؛ به‌شرط اینکه کنترل‌کننده دیگری قبل از آن به سطح LLT نرسیده باشد. سطح آستانه پایین زمانی اتفاق می‌افتد که حجم بار به کمتر از ۲۰ درصد توان کنترل‌کننده برسد.

برای کنترل بهتر بار ترافیک، چند عامل دیگر نیز باید در نظر گرفته شود. اول‌ازهمه، بازه زمانی گزارش دهی اطلاعات بار کنترل‌کننده‌ها به CLDF است. این بازه باید طوری انتخاب شود که تعاملی بین دقت توازن بار و سربار پیام‌دهی باشد. مثلاً اگر بازه زمانی خیلی کوتاه باشد، هر کنترل‌کننده با گزارش دهی مکرر، بار قابل‌توجهی را به شبکه تحمیل می‌کند. با توجه به این‌که بار ترافیکی هر کنترل‌کننده با مقدار بار ورودی به شبکه رابطه مستقیم دارد، زمان گزارش دهی متناسب با مقدار نرخ ارتباط جدید در شبکه در نظر گرفته می‌شود، بدین‌صورت که هر چه نرخ تماس (ارتباط) جدید بیشتر باشد، بازه گزارش دهی کوتاه‌تر می‌شود. این شیوه باعث می‌شود که کنترل‌کننده در برابر بار ناگهانی به‌خوبی پاسخ دهد و به‌سرعت 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 تجزیه‌وتحلیل اطلاعات بار ارسالی از کنترل‌کننده‌ها، CLDF درصد مشغولیت کنترل‌کننده‌ها را با حد آستانه اشباع آن کنترل‌کننده (HLT) مقایسه می‌کند. به‌محض اینکه سطح بار یک کنترل‌کننده از مقدار آستانه بالا (HLT) بیشتر شود، با تقریب اینکه هر دستگاه میانی چه کسری از بار کنترل‌کننده‌ها را ایجاد می‌کند، ابتدا آن کنترل‌کننده را (درصورتی‌که کنترل‌کننده‌ای در حالت غیراشباع وجود داشته باشد) موقتاً از خوشه خارج کرده و اقدام به اتصال مجدد تعداد مناسب تجهیزات میانی از کنترل‌کننده با بار زیاد به کنترل‌کننده با بار کم می‌کند. این روند تا زمانی که میزان بار پردازش نشده کنترل‌کننده به سطح LLT نرسد، ادامه می‌یابد. هنگامی‌که سطح بار پردازش نشده بر روی کنترل‌کننده به سطح LLT رسید، آن کنترل‌کننده مجدداً به خوشه معرفی شده و آماده دریافت ترافیک

[1] Controller Load Detection Function (CLDF)
[2] High Level Threshold (HLT)
[3] Low Level Threshold (LLT)

کنترل‌کننده خواهد بود. بدین طریق با توجه به مقادیر آستانه، توازن بار می‌تواند با دقت مطلوب انجام شود.

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

همان‌طور که بیان شد، اساس الگوریتم CLDF پیشنهادی، بر بررسی سطوح آستانه بالا و پایین کنترل‌کننده‌ها نسبت به بار ترافیکی پردازش نشده و موجود در کنترل‌کننده‌ها استوار است. نکته بسیار مهم در اینجا این است که در یک سخت‌افزار ممکن است چند کنترل‌کننده باهم به‌صورت ماشین مجازی فعالیت کنند (چه‌بسا بسیاری از کنترل‌کننده‌های تجاری موجود مانند Onus، Floodlight و OpenDeyLight قابلیت نصب بر روی مجازی‌ساز ESXi را دارند)، همچنین حتی اگر به‌عنوان تنها سیستم‌عامل موجود در سخت‌افزار باشند نیز پردازش‌های پس‌زمینه سیستم‌عامل سبب می‌شود توان پردازشی متغیری در اختیار کنترل‌کننده قرار بگیرد. این امر سبب خواهد شد تا نتوان حد آستانه ثابتی را برای کنترل‌کننده‌ها تعریف کرد (خصوصاً حد آستانه بالا).

با توجه به توضیحات فوق، جهت بهبود کارایی الگوریتم پیشنهادی، سازوکاری برای تعریف حد آستانه بالای پویا (D_HLT[1]) طراحی شده است. ازآنجایی‌که اطلاعات پیرامون به‌کارگیری منابع مانند عملکرد CPU، مصرف حافظه، سرعت ورود پیام‌های کنترل، سرعت پردازش ترافیک کنترل‌کننده و نوع ترافیک ورودی کنترلی از الگوی خاصی پیروی نمی‌کنند، همواره در مسائل تخصیص منابع، تعیین حد آستانه جزو مسائل NP[2]-Hard (مسائلی که پیچیدگی چندجمله‌ای غیرقطعی دارند) دسته‌بندی می‌شود. لذا برای حل این مسائل نمی‌توان از روش‌های الگوریتمی عادی بهره برد.

ب-۱) شرایط مسئله D-HLT

فرض کنیم که m تعداد کنترل‌کننده‌های موجود در خوشه شبکه کنترل‌کننده‌ها باشد:

$$CTRL = \{CTRL1, CTRL2, \ldots, CTRLm\} \qquad (۱)$$

جهت تعیین حد آستانه برای تابع توزیع بار CLDF، باید بتوان نقطه بهینه‌ای که تناسب مناسبی بین ترافیک کنترل‌کننده‌های خوشه داشته باشد، برقرار کرد. این تناسب را می‌توان با استفاده از توابع بهینه‌سازی ایجاد نمود. درواقع این تابع بهینه‌سازی سعی دارد زمان پردازش ترافیک (T_{exe}) و زمان انتقال ترافیک (T_{trans}) از یک کنترل‌کننده به کنترل‌کننده بعدی را کمینه سازد که می‌توان از آن به‌عنوان تابع هدف بهره برد. دو معیار زمان اجرا و زمان انتقال از طریق روابط زیر قابل‌محاسبه می‌باشند:

$$T_{exe} = \frac{Number\ of\ Trafic\ Tasks}{Processor\ Speed}\ (\text{sec}) \qquad (۲)$$

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

$$T_{trans} = \sum_{i=1}^{n}\sum_{k=1}^{m}\sum_{z=1}^{m}(1 - x_{ik}x_{iz})\frac{DT_{kz}}{B_{kz}} \qquad (۳)$$

که در آن، DT_{kz} حجم داده‌ای است که بین k امین و z امین کنترل‌کننده تبادل می‌شود. B_{kz} پهنای باند بین دو کنترل‌کننده k و z است. x_{iz}: اگر i امین ترافیک از کنترل kام به کنترل‌کننده zام 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 به‌صورت زیر بیان می‌گردد:

$$U = \alpha T_{exe} + (1 - \alpha)T_{trans} \rightarrow min \qquad (۴)$$

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 استفاده‌شده و سرعت پردازش ترافیک کنترلی، باید سطوح مختلف HLT موردبررسی قرارگرفته تا در نقاطی که سطح کمینه این تابع تأمین می‌گردد، HLT هر کنترل‌کننده انتخاب شود.

همان‌طور که بیان شد، کمینه‌سازی رابطه فوق، یک مسئله NP-HARD است که برای حل آن نیاز به روش‌های فرا ابتکاری است. بامطالعه در روش‌های 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 گام‌های زیر طی گردد:

گام ۱: جمع‌آوری اطلاعات راجع به کنترل‌کننده‌ها به‌عنوان ورودی الگوریتم PSO شامل تعداد و سرعت پردازش پردازنده هر کنترل‌کننده، میزان حافظه آزاد، تعداد وظایف ترافیکی در حال پردازش، زمان پردازش هر ترافیک ورودی، نرخ ترافیک ورودی و ...

[1] Dynamic HLT
[2] Non-deterministic Polynomial-time HARDness (NP-HARD)
[3] Particle Swarm Optimization (PSO)

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.

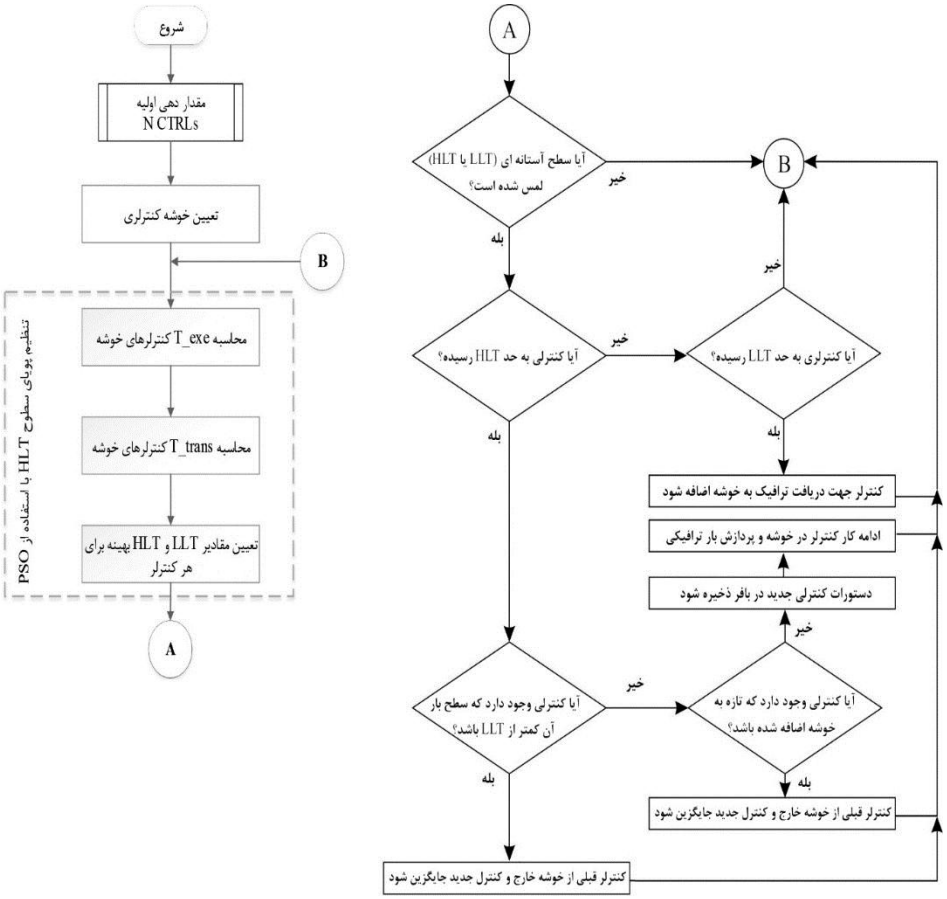
در روش پیشنهادی به‌کارگیری الگوریتم PSO، تطبیق پارامترها بدین‌صورت است که: ذرات، همان ترافیک کنترلی هستند که باید در کنترل‌کننده مورد پردازش قرار گیرند. هنگامی‌که سطح خاصی از بار ترافیکی به یک کنترل‌کننده اختصاص داده می‌شود، بر اساس خروجی الگوریتم PSO، ترافیک جدید به کنترل‌کننده‌های دیگر منتقل خواهد 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

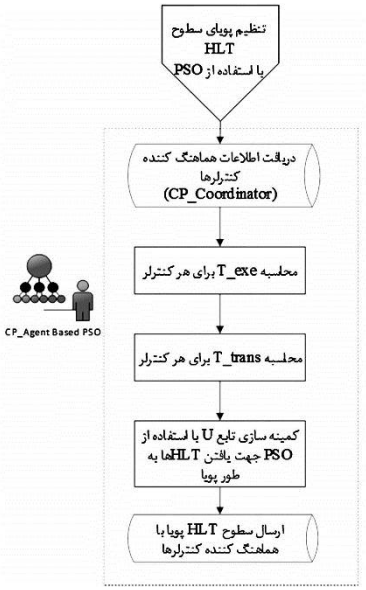
با توجه به توضیحات داده‌شده، نمای بلوکی الگوریتم پیشنهادی به‌صورت شکل (۳) است که در آن اثر D-HLT بر روی الگوریتم پیشنهادی قابل‌مشاهده است.



شکل (۳): الگوریتم پیشنهادی

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

با توجه به توضیحات فوق می‌توان گفت با استفاده از کنترل‌های توزیع‌شده می‌توان به حل چندین چالش اساسی SDN، مانند تک نقطه خرابی، مقیاس‌پذیری، تأخیر سراسری، قابلیت اطمینان و قابلیت دسترس‌پذیری پرداخت. هرچند تاکنون روش‌های مختلفی برای توزیع بار بین تجهیزات شبکه با استفاده از کنترل‌کننده مرکزی ارائه شده است، اما در این فصل یک الگوریتم جدید برای توزیع بار کنترلی بین کنترل‌کننده‌های توزیع‌شده شبکه که به‌صورت یک خوشه عمل می‌کنند، پیشنهاد شده است.



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

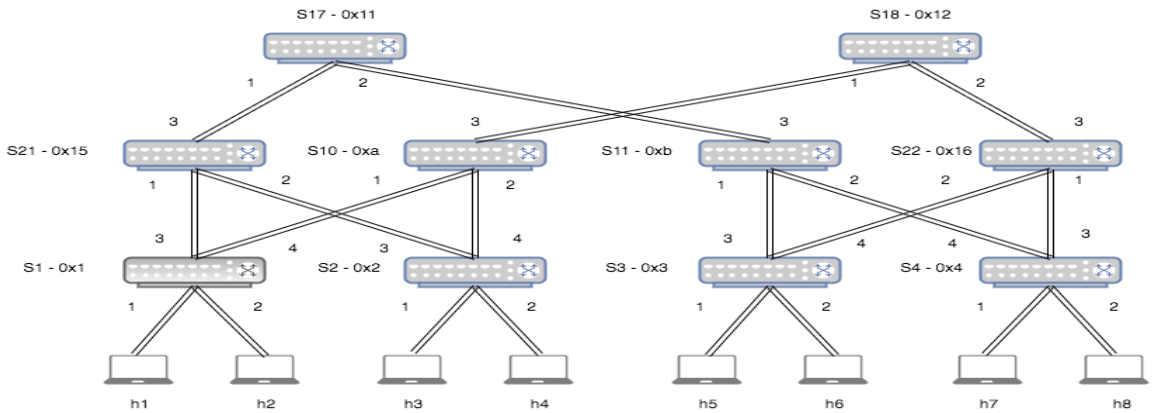
۴- شبیه‌سازی و ارزیابی

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

کنترل‌کننده مورداستفاده: برای انتخاب کنترل‌کننده، گزینه‌های زیادی وجود دارد. کنترل‌کننده Floodlight با جامعه کاربری و پشتیبانی بسیار فعال خود و واسط گرافیکی کاربرپسند و همچنین مستندات کاملی که دارد، می‌تواند انتخاب مناسبی باشد. در نسخه جدید این کنترل‌کننده، امکان تنظیم نقش کنترل‌کننده و استفاده از چندین کنترل‌کننده توزیعی نیز فراهم شده است.

بستر نرم‌افزاری شبیه‌سازی: شبیه‌سازی‌های صورت گرفته، در سیستم‌عامل اوبونتو ۱۴،۰۴و با استفاده از کنترل‌کننده Floodlight نسخه ۱/۲ و شبیه‌ساز Mininet که یک شبیه‌ساز اختصاصی کنترل‌کننده‌های SDN است انجام شده است. همچنین به‌منظور ارزیابی شبیه‌سازی‌ها، بررسی و رصد عملکرد و سرعت پورت‌ها و آدرس‌ها از ابزار WireShark استفاده شده است.

ساختار مورداستفاده: 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 به‌صورت شکل (۵) است.



شکل (۵): ساختار مد نظر شبکه در شبیه‌سازی‌های صورت گرفته

۴-۱- نتایج ارزیابی

ازآنجایی‌که هدف الگوریتم پیشنهادی در فرایند توزیع‌شدگی، بهبود مقیاس‌پذیری 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 موردبررسی قرار گرفته است. لازم به ذکر است 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) (CLDF در جدول (۲) آمده 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**

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.

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

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. ازآنجایی‌که 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.

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

۶- مراجع

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

[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 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.

Journal of Electronical & Cyber Defence
Vol. 9, No. 1, 2021, Serial No. 33

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