ITRC
ICT Research Institute
(Iran Telecommunication Research Center)

Journal of Information and Communication Technology

Iranian Society of Information and Communication Technology

Volume 16, Issue 61-62, Autumn and Winter 2024, pp. 55-71

Dynamic Load Balancing Improvement in Software-Defined Networks Using Fuzzy Multi-Objective Programming Algorithms

Mohammadreza Forghani[1], Mohammadreza Soltanaghaei[1*], Farsad Zamani Boroujeni[2]

[1] Department of Computer Engineering, Isfahan (Khorasgan) Branch, Islamic Azad University, Isfahan, Iran
[2] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

Received: 25 December 2023, Revised: 25 April 2024, Accepted: 28 April 2024
Paper type: Research

Abstract

Software-Defined Networking (SDN) has been recognized as an efficient approach in the field of communication technology, aiming to improve the performance and efficiency of computer networks, thus reducing costs. One of the key challenges in SDN is load balancing among nodes. Solving this challenge leads to improved response time and network performance. Nowadays, various methods have been proposed for load balancing in SDN, but they have not yet reached the ideal state. In this article, a new method is presented to enhance load balancing and reduce response time. This method utilizes multi-objective evolutionary algorithms and fuzzy weighting. In the proposed method, factors such as bandwidth, traffic status, link buffer, and desired router are taken into account, and the best path and router with desired load balancing for information flows are selected with the minimum time. One prominent advantage of this method is the possibility of performing load balancing automatically without the need for human intervention. Experimental results demonstrate that the proposed method shows a significant improvement of approximately 14.8% in response time compared to other methods, while maintaining load balancing in SDNs. By using the proposed method, in addition to improving service quality and user satisfaction, response time will also be enhanced. In summary, the proposed method is introduced as a viable approach in SDNs and exhibits superiority over existing methods.

Keywords: Software-Defined Networks (SDN), Load Balancing, Multi-Objective Optimization, Fuzzy Voting Algorithm, Response Time.

* Corresponding Author's email: soltan@khuisf.ac.ir

دوفصلنامه
فناوری اطلاعات و ارتباطات ایران

سال شانزدهم، شماره‌های ۶۱ و ۶۲، پاییز و زمستان ۱۴۰۳، صفحه ۵۵ الی ۷۱

بهبود توازن بار پویا و زمان پاسخ در شبکه‌های نرم‌افزارمحور با بهره‌گیری از الگوریتم‌های برنامه‌ریزی آرمانی چند منظورهٔ فازی

محمدرضا فرقانی[۱]، محمدرضا سلطان آقایی[۱*]، فرساد زمانی بروجنی[۲]

[۱] گروه مهندسی کامپیوتر، واحد اصفهان(خوراسگان)، دانشگاه آزاد اسلامی، اصفهان، ایران

[۲] گروه مهندسی کامپیوتر، واحد علوم و تحقیقات تهران، دانشگاه آزاد اسلامی، تهران، ایران

تاریخ دریافت: ۱۴۰۲/۱۰/۰۴ تاریخ بازبینی: ۱۴۰۳/۰۲/۰۶ تاریخ پذیرش: ۱۴۰۳/۰۲/۰۹

نوع مقاله: پژوهشی

چکیده

شبکه نرم‌افزارمحور به عنوان یک رویکرد کارآمد در حوزه فناوری ارتباطات شناخته شده و هدف آن بهبود عملکرد و افزایش بهره‌وری و کاهش هزینه در شبکه می‌باشد. یکی از چالش‌های اساسی در شبکه‌های نرم‌افزارمحور، توازن بار بین گره‌ها است. حل این چالش باعث بهبود زمان پاسخ و عملکرد شبکه می‌شود. امروزه روش‌های متعددی برای توازن بار در شبکه‌های نرم‌افزارمحور ارائه شده است، اما هنوز به وضعیت ایده‌آل نرسیده‌اند. در این مقاله، یک روش جدید برای بهبود توازن بار و کاهش زمان پاسخ ارائه می‌شود. این روش از الگوریتم‌های برنامه‌ریزی آرمانی چند منظوره و وزن‌دهی فازی بهره می‌برد. در روش پیشنهادی، فاکتورهایی مانند پهنای باند، وضعیت ترافیک و گذردهی گره و مسیرها ، بافر و مسیریاب مد نظر قرار می‌گیرند و بهترین مسیر و مسیریاب با توازن بار مطلوب برای جریان‌های اطلاعات با کمترین زمان پاسخ انتخاب می‌شوند. یکی از مزایای بارز این روش، امکان انجام توازن بار به صورت خودکار و بدون نیاز به مداخله انسان است. نتایج تجربی نشان می‌دهد که روش پیشنهادی نسبت به روش‌های دیگر، بهبود قابل توجهی در زمان پاسخ حدود ۱۴٫۸ درصد را نشان می‌دهد و همچنین توازن بار شبکه‌های نرم‌افزارمحور را حفظ می‌کند. با استفاده از روش پیشنهادی، علاوه بر بهبود کیفیت سرویس و رضایت کاربران، زمان پاسخ نیز بهبود خواهد یافت. به طور خلاصه، روش پیشنهادی به عنوان یک رویکرد قابل استفاده در شبکه‌های نرم‌افزارمحور مطرح است و نسبت به روش‌های موجود برتری دارد.

کلیدواژگان: شبکه نرم‌افزارمحور، توازن بار پویا، بهینه‌سازی چند منظوره، الگوریتم رأی‌گیری فازی، زمان پاسخ.

* رایانامهٔ نویسنده مسؤول: soltan@khuisf.ac.ir

۱- مقدمه

در سال‌های اخیر، با توسعه پیشرفت‌های فناوری اطلاعات و ارتباطات مانند تلفن همراه، سرویس‌های ابری، شبکه‌های اجتماعی، داده‌های حجیم و چندرسانه‌ای، مدیریت و پیکربندی این فناوری‌ها بسیار پیچیده، چالش‌برانگیز و زمان‌بر شده است. در حال حاضر، دسترسی به پهنای باند بالا و قابلیت گسترش و مدیریت پویا نیز از اهمیت ویژه برخوردار است [۱, ۲]. بسیاری از شبکه‌های سنتی به صورت سلسله‌مراتبی ساختاردهی می‌شوند، که با استفاده از گره‌های اترنت در یک ساختار شبکه‌ای درختی، تشکیل می‌شوند. این معماری در مورد ارتباطات مشتری یا سرویس‌دهنده قابل مشاهده است، اما برای ارتباطات پویا و نیازهای شرکت‌ها، مراکز داده و رسانه‌های سرویس‌دهنده کافی نیست [۳-۵]. معماری فعلی شبکه‌ها طوری طراحی نشده است که بتواند نیازهای کنونی شرکت‌ها، سرویس‌گیرنده‌های مخابراتی و کاربران را برآورده کند.

محدودیت‌هایی مانند پیچیدگی، سیاست‌های متناقض، عدم مقیاس‌پذیری و وابستگی به تولیدکننده محصول، هماهنگی نامناسب بین نیازهای بازار و قابلیت‌های شبکه را محدود می‌کند و صنعت فناوری اطلاعات را به یک نقطه انحراف می‌برد. به منظور جلوگیری از این روند، معماری شبکه‌های نرم‌افزارمحور[1] (SDN) معرفی شده‌اند و استانداردها، پروتکل‌ها و نرم‌افزارهای مرتبط با آنها توسعه یافته‌اند [۶].

با توجه به افزایش کاربران و نیاز به سرویس‌های جدید و با کیفیت بالاتر و همچنین گسترش و پیچیده‌تر شدن روزافزون سخت‌افزارها و نرم‌افزارها، فناوری شبکه‌های نرم‌افزارمحور به عنوان یک رویکرد کارآمد در حوزه فناوری ارتباطات مطرح می‌شود. هدف اصلی این فناوری بهینه‌سازی استفاده از منابع، بهبود عملکرد و بهره‌وری و کاهش هزینه‌ها در شبکه‌های کامپیوتری است [۶, ۷].

این فناوری به منظور افزایش هوشمندی شبکه‌ها و ارتقای عملکرد آن‌ها، با انتقال قسمت کنترل داده‌ها از سوئیچ و مسیریاب سخت‌افزاری به لایه‌های نرم‌افزار مجازی شبکه و استفاده از یک کنترل‌کننده نرم‌افزاری متمرکز، قابلیت‌هایی نظیر برنامه‌ریزی، مقیاس‌پذیری، انعطاف‌پذیری، اتوماسیون، هوشمندی و توسعه نرم‌افزاری شبکه را برای سازمان‌ها فراهم می‌کند. این ایده ابتدا در سال ۲۰۰۵ مطرح شد و از سال ۲۰۱۰ به شدت رشد کرد. با تشکیل بنیاد ONF[2] و عضویت بیش از ۸۰ شرکت بزرگ صنعت شبکه در آن و تدوین استاندارد OpenFlow، این فناوری وارد فاز جدیدی شد. اولین محصولات کاربردی این شبکه از سال ۲۰۱۳ به بازار عرضه شد [۸, ۹].

در سال‌های اخیر، توسعه شبکه‌های نرم‌افزارمحور شتاب بیشتری گرفته است و اکثر تجهیزات تجاری در حال حاضر پروتکل OpenFlow را که اساس شبکه‌های نرم‌افزارمحور است، پشتیبانی می‌کنند [۱۰, ۱۱]. این توسعه سریع شبکه‌های نرم‌افزارمحور منجر به تأسیس سازمان ONF توسط شرکت‌های بزرگی مانند گوگل، فیسبوک، یاهو، مایکروسافت، Verizon و Deutsche Telekom شد. هدف اصلی این سازمان، توسعه شبکه‌های نرم‌افزارمحور با استفاده از استانداردهای متن‌باز بوده است [۱۲, ۱۳].

تحقیق حاضر بهبود کیفیت سرویس در شبکه‌های نرم‌افزارمحور با تمرکز بر مهندسی ترافیک، توازن بار ترافیکی و کاهش زمان پاسخ را مورد بررسی قرار می‌دهد. در این تحقیق، با استفاده از تکنیک‌های بهینه‌سازی ابتکاری و توسعه الگوریتم‌های برنامه‌ریزی آرمانی چند منظورهٔ فازی، تلاش می‌شود تا توازن بار ترافیکی به صورت پویا بهبود یابد و زمان پاسخ در شبکه‌های نرم‌افزارمحور کاهش یابد [۱۴, ۱۵].

در شبکه‌های نرم‌افزارمحور، نقش لایه‌های نرم‌افزاری در طراحی و مدیریت شبکه بسیار حائز اهمیت است. با افزایش نقش لایه‌های نرم‌افزاری و کاهش نقش سخت‌افزار، امکان انعطاف‌پذیری و پیکربندی آسان‌تر شبکه‌ها فراهم می‌شود. این امر به شبکه‌ها اجازه می‌دهد تا با توجه به نیازهای مختلف، ترافیک را بهبود دهند و بار ترافیکی را به طور هوشمندانه توزیع کنند [۶, ۱۶].

در این تحقیق، با بهره‌گیری از الگوریتم‌های برنامه‌ریزی آرمانی چند منظورهٔ فازی و تکنیک‌های بهینه‌سازی ابتکاری، تلاش می‌شود تا توازن بار ترافیکی در شبکه‌های نرم‌افزارمحور به صورت پویا بهبود یابد. همچنین، با اعمال این الگوریتم‌ها و روش‌ها، زمان پاسخ در شبکه‌های نرم‌افزارمحور بهبود می‌یابد. این بهبودها می‌تواند منجر به کاهش تأخیر در ارائه سرویس‌ها، افزایش بهره‌وری شبکه و بهبود تجربه کاربران گردد [۶, ۱۷].

الگوریتم برنامه‌ریزی آرمانی چند منظورهٔ فازی[3] یک روش محاسباتی است که برای حل مسائل بهینه‌سازی چند هدفه، از منطق فازی و

[1] Software Defined Networks

[2] Open Networking Foundation

[3] Fuzzy Multi-Objective Optimization Algorithm

مجموعه‌های فازی بهره می‌برد. این الگوریتم‌ها قادر به پیدا کردن تعادل مطلوب بین اهداف متفاوت در یک مسئله بهینه‌سازی چند هدفه هستند. الگوریتم برنامه‌ریزی آرمانی چند منظورهٔ فازی توانایی مدل‌سازی و حل مسائلی را دارد که در آن هدف‌های متفاوت و مغایر مدنظر قرار دارند. این مسائل می‌توانند شامل بهینه‌سازی همزمان چندین هدف مانند کاهش هزینه، افزایش کیفیت، کاهش زمان، حداکثر سود و موارد دیگر باشند [۱۸-۲۱].

در الگوریتم برنامه‌ریزی آرمانی چند منظورهٔ فازی، ابتدا مسئله بهینه‌سازی چند هدفه به شکل یک مسئله فازی مدل می‌شود. سپس با استفاده از روش‌های بهینه‌سازی مانند الگوریتم‌های تکاملی، الگوریتم‌های گرانشی، الگوریتم‌های جستجوی مستعمل و غیره، جواب بهینه‌ای برای مسئله بهینه‌سازی چند هدفه با در نظر گرفتن تعادل بین اهداف به دست می‌آید. در هر مرحله از اجرای الگوریتم برنامه‌ریزی آرمانی چند منظورهٔ فازی، جمعیتی از نقاط کاندیدا (نقاطی که جواب ممکن به مسئله را نمایندگی می‌کنند) تولید می‌شود و سپس با استفاده از معیارهای تقریبی فازی و عملکردهای هدف، نقاط بهتر و بهینه‌تر انتخاب می‌شوند. در نهایت، جواب بهینه‌ای که حاوی تعادل مطلوب بین اهداف است، به عنوان جواب نهایی ارائه می‌شود [۲۲, ۲۳].

سازماندهی مقاله به این صورت است که، در بخش ۲، کارهای انجام شده در گذشته مورد بررسی قرار گرفته‌اند. در بخش ۳، روش پیشنهادی و جزئیات مربوط به آن تشریح شده‌اند. در نهایت، در بخش ۴، نتایج به‌دست‌آمده تحلیل می‌شوند و در نهایت در بخش ۵ نتیجه‌گیری ارائه می‌گردد.

۲- مرور ادبیات و کارهای گذشته

در این بخش به ارائه تحقیقات انجام شده در زمینه بهبود توازن بار در شبکه‌های نرم‌افزار محور پرداخته می‌شود.

پژوهش [۲۴]، روشی برای حل چالش‌های شبکه‌های تعریف شده نرم‌افزاری SDN توزیع شده معرفی می‌کند. در این روش، با استفاده از آستانه متغیری بر اساس کنترل‌کننده‌ها، از ترافیک غیرمتعادل و شلوغی در شبکه استفاده می‌شود و سپس با استفاده از ترکیب هوشمندانه الگوریتم ژنتیک و الگوریتم OPSO، بهترین کنترل‌کننده با ظرفیت مناسب برای مهاجرت انتخاب می‌شود. نتایج پیاده‌سازی و ارزیابی روش GOP-SDN، بهبود ۲۴,۷۲ درصدی در ظرفیت شبکه و کاهش ۱۳,۹۶ درصدی در تعداد مهاجرت‌ها را نشان داد. تحقیق [۲۵]، راهکاری کارآمد برای توازن بار با استفاده از الگوریتم الهام‌گرفته از زنبور عسل در محیط‌های ابری معرفی می‌کند. در این روش، با استفاده از انتخاب تصادفی ماشین‌های مجازی، توازن بار صورت می‌پذیرد. این مقاله راهکاری برای توازن بار بر اساس ترافیک بیش از حد ارائه می‌دهد. با جابجایی بار به ماشین مجازیی که بیشترین کارآیی را دارد، می‌توان زمان پاسخ کلی روش پیشنهادی را کاهش داد. نتایج حاصل نشان می‌دهد که رویکرد پیشنهادی باعث کاهش تأثیر بار حدود ۱۰ تا ۱۵ درصد می‌شود. همچنین، روش پیشنهادی باعث کاهش مدت زمان اجرا و زمان پاسخ ماشین‌های مجازی می‌شود.

مطالعه‌ی [۲۶]، به منظور توازن بار ترافیک، استفاده از کنترل‌کننده مجازی شبکه نرم‌افزارمحور (vSDN) به عنوان یک تابع شبکه مجازی (VNF) را هدف قرار داده است. با استفاده از vSDN، هنگامی که بار ترافیک نامتوازن و افزایش یابد، کنترل‌کننده‌های vSDN فرعی برای به اشتراک گذاری این بار اضافی، اضافه می‌شوند. نیاز به کنترل‌کننده‌های vSDN فرعی تعیین می‌شود و یک کپی vSDN با تنظیمات دقیقاً یکسان با کنترل‌کننده اصلی vSDN ایجاد می‌شود که با دقت عمل کرده و وظایف توازن بار ترافیک را با کنترل‌کننده vSDN اصلی به اشتراک می‌گذارد. نتایج نشان داد که بهبود ۵۰ درصدی در بار میانگین، بهبود ۴۱ درصدی در تأخیر میانگین و بهبود قابل توجهی در زمینه پاسخ ping، استفاده پهنای باند و توانایی سیستم وجود دارد. پژوهش [۲۷] چارچوب‌های DALB و CAMD را برای حل چالش عدم توازن بار در شبکه‌های SDN معرفی می‌کند. چارچوب CAMD کارآمدتر از DALB است، اما در صورت وجود بار فراگیر کارآیی آن کاهش می‌یابد. مطالعه حاکی است که الگوریتم ISMDA با استفاده از مدل مهاجرت کارآمد، بهبود قابل توجهی در توازن بار دارد. این الگوریتم با افزایش توان و کاهش زمان پاسخ و فضای مهاجرت، عملکرد بهتری نسبت به CAMD ارائه می‌دهد. نتایج نشان می‌دهد که ISMDA در برابر بار فراگیر عملکرد بهتری دارد.

تحقیق [۲۸]، با استفاده از شبکه تعریف شده با نرم‌افزار، یک چارچوب مسیریابی آنلاین با صرفه‌جویی در انرژی و توازن بار پیشنهاد می‌شود. سپس، مسئله بهینه‌سازی مصرف برق و توازن بار مدل‌سازی شده و الگوریتمی برای حل آن ارائه می‌شود. همچنین الگوریتمی برای به‌روزرسانی مسیرها به علت برنامه‌ریزی جریان‌ها پیشنهاد شده است. نتایج آزمایش‌ها نشان می‌دهد که الگوریتم پیشنهادی در صرفه‌جویی برق، توازن بار و کاهش احتمال اختلال در لینک‌ها عملکرد بهتری نسبت به روش‌های مقایسه‌ای دارد. پژوهش [۲۹] راهکاری برای توازن بار و کنترل اختلال در شبکه‌های مرکز داده ارائه می‌دهد. با توجه به تغییرات پویا در الگوهای ترافیک،

رویکردهای موجود نیازمند تغییرات بزرگ در تجهیزات شبکه مرکز داده هستند. در روش پیشنهادی، با تشخیص دقیق جریان‌های بزرگ و ارائه روش‌های مسیریابی متفاوت برای انواع مختلف جریان‌ها، راهکاری بر اساس OpenFlow برای توازن بار در شبکه‌های مرکز داده ارائه شده است. نتایج نشان می‌دهد که استراتژی پیشنهادی قوانین جریان را به طور کافی تولید کرده و عملکرد استفاده از پهنای باند را به طرز قابل توجهی نسبت به راهکارهای دیگر در ادبیات افزایش می‌دهد.

تحقیق [۳۰] درباره نوآوری‌ها در مدیریت ترافیک شبکه در شهرهای هوشمند با استفاده از اینترنت اشیاء است. پیاده‌سازی چندین کنترل‌کننده در SDN به همراه اینترنت اشیاء، امنیت، حریم خصوصی و انعطاف‌پذیری سیستم کنترل متمرکز را در برنامه‌های شهر هوشمند بهبود می‌بخشد. نتایج آزمایشات نشان می‌دهد که استفاده از الگوریتم پیشنهادی موجب بهبود استفاده از واحد پردازش مرکزی با توازن بار شده و باعث کاهش ۲۰ درصدی در استفاده از CPU نسبت به عدم توازن بار می‌شود. مطالعه‌ی [۳۱] رویکردی برای بهینه‌سازی مسئله اختصاص سوئیچ به کنترل‌کننده با پشتیبانی از توازن بار ارائه شده است. مسئله به عنوان یک مسئله بهینه‌سازی اختصاص حداقل هزینه تخصیص دو جانبه[1] فرموله شده است. الگوریتم جدید بر اساس مفهوم جریمه مبتنی بر بار معرفی شده است که هدف آن تعادلی بین زمان سفر رفت و برگشت و بار کنترل‌کننده است. در نهایت، پروتکل جدیدی به نام پروتکل (DHAP) به عنوان پیاده‌سازی راه‌حل پیشنهادی در محیط‌های چند کنترل‌کننده استفاده شده است. نتایج نشان می‌دهد که راه‌حل پیشنهادی در مقایسه با روش‌های موازی، بهترین عملکرد را در زمینه زمان برپا شدن جریان و توازن بار ارائه می‌دهد.

پژوهش [۳۲] راهکار جابجایی سوئیچ‌ها را برای مدیریت ترافیک ارائه کرده‌اند. اما جابجایی سوئیچ‌ها می‌تواند تأثیری بر کیفیت خدمات کاربران و زمان پاسخ پیام‌های کنترلی داشته باشد. راهکار RTSM برای بهینه‌سازی زمان پاسخ در جابجایی سوئیچ‌ها ارائه شده است. معماری SDWAN جدید با کنترل‌کننده لایه ۲، وابستگی به ارتباطات WAN را کاهش می‌دهد. شرایط Karush-Kuhn-Tucker برای انتخاب کنترل‌کننده هدف استفاده شده است. روش انتخاب سوئیچی به کمک کاربران در جابجایی دستگاه‌ها کمک می‌کند. نتایج ارزیابی نشان می‌دهد که الگوریتم پیشنهادی عملکرد بهتری در موازنه بار و زمان پاسخ شبکه دارد.

تحقیق [۳۳]، الگوریتم جستجوی مبتنی بر چندین رگرسیون (MRBS) برای انتخاب بهینه سرویس دهنده و مسیریابی در شبکه مرکز داده پیشنهاد شده است. MRBS بر اساس تحلیل رگرسیون، سرویس دهنده را بر اساس پیش‌بینی نوع ترافیک و زمان پاسخ مبتنی بر پارامترهای داده سرویس دهنده مانند بار، زمان پاسخ و پهنای باند و استفاده از سرویس دهندهانتخاب می‌کند. الگوریتم پیشنهادی با ترکیب الگوریتم هیوریستیک و مدل رگرسیون، انتخاب سرویس دهنده و مسیر را بهبود می‌بخشد. الگوریتم پیشنهادی با تخمین وزن‌های کاهنده گرادیان تصادفی، تأخیر و زمان را بیش از ۴۵ درصد کاهش می‌دهد و نشان می‌دهد که در مقایسه با الگوریتم‌های سنتی، استفاده از سرویس دهنده بهتری به نسبت ۸۳ درصد به دلیل برآورد وزن کاهنده گرادیان تصادفی را دارد. تحقیق [۳۴] یک روش احتمالی برای توازن بار بر اساس تحلیل واریانس ارائه می‌دهد. این روش قادر است به طور پویا جریان‌های ترافیک را مدیریت کند تا از تقاضاهای بازیابی توسط کاربران بزرگ جابجایی پشتیبانی کند. در این روش، از فناوری سوئیچینگ مجازی OpenFlow به جای فناوری سوئیچینگ سخت‌افزاری سنتی استفاده می‌شود. کنترل‌کننده SDN با استفاده از تحلیل واریانس ترافیک داده را در هر درگاه نظارت کرده و با استفاده از الگوریتم انتخاب مبتنی بر احتمال، ترافیک را به طور پویا با تکنولوژی OpenFlow هدایت می‌کند. در مقایسه با روش‌های توازن بار موجود که برای پشتیبانی از شبکه‌های سنتی طراحی شده‌اند، این روش هزینه کمتر، اعتمادپذیری بالاتر و قابلیت مقیاس‌پذیری بیشتری دارد.

پژوهش [۳۵] یک طرح تعاملی را در کنترل‌کننده SDN معرفی کرده است که از الگوریتم فراابتکاری میگو برای برنامه‌نویسی مبتنی بر بهینه‌سازی استفاده می‌کند. روش پیشنهادی بر تعامل بین متعادل‌سازی بار و بهره‌وری انرژی، استفاده از داده‌های صفحه کنترل و به‌روزرسانی‌های ماشین مجازی برای استخراج راه‌حل‌های بهینه تأکید دارد. نتایج شبیه‌سازی بهبود حدود ۳۰ درصدی راندمان انرژی و افزایش عملکرد تقریبی ۲۰ درصدی را نشان می‌دهد.

پژوهش [۳۶] درباره طرحی برای استقرار چندین کنترلر پویا با استفاده از توازن بار است. در روش پیشنهادی، درخواست‌های جریان به یک صف مدل تبدیل می‌شوند و تأخیر در انتشار ترافیک و ظرفیت کنترلرها به عنوان دو عامل اصلی در استقرار چندین کنترلر در نظر

[1] Minimum Cost Bipartite Assignment

گرفته می‌شوند. الگوریتم اصلاح شده انتشار پیوسته[1] (PSOAP) بر اساس بهینه‌سازی ذرات، در شبکه استاتیک اولیه برای حل مشکلات خوشه‌بندی (مشکلات تنظیم مقادیر اولیه پارامترهای بایاس و ضرایب همگرایی) و برنامه‌ریزی مناسب شبکه پیشنهاد شده است. با استفاده از شبکه ترافیک پویا، راه‌گزینی‌ها در زیردامنه‌های مختلف با استفاده از الگوریتم جستجوی Breadth-First Search (BFS) مجدداً تنظیم می‌شوند تا به توازن بار کنترلرها برسند. ارزیابی نشان می‌دهد که این طرح می‌تواند در مقایسه با انتشار پیوسته (AP) و الگوریتم‌های مبتنی بر ژنتیک، استقرار چندین کنترلر با ثبات، دقت و توازن بهتری داشته باشد.

تحقیقات گسترده‌ای در زمینه توازن بار انجام شده است؛ با این حال، هنوز تا دستیابی به وضعیت مطلوب فاصله زیادی وجود دارد و نیاز به تحقیقات جامع‌تری می‌باشد [۳۵, ۳۷]. در بخش بعدی روش پیشنهادی برای توازن بار پویا مورد بررسی قرار خواهد گرفت.

۳- روش پیشنهادی

در این بخش، روش پیشنهادی برای بهبود توازن بار پویا و زمان پاسخ در SDN با استفاده از الگوریتم‌های برنامه‌ریزی آرمانی چند منظورهٔ فازی را بررسی می‌کنیم.

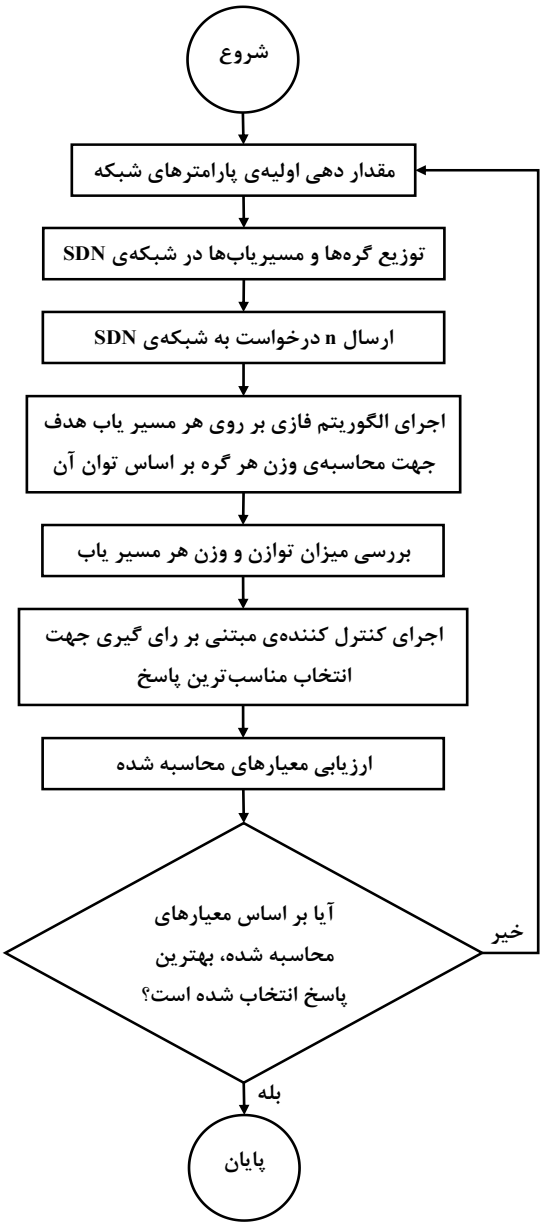
با توجه به فلوچارت روش پیشنهادی در شکل ۱، در اولین مرحله، پارامترهای شبکه SDN تنظیم می‌شوند. این پارامترها ابعاد شبکه، پهنای باند مسیریاب‌ها و توان پردازشی آن‌ها می‌باشند. سپس تعداد گره‌ها و مسیریاب‌ها (G) تعیین می‌شود و این گره‌ها در کل شبکه‌ی SDN توزیع می‌شوند. سپس، تعداد درخواست‌ها (n) تعیین می‌شود و این درخواست‌ها به مسیریاب‌ها و گره‌ها ارسال می‌شوند. الگوریتم فازی بر روی هر مسیریاب اجرا می‌شود و وزن گره‌ها بر اساس توان آن محاسبه می‌شود. در نهایت، با توجه به وزن‌های محاسبه شده، بهترین پاسخ بر اساس ارزیابی معیارها انتخاب می‌شود و مسیریاب مورد نظر، عملیات درخواست یا درخواست‌های دریافت شده را انجام می‌دهد. در این حالت، مسیریابی که وزن بیشتری دارد، به عنوان مسیریاب مطلوب انتخاب می‌شود. بنابراین با انتخاب بهترین و کوتاه‌ترین مسیر برای انجام هر عمل (Task) زمان پاسخ کاهش پیدا می‌کند و با کاهش زمان پاسخ می‌توان ادعا کرد که توازن بار نیز بهبود خواهد یافت. در ادامه، مراحل مختلف روش پیشنهادی توضیح داده شده و با جزئیات بیشتری مورد بررسی قرار می‌گیرند.

۳-۱- مراحل کلیدی روش پیشنهادی

روش پیشنهادی شامل چهار مرحله‌ی کلیدی زیر است:

۱. اجرای الگوریتم فازی جهت تعیین وزن مطلوب
۲. فرموله کردن مدل فازی پیشنهادی
۳. مکانیسم توازن بار مسیر (Path Load)
۴. اجرای تکنیک رأی گیری آرمانی

در ادامه هر مرحله بصورت کامل توضیح داده می‌شود و جزئیات آن مورد بررسی قرار می‌گیرد.



شکل ۱. فلوچارت روش پیشنهادی

[1] Particle Swam Optimization base on Affinity Propagation

۳-۱-۱- اجرای الگوریتم فازی جهت تعیین وزن مطلوب

پس از اینکه گره‌ها در شبکه توزیع شدند و درخواست‌ها نیز به شبکه جهت اجرا ارسال شدند، می‌بایست فرآیند پردازش این درخواست‌ها صورت پذیرد. بدین منظور، ابتدا با کمک منطق فازی اقدام به محاسبه وزن هر مسیریاب به منظور اجرای درخواست‌ها می‌گردد. زمانی که کنترلر می‌خواهد یک گره مطلوب را انتخاب کند تا سایر گره‌ها بتوانند اطلاعات مربوطه را ارسال نمایند، سیستم فازی نیز اعمال می‌گردد. جهت انتخاب گره مطلوب از یک سری قوانین فازی به صورت زیر استفاده می‌شود. دو پارامتر اصلی که برای انتخاب گره‌های مطلوب، استفاده می‌شود عبارتند از میزان گذردهی و پهنای باند یا سربار داده.

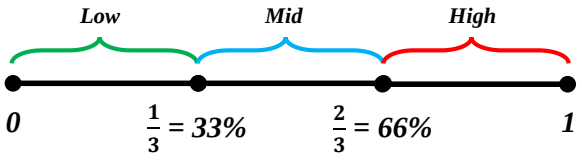
برای انتخاب درست، باید قوانین فازی را تشکیل داده و بر اساس آن‌ها عمل کنیم. جدول ۱، قوانین انتخاب گره مطلوب بر اساس میزان نرخ گذردهی و همچنین بر اساس پهنای باند / سربار داده در سه حالت $\{Low, Mid, High\}$ را مشخص می‌کند. همچنین در جدول ۱، مقدار Bad انتخاب ضعیف گره از نظر پهنای باند / سربار داده و مقدار Worst انتخاب نامطلوب و غیرقابل قبول گره از بدترین میزان از پهنای باند / سربار داده را از نظر فازی نشان می‌دهد.

با توجه به جدول ۱ و قوانین فازی ارائه شده، توابع عضویت مربوط به میزان گذردهی و میزان پهنای باند/ سربار داده و فاکتور قطبیت بر اساس میزان گذردهی، به صورت شکل ۱ تعریف می‌گردد.

انتخاب یک گره مطلوب بر اساس قوانین بالا قابل انجام است. به عنوان مثال، فرض کنید گره ۱ دارای پهنای باند / سربار داده‌ای برابر با ۸۰٪ باشد و گره ۲ دارای پهنای باند / سربار داده‌ای برابر با ۶۰٪ باشد. همچنین، میزان گذردهی از گره ۱ به گره بعدی حدوداً ۵۰ کیلوبایت است و میزان گذردهی از گره ۲ به گره بعدی حدوداً ۷۰ کیلوبایت است. بنابراین، احتمال انتخاب هر گره به شرح زیر است: به دلیل داشتن پهنای باند / سربار داده بالا (۸۰٪) و میزان گذردهی تا گره بعدی معادل ۵۰ کیلوبایت، احتمال انتخاب گره ۱ در حالت خوب (Good) است. همچنین، به دلیل داشتن پهنای باند / سربار داده متوسط (۶۰٪) و میزان گذردهی تا گره بعدی معادل ۷۰ کیلوبایت، احتمال انتخاب گره ۲ در حالت ضعیف است. در نهایت، گره شماره ۱ انتخاب می‌شود.

جدول ۱. قوانین و شرایط موجود برای انتخاب گره مطلوب

میزان گذردهی (Gateway) (Throughput)	پهنای باند / سربار داده (Overload/Bandwidth)		
	Low	Mid	High
Low	Mid	Worst	Worst
Mid	Normal	Mid	Bad
High	Good	Normal	Mid



شکل ۲. فاکتور قطبیت بر اساس پهنای باند یا سربار داده و همچنین فاکتور قطبیت بر اساس میزان گذردهی

انتخاب گره مطلوب بر اساس شروط زیر انجام می‌شود:

R1=if (Overloading/BW = Low & Gateway = Low) then select_ R node = Mid

R2=if (Overloading/BW = Mid & Gateway = Low) then select_ R node = Worst

R3=if (Overloading/BW = High & Gateway = Low) then select_ R node = Worst

R4=if (Overloading/BW = Low & Gateway = Mid) then select_ R node = Normal

R5=if (Overloading/BW = Mid & Gateway = Mid) then select_ R node = Mid

R6=if (Overloading/BW = High & Gateway = Mid) then select_ R node = Bad

R7=if (Overloading/BW = Low & Gateway = High) then select_ R node = Good

R8=if (Overloading/BW = Mid & Gateway = High) then select_ R node = Normal

R9=if (Overloading/BW = High & Gateway = High) then select_ R node = Mid

این نکته حائز اهمیت است که در صورتی که دو گره با شرایط مساوی وجود داشته باشند، برای انتخاب یکی از آنها باید از تابع برازندگی (Fitness) که از رابطهٔ ۱ محاسبه می‌شود استفاده نمود.

$$Fitness = \frac{Overloading}{BW} + \frac{1}{Gateway} \qquad (۱)$$

$$Selectnode = Max\left(Fitness1,\ Fitness2\right)$$

در اینجا Overloading بار پردازشی، BW پهنای باند و Gateway میزان گذردهی یا نرخ ترافیک آزاد در درگاه شبکه است. با استفاده از روش پیشنهادی، می‌توان بهبود قابل توجهی در عملکرد SDN

داشت. در جدول ۲ نمادهای پرکاربرد مورد استفاده در این مقاله و توضیحات آنها آورده شده است.

۳-۱-۲- فرموله کردن مدل فازی پیشنهادی

در یک محیط شبکه پویا، انتخاب بهترین مسیر برای جریان داده‌ها چالشی پیچیده است، زیرا تأثیرات فازی متعددی که بر این انتخاب تأثیر می‌گذارند، وجود دارد. هدف‌ها و محدودیت‌ها در این مسئله به طور طبیعی فازی هستند. توازن بار در مسیریابی، یک مسئله تصمیم‌گیری در محیط شبکه با خصوصیات فازی است. برای مقابله با ابهامات، می‌توان از تئوری فازی بهره‌برداری کرده و اهداف و محدودیت‌های فازی را به عنوان مجموعه‌های فازی دقیق تعریف کرده و با استفاده از ترکیب اهداف و محدودیت‌های داده‌شده، تصمیم‌گیری فازی انجام داد. بنابراین، در این مقاله، استفاده از مکانیسم ارزیابی ترکیبی فازی[1] (FSEM) برای کنترل توازن بار در مسیریابی شبکه‌های نرم‌افزارمحور پیشنهاد شده است.

جدول ۲. نمادهای پرکاربرد

نماد	توضیحات
Overloading	بار پردازشی در شبکه
BW	پهنای باند[2]
Gateway	میزان گذردهی در درگاه شبکه
Top-K	مسیرهای برتر برای انتخاب بهترین مسیر
h	تعداد واسطه ها یا پرش ها[3] در مسیر
p	تعداد بسته‌های انتقال داده شده
b	تعداد بایت‌های سوئیچ بحرانی
r	نرخ ارسال و دریافت سوئیچ (مسیریاب) مورد نظر
U	دامنه فاکتور
V	دامنه رتبه
A	بردار وزن
R	ماتریس رابطه فازی
N	مجموعه فازی منفی
t	نرخ ارسال و دریافت مسیریاب یا سوئیچ
M	جابجایی یا انتقال داده‌ها
ZE	مجموعه فازی صفر
P	مجموعه فازی مثبت
r_{ij}	درجه وابسته به فاکتور u_i در دامنه‌ی U و فاکتور v_j در دامنهٔ V
b_k	ارزیابی نهایی بهترین مسیر k

با توجه به تعداد زیادی مسیر ممکن بین یک گره مبدأ و یک گره مقصد، برای اطمینان از کارآیی، طرح پیشنهادی به دو مرحله تقسیم شده است.

مرحله اول: الگوریتم انتخاب مسیرهای Top-K

مرحله دوم: ارزیابی ترکیبی فازی[4]

بنابراین ابتدا مسیرهای Top-K انتخاب می‌شوند و سپس هنگام ورود جریان جدید یا تغییر وضعیت شبکه، با استفاده از روش ارزیابی ترکیبی فازی، بهترین مسیر از بین مسیرهای Top-K به صورت پویا انتخاب می‌شود.

مرحله اول: الگوریتم انتخاب مسیرهای Top-K

معمول است که بیش از یک مسیر بین دو گره در یک شبکه وجود داشته باشد، به ویژه برای گره‌های مهم یا با تقاضای اطمینان بالا. در صورت وجود تعداد زیادی مسیر، انتخاب بهترین مسیر در زمان واقعی به صورت بهینه غیرکارآمد خواهد بود. بنابراین، در ابتدا مسیرهای Top-K که به عنوان کوتاه‌ترین مسیرها در نظر گرفته می‌شوند، انتخاب می‌شوند. این انتخاب می‌تواند به دو صورت انجام شود: یکی به صورت اولیه و دیگری به صورت دوره‌ای.

در حالت اولیه، هیچ ترافیکی در شبکه وجود ندارد، بنابراین مسیرهای Top-K کوتاه‌ترین مسیرها خواهند بود. از الگوریتم کوتاه‌ترین مسیر K برای پیدا کردن K مسیر کوتاه‌تر استفاده می‌شود. این الگوریتم، یک توسعه از الگوریتم کوتاه‌ترین مسیر است که K مسیر کوتاه‌تر را مشخص می‌کند. در این مقاله، از الگوریتم کوتاه‌ترین مسیر K بر اساس الگوریتم Floyd که یک الگوریتم مبتنی بر برنامه نویسی پویا برای محاسبه کوتاه‌ترین مسیر بین تمامی گره ها در توپولوژی شبکه می باشد ، استفاده شده است. اصل اساسی این الگوریتم به این صورت است که یک نقشه ماتریسی را برای جستجوی کوتاه‌ترین مسیر ثبت می‌کند. برای هر جفت از گره‌ها i و j ، الگوریتم به‌تدریج گره‌های جدیدی را به عنوان گره میانی به مسیر اصلی اضافه می‌کند. اگر طول مسیر جدید کوتاه‌تر از مسیر اصلی باشد، مسیر اصلی و نقشه ماتریسی با مسیر جدید به‌روزرسانی می‌شوند. این فرایند با استفاده از رابطه‌ی ۲ انجام می‌شود.

$$map[i, j] = min\big(\big(map[i, k] + map[k, j]\big), map[i, j]\big) \quad (۲)$$

در اینجا i و j به‌ترتیب مقادیر سطر و ستون نقشه‌ی ماتریسی هستند. اگرچه مسیرهای Top-K در ابتدا بر اساس تعداد واسطه‌ها

[1] Fuzzy Synthetic Evaluation Mechanism

[2] Band Width

[3] Hop

[4] Fuzzy Synthetic Evaluation

(هاپ) انتخاب می‌شوند، اما مسیر ترافیکی در شبکه برای مسیرهای Top-K با استفاده از روش ارزیابی ترکیبی فازی انتخاب می‌شود. فاصله زمانی می‌تواند به مدت پنج دقیقه باشد. به این ترتیب، مسیرهای Top-K قادر به سازگاری با تغییرات ترافیک شبکه می‌باشند.

مرحله دوم: ارزیابی ترکیبی فازی

برای ارزیابی مسیرها در شبکه‌های نرم‌افزارمحور، عوامل متعددی تأثیر می‌گذارند. ارزیابی مسیر می‌تواند به عنوان یک مسئله تصمیم‌گیری چندمعیاری در نظر گرفته شود. بنابراین، یک مدل ارزیابی ترکیبی فازی با ویژگی‌های چندگانه به عنوان یک مدل تصمیم‌گیری فازی برای توازن بار مسیر پیشنهاد شده است. از دیدگاه عوامل چندگانه، درجات رضایت از ویژگی‌ها به وسیله مجموعه‌های فازی مرتبط به طور کامل از طریق تابع عضویت ارزیابی می‌شود. علاوه بر این، این امکان را فراهم می‌کند تا درجهٔ خاصی از اهمیت به هر معیار اختصاص داده شود. یک تصمیم بهینه به عنوان نقطه‌ای در فضای گزینه‌های ممکن تعریف شده است که در آن عملکرد یک تصمیم فازی به حداکثر مقدار خود می‌رسد. فاکتور مقدار دامنه U برای ارزیابی مسیر از رابطهٔ ۳ محاسبه می‌شود.

$$U = (u_1, u_2, u_3, ...u_n) \quad (۳)$$

برای ارزیابی مسیرها، دو جنبه در نظر گرفته می‌شود. یکی طول مسیر است که از شمارش تعداد واسطه ها بدست می‌آید و دیگری، بار ترافیکی در طول مسیر است که از ترافیک سوئیچ‌ها و پیوندهای مسیر استخراج می‌شود. از آنجا که یک مسیر از تعداد زیادی سوئیچ و پیوند تشکیل شده است، میانگین یا کل ترافیک نمی‌تواند وضعیت واقعی مسیر را نشان دهد، بنابراین سوئیچ‌ها و پیوندها برای نشان دادن وضعیت مسیر انتخاب می‌شوند. ترافیک سوئیچ با استفاده از تعداد بسته‌ها و تعداد بایت‌هایی که توسط سوئیچ ارسال می‌شود، اندازه‌گیری می‌گردد. بنابراین میزان ترافیک پیوند و میزان ارسال درگاه مربوطه اندازه‌گیری می‌شود. دامنه فاکتور u برای ارزیابی مسیر از رابطه‌ی ۴ محاسبه می‌شود.

$$u = (h,\ p,\ b,\ t) \quad (۴)$$

در اینجا، h تعداد واسطه ها (هاپ) ، p تعداد بسته‌های انتقال داده شده و b تعداد بایت‌های سوئیچ بحرانی و t نرخ ارسال و دریافت مسیریاب یا سوئیچ مورد نظر است. برای تعیین دامنه رتبه‌ی V برای هر عامل از رابطه‌ی ۵ استفاده می‌شود.

$$V = (v_1, v_2, v_3, ..., v_n) \quad (۵)$$

در روش پیشنهادی، دامنه رتبه به وسیلهٔ یک یا چند مجموعه فازی تشکیل می‌شود که نشان‌دهندهٔ رتبه‌بندی گزینه‌های مختلف است. این مجموعه‌ها شامل منفی (N)، صفر (ZE) و مثبت (P) می‌باشند. با توجه به اینکه هدف اصلی، انتخاب بهترین مسیر است، می‌توان $V = (P)$ تعریف کرد، که در اینجا P نشان‌دهندهٔ مجموعه فازی مثبت است. در روش پیشنهادی، فاکتورها به وسیلهٔ ماتریس رابطه فازی R ارزیابی شده و از رابطه‌ی ۶ محاسبه می‌شود.

$$R = \begin{bmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nm} \end{bmatrix} \quad (۶)$$

در این ماتریس، r_{ij} نشان‌دهندهٔ درجه‌ای است که به فاکتور u_i در دامنه‌ی U و فاکتور v_j در دامنه‌ی V وابسته است. در روش پیشنهادی، عضویت در مجموعه P برای هر ویژگی U محاسبه می‌شود. عملکرد عضویت h ، p ، b و t از رابطهٔ ۷ بدست می‌آید [۳۸].

$$\begin{aligned} r_h &= 1.0/e^h \\ r_p &= 1.0/\log(p + 0.1) \\ r_b &= 1.0/\log(b + 0.1) \\ r_t &= 1.0/\left(1 + e^{\frac{-t}{50.0}}\right) \end{aligned} \quad (۷)$$

در اینجا r_t، نرخ ارسال و دریافت مسیریاب یا سوئیچ است. برای محاسبه r_t، از ماتریس رابطه فازی R استفاده می‌شود که در آن هر درایه ماتریس مقداری را برای درجه ارتباط بین فاکتورهای مختلف نشان می‌دهد. در این روش، ابتدا عضویت در مجموعه فازی مثبت (P) برای هر ویژگی U محاسبه می‌شود، سپس با استفاده از ماتریس رابطه فازی R، مقادیر درجه ارتباط بین ویژگی‌ها و وابستگی آن‌ها به هم، به عنوان مقادیر r_t محاسبه می‌شوند.

سپس، ماتریس R می‌تواند به عنوان بردار ستونی برای یک مسیر مشخص از رابطه‌ی ۸ محاسبه شود.

$$R = \begin{pmatrix} r_h \\ r_p \\ r_b \\ r_t \end{pmatrix} \quad (۸)$$

برای تعیین بردار وزن $A = (a_1, a_2, a_3, ..., a_n)$، باید درجه اهمیت هر عامل در دامنه U مشخص شود. در اینجا، برای متعادل‌سازی بار مسیر، A به صورت (0.4, 0.15, 0.15, 0.3) تعریف

شده است، به این معنی که طول مسیر اهمیت بیشتری دارد و حجم کار سوئیچ و پیوند بحرانی مشترک هستند. نتایج ارزیابی نهایی، با ترکیب بردار A و ماتریس R از رابطه‌ی ۹ قابل محاسبه است.

$$B=(b_1,b_2,b_3,...,b_i)$$
$$b=A\otimes R$$
$$b=(a_1,a_2,a_3,...,a_n)\otimes\begin{bmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nm} \end{bmatrix} \quad (۹)$$

در اینجا، نماد $\otimes$ ضرب تک تک درایه‌های ماتریس خطی A در تک تک ستون‌های ماتریس R و B مقدار ارزیابی است. در نتیجه ارزیابی مسیرهای K، از رابطه‌ی ۱۰ محاسبه می‌شود.

$$b_k=(0.4,0.15,0.15,0.3)\otimes\begin{bmatrix} r_{h1} & r_{h2} & \cdots & r_{hk} \\ r_{p1} & r_{p2} & \cdots & r_{pk} \\ r_{b1} & r_{b2} & \cdots & r_{bk} \\ r_{t1} & r_{t2} & \cdots & r_{tk} \end{bmatrix} \quad (۱۰)$$

در اینجا b_k ارزیابی نهایی مسیر k است که بصورت زیر محاسبه می‌شود:

$$b_k=(0.4\times r_{hk})+(0.15\times r_{pk})+(0.15\times r_{bk})+(0.3\times r_{tk})$$

۳-۱-۳- مکانیسم توازن بار مسیر[1]

راه‌حل پیشنهادی در این مقاله برای توازن بار مسیر در شبکه‌های نرم‌افزارمحور از معماری لایه‌بندی شبکه‌های نرم‌افزارمحور استفاده می‌کند. این معماری شامل لایه‌ی کنترل و لایه‌ی داده است. مسیری که ترافیک شبکه را از گره منبع به گره مقصد منتقل می‌کند، باید بهینه باشد تا بار را به طور متعادل بین مسیرها تقسیم کند و همچنین از منابع شبکه به طور کامل استفاده کند. برای این منظور، کنترلر با استفاده از الگوی ارزیابی مسیر، مسیرهایی را انتخاب می‌کند که برای بارگذاری مسیر مفید هستند. این عملیات توسط سه ماژول انجام می‌شود:

۱. ماژول جمع‌آوری داده‌ها
۲. ماژول ارزیابی مسیر
۳. ماژول نصب جدول جریان

در ادامه هر یک از ماژول‌ها توضیح داده می‌شوند.

۱. ماژول جمع‌آوری داده‌ها

این ماژول شامل بررسی توپولوژی و جمع‌آوری داده‌های ترافیک است. بررسی توپولوژی به کنترلر کمک می‌کند تا با داشتن اطلاعات توپولوژی شبکه، آگاهی کاملی از موقعیت میزبان‌ها در سراسر شبکه داشته باشد. این اطلاعات با استفاده از پیوندهایی که با ارسال بسته‌های داده LLDP[2] تشکیل شده‌اند، جمع‌آوری می‌شود. پروتکل LLDP یک پروتکل استاندارد در لایه ۲ مدل OSI با هدف شناسایی شبکه است که به کلیه تجهیزات شبکه (مستقل از برند) اجازه می‌دهد اطلاعاتی درباره خود را به دستگاه‌های همسایه ارسال و مبادله کنند. این پروتکل به‌ویژه در شبکه‌های بزرگ که شامل تجهیزات مختلف از برندهای گوناگون هستند، کاربرد دارد و به مدیریت و نگهداری شبکه کمک می‌کند. هنگامی که پیوندها بین گره‌ها ایجاد یا خراب می‌شوند، رویدادهای پیوند ایجاد می‌شوند. جمع‌آوری داده‌های ترافیک اطلاعات آماری مربوط به تعداد بسته‌ها، تعداد بایت‌های ارسال شده توسط سوئیچ و میزان ارسال درگاه مربوطه را از سوئیچ‌های OpenFlow جمع‌آوری می‌کند.

۲. ماژول ارزیابی مسیر

این ماژول از الگوریتم مسیرهای Top-K و الگوریتم ارزیابی ترکیبی فازی استفاده می‌کند. با توجه به اینکه ممکن است تعداد زیادی مسیر اضافی در شبکه وجود داشته باشد، کنترلر ابتدا باید مسیرهای بهینه Top-K را انتخاب کند. این الگوریتم بر اساس معیارهایی مانند پهنای باند، تأخیر و هزینه‌های مسیریابی، مسیرهای بهینه را انتخاب می‌کند. سپس بر اساس این مسیرهای بهینه، مسیرهای بیشتری را برای ارزیابی انتخاب می‌کند. الگوریتم ارزیابی ترکیبی فازی بر اساس منطق فازی و قوانین قابلیت دسترسی به منابع شبکه، بارگذاری مسیر و توازن بار مسیر، مسیرهای بهینه را ارزیابی می‌کند. با استفاده از این الگوریتم، کنترلر می‌تواند تصمیم بگیرد که کدام مسیرها برای بارگذاری مسیر استفاده شوند.

۳. ماژول نصب جدول جریان

این ماژول بر اساس مسیرهایی که توسط ماژول ارزیابی مسیر انتخاب شده‌اند، جداول جریان را در سوئیچ‌ها نصب می‌کند. این جداول جریان مشخص می‌کنند که بسته‌هایی که از یک مسیر خاص

[1] Path load balancing mechanism

[2] Link Layer Discovery Protocol

عبور می‌کنند، به کدام درگاه‌ها هدایت شوند. روش مسیریابی پیشنهادی Top-K در الگوریتم ۱ نشان داده شده است.

الگوریتم ۱: یافتن کوتاه‌ترین مسیر از مبدا به مقصد با روش پیشنهادی

Input: Graph G represented by a set of nodes and edges, starting node S, destination node T
Output: Shortest path from node S to node T

Initialize $A[0]$ to represent the shortest path from node S to node T
For each k in the range {1, 2, …, K-1}:
  For each i in the range {0, 1, 2, …, length($A[k\text{-}1]$) - 2}:
    Set *currentNode* as the i-th node in $A[k\text{-}1]$
    For each j in the range {0, 1, 2, …, K-1}:
      Set *previousNode* as the j-th node in $A[k\text{-}1]$
      If *currentNode* is equal to *previousNode*:
        Set the distance between *currentNode* and the (i+1)-th node as infinity
      End if
    EndFor inner loop
    Calculate $S[i]$ representing the shortest path from *currentNode* to node T
    Calculate $R[i]$ as the shortest path from node S to *currentNode*
    Calculate $B[i]$ as the sum of $R[i]$ and $S[i]$
  EndFor outer loop
  Update $A[k]$ to represent the shortest path with the minimum length among all $B[i]$
  Restore the distance between *currentNode* and *previousNode* to its original value
EndFor outer loop

در این الگوریتم، برای یافتن کوتاه‌ترین مسیر از یک گره به گره دیگر در یک گراف، مراحل زیر انجام می‌شود: ابتدا، کوتاه‌ترین مسیر از گره شروع (S) به گره مقصد (T) در یک لیست با نام $A[0]$ ذخیره می‌شود. در مراحل بعدی، برای هر مرحله k (که نشان‌دهنده تعداد مراحل محاسبه کوتاه‌ترین مسیر است)، به ازای هر گره i در مسیر قبلی ($A[k\text{-}1]$)، مسیر جدیدی که از گره فعلی تا گره مقصد (T) را محاسبه می‌کنیم (که با $S[i]$ نشان داده می‌شود). همچنین، کوتاه‌ترین مسیر از گره S به گره فعلی را نیز محاسبه می‌کنیم و در لیست $R[i]$ ذخیره می‌کنیم. سپس مجموع این دو مسیر، به عنوان کوتاه‌ترین مسیر از گره S به گره مقصد (T) را در لیست $B[i]$ ذخیره می‌کنیم. پس از پایان مراحل برای هر k، مسیری که کوتاه‌ترین مسیر از گره S به گره مقصد (T) است را در لیست $A[k]$ ذخیره می‌کنیم. در پایان هر مرحله، مقادیر فاصله‌های میان گره های متصل را به حالت اولیه برمی‌گردانیم و مراحل برای مرحله k بعدی را انجام می‌دهیم. این مراحل با استفاده از ارتباطات میان مسافت مسیرها و محاسبات فاصله‌ی بین مسیرهای مختلف، در نهایت منجر به یافتن کوتاه‌ترین مسیر از یک گره به گره دیگر در گراف می‌شود.

الگوریتم ۲ برای ارزیابی ترکیبی فازی برای ارزیابی مسیرهای بهینه Top-K در شبکه استفاده می‌شود. شاخص‌های بارگذاری شامل هاپ، تعداد بسته‌ها، تعداد بایت و نرخ ارسال درگاه سوئیچ هستند. این شاخص‌ها از اطلاعات سراسری شبکه، شامل اطلاعات گره‌ها و پیوندها، جمع‌آوری می‌شوند. بردار وزن A برابر با (0.4, 0.15, 0.15, 0.3) است که عناصر آن به ترتیب با شاخص‌های هاپ، تعداد بسته‌ها، تعداد بایت و نرخ ارسال بسته مطابقت دارند. سپس چهار عامل عملکرد با عضویت تعریف شده است. در نهایت، بردار وزن A با ماتریس R ترکیب می‌شود تا به نتایج ارزیابی برسیم. نمره ارزیابی نهایی مسیر بهینه را نشان می‌دهد. مسیریابی روش پیشنهادی در الگوریتم ۲ نشان داده شده است.

الگوریتم ۲: مسیریابی روش پیشنهادی Top-K

Input: Top-K paths K_PATH determined
Output: The optimal path BEST_PATH

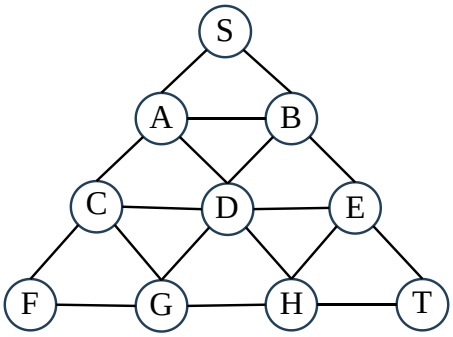
Define the factor domain U (hops, bytes, packets, port_rate) and rank domain $V(P)$ for a path
For each i in the range {0, 1, 2, …, K-1}:
  Calculate *hops*[i] as the length of K_PATH[i]
  Calculate *bytes*[i] as the total bytes of K_PATH[i]
  Calculate *packets*[i] as the total packets of K_PATH[i]
  Calculate *port_rate*[i] as the transfer rate of K_PATH[i]
EndFor
For each j in {0, 1, 2, …, K-1}:
  Transform *hops*[j] by applying the function 1.0 / exp(*hops*[j])
  Transform *bytes*[j] by applying the function 1.0 / log(*bytes*[j] + 0.1)
  Transform *packets*[j] by applying the function 1.0 / log(*packets*[j] + 0.1)
  Transform *port_rate*[j] by applying the function 1.0 / (1 + exp(-*port_rate*[j]/50.0))
EndFor
Define weight vector A (*weight_hops*, *weight_bytes*, *weight_packets*, *weight_port_rate*)
For each k in {0, 1, 2, …, K-1}:
  Calculate *temp_hops*[k] as the product of weight_hops and *hops*[k]
  Calculate *temp_bytes*[k] as the product of *weight_bytes* and *bytes*[k]
  Calculate *temp_packets*[k] as the product of *weight_packets* and *packets*[k]
  Calculate *temp_port_rate*[k] as the product of *weight_port_rate* and *port_rate*[k]
  Calculate *score*[k] as the sum of *temp_hops*[k], *temp_bytes*[k], *temp_packets*[k], and *temp_port_rate*[k]
EndFor
BEST_PATH is equal to the path with the maximum score among all scores

این الگوریتم برای مسیریابی با استفاده از روش Top-K طراحی شده است. در این روش، مجموعه‌ای از K بهترین مسیر از بین مسیرهای موجود انتخاب می‌شود. مراحل این الگوریتم به شرح زیر است: ابتدا، دامنه عوامل (مانند تعداد واسطه ها، حجم داده‌ها، تعداد بسته‌ها و نرخ درگاه) و دامنه رتبه‌بندی برای مسیرها تعیین می‌شود. سپس برای هر مسیر از 0 تا K-1، ویژگی‌های مختلف محاسبه شده و مقادیر

آنها تبدیل می‌شوند. سپس بردار وزن برای ارزیابی مسیرها با استفاده از وزن‌های مختلف مانند تعداد پرش‌ها، حجم داده‌ها، تعداد بسته‌ها و نرخ درگاه تعیین می‌شود. سپس برای هر مسیر، با استفاده از وزن‌های تعیین شده و مقادیر ویژگی‌های محاسبه شده، یک امتیاز محاسبه می‌شود. در نهایت، بهترین مسیر برابر با مسیری که بیشترین امتیاز را دارد، انتخاب می‌شود. این الگوریتم با استفاده از یک ترکیب خطی از ویژگی‌های مختلف مسیرها و وزن‌های مشخص شده، بهترین مسیر را از بین مسیرهای Top-K انتخاب می‌کند.

در الگوریتم ۲، K_PATH[i] نشان‌دهنده مسیرهای برتر (Top-K paths) می‌باشد. این مسیرها در واقع کوتاه‌ترین مسیرها از یک گره به گره دیگر در گراف است که با در نظر گرفتن ویژگی‌های مختلف مانند تعداد پرش‌ها، حجم داده‌ها، تعداد بسته‌ها و نرخ درگاه محاسبه می‌شوند. مقدار K_PATH[i]، می‌تواند تا K مسیر مختلف باشد که در لیست K_PATH ذخیره شده‌اند. به عبارت دیگر، K_PATH[i] می‌تواند مجموعه‌ای از مسیرهای مختلف باشد که به عنوان مسیرهای برتر از نظر کوتاه‌ترین مسیرها در گراف در نظر گرفته شده‌اند و K، تعداد مسیرهای برتر است.

تفاوت الگوریتم ۱ و الگوریتم ۲ در حل مسئلهٔ یافتن مسیر بهینه در یک گراف در این است که، در الگوریتم ۱، به صورت تکاملی و دقیق، کوتاه‌ترین مسیر از گره شروع به گره مقصد محاسبه می‌شود. در هر گام از محاسبه، از نتایج قبلی استفاده می‌شود و مسیرها به صورت دقیق و بر اساس داده‌های قبلی محاسبه می‌شوند. اما در الگوریتم ۲، ابتدا ویژگی‌های مسیرهای مختلف، مانند تعداد پرش‌ها، حجم داده‌ها، تعداد بسته‌ها و نرخ درگاه، محاسبه می‌شود و سپس با توجه به وزن‌های مشخص شده، مسیر بهینه انتخاب می‌شود. این روش انتخاب مسیر بر اساس امتیازدهی به ویژگی‌های مسیرها انجام می‌شود، به‌طوری‌که مسیری با بیشترین امتیاز به عنوان مسیر بهینه انتخاب می‌شود. به عنوان مثال گراف شکل ۳ را در نظر بگیرید. فرض کنید می‌خواهیم مسیر بهینه از گره S به گره T را پیدا کنیم. این گراف شامل مجموعه از گره‌های S، A، B، C، D، E، F، G، H و T می‌باشد.



شکل ۳. مثالی از عملکرد الگوریتم‌های مسیریابی ۱ و ۲ برای پیدا کردن یک مسیر از S به T

حالا با استفاده از الگوریتم ۱، مراحل اجرای الگوریتم به صورت زیر است:

- در مرحله‌ی اول، مسیر از S به T به صورت (S, A, B, E, T) محاسبه می‌شود.
- در مرحله دوم، مسیری جدید از A به T به صورت (A, B, E, T) به عنوان کوتاه‌ترین مسیر محاسبه می‌شود.
- در مرحله سوم، مسیری جدید از B به T به صورت (B, E, T) به عنوان کوتاه‌ترین مسیر محاسبه می‌گردد.
- در نهایت، کوتاه‌ترین مسیر از S به T برابر با (S, A, B, E, T) به دست می‌آید.

اما با استفاده از الگوریتم ۲، مراحل اجرای الگوریتم به صورت زیر است:

ابتدا ویژگی‌های هر مسیر محاسبه می‌شود. برای مسیرهای ممکن، ویژگی‌هایی مانند تعداد پرش‌ها، حجم داده‌ها، تعداد بسته‌ها و نرخ درگاه محاسبه می‌شود. سپس ویژگی‌ها با توجه به وزن‌های مشخص شده تبدیل می‌شوند. از بین مسیرهای محاسبه شده، مسیری که بیشترین امتیاز را دارد به عنوان مسیر بهینه انتخاب می‌شود. بنابراین ممکن است با توجه به ویژگی‌ها و وزن‌های مشخص شده، مسیر (S, A, C, D, E, T) به عنوان مسیر بهینه انتخاب شود.

در مثال بالا، الگوریتم ۱ به صورت دقیق و تکاملی مسیرهای کوتاه را محاسبه می‌کند، در حالی که الگوریتم ۲ بر اساس امتیازدهی به ویژگی‌های مسیرها عمل می‌کند و ممکن است مسیری که به طور دقیق کوتاه‌ترین نباشد را به عنوان مسیر بهینه انتخاب کند.

تکنیک رای گیری آرمانی، آخرین تکنیک بهبود توازن بار پویا و کاهش زمان پاسخ در روش پیشنهادی است که در ادامه توضیح داده می‌شود.

۳-۱-۴- اجرای تکنیک رای‌گیری آرمانی

روش پیشنهادی در این مقاله با استفاده از تکنیک رای‌گیری آرمانی، بهبود توازن بار پویا و کاهش زمان پاسخ در شبکه‌های نرم‌افزارمحور را هدف قرار می‌دهد. در این روش، توازن بار بین مسیریاب‌ها بر اساس وزن‌هایی که از آنها بدست می‌آید، تنظیم می‌شود. میزبان هدف، سیگنال‌هایی به کنترلر ارسال می‌کند و مسیریاب‌های همسایه وزن‌هایی بین صفر تا یک بر اساس بازدهی خود ارسال می‌کنند. سپس با رأی‌گیری آرمانی بر اساس حداکثر آرا، مسیریاب با کمترین وزن به عنوان مسیریاب مطلوب انتخاب می‌شود.

معادله ۱۱ نحوه رأی‌گیری مبتنی بر حداکثر آرا را نشان می‌دهد. با استفاده از این معادله، بازدهی مسیریاب مورد نظر بر اساس وزن‌ها تعیین می‌شود و مسیریاب مطلوب انتخاب می‌شود.

$$B_s = Max\left(w_1, w_2, w_3, ..., w_n\right) \qquad (۱۱)$$

در اینجا B_s وزن ماکزیمم و w_n وزن مسیرها می‌باشند. با استفاده از تکنیک رای‌گیری آرمانی، این روش قادر است بهبود قابل توجهی در توازن بار و زمان پاسخ شبکه‌های نرم‌افزارمحور را ایجاد کند. با انتخاب مسیریابی مناسب بر اساس بازدهی و وزن‌ها، بار شبکه بهینه توزیع می‌شود و زمان پاسخ به درخواست‌ها بهبود می‌یابد. این روش می‌تواند در شبکه‌های SDN و شبکه‌های نرم‌افزارمحور کارآمدی بالا را ارائه دهد.

۳-۲- معیارهای ارزیابی در روش پیشنهادی

در این بخش برخی از مهمترین معیارهای ارزیابی عملکرد روش پیشنهادی مورد بررسی قرار می‌گیرد. این معیارهای ارزیابی عبارتند از:

الف- نرخ ارسال داده در شبکه: میزان داده‌هایی است که از یک دستگاه به دستگاه دیگر در شبکه ارسال می‌شود، معمولاً به صورت بیت یا بایت در ثانیه اندازه‌گیری می‌شود، طبق رابطهٔ ۱۲.

$$S = \frac{A(mb)}{T(s)} \qquad (۱۲)$$

در اینجا S نرخ ارسال داده بر حسب مگابایت در ثانیه، مقدار داده A بر حسب مگابایت و زمان انتقال T بر حسب ثانیه است.

ب- نرخ دریافت داده شبکه نسبت به ارسال: نسبت داده‌هایی که یک دستگاه یا یک رابط شبکه دریافت می‌کند به داده‌هایی که از همان دستگاه یا رابط شبکه به سایر دستگاه‌ها یا رابط‌های شبکه ارسال می‌شود را نرخ دریافت داده شبکه نسبت به ارسال می گویند و مقدار آنرا با R/S بیان می‌کنند. این نرخ معمولاً به صورت تعداد بیت‌ها یا بایت‌های دریافتی به تعداد بیت‌ها یا بایت‌های ارسالی در یک بازه زمانی خاص محاسبه می‌شود، و به عنوان یک نسبت یا درصد بیان می‌شود و مقدار آن از رابطه‌ی ۱۳ محاسبه می‌گردد. در شبکه‌های مختلف، این نرخ‌ها ممکن است متفاوت باشند و بستگی به نوع و تنظیمات شبکه دارند.

$$R/S = \frac{R(mb)}{S(mb)} \times 100 \qquad (۱۳)$$

در اینجا R/S نسبت دریافت به ارسال بر حسب درصد، R(mb) و S(mb) به ترتیب حجم داده‌های ارسالی و دریافتی بر حسب مگابایت است.

تأخیر تبادلات انتها به انتها[1]: زمانی است که طول می‌کشد تا یک پیام از یک دستگاه ارسال کننده به دستگاه گیرنده برسد، شامل زمان پردازش، انتقال و تأخیر در دستگاه‌های مختلف است و از رابطه‌ی ۱۴ محاسبه می‌شود.

$$D = \sum_{i=1}^{i=n}\left(RT_i - ST_i\right) \qquad (۱۴)$$

در اینجا D میزان تأخیر تبادل، RT زمان دریافت و ST زمان ارسال داده در شبکه است.

زمان پاسخ شبکه: زمان پاسخ شبکه به مدت زمانی اشاره دارد که یک بسته داده برای رسیدن از یک دستگاه به دستگاه دیگر در شبکه نیاز دارد. این زمان معمولاً به صورت میلی‌ثانیه (ms) یا میکروثانیه (µs) اندازه‌گیری می‌شود که از رابطه‌ی ۱۵ محاسبه می‌شود.

$$RT = MT + PT + DT + \delta T \qquad (۱۵)$$

در اینجا RT زمان پاسخ، MT زمان جابجایی یا انتقال، PT زمان پردازش، DT زمان تأخیر در پردازش داده و δT زمان‌های ناشی از اختلالات شبکه هستند.

با بررسی این معیارها، کوتاه‌ترین مسیرها برای SDN شناسایی شده و با اجرای آنها توازن بار در شبکه بهبود یافته و زمان پاسخ شبکه کاهش خواهد یافت.

[1] End-to-End Delay

۴- نتایج تجربی

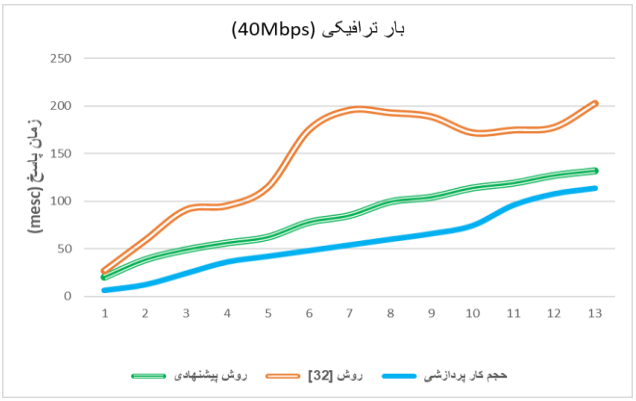
در این بخش، نتایج روش پیشنهادی و شبیه‌سازی آن ارائه می‌شود و با یک روش مشابه دیگر مقایسه می‌شود.

۴-۱- نتایج شبیه‌سازی روش پیشنهادی

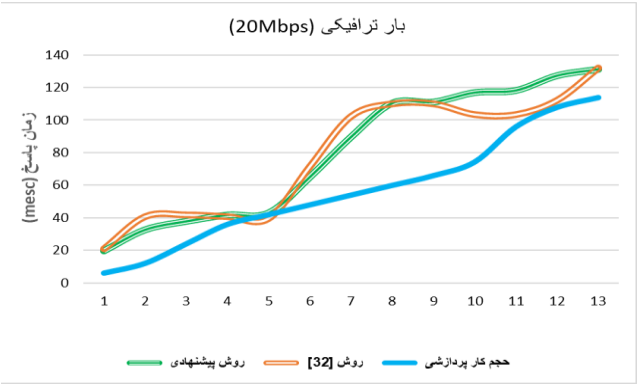
در این بخش، نتایج شبیه‌سازی روش پیشنهادی جهت بهبود توازن بار و زمان پاسخ با یک روش مشابه دیگر مورد مقایسه قرار گرفته است. برای توصیف توازن بار، از عوامل مختلفی شامل نرخ ارسال و دریافت داده در شبکه، تأخیر تبادلات و زمان پاسخ شبکه استفاده شده است. در این تحلیل، حجم درخواست‌های ورودی به عنوان ترافیک بار مد نظر قرار گرفته و حجم پردازش CPU نشان می‌دهد چند درخواست ورودی در حال پردازش هستند. برای توصیف توازن بار، یک مسئله کاهش زمان پاسخ مورد تعریف قرار گرفته است و برای بهترین مقایسه با روش پیشنهادی، از مدل صف درخواست‌های M/M/c با توزیع بار مشخص استفاده شده است [۳۲].

در این شبیه‌سازی، از آخرین نسخه‌های Mininet V2.3.0 و Floodlight V1.2 به عنوان دو ابزار برجسته در زمینه شبیه‌سازی شبکه‌های SDN استفاده شده است. این ابزارها به خوبی قابلیت تنظیم پارامترهای مختلف شبیه‌سازی براساس سناریوهای روش پیشنهادی را دارند. همچنین، برای تست عملکرد و توازن بار پویا، از ابزار Locust به عنوان پایگاه داده استفاده شده است که یک ابزار متن‌باز با قابلیت تعریف سناریوهای تست با تنظیم پارامترهای مربوطه است و برای ارزیابی و تجزیه و تحلیل نتایج تست، امکان گزارش‌دهی جامع را فراهم می‌کند. اعتبار و قابلیت‌های این ابزارها، در انجام شبیه‌سازی‌ها و آزمایشات مربوط به توازن بار و عملکرد شبکه و SDN بسیار مفید و حائز اهمیت هستند. سناریویی که در شبیه‌سازی روش پیشنهادی استفاده شده به این صورت است که در یک SDN شامل سوئیچ‌ها و یک کنترل کنندهٔ مرکزی، با اعمال یک بار ترافیک پویا و متغیر، میزان بار و زمان پاسخ در نقاط مختلف شبکه اندازه‌گیری شده است. تنظیمات شبکه شامل تعداد 100 عدد سوئیچ و یک کنترل‌کنندهٔ مرکزی و پارامترهای برنامه‌ریزی SDN شامل پروتکل ارتباط بین کنترل‌کننده و سوئیچ gRPC[1] و الگوریتم مسیریابی OSPF[2] می‌باشد. همچنین، ترافیک با الگوی پویا و متغیر شبیه‌سازی شده و نرخ ارسال درخواست‌ها و حجم داده‌ها تعیین شده است. با استفاده از ماژول‌های اندازه‌گیری بار و زمان پاسخ در نقاط مختلف شبکه، میزان بار و زمان پاسخ در سوئیچ‌ها و کنترل‌کننده‌ها اندازه‌گیری شده است. در نهایت، با تحلیل داده‌های جمع‌آوری شده، توازن بار در سوئیچ‌ها و کنترل‌کننده‌ها مورد بررسی قرار گرفته و زمان پاسخ در نقاط مختلف شبکه مقایسه و تحلیل شده است. نتایج با استفاده از روش پیشنهادی و روش‌های مقایسه‌ای [۳۲] مورد ارزیابی قرار گرفته است.

در شکل ۴، زمان پاسخ روش پیشنهادی و روش [۳۲] با سه بار ترافیکی مختلف نشان داده شده است. نتایج نشان می‌دهند که در بار ترافیکی (40Mbps)، در روش پیشنهادی زمان پاسخ کمتری نسبت به روش [۳۲] دارد. همچنین، در بار ترافیکی (20Mbps)، نیز روش پیشنهادی زمان پاسخ کمتری دارد. اما در شرایط بار ترافیکی خیلی کم و نزدیک به (0Mbps)، روش [۳۲] زمان پاسخ کمتری نسبت به روش پیشنهادی دارد. بررسی دقیق‌تر نشان می‌دهد روش پیشنهادی در مقابله با بار ترافیکی بالا و درخواست‌های پردازشی سنگین عملکرد بهتری دارد. در حالی که روش [۳۲] در شرایط بار ترافیکی کم می‌تواند زمان پاسخ کمتری داشته باشد. از آنجایی که در اکثر موارد، بار ترافیکی بالاست می‌توان ادعا کرد که روش پیشنهادی عملکرد بهتری نسبت به روش مشابه دارد.



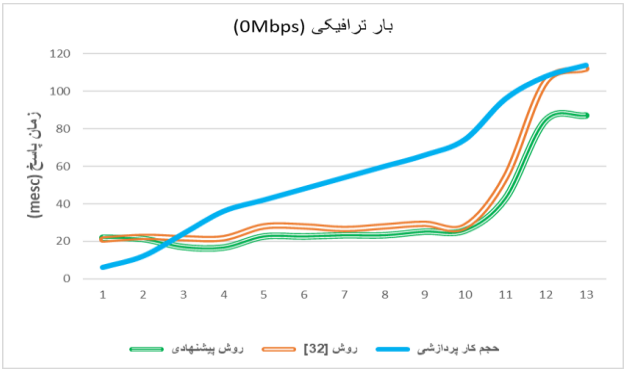
(الف)



(ب)

[1] Google Remote Procedure Call

[2] Open Shortest Path First



(ج)

شکل ۴. مقایسه‌ی زمان پاسخ روش پیشنهادی با روش [۳۲] در سه بار ترافیکی (الف) (40Mbps)، (ب) (20Mbps) و (ج) (0Mbps)

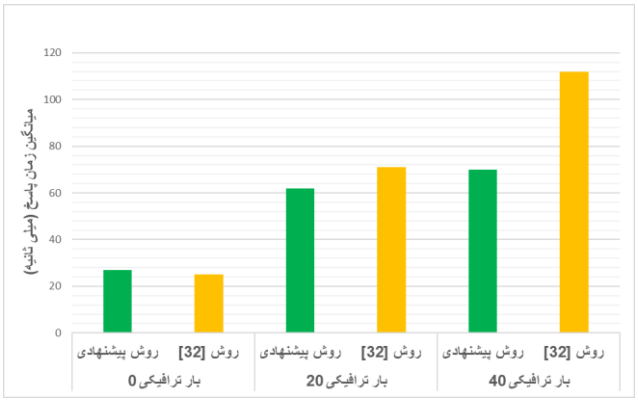
در شکل ۵، نتایج مقایسه‌ای از میانگین زمان پاسخ روش پیشنهادی و روش [۳۲] با بارهای ترافیکی (۴۰, ۲۰, ۰) نمایش داده شده است. نتایج آماری نمودار شکل ۶ نشان می‌دهد که در شرایط بار ترافیکی کم، روش [۳۲] با کمی تفاوت، عملکرد بهتری نسبت به روش پیشنهادی دارد. اما در شرایط بار ترافیکی متوسط و بالا، روش پیشنهادی عملکرد بهتری نسبت به روش [۳۲] ارائه می‌دهد. این نتایج همچنین برتری روش پیشنهادی را نسبت به روش مشابه، همانند جدول ۱ نشان می‌دهند. در آزمایشی دیگر، درصد توازن بار روش پیشنهادی با سه روش [۲۹]، [۳۲] و [۳۴]، مقایسه شده است. نمودار مقایسه درصدها در شکل ۶، نشان داده شده است. بر اساس نتایج نمودار، روش پیشنهادی دارای درصد توازن بار بالایی برابر با ۸۷ درصد است. در مقابل، روش [۲۹] با درصد توازن بار ۵۸ درصد، روش [۳۲] با درصد توازن بار ۷۵ درصد و روش [۳۴] با درصد توازن بار ۷۱ درصد عملکرد خود را نشان داده‌اند. روش پیشنهادی در توازن بار بهبود قابل توجهی نسبت به روش [۲۹] دارد و به طور معمول از روش‌های [۳۲] و [۳۴] عملکرد بهتری را ارائه می‌دهد. این نتایج نیز برتری روش پیشنهادی را نسبت به سایر روش‌های مقایسه شده در این تحقیق نشان می‌دهد.

آخرین آزمایش حجم درخواست‌های پردازش شده در یک زمان ثابت در روش پیشنهادی و روش‌های [۲۹]، [۳۲] و [۳۴]، است که نتایج مقایسه در شکل ۷، نشان داده شده است. با توجه به نمودار نشان داده شده، روش پیشنهادی در پردازش حجم درخواست‌ها عملکرد بهتری نسبت به روش‌های [۲۹] و [۳۴] دارد و همچنین از روش‌های [۳۲] حجم بیشتری را پردازش می‌کند. این نتایج می‌توانند به عنوان راهنمایی برای انتخاب مناسب‌ترین روش در مدیریت حجم درخواست‌ها در شبکه و SDN استفاده شوند.

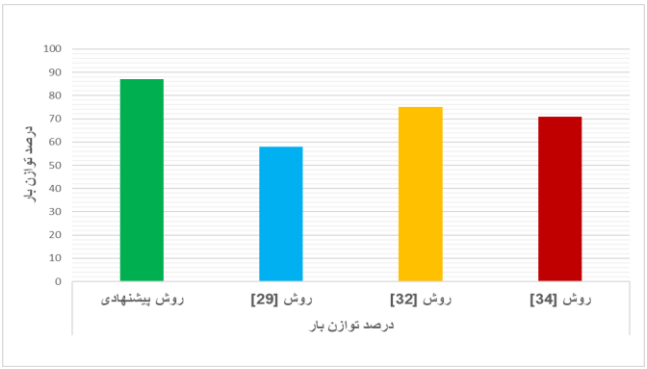
با توجه به نتایج تجربی به دست آمده، روش پیشنهادی در زمینه SDN عملکرد بهتری نسبت به سایر روش‌ها از نظر زمان پاسخ، میانگین زمان پاسخ، تعادل بار و حجم درخواست‌های پردازش شده دارد. این نتایج نشان می‌دهند که روش پیشنهادی می‌تواند به عنوان یک روش بهینه در این حوزه معرفی شود.

۵- نتیجه‌گیری

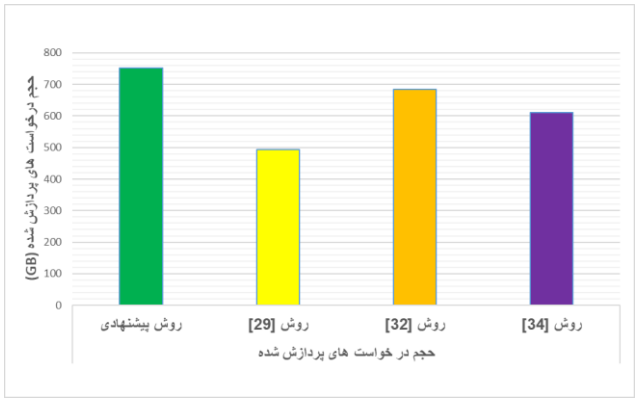
با توجه به نتایج و پیشنهادات این مطالعه، مشخص می‌شود که مدیریت حجم درخواست‌های پردازش شده در SDN امری بحرانی و حیاتی است که نیازمند ابزارها و روش‌های بهینه و کارآمد می‌باشد.



شکل ۵. مقایسه میانگین زمان پاسخ روش پیشنهادی و روش [۳۲] با بار ترافیکی (40, 20, 0).



شکل ۶. مقایسه درصد تعادل بار روش پیشنهادی با سه روش [۲۹]، [۳۲] و [۳۴].



شکل ۷. مقایسه حجم درخواست‌های پردازش شده در روش پیشنهادی و سایر روش‌ها.

در این مطالعه، یک روش پیشنهادی با عملکرد برتر در زمینه زمان پاسخ، میانگین زمان پاسخ، توازن بار و حجم درخواست‌ها ارائه شده است که می‌تواند به عنوان یک راهکار مناسب برای مدیریت SDN مورد استفاده قرار گیرد.

با اجرای روش پیشنهادی، امکان بهبود کارآیی و بهره‌وری در شبکه و SDN وجود دارد. این بهبود در نتایج عملکرد و اجرای شبکه‌ها، توازن بار بین منابع، کاهش زمان پاسخ و ارتقای تجربه کاربران تأثیر مثبتی خواهد داشت. بنابراین، روش پیشنهادی می‌تواند به‌عنوان یک انتخاب مناسب برای مدیران شبکه و متخصصان SDN در بهبود کارآیی و عملکرد شبکه و SDN مورد استفاده قرار گیرد. برای پیشرفت و توسعه بیشتر در این حوزه، پژوهش‌های آینده می‌توانند به بهبود و گسترش روش پیشنهادی، بررسی تأثیرات و عملکرد آن در شرایط مختلف، و بررسی قابلیت اعمال آن در شبکه‌های بزرگتر و پیچیده‌تر متمرکز شوند. همچنین، با بررسی ترکیب روش پیشنهادی با روش‌های دیگر و ارائه روش‌های ترکیبی، می‌توان دقت و کارآیی را بهبود بخشید. این تحقیقات آینده می‌توانند به پیشرفت و توسعه حوزه مدیریت حجم درخواست‌ها در SDN کمک کرده و بهبودی مداوم در عملکرد و کارآیی این سیستم‌ها را فراهم نمایند.

مراجع

[1] B. Isyaku, K. B. Abu Bakar, N. M. Yusuf, M. Abaker, A. Abdelmaboud, and W. Nagmeldin, "Software defined wireless sensor load balancing routing for internet of things applications: Review of approaches," (in eng), Heliyon, vol. 10, no. 9, p. e29965, May 15 2024, doi: 10.1016/j.heliyon.2024.e29965.

[2] M. Hussain, R. Amin, R. Gantassi, A. H. Alshehri, J. Frnda, and S. M. Raza, "Efficient handling of ACL policy change in SDN using reactive and proactive flow rule installation," (in eng), Sci Rep, vol. 14, no. 1, p. 14976, Jun 28 2024, doi: 10.1038/s41598-024-65721-x.

[3] K. Rui, H. Pan, and S. Shu, "Secure routing in the Internet of Things (IoT) with intrusion detection capability based on software-defined networking (SDN) and Machine Learning techniques," Scientific Reports, vol. 13, no. 1, p. 18003, 2023/10/21 2023, doi: 10.1038/s41598-023-44764-6.

[4] K. Luo, "A distributed SDN-based intrusion detection system for IoT using optimized forests," (in eng), PLoS One, vol. 18, no. 8, p. e0290694, 2023, doi: 10.1371/journal.pone.0290694.

[5] I. Smołka and J. Stój, "Utilization of SDN Technology for Flexible EtherCAT Networks Applications," Sensors, vol. 22, no. 5, p. 1944, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/5/1944.

[6] J. Gong and A. Rezaeipanah, "A fuzzy delay-bandwidth guaranteed routing algorithm for video conferencing services over SDN networks," (in eng), Multimed Tools Appl, pp. 1-30, Jan 23 2023, doi: 10.1007/s11042-023-14349-6.

[7] Z. Liu, X. Dong, L. Wang, J. Feng, C. Pan ,and Y. Li, "Satellite Network Task Deployment Method Based on SDN and ICN," (in eng), Sensors (Basel), vol. 22, no. 14, Jul 21 2022, doi: 10.3390/s22145439.

[8] D. Yang and W. T. Tsai, "SDN-Based Congestion Control and Bandwidth Allocation Scheme in 5G Networks," (in eng), Sensors (Basel), vol. 24, no. 3, Jan 24 2024, doi: 10.3390/s24030749.

[9] H. Xu, W. D. Liu, L. Li, and Q. Zhou, "An IoT-based low-cost architecture for smart libraries using SDN,", Sci Rep, vol. 14, no. 1, p. 7022, Mar 25 2024, doi: 10.1038/s41598-024-57484-2.

[10] S. E. dashti and S. Shabooei, "Improving resource allocation in mobile edge computing using gray wolf and particle swarm optimization algorithms," Journal of Information and Communication Technology, vol. 59, no. 16, pp. 108-124, 2024. [Online]. Available: http://rimag.ir/fa/Article/39782.

[11] S. Naderi, "Application identification through intelligent traffic classification," Journal of Information and Communication Technology, vol. 59, no. 16, pp. 264-278, 2024. [Online]. Available: http://rimag.ir/fa/Article/42244.

[12] D. Wang et al., "DoSDefender: A Kernel-Mode TCP DoS Prevention in Software-Defined Networking," (in eng), Sensors (Basel), vol. 23, no. 12, Jun 8 2023, doi: 10.3390/s23125426.

[13] Z. B. Zuo, R. Y. He, X. W. Zhu, and C. W. Chang, "A novel software-defined network packet security tunnel forwarding mechanism," (in eng), Math Biosci Eng, vol. 16, no. 5, pp. 4359-4381, May 17 2019, doi: 10.3934/mbe.2019217.

[14] Y. Guo et al., "Traffic Management in IoT Backbone Networks Using GNN and MAB with SDN Orchestration," (in eng), Sensors (Basel), vol. 23, no. 16, Aug 10 2023, doi: 10.3390/s23167091.

[15] L. Li, K. Li, X. Meng, Y. Wang, and X. Wang, "Dynamic weight routing and optical-code algorithm based on SDN," (in eng), Heliyon, vol. 9, no. 1, p. e12407, Jan 2023, doi: 10.1016/j.heliyon.2022.e12407.

[16] M. Hussain, N. Shah, R. Amin, S. S. Alshamrani, A. Alotaibi, and S. M. Raza, "Software-Defined Networking: Categories, Analysis, and Future Directions ",(in eng), Sensors (Basel), vol. 22, no. 15, Jul 25 2022, doi: 10.3390/s22155551.

[17] M. Hamzei, S. Khandagh, and N. Jafari Navimipour, "A Quality-of-Service-Aware Service Composition Method in the Internet of Things Using a Multi-Objective Fuzzy-Based Hybrid Algorithm," (in eng), Sensors (Basel), vol. 23, no. 16, Aug 17 2023, doi: 10.3390/s23167233.

[18] G. Yuan, Y. Yang, G. Tian, and A. M. Fathollahi-Fard, "Capacitated multi-objective disassembly scheduling with fuzzy processing time via a fruit fly optimization algorithm," (in eng), Environ Sci Pollut Res Int, Jan 31 2022, doi: 10.1007/s11356-022-18883-y.

[19] H. Li, D. Ou, and Y. Ji, "An Environmentally Sustainable Software-Defined Networking Data Dissemination Method for Mixed Traffic Flows in RSU Clouds with Energy Restriction," (in eng), Int J Environ Res Public Health, vol. 19, no. 22, Nov 16 2022, doi: 10.3390/ijerph192215112.

[20] K. Salimi and M. Mollamotalebi, "Improving the load balancing in Cloud computing using a rapid SFL algorithm (R-SFLA ",(Journal of Information and Communication Technology, vol. 57, no. 57, p. 191, 2023.

[21] A. Ghorbannia Delavar and R. Akraminejad, "WSTMOS: A Method For Optimizing Throughput, Energy, And Latency In Cloud

Workflow Scheduling," Journal of Information and Communication Technology, vol. 57, no. 57, p. 62, 2023.

[22] H. Xue, K. T. Kim, and H. Y. Youn, "Dynamic Load Balancing of Software-Defined Networking Based on Genetic-Ant Colony Optimization," (in eng), Sensors (Basel), vol. 19, no. 2, Jan 14 2019, doi: 10.3390/s19020311.

[23] X. Xu, W. K. Jia, Y. Wu, and X. Wang, "On the Optimal Lawful Intercept Access Points Placement Problem in Hybrid Software-Defined Networks," (in eng), Sensors (Basel), vol. 21, no. 2, Jan 9 2021, doi: 10.3390/s21020428.

[24] Z. Kabiri, B. Barekatain, and A. Avokh, "GOP-SDN: an enhanced load balancing method based on genetic and optimized particle swarm optimization algorithm in distributed SDNs," Wireless Networks, vol. 28, no. 6, pp. 2533-2552, 2022.

[25] R. Sharma, I. Sharma, and A. Sharma, "Load Balancing and Resource Utilization Approach in Cloud Computing Using Honey Bee-Inspired Algorithm," in International Conference on Mobile Computing and Sustainable Informatics: ICMCSI 2020, 2021: Springer, pp. 811-820.

[26] S. Ejaz, Z. Iqbal, P. A. Shah, B. H. Bukhari, A. Ali, and F. Aadil, "Traffic load balancing using software defined networking (SDN) controller as virtualized network function," IEEE Access, vol. 7, pp. 46646-46658, 2019.

[27] O. Adekoya, A. Aneiba, and M. Patwary, "An improved switch migration decision algorithm for SDN load balancing," IEEE Open Journal of the Communications Society, vol. 1, pp. 1602-1613, 2020.

[28] Y. Zhao, X. Wang, Q. He, C. Zhang, and M. Huang, "PLOFR: An online flow route framework for power saving and load balance in SDN," IEEE Systems Journal, vol. 15, no. 1, pp. 526-537, 2020.

[29] X. Shi et al., "An openflow-based load balancing strategy in SDN," Comput. Mater. Contin, vol. 62, no. 1, pp. 385-398, 2020.

[30] H. Babbar, S. Rani, D. Gupta, H. M. Aljahdali, A. Singh, and F. Al-Turjman, "Load balancing algorithm on the immense scale of internet of things in SDN for smart cities," Sustainability, vol. 13, no. 17, p. 9587, 2021.

[31] A. El Kamel and H. Youssef, "Improving switch-to-controller assignment with load balancing in multi-controller software defined WAN (SD-WAN)," Journal of Network and Systems Management, vol. 28, pp. 553-575, 2020.

[32] K. S. Sahoo, M. Tiwary, B. Sahoo, B. K. Mishra, S. RamaSubbaReddy, and A. K. Luhach, "RTSM: Response time optimisation during switch migration in software-defined wide area network," IET wireless sensor systems, vol. 10, no. 3, pp. 105-111, 2020.

[33] G. S. Begam, M. Sangeetha, and N. Shanker, "Load balancing in dcn servers through sdn machine learning algorithm," Arabian Journal for Science and Engineering, pp. 1-12, 2022.

[34] K. A. Jadhav, M. M. Mulla, and D. Narayan, "An efficient load balancing mechanism in software defined networks," in 2020 12th international conference on computational intelligence and communication networks (CICN), 2020: IEEE, pp. 116-122.

[35] M. Forghani, M. Soltanaghaei, and F. Z. Boroujeni, "Dynamic optimization scheme for load balancing and energy efficiency in software-defined networks utilizing the krill herd meta-heuristic algorithm," Computers and Electrical Engineering, vol. 114, p. 109057, 2024.

[36] G. Li, X. Wang, and Z. J. I. A. Zhang, "SDN-based load balancing scheme for multi-controller deployment," vol. 7, pp. 39612-39622, 2019.

[37] Z. Li and E. Peng, "Software-Defined Optimal Computation Task Scheduling in Vehicular Edge Networking," (in eng), Sensors (Basel), vol. 21, no. 3, Feb 1 2021, doi: 10.3390/s21030955.

[38] T. Han and N. Ansari, "A traffic load balancing framework for software-defined radio access networks powered by hybrid energy sources," IEEE/ACM Transactions on Networking, vol. 24, no. 2, pp. 1038-1051, 2015.