طبقه‌بندی و بررسی جامع تکنیک‌های بهینه‌سازی تعادل بار شبکه‌های نرم‌افزار محور

سمیه ایمان‌پور[1*]، دانشجوی کارشناسی ارشد، احمدرضا منتظرالقائم[2]، استادیار

[1] دانشکده مهندسی کامپیوتر- دانشگاه اصفهان - اصفهان - ایران - s.imanpour@ eng.ui.ac.ir

[2] دانشکده مهندسی کامپیوتر- دانشگاه اصفهان - اصفهان - ایران - a.montazerolghaem@comp.ui.ac.ir

چکیده: ترافیک شبکه‌ها روزبه‌روز در حال افزایش است؛ به همین دلیل برای مدیریت شبکه‌ها از تکنولوژی شبکه‌های نرم‌افزار محور استفاده می‌شود؛ زیرا تکنولوژی شبکه‌های نرم‌افزار محور یک نمای کلی از شبکه ارائه می‌دهد و مدیریت پیشرفته را امکان‌پذیر می‌کند. در شبکه‌های نرم‌افزار محور نیز برای بهبود عملکرد نیاز به تعادل بار هست. برای ایجاد تعادل بار در شبکه‌های نرم‌افزار محور رویکردهای بسیاری پیشنهاد شده است. این رویکردها را می‌توان طبقه‌بندی کرد؛ اما طبقه‌بندی‌های ارائه شده تاکنون دقیق نیستند. در این مقاله طبقه‌بندی دقیقی برای تعادل بار شبکه‌های نرم‌افزار محور ارائه شده است. سپس رویکردهایی که برای حل مسئله تعادل بار شبکه‌های نرم‌افزار محور از الگوریتم‌های بهینه‌سازی مبتنی بر هوش مصنوعی استفاده می‌کنند، توضیح داده شده است. در نهایت روش‌های پیش‌بینی تعادل بار شبکه‌های نرم‌افزار محور و اینکه چگونه در کاهش مصرف انرژی کمک می‌کند ارائه شده است.

واژه‌های کلیدی: تعادل بار، شبکه‌های نرم‌افزار محور، بهینه‌سازی، پیش‌بینی، مجازی‌سازی توابع شبکه

* احمدرضا منتظرالقائم، **a.montazerolghaem@comp.ui.ac.ir**

© پیش نویس مجله علمی محاسبات نرم، برای داوری در مجله

Taxonomy and comprehensive review of optimization techniques of load balancing software defined networks

Somaye Imanpour [1*], Masters student, Ahmadreza Montazerolghaem [2], Assistant professor

[1] Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran, s.imanpour@ eng.ui.ac.ir

[2] Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran, a.montazerolghaem@comp.ui.ac.ir

Abstract: Traffic is increasing day by day; for this reason, software defined network technology is used to manage network; because software defined network technology provides an overview of the network and enables advanced management. In software defined networks, load balancing is also needed to improve performance. Many approaches have been proposed for load balancing in software defined networks. These approaches can be the taxonomy; but the taxonomies presented so far are not exact. In this article, a detailed taxonomy for load balancing of software defined networks is provided. Then, the approaches that use optimization algorithms based on artificial intelligence to solve the problem of load balancing of software defined networks are explained. Finally, the methods of predicting the load balance of software defined networks and how it helps in reducing energy consumption are provided.

Keywords: *Load Balancing; Software Defined Network; Optimization; prediction; Network Function Virtualization*

*** Ahmadreza Montazerolghaem, a.montazerolghaem@comp.ui.ac.ir**

۱. مقدمه

با گسترش شبکه ها و افزایش تعداد کاربران و در نهایت افزایش ترافیک موجود، شبکه‌های سنتی از مشکلات مدیریت شبکه رنج می‌برند، زیرا دستگاه‌های سخت‌افزاری شبکه، مانند سوئیچ‌ها، روترها و متعادل‌کننده‌های بار، همگی خاص فروشنده هستند. در چنین شرایطی مدیریت شبکه دشوار می‌شود[۱]. به همین دلیل از شبکه‌های نرم‌افزار محور[1] (SDN) استفاده می‌شود که می‌توانند به‌صورت متمرکز یک نمای کلی از تمام منابع شبکه را ارائه دهند. شبکه‌های مبتنی بر نرم‌افزار با جداکردن لایه کنترل از لایه داده مدیریت پیشرفته را امکان‌پذیر کرده است.

تعادل بار در شبکه‌های نرم‌افزار محور به معنای توزیع ترافیک در شبکه است. شبکه‌های معمولی غیر شبکه‌های نرم‌افزار محور قادر به ارائه یک نمای کلی از توپولوژی و منابع شبکه نیستند؛ بنابراین، مکانیسم‌های تعادل بار به‌صراحت تعریف نشده‌اند. در نتیجه، شبکه‌های نرم‌افزار محور امکانات جدیدی را برای بهبود تعادل بار شبکه معمولی به ارمغان می‌آورد [۳]. حجم بالای ترافیک شبکه‌های امروزی به سمت سرورها و سوئیچ‌ها در حال حرکت است که باعث ازدحام و اضافه‌بار در شبکه می‌شود. حال کنترلر باید با استفاده از استراتژی‌های تعادل بار، بار را بین سرورهای مختلف توزیع کند. اما با یک کنترلر نمی‌توان کل شبکه را مدیریت کرد؛ چون مقیاس‌پذیری و دردسترس‌بودن را کاهش می‌دهد. در نتیجه از معماری کنترلر توزیع شده، استفاده شده است. در معماری کنترلر توزیع شده هر کنترلر بر اساس ظرفیتی که دارد تعدادی سوئیچ و سرور می‌تواند مدیریت کند. سوئیچ‌ها و سرورهایی که هر کنترلر مدیریت می‌کند در دامنه آن کنترلر قرار می‌گیرد. در نتیجه دامنه هر کنترلر شامل چند سوئیچ و سرور که توسط آن کنترلر مدیریت می‌شود. شبکه‌های چند کنترلری نیز ممکن است دچار اضافه‌بار شوند، در نتیجه از الگوریتم مهاجرت سوئیچ استفاده می‌کنیم. مهاجرت سوئیچ به معنای انتقال یک سوئیچ از دامنه کنترلر اضافه‌بار به دامنه کنترلر کم‌بار برای ایجاد تعادل بار کنترلرها در شبکه نرم‌افزار محور است [۱-۳].

به‌صورت کلی تعادل بار شبکه‌های نرم‌افزار محور به تعادل بار سرور، تعادل بار لینک و تعادل بار کنترلر تقسیم می‌شود. برای حل مشکل تعادل بار شبکه‌های نرم‌افزار محور برخی از مقالات از روش بهینه‌سازی استفاده کرده‌اند. در مسائل بهینه‌سازی، مسئله تعادل بار شبکه‌های نرم‌افزار محور از نظر پیچیدگی محاسبات NP-Hard [۴] است. به همین دلیل در سال‌های اخیر برای حل مسئله تعادل بار شبکه‌های نرم‌افزار محور مبتنی بر بهینه‌سازی از الگوریتم‌های هوش مصنوعی استفاده می‌کنند [۵]. اما گاهی اوقات ایجاد تعادل بار شبکه‌های نرم‌افزار محور کافی نیست و تجهیزات موجود حتی با وجود تعادل بار پاسخگوی نیاز کاربران نیست و تجهیزات جدید باید اضافه شود و شبکه گسترش داده شود؛ اما اضافه‌کردن تجهیزات خود هزینه‌هایی را به دنبال دارد. به همین دلیل از مجازی‌سازی توابع شبکه استفاده می‌شود [۶].وقتی یک شبکه از مجازی‌سازی توابع شبکه استفاده می‌کند نیازی به نصب تجهیزات جدید نیست و این باعث کاهش هزینه‌ها می‌شود. به همین دلیل برای تعادل بار، علاوه بر شبکه‌های نرم‌افزار محور به مجازی‌سازی توابع شبکه نیز نیاز داریم. هرکدام از این مفهوم‌های شبکه نرم‌افزار محور و مجازی‌سازی توابع شبکه بدون نیاز به دیگری به کار خود ادامه می‌دهد؛ اما مکمل یکدیگر هستند. به همین دلیل مفهوم جدیدی به نام مجازی‌سازی توابع شبکه‌های نرم‌افزار محور به وجود آمد [۷].در برخی از مقالات تعادل بار شبکه‌های نرم‌افزار محور برای مدیریت بهتر منابع، از پیش‌بینی میزان مصرف منابع در آینده استفاده می‌شود. تا بر اساس پیش‌بینی، منابع شبکه کاهش یا افزایش داده شود. این هم باعث می‌شود مدیریت بهتری روی منابع داشته باشم و هم باعث کاهش مصرف انرژی می‌شود.

[1]Software Defined Network(SDN)

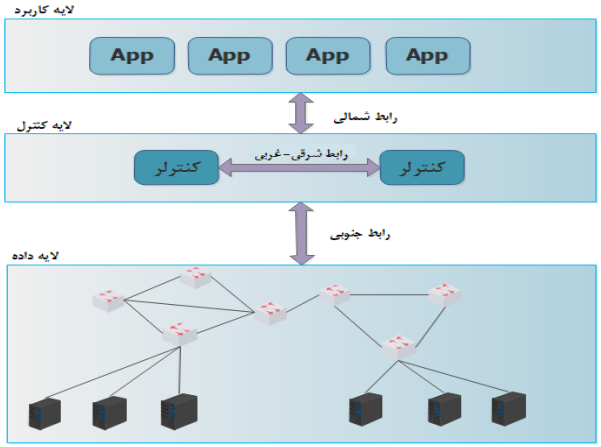
در بخش ۲ ادبیات موضوع و مفاهیم مورداستفاده در مقاله توضیح داده شده است. در بخش ۳ طبقه‌بندی دقیق‌تری از تعادل بار شبکه‌های نرم‌افزار محور ارائه شده است. در بخش ۴ تعادل بار شبکه‌های نرم‌افزار محور مبتنی بر بهینه‌سازی که از الگوریتم‌های هوش مصنوعی استفاده کرده‌اند و چگونگی استفاده از این الگوریتم‌ها توضیح داده شده است. در بخش ۵ روش‌های تعادل بار شبکه‌های نرم‌افزار محور که از پیش‌بینی منابع استفاده می‌کنند شرح داده شده است. در نهایت در بخش ۶ نتیجه‌گیری بیان شده است.

۲. ادبیات موضوع

۱،۲. شبکه‌های نرم‌افزار محور

شبکه نرم‌افزار محور یک تکنولوژی جدید است برای مدیریت پیشرفته شبکه‌های امروزی [۱]. معماری شبکه‌های نرم‌افزار محور همان‌طور که در شکل ۱ نشان‌داده‌شده است، شامل سه‌لایه است: ۱. لایه کاربرد ۲. لایه کنترل ۳. لایه داده

برای ارتباط بین لایه‌ها رابط‌های متفاوتی تعریف شده است. رابط شمالی برای ارتباط بین لایه کنترل و لایه کاربرد است. رابط جنوبی برای ارتباط بین لایه کنترل و لایه داده است [۲]. OpenFlow متداول‌ترین رابط جنوبی برای ارتباط بین کنترلر و سوئیچ‌ها است و می‌تواند مدیریت شبکه را تسریع کند [۸]. رابط شرقی – غربی برای ارتباط بین کنترلرها در لایه کنترلر است [۲]. کنترلر مهم‌ترین جزء شبکه‌های نرم‌افزار محور است که مدیریت ترافیک لایه داده را به عهده دارد.



شکل ۱: معماری شبکه نرم‌افزار محور

۲،۲. مجازی‌سازی توابع شبکه

مجازی‌سازی توابع شبکه (NFV)[1] با مجازی‌سازی، توابع شبکه را از زیرساخت سخت‌افزاری جدا می‌کند و در شبکه دیگر نیازی به نصب تجهیزات جدید نیست. در نتیجه باعث کاهش هزینه و مصرف انرژی می‌شود [۶]. معماری مجازی‌سازی توابع شبکه در شکل ۲ نشان‌داده‌شده است [۹] شامل سه بخش است:

- **زیرساخت مجازی‌سازی توابع شبکه[2] (NFVI)**: مرتبط با صفحه داده است و منابع شبکه سخت‌افزاری را با کمک لایه مجازی هایپروایزر از منابع مجازی جدا می‌کند. همچنین یک محیط مجازی برای استقرار توابع شبکه مجازی فراهم می‌کند.
- **توابع شبکه مجازی[3] (VNF)**: توابع شبکه مجازی برنامه‌هایی هستند که می‌توانند عملکرد شبکه را که توسط دستگاه‌های شبکه ارائه شده است، فراهم کنند.
- **مدیریت و ارکستراسیون مجازی‌سازی توابع شبکه[4] (MANO)**: مسئول مدیریت و تنظیم زیرساخت

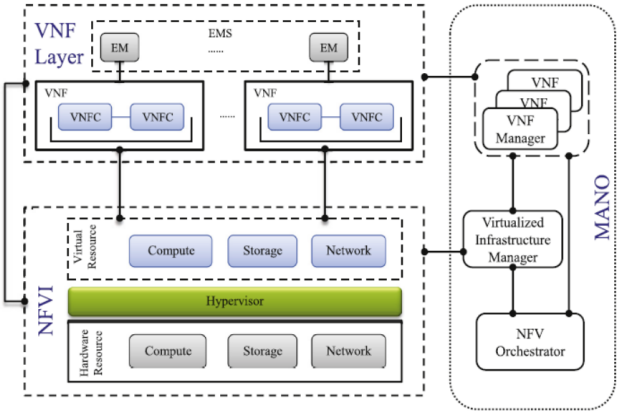
[1] Network Function Virtualization

[2] Network Function Virtualization Infrastructure

[3] Virtual Network Function

[4] Management and orchestration

مجازی‌سازی توابع شبکه و مدیریت و تهیه توابع شبکه مجازی است [۹].

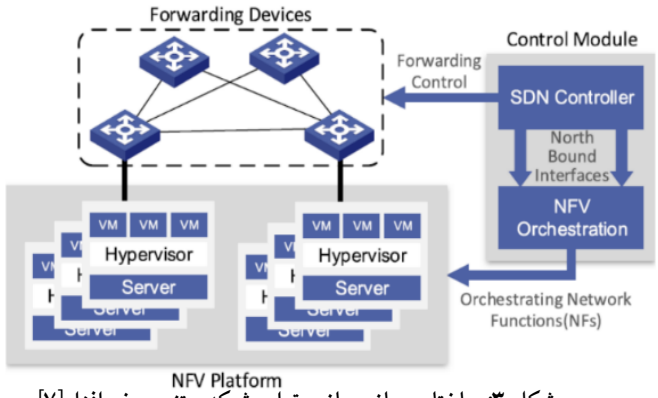


شکل ۲: معماری SDN [۹]

۳،۲. مجازی‌سازی توابع شبکه نرم‌افزار محور

مجازی‌سازی توابع شبکه نرم‌افزار محور ترکیب شبکه‌های نرم‌افزار محور و مجازی‌سازی توابع شبکه است. شبکه‌های نرم‌افزار محور ترافیک شبکه را مدیریت می‌کند و مجازی‌سازی توابع شبکه منابع را مدیریت می‌کند. هرکدام به‌صورت مستقل کار خود را به‌خوبی انجام می‌دهد؛ اما این دو مکمل یکدیگرند. مجازی‌سازی توابع شبکه نرم‌افزار محور همان‌طور که در شکل ۳ نشان‌داده‌شده است شامل سه بخش است [۷]:

- **پلتفرم مجازی‌سازی توابع شبکه:** سرورها توسط مجازی‌سازی توابع شبکه، مجازی شده‌اند که در نتیجه پهنای باند بیشتر و هزینه کمتر می‌شود.
- **دستگاه‌های انتقال:** مدیریت ترافیک شبکه توسط کنترلر شبکه نرم‌افزار محور انجام می‌شود و از پروتکل OpenFlow برای ارتباط بین کنترلر و دستگاه انتقال استفاده شده است.



شکل ۳: ساختار مجازی‌سازی توابع شبکه مبتنی بر نرم‌افزار[۷].

- **ماژول کنترلر:** این ماژول برای مدیریت کل شبکه است که شامل دو بخش کنترلر شبکه‌های نرم‌افزار محور و ارکستراسیون مجازی‌سازی توابع شبکه. ارکستراسیون مجازی‌سازی توابع شبکه وظیفه مدیریت شبکه مجازی را به عهده دارد و توسط کنترلر شبکه‌های نرم‌افزار محور از طریق رابط‌هایی استاندارد کنترل می‌شود [۷].

۴،۲. تعادل بار شبکه‌های نرم‌افزار محور

تعادل بار روشی است برای توزیع ترافیک در شبکه‌ها. طرح‌های تعادل بار سنتی [۱۰] محدودیت‌های سخت‌افزاری بسیاری دارند و پرهزینه هستند و مقیاس‌پذیر نیستند. اما شبکه‌های نرم‌افزار محور با داشتن نمای کلی از شبکه باعث تعادل بار بهینه می‌شود. در نتیجه می‌تواند امکانات جدیدی را برای بهبود تعادل بار در شبکه به ارمغان آورد [۳].

۳. طبقه‌بندی تعادل بار شبکه‌های نرم‌افزار محور

در این بخش به طبقه‌بندی تعادل بار شبکه‌های نرم‌افزار محور می‌پردازیم. در مقاله [۳] تعادل بار شبکه‌های نرم‌افزار محور به دو بخش تعادل بار صفحه داده و تعادل بار صفحه کنترل تقسیم شده است. همان‌طور که در شکل ۴ نشان‌داده‌شده است در نمودار درختی قسمت‌هایی که به رنگ سبز هستند، نمودار درختی مقاله [۳] هستند.

اما مقالاتی وجود دارند که به‌صورت هم‌زمان روی تعادل بار صفحه داده و صفحه کنترل کار کرده‌اند. به همین دلیل در این مقاله، تعادل بار شبکه‌های نرم‌افزار محور به سه دسته تعادل بار صفحه داده، تعادل بار صفحه کنترل، تعادل بار هم‌زمان صفحه داده و صفحه کنترل تقسیم شده است. همان‌طور که در نمودار درختی نشان‌داده‌شده است قسمت‌هایی که به رنگ نارنجی هستند در این مقاله اضافه شده‌اند. این مقاله ساختار دقیق‌تری از تعادل بار شبکه‌های نرم‌افزار محور ارائه داده است.

۱،۳. تعادل بار صفحه داده

شکل ۴: نمودار درختی طبقه بندی تعادل بار شبکه های نرم افزار محور

شبکه برای دریافت و عبور ترافیک است. تعادل بار در صفحه داده به سه دسته تقسیم می‌شود:

۱،۱،۳. تعادل بار سرورها:

ترافیک شبکه‌ها روزبه‌روز در حال افزایش است و این ترافیک به سمت سرورها حرکت می‌کند. حال ممکن است توزیع بار بین سرورها به‌درستی انجام نشود و سروری دچار اضافه‌بار شود. به همین دلیل به تعادل بار بین سرورها نیاز داریم [۳]. تعادل بار سرورها خود به دودسته تعادل بار سرورهای ایستا [۱۱] و تعادل بار سرورهای پویا تقسیم می‌شود.

الف. تعادل بار سرورهای ایستا:

الگوریتم‌های تعادل بار ایستا، برای شبکه‌هایی که سرورهای کاملاً یکسان دارند مناسب است؛ اما برای شبکه‌های پویا مناسب نیست؛ چون انعطاف‌پذیری ندارد [۱۲].

کائور و همکارانش [۱۳] با استفاده از شبکه‌های نرم‌افزار محور و استراتژی Round Robin ، بار را بین سرورهای مختلف به‌صورت مساوی تقسیم می‌کند تا سروری دچار اضافه‌بار نشود. معماری تعادل بار شامل یک سوئیچ Open Flow و یک کنترلر POX و چندین سرور متصل به سوئیچ است با آدرس IP ثابت و کنترلر، لیستی از سرورها را دارد. زمانی که بارهای ورودی وارد می‌شوند کنترلر طبق ترتیب بار را بین سرورها تقسیم می‌کند و میانگین زمان پاسخ این روش از روش Random کمتر است.

ب. تعادل بار سرورهای پویا:

الگوریتم‌های تعادل بار سرورهای پویا ابتدا وضعیت فعلی سرورها را به دست می‌آورد سپس بر اساس وضعیت به‌دست‌آمده و الگوریتم تعادل بار شبکه‌های نرم‌افزار محور، سرور مناسبی را برای دریافت ترافیک انتخاب می‌کند [۳].

ژانگ و همکارانش [۱۴] یک طرح تعادل بار پویا مبتنی بر زمان پاسخ سرور (LBBSRT) را در معماری شبکه‌های نرم‌افزار محور ارائه کرده‌اند. مدل سیستمی که طراحی کرده است شامل سه بخش، سرور و مشتری، سوئیچ Open Flow و کنترلر شبکه‌های نرم‌افزار محور است. برای تعادل بار بین سرورها ابتدا زمان پاسخ تمام سرورها را به دست می‌آورد و بر اساس زمان پاسخ بار را بین سرورها تقسیم می‌کند. یعنی هر سرور که زمان پاسخ کمتر یا پایدارتری داشته باشد برای دریافت بار در اولویت قرار می‌گیرد. به این صورت که حداقل و حداکثر زمان پاسخ

سرورها را به دست می‌آورد و اختلاف حداکثر زمان پاسخ و حداقل زمان پاسخ را به دست می‌آورد. این مقدار اختلاف اگر کم باشد نشان‌دهنده این است که سرورها بار مشابه دارند و سروری با حداقل زمان پاسخ انتخاب می‌کند. در غیر این صورت انحراف استاندارد را به دست می‌آورد و هر سرور که انحراف استاندارد کمتری داشته باشد برای دریافت بار ورودی در اولویت قرار می‌گیرد. یعنی هر سرور که زمان پاسخ پایدارتری داشته باشد انتخاب می‌شود. این روش نسبت به روش Round Robin، Random عملکرد بهتری دارد و به دلیل کاهش نیازهای سخت‌افزاری و روش سفارشی‌سازی نرم‌افزار مقرون‌به‌صرفه‌تر از راه‌های سنتی است گرچه این طرح به طور مؤثر بر بسیاری از مسائل تعادل بار غلبه می‌کند؛ اما صرفه‌جویی در انرژی سرور را در نظر نمی‌گیرد.

منتظرالقائم[۱۵] مسئله توزیع حالت بین چندین گره تعریف شده است و هدف از حل این مسئله افزایش گذردهی کل تماس ها و دسترسی پذیری سرورهاست. در این مقاله ابتدا میزان مصرف CPU در یک سرورSIP بررسی شده است که با توجه به نتایج بررسی شده میزان مصرف CPU توسط سرور حالتمند با افزایش نرخ تماس با سرعت بیشتری نسبت به سرور بدون حالت افزایش می یابد، زیرا در سرورهای SIP که حالتمند هستند میزان مصرف CPU به دلیل نگه داشتن حالت در زمانی که تماس افزایش می یابد بسیار زیاد است در نتیجه در این مقاله هر سرور تنها برای بخشی از درخواست ها حالت را نگه می دارد و برای بقیه فاقد حالت است. در مرحله بعد که طراحی چارچوب مبتنی بر شبکه نرم افزار محور است که برای توزیع حالت در شبکه SIP در لایه کنترل سه ماژول اضافه شده است و در نهایت با توجه به ارزیابی های انجام گرفته این مدل به افزایش ۱۵ درصد تا ۲۰ درصدی بیشینه گذردهی تماس دست یافته است.

منتظرالقائم و همکارشان [16] یک الگوریتم تطبیقی توزیع شده و انتها به انتهای مبتنی بر پنجره را برای کنترل اضافه بار پیشنهاد کرده است. این مقاله یک الگوریتم توزیع شده کنترلر اضافه بار SIP که مبتنی بر بازخورد ضمنی است و البته مبتنی بر شبکه های نرم افزار محور نیست ،پیشنهاد کرده است. دلیل اصلی اضافه بار سرور SIP پیام های درخواست است و برای محاسبه بیشینه جدید اندازه پنجره با تحلیل نسبت پیام های درخواست به پیام های تایید است. نتایج بدست آمده نشان می دهد که گذردهی بهبود یافته است و تاخیر بسیار کمتر شده است و نرخ ارسال مجدد کاهش یافته است.

۲،۱،۳. تعادل بار لینک‌ها

ترافیک از طریق لینک‌ها به سمت سرورها در حال حرکت است؛ اما لینک‌ها هم ممکن است دچار اضافه‌بار شوند. در نتیجه برای عبور ترافیک، لینکی انتخاب می‌شود که کمترین ترافیک را داشته باشد [۳].

شیانگی و همکارانش [۱۷] یک چارچوب بهینه‌سازی کارایی انرژی مبتنی بر پیش‌بینی ترافیک[1] (TPEO) در شبکه‌های نرم‌افزار محور پیشنهاد می‌کند. ایده اصلی بهینه‌سازی کارایی انرژی مبتنی بر پیش‌بینی ترافیک بهبود بهره‌وری انرژی شبکه بر اساس پیش‌بینی ترافیک، با درنظرگرفتن موازنه بین بار شبکه و مصرف انرژی است. ابتدا یک مکانیسم نظارت بر ترافیک تطبیقی را طراحی می‌کند که می‌تواند دوره نظارت ترافیک را به‌صورت پویا تنظیم کند و بار ترافیک شبکه‌های نرم‌افزار محور را در زمان واقعی به دست آورد. هزینه نظارت بر شبکه کاهش می‌یابد درحالی‌که دقت نظارت تضمین می‌شود. بر اساس روش پیش‌بینی شبکه عصبی، ترافیک بی‌درنگ شبکه‌های نرم‌افزار محور را بر اساس یادگیری عمیق طراحی می‌کنند. یک مدل پیش‌بینی را از طریق آموزش آفلاین ایجاد می‌کنند، سپس ترافیک شبکه را در زمان واقعی توسط پنجره کشویی سری‌های زمانی ثبت و پیش‌بینی می‌کنند. سپس یک الگوریتم اکتشافی آنلاین جدید برای بهینه‌سازی بازده انرژی شبکه نرم‌افزار محور بر اساس پیش‌بینی ترافیک پیشنهاد می‌کنند که منابع لینک را با تغییرات شبکه برای دستیابی به تعادل بار پویا و صرفه‌جویی در انرژی هماهنگ می‌کند. مزیت الگوریتم پیشنهادی این است که می‌تواند بار شبکه را متعادل کند و به یک اثر صرفه‌جویی انرژی

[1] Traffic Prediction-based Energy efficiency Optimization(TPEO)

دست یابد، و مصرف انرژی کاهش یابد. بااین‌حال، محدودیت این الگوریتم این است که استراتژی‌های صرفه‌جویی در انرژی منجر به تمرکز نسبی ترافیک شبکه می‌شود که مستعد خرابی لینک‌ها است.

دانگ و همکاران [۱۸] یک الگوریتم مسیریابی متعادل‌کننده بار جدید را تحت محدودیت‌های کیفیت انتقال برای شبکه‌های مش بی‌سیم پیشنهاد کرده است. الگوریتم پیشنهادی مبتنی بر شبکه‌سازی نرم‌افزار محور است که مسیریابی تعادل بار تحت محدودیت‌های کیفیت انتقال[1] (LBRCQT) نامیده می‌شود. الگوریتم مسیریابی تعادل بار تحت محدودیت‌های کیفیت انتقال بر اساس اصل مسیریابی متمرکز کار می‌کند و تابع مسیریابی در کنترلر شبکه نرم‌افزار محور پیاده‌سازی می‌شود. الگوریتم پیشنهادی می‌تواند عملکرد شبکه را از نظر نسبت سیگنال به نویز[2] (SNR)، نسبت تحویل بسته[3] (PDR) و توان بهبود بخشد.

وانگ و همکارانش [۱۹] یک چارچوب مدیریت مسیر متعادل برای بهبود عملکرد شبکه مرکز داده‌های درخت چاق[4] با بهره‌برداری از شبکه‌های نرم‌افزار محور با هزینه کم پیشنهاد کرده است. این مقاله ابتدا وضعیت شبکه را با استفاده از سه جدول ذخیره‌سازی اطلاعات مسیرها، بارها، سوئیچ‌ها نظارت می‌کند. سپس در صورت وجود تراکم احتمالی، مکانیسم اصلاح مسیر تطبیقی راه‌اندازی می‌شود. برای صرفه‌جویی در هزینه‌های سربار کنترلر مکانیسم پویایی اطلاعات توسعه داده شده است تا در صورت تطبیق اطلاعات سوئیچ‌ها را با پیام‌های کمتر جستجو کند. مکانیسم اصلاح تطبیقی دارای دو ویژگی است: ۱. تنها در صورت لزوم فراخوانی می‌شود که در نتیجه در هزینه و مصرف انرژی صرفه‌جویی می‌شود. ۲. حذف ورودی‌های قدیمی از جدول سوئیچ برای جلوگیری از سرریز بافر.

هان و همکارانش [۲۰] یک روش استقرار زنجیره تابع سرویس بر اساس تئوری بدهی شبکه پیشنهاد کرده‌اند. ابتدا یک مجموعه گره کلید کاندید برای تشکیل شبکه هدایت شده تعیین می‌شود. در مرحله دوم مسئله استقرار به وظیفه‌ای برای یافتن مسیر بهینه در شبکه هدایت شده تبدیل می‌شود. در مرحله سوم با درنظرگرفتن هزینه انتقال لینک به‌عنوان هزینه و تقاضای پهنای باند به‌عنوان محدودیت، یک الگوریتم حداقل هزینه – حداکثر جریان برای تعیین مسیر بهینه استقرار و تکمیل استقرار زنجیره تابع سرویس استفاده می‌شود.

اندجامبا و همکارانش[۲۱] پروتکل مسیریابی متعادل‌کننده بار[5](LBRP) مبتنی بر شبکه‌های نرم‌افزار محور را برای تضمین کیفیت ویدئو هنگام پخش در نرم‌افزار تعریف شده، معرفی می‌کند. این پروتکل از استراتژی مهندسی ترافیک برای دستیابی به کیفیت سرویس استفاده می‌کند. به این صورت که جریان‌های ترافیکی را به دودسته تقسیم می‌کند:۱. جریان‌های حساس به تأخیر (جریان واقعی) ۲. جریان‌هایی که به تأخیر حساس نیستند (جریان‌های غیرواقعی). جریان‌های واقعی اولویت بیشتری نسبت به جریان‌های غیرواقعی دارند و تلاش می‌کنند جریان‌های واقعی را از لینک‌هایی که پهنای باند تضمین شده دارند عبور دهند و جریان‌های غیرواقعی را از بقیه لینک‌های جایگزین عبور دهد. این کار باعث بهبود کیفیت تجربه کاربر از برنامه‌های پخش ویدئوی ذخیره شده، می‌شود.

گالان و همکارانش [۲۲] به مشکل تعادل بار ترافیک و حداقل‌سازی مصرف انرژی در یک شبکه IP/SDN می‌پردازد. برای مهاجرت شبکه‌های IP قدیمی به شبکه نرم‌افزار محور باید تجهیزات شبکه IP با گره‌های سازگار با شبکه نرم‌افزار محور جایگزین شود. اما چنین مهاجرتی برای شبکه‌های بزرگ ساده یا سریع نیست در نتیجه باید به‌صورت تدریجی مهاجرت کرد. چنین مهاجرت‌هایی باعث ایجاد شبکه ترکیبی IP/SDN می‌شود . این مقاله برای تعادل بار لینک‌ها از الگوی شبکه نرم‌افزار محور برای تقسیم ترافیک بین گره‌های شبکه نرم‌افزار محور استفاده می‌کند. همچنین از الگوریتم ژنتیک برای بهبود مصرف انرژی

[1] Load Balancing Routing under Constraints of Quality of Transmission(LBRCQT)

[2] Signal to Noise Ratio(SNR)

[3] Package Delivery Ratio(PDR)

[4] Fat Tree

[5] Load Balancing Routing Protocol(LBRP)

استفاده می‌کند. به این صورت که با استفاده از الگوریتم ژنتیک بهترین پیکربندی شبکه را که شامل کمترین تعداد لینک‌های فعال باشد با درنظرگرفتن ظرفیت لینک‌ها و ترافیک موجود، می‌یابد. این رویکرد باعث کاهش ۵۰ درصدی حداکثر استفاده از لینک و ۶۰ درصدی مصرف انرژی می‌شود.

خراسانی فردوانی و همکارانشان [۲۳] یک مسیریابی انرژی کارا مبتنی بر اینترنت اشیا که مبتنی بر شبکه های نرم افزار محور نیست ارائه کرده است. برای مسیریابی از پارامترهای تعداد گام، فاصله و انرژی گره های میانی برای انتخاب مسیر استفاده می کند.روش پیشنهادی ار نظر پارامترهای نرخ گم شدن بسته و انرژی باقیمانده بهبود یافته است.

۳،۱،۳. تعادل بار هم‌زمان سرورها و لینک‌ها

در تعادل بار هم‌زمان سرور و لینک، ترافیک را از مسیر منتخب الگوریتم تعادل بار لینک به سرور منتخب الگوریتم تعادل بار سرور ارسال می‌کند تا به‌صورت هم‌زمان تعادل بار سرور و لینک را داشته باشیم.

منتظرالقائم [۲۴] یک چارچوب جدید مبتنی بر شبکه‌های مبتنی بر نرم‌افزار برای برآورده‌کردن الزامات کیفیت سرویس سرویس‌های مختلف اینترنت اشیا و ایجاد تعادل ترافیک بین سرورهای اینترنت اشیا به طور هم‌زمان پیشنهاد کرده‌اند. ابتدا ثابت کرده است که این دو مسئله به طور هم‌زمان NP-hard هستند و برای کاهش پیچیدگی زمانی مسئله را به دو مشکل فرعی تقسیم کرده‌اند: (۱) انتخاب سرور (۲) انتخاب مسیر. در این راستا یک چارچوب مبتنی بر شبکه‌های نرم‌افزار محور طراحی شده است. این چارچوب از سه‌لایه تشکیل شده است که از طریق یک‌سری رابط با یکدیگر تعامل دارند. کنترلر ابتدا آمار منابع مصرفی (CPU و حافظه) سرورهای اینترنت اشیا را جمع‌آوری می کندو سپس با استفاده از الگوریتم حداقل میانگین مربعات نرمال شده (NLMS)[1] میزان مصرف منابع را در آینده پیش‌بینی می‌کند و باتوجه‌به خروجی این الگوریتم و با استفاده از سیستم فازی ظرفیت سرورها برای دریافت ترافیک مشخص می‌شود. از طرف دیگر ترافیک اینترنت اشیا در سه کلاس دسته‌بندی‌شده است. هرکدام از این کلاس‌های ترافیک کیفیت سرویس متفاوتی دارند. در نتیجه بر اساس کیفیت سرویس ترافیک و ظرفیت سرور بار بین سرورها تقسیم می‌شود. یعنی ترافیک‌های با کیفیت سرویس بیشتر به سمت سرورهای با ظرفیت بیشتر مسیریابی می‌شوند. در نتیجه هم کیفیت سرویس ترافیک‌ها حفظ می‌شود و هم تعادل بار بین سرورها ایجاد می‌شود.

مقاله بعدی تعادل بار مبتنی شبکه‌های نرم‌افزار محور است و از تکنیک‌های یادگیری ماشینی مانند شبکه‌های عصبی مصنوعی (ANN)[2] و یادگیری تقویتی استفاده می‌کند که می‌تواند برای متعادل‌سازی بار سرور و مسیر محیط‌های شبکه‌های نرم‌افزار محور اعمال شود. توازن بار سرور از طریق یک شبکه عصبی مصنوعی به دست می‌آید و شبکه عصبی مصنوعی یاد می‌گیرد که ترافیک را بین سرورها بر اساس یادگیری‌های مختلف از تجربیات قبلی خود توزیع کند. در ابتدا الگوریتم‌های Random Round Robin, Weighted Round Robin, Least Connection روی متعادل‌کننده بار برای به‌دست‌آوردن پارامترهای سرور مانند: (۱) استفاده از پردازنده (۲) استفاده از حافظه (۳) نوع فایل درخواستی (٤) بایت ارسال شده به مشتری در صورت درخواست (۵) بایت‌های دریافت شده، استفاده می‌شود. سپس بر اساس همین پارامترها و الگوریتم‌ها، الگوهایی را یاد می‌گیرد و از طریق آن مدل می‌سازد. برای متعادل‌سازی بار مبتنی بر مسیر از توپولوژی Abilene استفاده می‌کند که برای متعادل‌سازی بار میان مسیرهای با تأخیر کم برای یافتن بهترین مسیرهای مناسب از مشتری به سرور، پیاده‌سازی شده است [۲۵].

منتظرالقائم [۲۶] برای حل بحران مدیریت منابع شبکه‌های اینترنت اشیا چندرسانه‌ای[3] (IoMT) راه‌حلی طراحی کرده‌اند. زیرساخت اینترنت اشیا چندرسانه‌ای از دو جنبه با بحران مدیریت منابع مواجه است: تعادل بار و بهینه‌سازی انرژی که

[1] Normalised Least Connection Mean Squares

[2] Artifical Neural Network (ANN)

[3] Internet of Multimedia Things(IoMT)

محدودیت‌های انرژی و بار به طور هم‌زمان یک مسئله NP-hard است و پیچیدگی زمانی بالایی دارد و به طور دقیق‌تر یک مسئله برنامه‌ریزی غیرخطی عدد صحیح مختلط (MINLP)[1] است.چون این مسئله از پیچیدگی بالایی برخوردار است به سه مسئله فرعی تقسیم شده است: (۱) اندازه شبکه (۲) انتخاب سرور مناسب (۳) انتخاب مسیر مناسب. با تعیین اندازه دقیق شبکه مصرف انرژی کاهش می‌یابد و با انتخاب سرور و مسیر مناسب تعادل بار شبکه محقق می‌شود. برای مدیریت یکپارچه منابع ابری اینترنت اشیا چندرسانه‌ای یک چارچوب جدید طراحی کرده است. این چارچوب مبتنی بر محاسبات ابری از دولایه ابر لبه (Edge cloud) و ابر هسته (Core cloud) تشکیل شده است. که این دولایه‌ای بودن از منطق شبکه‌های نرم‌افزار محور سرچشمه می‌گیرد.Core cloud شامل طراح کنترلر منابع کم‌مصرف و بار متعادل (ELRC)[2] است و ابر لبه هم شامل سرورها و سوئیچ‌های مجازی است. سرورها همگی مبتنی بر فناوری مجازی‌سازی توابع شبکه هستند که کنترلر بتواند اندازه شبکه را تغییر دهد. سپس با استفاده از روش پیش‌بینی حداقل میانگین مربعات نرمال شده میزان مصرف سرورها را باتوجه‌به تاریخچه مصرف منابع سرورها (یعنی میزان مصرف CPU و حافظه و هارددیسک) تخمین می‌زند و بر اساس پیش‌بینی و منطق فازی، سرورها در سه کلاس دسته‌بندی می‌شوند. بر همین اساس وضعیت کل شبکه مشخص می‌شود و بر اساس وضعیت شبکه اندازه شبکه تغییر می‌کند. اگر شبکه در وضعیت اضافه‌بار باشد اندازه شبکه بزرگ‌تر می‌شود و درصورتی‌که شبکه در حالت کم‌بار باشد اندازه شبکه کوچک‌تری شود. در نهایت با استفاده از الگوریتم حداقل بار، بهترین سرور انتخاب می‌شود و با استفاده از الگوریتم کوتاه‌ترین مسیر بهترین مسیر مشخص می‌شود.

ژانگ و همکارانش [۲۷] الگوریتم تعادل بار شبکه‌های نرم‌افزار محور بر اساس بهینه‌سازی کلونی مورچه‌ها (IACO-LB)[3] پیشنهاد کرده است. این مقاله ابتدا اطلاعات کامل سرورها و لینک‌ها را جمع‌آوری کرده و این اطلاعات را به‌عنوان ورودی به الگوریتم کلونی مورچه‌ها داده می‌شود. الگوریتم کلونی مورچه‌ها کم‌بار ترین سرور و مسیر را انتخاب می‌کند. این رویکرد سرعت همگرایی پایدار و قابل‌توجهی دارد. اما عیب این رویکرد این است که هزینه پردازش کنترلر در نظر گرفته نشده است.

همان‌طور که در جدول ۱ مشاهده می‌کنید مروری بر مقالات تعادل بار صفحه داده جمع‌آوری شده است و جدول ۲ مقایسه‌ای بین پارامترهای مقالات است. در جدول ۲ و بقیه جداول مشابه سلول‌هایی که با علامت ✓ پر شده‌اند به این معنا است که مقاله موردنظر آن پارامتر خاص را در نظر گرفته است و سلول‌های خالی به این معنا هستند که مقاله موردنظر آن پارامترها را در نظر نگرفته است. مقالاتی که برای تعادل بار سرور هستند تعداد کمی دارند و مصرف انرژی نادیده گرفته شده است و پارامتر زمان پاسخ برای این مقالات بسیار حائز اهمیت است. مقالاتی که برای تعادل بار لینک هستند کنترلر منتخبشان Ryu بوده .زیرا در این مقالات طبق جدول ۲ دو پارامتر گذردهی و تأخیر اهمیت دارد و این کنترلر برای این دو پارامتر به نسبت بقیه کنترلرها بهتر عمل می‌کند. مقالاتی که برای تعادل بار هم‌زمان سرور و لینک هستند بیشتر در محیط واقعی انجام‌گرفته‌اند و پارامترهای گذردهی و نسبت استفاده از منابع دارای اهمیت هستند.

به‌صورت کلی در مقالات صفحه داده به‌صرفه جویی در مصرف انرژی به نسبت خوبی پرداخته شده است و پارامترهای گذردهی و تأخیر بسیار مهم بوده‌اند و به پارامترهای RMSE و زمان اتمام جریان پرداخته نشده است.

[1] mixed-integer nonlinear programming (MINP)
[2] Energy-efficient and Load-balanced Resource Controller (ELRC)
[3] Improved Ant Colony Optimization-Load Balancing (IACO-LB)

جدول ۱: مروری بر مقالات مربوط به تعادل بار صفحه داده

مرجع	سال	ویژگی	پارامتر ورودی	کنترلر	محیط شبیه‌سازی / محیط واقعی	تعادل بار	NFV	مصرف انرژی
[۱۳]	۲۰۱۵	تقسیم بار ورودی بین سرورها طبق ترتیب	توان عملیاتی، زمان پاسخ	POX	Mininet	سرور ایستا	×	×
[۱۴]	۲۰۱۸	تعادل بار پویا مبتنی بر زمان پاسخ در معماری SDN	زمان پاسخ	FloodLight	-	سرور پویا	×	×
[۱۵]	۲۰۲۱	توزیع حالت بین چندین گره	میزان مصرف CPU	-	محیط واقعی	سرور پویا	×	✓
[۱۶]	۲۰۲۱	الگوریتم تطبیقی توزیع شده و انتها به انتهای مبتنی بر پنجره را برای کنترل اضافه بار	نوع و تعداد پیام ها	-	محیط واقعی	سرور پویا	×	×
[۱۷]	۲۰۲۰	بهینه‌سازی کارایی انرژی مبتنی بر پیش بینی ترافیک	پهنای باند و تأخیر لینک	Ryu	Mininet	لینک	×	✓
[۱۸]	۲۰۲۱	مسیریابی تعادل بار تحت محدودیت‌های کیفیت انتقال (LBRCQT)	پیام Packet-In	-	OMNet ++	لینک	×	×
[۱۹]	۲۰۱۸	مدیریت مسیر متعادل برای بهبود عملکرد شبکه	بار ورودی سوئیچ ها	Ryu	Mininet	لینک	×	✓
[۲۰]	۲۰۲۰	روش استقرار زنجیره تابع سرویس بر اساس تئوری بدهی شبکه	استقرار نسبت موقعیت، متوسط بلند مدت درآمد نسبت به هزینه، ضریب گسترش طول لینک، میانگین تأخیر انتقال، نسبت گره/لینک گلوگاه	-	Matlab	لینک	✓	×
[۲۱]	۲۰۲۳	پروتکل مسیریابی متعادل کننده بار (LBRP)	تأخیر، پهنای باند، توان عملیاتی	Ryu	Mininet	**لینک**	×	×
[۲۲]	۲۰۲۳	تعادل بار ترافیک و به حداقل رساندن مصرف انرژی در شبکه IP/SDN	استفاده از لینک و میزان مصرف انرژی	-	-	**لینک**	✓	×
[۲۳]	۲۰۲۱	مسیریابی انرژی کارا مبتنی بر اینترنت اشیا	تعداد گام ، فاصله و انرژی گره میانی	-	NS-2	**لینک**	×	✓
[۲۴]	۲۰۲۰	برآورده کردن الزامات QoS سرویس‌های مختلف و ایجاد تعادل بار بین سرورهای اینترنت اشیا	QoS(تأخیر و توان عملیاتی) میزان مصرف CPU و حافظه	FloodLight	محیط واقعی	سرور و لینک	×	✓
[۲۵]	۲۰۲۲	تعادل بار سرور و مسیر در SDN با استفاده از تکنیک‌های یادگیری	میزان مصرف CPU و حافظه ، توان عملیاتی، زمان پاسخ	POX	محیط	سرور و	×	×

		ماشین			واقعی	لینک		
[۲۶]	۲۰۲۲	حل بحران تعادل بار و میزان مصرف انرژی سرورها و سوئیچ‌های شبکه IoMT	میزان مصرف CPU و حافظه و هارددیسک	FloodLight	محیط واقعی	سرور و لینک	✓	✓
[۲۷]	۲۰۲۳	الگوریتم تعادل بار شبکه‌های نرم‌افزار محور بر اساس کلونی مورچه‌ها (IACO-LB)	پهنای باند لینک و میزان مصرف CPU و حافظه سرور	-	Matlab	سرور و لینک	×	×

جدول ۲: پارامترهای مقالات مربوط به تعادل بار صفحه داده

مراجع	از دست دادن بسته	گذردهی	زمان پاسخ	تأخیر	نسبت استفاده از منابع	سرریز	تعداد پرش انتقال	RMSE	زمان اتمام جریان	نوع ترافیک
[۱۳]			✓							
[۱۴]			✓							
[۱۵]		✓								
[۱۶]		✓		✓						
[۱۷]		✓		✓			✓			
[۱۸]		✓								
[۱۹]	✓	✓				✓				
[۲۰]				✓	✓					
[۲۱]		✓		✓						✓
[۲۲]		✓		✓						
[۲۳]	✓						✓			
[۲۴]		✓	✓	✓	✓					✓
[۲۵]		✓	✓							
[۲۶]	✓	✓		✓	✓					
[۲۷]		✓			✓					

۲,۳. تعادل بار صفحه کنترل

تعادل بار در صفحه کنترل به سه دسته تعادل بار کنترلر با تمرکز منطقی / توزیع فیزیکی، تعادل بار کنترلر توزیع شده و تعادل بار کنترلر مجازی‌سازی تقسیم کرد.

۱,۲,۳.تعادل بار کنترلر توزیع شده

تعادل بار کنترلر توزیع شده به دودسته سلسله‌مراتبی عمودی و سلسله‌مراتبی افقی تقسیم می‌شود.

الف. تعادل بار کنترلر توزیع شده سلسله‌مراتبی عمودی

در کنترلر سلسله‌مراتبی عمودی هر کنترلر در لایه‌های مختلف قرار می‌گیرد و ارتباط کنترلرهای درون لایه‌ها غیرممکن است و همه کنترلرها توسط یک کنترلر ریشه مدیریت می‌شوند [۲۸]. ایجاز و همکارانش [۲۹] به تعادل بار ترافیکی با استفاده از یک کنترلر شبکه مجازی نرم‌افزار محور (vSDN) و به‌عنوان یک تابع شبکه مجازی دست می‌یابند. مجازی‌سازی توابع شبکه معادله مؤثر تابع شبکه مجازی و خدمات مرتبط در زیرساخت شبکه پویا را ارائه می‌دهد. هنگام استفاده از کنترلر مجازی نرم‌افزار محور به‌عنوان یک تابع شبکه مجازی این امکان هست که

می‌توان توابع شبکه مجازی مشابه بیشتری برای همان اضافه کرد و در صورت افزایش بار ترافیکی می‌توان شبکه‌های مجازی نرم‌افزار محور ثانویه را برای به‌اشتراک‌گذاری بار اضافه کرد. از جایی که تمام منابع مجازی می‌شوند بنابراین می‌توان منابع سخت‌افزاری را بنا به نیاز اختصاص داد/ اضافه کرد. هنگامی که نیاز به یک کنترلر مجازی نرم‌افزار محور ثانویه تعیین می‌شود و یک کپی از کنترلر شبکه مجازی نرم‌افزار محور با پیکربندی مشابه کنترلرشبکه مجازی نرم‌افزار محور اصلی ایجاد می‌شود که به‌دقت کار می‌کند و بار ترافیک را به اشتراک می‌گذارد. هر دو کنترلر شبکه مجازی نرم‌افزار محور به طور مستقل در فضای ابری با شفافیت قرار می‌گیرند و اطمینان می‌دهند که هر مشتری در شبکه با وجود کنترلر ثانویه شبکه مجازی نرم‌افزار محور تازه ایجاد شده آشنا شده است. نتایج تجربی نشان می‌دهد که روش پیشنهادی می‌تواند از روش‌های پیشرفته پیشی گرفته و متعادل‌سازی بار را تا ۱۴ درصد در موارد خاص افزایش می‌دهد.

ب. تعادل بار کنترلر توزیع شده سلسله‌مراتبی افقی

در کنترلر سلسله‌مراتبی افقی همه کنترلرها روی یک‌لایه در کنترلر افقی قرار می‌گیرند؛ بنابراین اگرچه همه کنترلرها دامنه‌های متنوعی پیدا می‌کنند؛ اما همه کنترلرها به‌عنوان کنترلر ریشه در نظر گرفته می‌شوند که با کل وضعیت شبکه آشنا هستند[۳۰].

[۳۱] یک استراتژی متعادل‌سازی بار کنترلر چندگانه نرم‌افزار محور بر اساس زمان پاسخ[1] (SMCLBRT) پیشنهاد شده است. این روش تعادل بار شبکه‌های نرم‌افزار محور بر اساس زمان پاسخ است که با درنظرگرفتن ویژگی‌های متغیر زمان‌های پاسخ بلادرنگ در مقابل بارهای کنترلی با انتخاب آستانه[2] زمان پاسخ مناسب و برخورد هم‌زمان با چندین کنترلر حل می‌کند. یعنی ابتدا زمان پاسخ تمام کنترلرها را به دست می‌آورد و بر اساس زمان پاسخ هر کنترلر برای به‌دست‌آوردن آستانه استفاده کرده است که ابتدا زمان پاسخ را تحت بارهای مختلف آزمایش کرده و سپس آستانه زمان پاسخ را باتوجه‌به ویژگی‌های متنوع آن انتخاب کرده است. با افزایش بار کنترلر از حالت عادی به‌اضافه بارزمان پاسخ سریع‌تر افزایش می‌یابد؛ ولی قبل از این مرحله گرچه زمان پاسخ با افزایش بار، افزایش می‌یابد؛ ولی به‌آرامی تغییر می‌کند. بر اساس همین افزایش ناگهانی و مداوم زمان پاسخ آستانه را انتخاب شده است و سپس زمان پاسخ تمام کنترلرها را با آستانه مناسب مقایسه کرده است و بر اساس این مقایسه سرورها را در دودسته اضافه‌بار و کم‌بار تقسیم می‌کند. حال سوئیچ کنترلری که بیشترین حجم کاری را دارد به کنترلری با کمترین حجم کاری مهاجرت می‌یابد. این طرح می تواند توزیع بار ناهموار را از قبل متعادل کند، و از زمان کمتری برای مقابله با چندین کنترلر اضافه‌بار استفاده کند. از دیدگاه کاربر، این طرح می‌تواند زمان پاسخگویی را کاهش دهد. مزایای این روش این است که می‌تواند تعادل بار چندین کنترلر نرم‌افزار محور را به طور مؤثر و سریع به دست آورد و محدودیت این روش، رویکرد متعادل‌سازی بهتر کنترلرهای بارگذاری شده متعدد هنگام درنظرگرفتن هزینه مهاجرت است.

مختار و همکارانش [۳۲] طرح مهاجرت سوئیچ تعادل بار آستانه چندگانه[3] (MTLB) را برای دستیابی به بار پیوسته در بین کنترلرها پیشنهاد کرده‌اند. این طرح به طور مستقیم وضعیت بار را به طور دقیق متمایز می‌کند و به‌صورت پویا مقدار آستانه را هنگامی که سطح بار افزایش یا کاهش می‌یابد تنظیم می‌کند. بار را به چندین سطح تدریجی طبقه‌بندی می‌کند که نشان‌دهنده مبنایی برای مهاجرت سوئیچ است. سپس یک استراتژی همگام‌سازی اطلاعات بار بر اساس مفهوم آستانه چندسطحی طراحی می‌کند. هنگامی که بار یک کنترلر از این آستانه فراتر رفت یا نزدیک به آن شد، به کنترلرهای دیگر اطلاع می‌دهد تا اطلاعات بار بر اساس وضعیت جدید بروز شود که به طور قابل‌ملاحظه‌ای سربار ناشی از همگام‌سازی اطلاعات بار ناخواسته را کاهش می‌دهد. در نهایت با مطالعه دقیق رفتار سوئیچ مهاجر و کنترلر هدف، روشی برای عملیات مهاجرت ارائه کرده است. درنتیجه تعداد دفعات مهاجرت را کاهش

[1] SDN Multiple Controller Load Balancing strategy based on Response Time(SMCLBRT)

[2] Threshold

[3] Multiple-level Threshold Load Balancing(MTLB)

می‌دهد. نتایج شبیه‌سازی نشان می‌دهد که طرح تعادل بار آستانه چندگانه دارای زمان راه‌اندازی جریان کم، سربار کنترل پایین، زمان پاسخ کم و نرخ توان بالا است. برای تعیین دقیق فواصل و شناسایی سطوح با راندمان بالا لازم است مطالعات بیشتری انجام شود.

شیانگ و همکارانش [۳۳] یک معماری یادگیری تقویتی عمیق[1](DRL) برای شبکه‌های نرم‌افزار محور برای حل مسئله مهاجرت سوئیچ[2](SMP) معرفی کرده‌اند، که در آن حالت شبکه به‌عنوان یک ماتریس دوبعدی رسمیت می‌یابد. یادگیری تقویتی عمیق ماتریس دوبعدی را به‌عنوان ورودی می‌گیرد و تصمیمات مهاجرت را به‌عنوان خروجی تولید می‌کند. برای حل مشکل تخمین بیش از حد و نوسان مدل ناشی از استفاده از Q_Network، یک مدل DDQN[3] طراحی شده است و DDQN با استفاده از مکانیزم بازپخش تجربه آموزش داده می‌شود. پس از مرحله آموزش، مدل DDQN می‌تواند به‌سرعت و بادقت تصمیم بگیرد که چگونه سوئیچ‌ها را بین کنترلرها منتقل کند. نتایج نشان می‌دهد که راه‌حل مبتنی بر DDQN می‌تواند به طور مؤثر اثر تعادل بار را بهبود بخشد و زمان تعادل را کاهش دهد.

شائو و همکارانش [۳۴] یک چارچوب کارآمد موازنه بار مبتنی بر مهاجرت سوئیچ[4](ESMLB) معرفی کرده‌اند. که هدف آن اختصاص سوئیچ‌ها به یک کنترلر کم استفاده به طور مؤثر است. برای انتخاب کنترلر هدف، تکنیک اولویت سفارش با شباهت به یک راه‌حل ایده‌آل[5] در چارچوب استفاده شده است. این چارچوب فرایندهای تصمیم‌گیری انعطاف‌پذیر را برای انتخاب کنترلرهایی با ویژگی‌های منابع مختلف امکان‌پذیر می‌کند. TOPSIS یک روش تجزیه‌وتحلیل تصمیم برای رتبه‌بندی کنترلرها بر اساس معیارهای خاصی مانند میزان مصرف فعلی حافظه، بار CPU، وضعیت پهنای باند کنترلر و تعداد پرش استفاده می‌شود. مزایای این روش مهاجرت سوئیچ کارآمد، تعادل بار بهتر، کاهش اضافه‌بار، کاهش زمان مهاجرت است. محدودیت‌های این روش این است که تقاضای ترافیک در نظر گرفته نشده است، برای محیط با مقیاس کوچک مناسب است.

ژانگ و همکارانش [۳۵] برای چندین کنترلر توزیع شده، یک طرح مهاجرت سوئیچ دو وزنه متعادل‌کننده بار نرم‌افزار محور مبتنی بر پیش‌بینی پیشنهاد کرده‌اند. این طرح بار ترافیک گذشته را به‌عنوان داده‌های تاریخی برای پیش‌بینی بار ترافیک آینده در نظر می‌گیرد. از طریق فناوری پیش‌بینی، زمان اضافه‌بار کنترلر به دست می‌آید، به‌طوری‌که عملیات مهاجرت سوئیچ می‌تواند از قبل انجام شود. همچنین یک الگوریتم اطلاعات بار راه‌اندازی را برای حل پردازش اضافی و سربار ارتباطی صفحه کنترل موردنیاز برای اطلاعات بار فعال دوره‌ای بین کنترلرهای توزیع شده پیشنهاد شده است. باتوجه‌به اطلاعات گذشته، این طرح پیشنهاد می‌کند که مدیریت سوئیچ‌های خاص بین کنترلرها منتقل شود. یک طرح مهاجرت سوئیچ با وزن دوگانه برای درنظرگرفتن وضعیت بار سوئیچ در آینده و جلوگیری از مهاجرت مکرر سوئیچ پیشنهاد شده است که فرکانس مهاجرت سوئیچ را کاهش می‌دهد. آزمایش‌ها ثابت کرده‌اند که این طرح می‌تواند به‌سرعت بار را بین کنترلرها متعادل کند و تعداد مهاجرت سوئیچ‌ها را کاهش دهد.

ظفر و همکارانش [۳۶] یک رویکرد موازنه بار مبتنی بر مهاجرت سوئیچ پویا[6] (DSMLB) را برای جلوگیری از سربار کنترل و توزیع کارآمد ترافیک با درنظرگرفتن دستگاه‌های ناهمگن اینترنت اشیا پیشنهاد کرده‌اند. طرح موازنه بار مبتنی بر مهاجرت سوئیچ پویا پیشنهادی نسبت بار بلادرنگ به میانگین بار هر کنترلر را با توصیف معیارهای بار مختلف و انتخاب کنترلر هدف با بهینه‌سازی بازده مهاجرت با استفاده از منابع کنترلر باقی‌مانده اندازه‌گیری می‌کند. هنگام انتخاب سوئیچ‌های

[1] Deep Reinforcement Learning (DRL)

[2] Switch Migration Problem (SMP)

[3] Double Deep Q-Network (DDQN)

[4] Efficient Switch Migration based Load Balancing(ESMLB)

[5] Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS)

[6] Dynamic Switch Migration-based Load Balancing (DSMLB)

کاندید از کنترلر اضافه‌بار مرتبط، موازنه بار مبتنی بر مهاجرت سوئیچ پویا بازده مهاجرت و درجه متعادل‌کننده بار را برای سریع‌ترین کاهش بار به طور هم‌زمان در نظر می‌گیرد و عملکرد متعادل‌کننده بار را افزایش می‌دهد. نتایج شبیه‌سازی نشان می‌دهد که موازنه بار مبتنی بر مهاجرت سوئیچ پویا از روش‌های مرسوم برای عملکرد متعادل بار کارآمد در یک صفحه کنترل توزیع‌شده با کاهش زمان پاسخ کنترلر تا ۳۸/۶ درصد و هزینه مهاجرت حدود ۴۵/۵ درصد به طور متوسط عملکرد بهتری دارد. علاوه بر این، موازنه بار مبتنی بر مهاجرت سوئیچ پویا همچنین نرخ تعادل بار کنترلر را بهبود می‌بخشد و سربار ارتباط را کاهش می‌دهد.

در جدول ۳ مروری بر مقالات صفحه کنترل آورده شده است. مقالات در تعادل بار کنترلر سلسله‌مراتبی افقی هستند خیلی بیشتر هستند و این مقالات همگی از کنترلر Floodlight یاRyu استفاده کرده‌اند و از محیط شبیه‌سازی Mininet استفاده کرده‌اند و هیچ‌کدام از تکنولوژی مجازی‌سازی توابع شبکه استفاده نکرده و مصرف انرژی را نادیده گرفته‌اند. در جدول ۴ جمع‌بندی مقالات صفحه کنترل را مشاهده می‌کنید و در این مقالات پارامتر زمان پاسخ بسیار حائز اهمیت است.

۳،۳. تعادل بار هم‌زمان صفحه داده و صفحه کنترل

در شبکه‌های امروزی با حجم بالای ترافیک مواجه هستیم. این ترافیک باعث ازدحام در صفحه داده می‌شود. در تعادل بار صفحه داده مشکل این است که یک کنترلر واحد را در نظر گرفته‌اند. درصورتی‌که این تک کنترلر بودن خود باعث ایجاد گلوگاه می‌شود. از طرف دیگر شبکه‌ها روزبه‌روز در حال گسترش هستند. پس با یک کنترلر نمی‌توان کل شبکه را مدیریت کرد؛ چون مقیاس‌پذیری و دردسترس‌بودن را کاهش می‌دهد. در نتیجه از معماری کنترلر توزیع شده استفاده شده است. شبکه‌هایی که در صفحه کنترلرشان چندین کنترلر وجود دارد نیز ممکن است تعادل بار بین کنترلرها استفاده می‌شده. مهاجرت سوئیچ باعث تعادل بار در لایه کنترل می‌شود؛ اما باعث عدم تعادل بار در سرورها می‌شود. به همین دلیل مقالاتی نوشته شده است که تلاش کرده‌اند تعادل بار صفحه داده و صفحه کنترلر را به‌صورت هم‌زمان انجام دهند. تعادل بار هم‌زمان صفحه داده و صفحه کنترل به دودسته تقسیم می‌شود: تعادل بار سرور و کنترلر و تعادل بار لینک و کنترلر.

جدول ۳: مروری بر مقالات مربوط به تعادل بار صفحه کنترل

پذیرفته شده در مجله محاسبات نرم

مرجع	سال	ویژگی	پارامتر	کنترلر	محیط شبیه‌سازی / محیط واقعی	تعادل بار	NFV	مصرف انرژی
[۲۹]	۲۰۱۹	تعادل بار ترافیک SDN با استفاده از با استفاده از یک کنترلر مجازی SDN به‌عنوان یک VNF	تأخیر، پهنای باند، توان عملیاتی سیستم	OpenDayLight	Mininet	کنترلر توزیع شده سلسله‌مراتبی عمودی	✓	×
[۳۱]	۲۰۱۸	متعادل‌سازی بار کنترلر چندگانه بر اساس زمان پاسخ (SMCLBRT)	زمان پاسخ	FloodLight	Mininet	کنترلر توزیع شده سلسله‌مراتبی افقی	×	×
[۳۲]	۲۰۲۲	طرح مهاجرت سوئیچ تعادل بار آستانه چندگانه(MTLB)	Packet –in ورودی به کنترلر	FloodLight	Mininet	کنترلر توزیع شده سلسله‌مراتبی افقی	×	×
[۳۳]	۲۰۲۲	یادگیری تقویتی عمیق برای حل مسئله مهاجرت سوئیچ	Packet –in ورودی به کنترلر	Ryu	Mininet	کنترلر توزیع شده سلسله‌مراتبی افقی	×	×
[۳۴]	۲۰۱۹	چارچوب کارآمد موازنه بار مبتنی (ESMLB) سوئیچ بر مهاجرت	میزان مصرف فعلی حافظه، بار CPU، وضعیت پهنای باند کنترلر و تعداد پرش	FloodLight	Mininet	کنترلر توزیع شده سلسله‌مراتبی افقی	×	×
[۳۵]	۲۰۲۲	طرح مهاجرت سوئیچ دو وزنه متعادل‌کننده بار SDN مبتنی بر پیش‌بینی	زمان پاسخ	Ryu	Mininet	کنترلر توزیع شده سلسله‌مراتبی افقی	×	×
[۳۶]	۲۰۲۲	رویکرد موازنه بار مبتنی بر مهاجرت سوئیچ پویا (DSMLB)	استفاده از حافظه، استفاده از CPU و پهنای باند کنترلر	Ryu	Mininet	کنترلر توزیع شده سلسله‌مراتبی افقی	×	×

جدول ۴: پارامترهای مقالات مربوط به تعادل بار صفحه داده

مراجع	از دست دادن بسته	گذردهی	زمان پاسخ	تأخیر	نسبت استفاده از منابع	سرریز	تعداد پرش انتقال	RMSE	زمان اتمام جریان	نوع ترافیک
[۲۹]		✓	✓	✓						
[۳۱]			✓							
[۳۲]		✓	✓			✓				
[۳۳]			✓							
[۳۴]				✓	✓		✓			
[۳۵]			✓							
[۳۶]			✓			✓				

۱،۳،۳. تعادل بار سرور و کنترلر

الف. تعادل بار سرور پویا و کنترلر با تمرکز منطقی /توزیع فیزیکی

گاوو و همکارانش [۳۷] یک روش همگام‌سازی حالت کنترلر به نام همگام‌سازی مبتنی بر واریانس بار (LVS)[1] را پیشنهاد کرده‌اند تا عملکرد تعادل بار را در شبکه SDN چند دامنه‌ای چند کنترلر افزایش دهد.ابتدا مشکل حلقه ارسال به دلیل اطلاعات متناقض سرور با کمترین بارگذاری بین کنترلرها ایجاد می‌شود که می‌تواند منجر به از دست دادن بسته ها و اختلالات قابل توجهی در ارتباط شود. سپس مشکل سربار همگام‌سازی کنترلر ناشی از همگام‌سازی های مکرر را حل می‌کند.این دو مشکل را با دو طرح خاص مبتنی بر LVS حل می‌کند: (۱) همگام‌سازی تغییرات سرور با حداقل بار[2] (LSVS) برای حل مشکل حلقه ارسال. (۲)همگام‌سازی تنوع دامنه با حداقل بار[3] (LDVS) برای حل مشکل سربار همگام‌سازی کنترلر. در مقایسه با روش‌های مبتنی بر همگام‌سازی دوره‌ای ، روش‌های مبتنی بر LVS همگام‌سازی حالت واقعی را در بین کنترلرها انجام می‌دهند درحالی‌که بار یک سرور از یک آستانه مشخص فراتر می‌رود که به طور قابل‌توجهی سربار همگام‌سازی کنترلرها را به حداقل می‌رساند. نتایج تجربی ثابت کرده‌اند که LVS در مقایسه با روش‌های موجود، عملکرد تعادل بار مناسب و ارسال بدون حلقه را با سربار همگام‌سازی کمتر به دست می‌آورد. بااین‌حال، دو روش پیشنهادی مبتنی بر LVS در یک بستر آزمایشی واقعی ارزیابی نشده‌اند. همچنین مصرف انرژی و تأخیر روش مورد ارزیابی قرار نگرفته است.

ب. تعادل بار سرور پویا و کنترلر توزیع شده

- تعادل بار سرور پویا و کنترلر توزیع شده سلسله‌مراتبی عمودی

شائو و همکارانش [۳۸] یک روش قراردادن کنترلر قابل اعتماد بر اساس بهینه‌سازی تأخیر و بار پیشنهاد کرده‌اند. روش بهینه‌سازی چند هدفه تأخیر بین کنترلر، تأخیر کنترلر به سوئیچ و متعادل‌سازی بار را در نظر می‌گیرد. ابتدا هدف بهینه‌سازی معرفی می‌شود و با استفاده از الگوریتم LOCP[4] یک مدل چیدمان کنترلر قابل اعتماد بر اساس تأخیر و متعادل‌سازی بار برای هدف بهینه‌سازی ساخته می‌شود. برای تخصیص منطقی منابع محاسباتی،الگوریتم HACA[5] یک روش تخصیص منبع بر اساس محدودیت‌های تأخیر کار و قابلیت اطمینان است و یک روش محدودیت چند هدفه مبتنی بر تأخیر انتقال و قابلیت اطمینان نیز در نظر گرفته شده است. ابتدا هدف محدودیت معرفی می‌شود و یک مدل تخصیص منبع بر اساس محدودیت‌های تأخیر کار و قابلیت اطمینان برای هدف محدودیت ساخته می‌شود. نتایج تجربی نشان می‌دهد که الگوریتم LOCP در مقایسه با سایر الگوریتم ها، تعادل بار کنترل کننده را ۱۸/۳۶٪ افزایش می‌دهد و در عین حال تأخیر انتشار کمتر و تأخیر در صف را تضمین می‌کند. الگوریتم HACA می‌تواند به طور مؤثر مشکل تخصیص بار منبع کار را حل کند و به طور مؤثر زمان پاسخ منبع کاربر و همچنین متوسط تأخیر تکمیل را کاهش دهد. معایب این روش این است که (۱) هنگام درنظرگرفتن معیارهای بهینه‌سازی برای قراردادن کنترلر، تعداد معیارهای بهینه‌سازی در نظر گرفته شده به طور جامع کم است. (۲) در درنظرگرفتن مشکل قراردادن کنترلر قابل‌اعتماد، خرابی اجزای شبکه را می‌توان بیش از خرابی‌های تک لینک در نظر گرفت.

- تعادل بار سرور پویا و کنترلر توزیع شده سلسله‌مراتبی افقی

در پژوهش دیگر یک الگوریتم مهاجرت سوئیچ متعادل‌کننده بار آگاه از QoS پویا (LBSMT) پیشنهاد شده است. هنگامی که بین سوئیچ‌ها و دامنه‌های مختلف ارتباط درونی وجود دارد. مهاجرت سوئیچ‌ها صورت می‌گیرد و بار روی سرورها نامتعادل است. عدم تعادل بار باعث افزایش زمان پاسخ و کاهش توان عملیاتی می‌شود. هر سوئیچ در معماری کنترلر توزیع شده توسط یک کنترلر در دامنه محلی مدیریت می‌شود. کنترلرها جریان‌های

[1] Load Variance-based Synchronization (LVS)

[2] Least loaded server variation synchronization (LSVS)

[3] Least loaded domain variation synchronization (LDVS)

[4] multi-controller placement

[5] heuristic ant colony algorithm(HACA)

ترافیک را مدیریت می‌کنند، و با افزایش ترافیک اینترنت، جریان بین کنترلر و سوئیچ overload می‌شود، درحالی‌که تعداد کمی از کنترلرها underload می‌شوند. به دلیل کاهش توان و زمان پاسخ طولانی، چنین عدم تعادلی بر اثربخشی شبکه‌های نرم‌افزار محور تأثیر می‌گذارد. مهاجرت سوئیچ معمولاً برای تغییر تخصیص بارهای کنترلر با جابه‌جایی سوئیچ از یک کنترلر overload به یک کنترلر underload استفاده می‌شود. در شبکه چندین دامنه داریم که هر دامنه از یک کنترلر و چند سوئیچ تشکیل شده است. به بار تمام دامنه‌ها مقادیر آستانه در KB/sec اختصاص داده می‌شود و اگر تعداد بازدیدها در یک دامنه از یک سوئیچ خاص از مقدار آستانه عبور کرده باشد، بار به سوئیچ دیگری از آن دامنه منتقل می‌شود و اگر یک دامنه دچار اضافه‌بار شود؛ یعنی تمام سوئیچ‌های دامنه از مقدار آستانه عبور کرده باشند. در این صورت مهاجرت سوئیچ از یک دامنه به دامنه دیگر را داریم. هدف این طرح نشان‌دادن مهاجرت سوئیچ برای یافتن هزینه مهاجرت و نرخ تعادل بار است. این محاسبات به دو صورت انجام می‌شود (۱) درون دامنه (۲) بین دامنه. رویکرد پیشنهادی زمان پاسخگویی را ۲ برابر نسبت به LBBSRT , round robin بهبود بخشیده است. توان عملیاتی را ۰/۴ برابر نسبت به round robin ، ۰/۵برابر از LBBSRT بهبود بخشیده است. استفاده از CPU را ۰/۶ برابر از round robin ، LBBSRT بهبود بخشیده است. استفاده از حافظه را ۲ /۰ برابر از round robin ، LBBSRT بهبود بخشیده است. اما در این طرح سرورهای مختلف در دامنه‌ها را به‌صورت همگن در نظر گرفته است[۳۹].

۲,۳,۳. تعادل بار لینک و کنترلر

وانگ و همکارانش [۴۰] مسیریابی تعادل بار را برای مشکل لینک‌ها و کنترلرها[1] (LBR-LC) در یک شبکه‌های مبتنی بر نرم‌افزار فرموله شده است و NP-hard آن را ثابت کرده‌اند. یک الگوریتم مبتنی بر گرد کردن[2] برای حل این مشکل پیشنهاد شده است و عملکرد تقریبی نیز تحلیل می‌شود. علاوه بر این، ما در مورد مکانیسم کارآمد برای نگهداری وضعیت شبکه در میان کنترلرهای توزیع شده بحث شده است. نتایج شبیه‌سازی گسترده نشان می‌دهد که الگوریتم پیشنهادی می‌تواند حداکثر زمان پاسخ کنترلر را تا ۷۰ درصد در مقایسه با راه‌حل قبلی کاهش دهد، درحالی‌که تنها حداکثر بار لینک را تا ۳ درصد افزایش می‌دهد. مزایای این روش تعادل بار بهتر، کیفیت سرویس بهتر است.محدودیت این روش این است که بار فراتر از آستانه افزایش می‌یابد. همان‌طور که در جدول ۵ و جدول ۶ مشاهده می کنید جمع بندی مقالات تعادل بار همزمان صفحه داده و صفحه کنترل آورده شده است. این مقالات مصرف انرژی را نادیده گرفته‌اند و همگی از کنترلرهای POX و NOX و Ryu استفاده کرده‌اند که همگی این کنترلرها معماری متمرکز یا توزیع شده دارند.

[1] Load-Balancing Routing for both Links and Controllers (LBR-LC)
[2] rounding-based

پذیرفته شده در مجله محاسبات نرم

پذیرفته شده

جدول ۵: مروری بر مقالات تعادل بار صفحه داده و صفحه کنترل

مرجع	سال	ویژگی	پارامتر	کنترلر	محیط شبیه‌سازی / محیط واقعی	تعادل بار	NFV	مصرف انرژی
[۳۷]	۲۰۱۹	روش همگام‌سازی حالت کنترل‌کننده به نام همگام‌سازی مبتنی بر واریانس بار (LVS)	بار سرور و کنترلر	POX	Mininet	تعادل بار سرور پویا و کنترلر با تمرکز منطقی متمرکز /توزیع فیزیکی	×	×
[۳۸]	۲۰۱۸	روش قراردادن کنترلر قابل‌اعتماد بر اساس بهینه‌سازی تأخیر و بار	تأخیر	Ryu	محیط واقعی	تعادل بار سرور پویا و کنترلر توزیع شده سلسله‌مراتبی	×	×
[۳۹]	۲۰۲۲	مهاجرت سوئیچ متعادل‌کننده بار آگاه از QoS پویا (LBSMT)	ترافیک موجود در دامنه کنترلر	-	محیط واقعی	تعادل بار سرور پویا و کنترلر توزیع شده افقی	✓	×
[۴۰]	۲۰۱۸	مسیریابی تعادل بار را برای مشکل لینک‌ها و کنترلرها (LBR-LC)	زمان پاسخ کنترلر	NOX	Mininet	تعادل بار لینک و کنترلر	×	×

جدول ۶: پارامترهای مقالات تعادل بار صفحه داده و صفحه کنترل

مرجع	از دست دادن بسته	گذردهی	زمان پاسخ	تأخیر	نسبت استفاده از منابع	سرریز	تعداد پرش انتقال	RMSE	زمان اتمام جریان	نوع ترافیک
[۳۷]	✓							✓		
[۳۸]		✓		✓						
[۳۹]		✓	✓		✓					
[۴۰]			✓							

۴.تعادل بار شبکه‌های نرم‌افزار محور مبتنی بر بهینه‌سازی

در تعادل بار شبکه‌های نرم‌افزار محور به دنبال حل مسائل تعادل بار سرور و لینک و کنترلر هستیم. در برخی از مقالات سعی شده

است که این مسائل را با کمک بهینه‌سازی حل کنند. زیرا بهینه‌سازی فرایندی است که به کمک آن بهترین پاسخ برای یک مسئله به دست می‌آید. الگوریتم‌های بهینه‌سازی به دودسته کلی تقسیم می‌شوند: ۱. الگوریتم‌های هوش مصنوعی ۲. الگوریتم‌های

ریاضی. در این مقاله تأکید ما روی الگوریتم‌های هوش مصنوعی در تعادل بار شبکه‌های نرم‌افزار محور است.

۱,۴. الگوریتم‌های هوش مصنوعی در تعادل بار شبکه‌های نرم‌افزار محور

الگوریتم‌های هوش مصنوعی می‌توانند در زمان کوتاه و قابل‌قبولی پاسخ نزدیک به بهینه‌ای برای مسائل پیچیده پیدا کنند. به همین دلیل از الگوریتم‌های هوش مصنوعی بری حل مسئله تعادل بار در شبکه‌های نرم‌افزار محور استفاده می‌شود. چون مسئله تعادل بار خود NP-Hard است و با استفاده از الگوریتم‌های هوش مصنوعی می‌توان به پاسخ قابل‌قبولی دست‌یافت. باتوجه‌به توضیحات داده شده الگوریتم‌های هوش مصنوعی به دودسته، یادگیری ماشین و فراابتکاری تقسیم می‌شود [۵].

۱,۱,۴.یادگیری ماشین

الگوریتم های یادگیری ماشین ، برنامه‌هایی هستند که بر اساس داده‌های موجود در محیط و تجربیات گذشته خودآموزش داده می‌شوند [۵]. سان و همکارانش [۴۱] یک طرح متعادل‌کننده کنترلر پویا با استفاده از تکنیک یادگیری ماشین پیشنهاد شده است. هدف الگوریتم یادگیری ماشین در این مقاله، مقابله مهاجرت سوئیچ است به این صورت که ورودی این الگوریتم صفحه کنترل است و خروجی آن تصمیم‌گیری درمورد مهاجرت سوئیچ است. این الگوریتم در دو مرحله کار می‌کند.در مرحله که یاد می‌گیرد چگونه سوئیچ‌ها را منتقل کند ودر مرحله دوم در مورد مهاجرت سوئیچ‌ها تصمیم‌گیری می‌کند

الف. یادگیری تقویتی

یک تکنیک یادگیری ماشین است که از طریق تعامل با محیط یاد می‌گیرد تا به هدف مشخص برسد [۴۲]. در مقاله [۲۵] از روش یادگیری تقویتی برای تعادل بار بین لینک‌ها استفاده می‌کند. به این صورت که ابتدا لینک‌ها در جدول یادگیری قرار می‌گیرند. سپس برای انتخاب بهترین لینک برای جریان جدید، در جدول بر اساس پارامتر تأخیر انتخاب می‌شود. یعنی مسیری که تأخیر کمتری داشته باشد در اولویت بالاتری برای انتخاب قرار دارد.

فیلالی و همکارانش [۴۳] الگوریتم یادگیری تقویتی برای حل مشکل تعادل بار بین کنترلرها و هزینه‌های عملیات مهاجرت که بر اساس زمان پاسخ کنترلر است پیشنهاد کرده‌اند. در الگوریتم پیشنهادی دو عمل برنده و یک عمل بازنده در نظر گرفته شده است. عمل برنده، عملی است که پاداش دارد و عمل بازنده، عملی است که جریمه دارد. عمل برنده اول تعادل بارکم و عدم مهاجرت سوئیچ است و عمل برنده دوم تعادل بارکم و انجام مهاجرت سوئیچ است. اولویت عمل برنده اول بیشتر ازعمل برنده دوم است و عمل بازنده تعادل بار زیاد و انجام مهاجرت زیاد است که در این صورت تعادل با کنترلر در شبکه نرم‌افزار محور را نداریم؛ اما هزینه مهاجرت بسیار زیادی داریم. این الگوریتم بر اساس این اعمال تصمیماتی را می‌گیرد. بر اساس این تصمیم یا پاداش می‌گیرد، یا جریمه می‌شود. بر اساس همین پاداش‌ها و جریمه‌ها تجربه کسب می‌کند.

ب. یادگیری عمیق

یادگیری عمیق زیرشاخه ای از یادگیری ماشین است که هدف آن ابداع ماشین هایی است که حتی در موقعیت های غیر منتظره با محیط تعامل داشته باشد[۵].در مقاله [۳۳] از یادگیری عمیق برای حل مسئله مهاجرت سوئیچ استفاده شده است. به این

صورت که حالت شبکه به‌عنوان ورودی بـه الگـوریتم یـادگیری عمیق داده می‌شود و تصمیمات مهاجرت به‌عنوان خروجی تولید می‌شود.

ج.شبکه عصبی مصنوعی

شبکه عصبی مصنوعی عمدتا از مغز انسـان بـرای حـل مسـائل پیچیده الهام می‌گیرد[۴۴]. تعادل بار شبکه‌های نرم‌افـزار محـور نیز از همین ویژگی شبکه‌های عصبی برای حل مسـائل اسـتفاده می‌کند. محبوب تریم تکنیـک یـادگیری عمیـق، شـبکه عصـبی مصنوعی است[۵]. در مقاله [۲۵] بـرای تعـادل بـار سـرورها در شبکه‌های نرم‌افزار محور از الگـوریتم شـبکه عصـبی مصـنوعی استفاده شـده اسـت. بـه ایـن صـورت کـه داده هـای سـرورها جمـع‌آوری مـی‌شـود. سـپس الگـوریتم‌هـای , Random RoundRobin, Weighted Round Robin , LeastConnection روی متعادل کننده بار بری به دست آوردن پارامترهای سـرور و برچسب به سرور اختصاص داده می‌شود. سپس همه پارامترها به‌عنوان ورودی به الگوریتم داده می‌شود تا الگویی را بر اسـاس داده ها یـاد بگیـرد و از طریـق آن مـدل بسـازد. در مقالـه [۴۵] تعادل بار هوشمند را پیشنهاد می‌کند که شامل تعادل بـار لینـک، تعادل‌سازی بار سرور و طبقه‌بندی ترافیـک اسـت. بـرای طبقـه بندی ترافیک از شبکه عصبی مصنوعی استفاده مـی‌کند.بـه ایـن صورت که برای جدا کردن جریان هـای فیـل در شـبکه از ایـن الگوریتم استفاده می‌کند.

۲،۱،۴. الگوریتم‌های فراابتکاری

الگوریتم‌های فراابتکاری برای حل مسائل بهینه‌سازی پیچیده که در زمان معقول قابل‌حل نیستند مناسب است [۴۶].به همین دلیل برای حل مسائل تعادل بار شبکه‌های نرم‌افزار محور که پیچیده هستند مناسب هستند. الگوریتم فراابتکاری انواع مختلفی دارد:

- الگوریتم ژنتیک[1](GA)

الگوریتم ژنتیک یک تکنیک برنامه‌نویسـی اسـت کـه مـی‌توانـد فضای راه‌حل را با استفاده از مفهوم برگرفته از ژنتیک طبیعـی و نظریه تکامل کاوش کند. [۵،۴۷]. سـانر از الگـوریتم ژنتیـک در [۴۸] برای یافتن بهترین موقعیت کنترلر برای به حداقل رسـاندن تأخیر سوئیچ برای کنترلر اسـتفاده مـی‌کنـد.در مقالـه[۲۲] بـرای صرفه جویی در مصرف انرژی از الگوریتم ژنتیک اسـتفاده شـده است. به این صورت کـه پیکربنـدی شـبکه در یـک کرومـوزوم نگاشت می‌شود و محدودیت ها شامل ظرفیت لینک‌ها و ترافیک موجود است . هدف به حداقل رساندن تعداد لینک هـای فعـال برای اطمینان از صرفه جوییی در انرژی است.

- الگوریتم کلونی مورچه‌ها[2] (ACO)

این الگوریتم به شبیه‌سازی رفتار مورچـه‌هـا بـرای حـل مسـائل می‌پردازد [۴۹] در مقالـه[۲۷] بـرای تعـادل بـار سـرور و لینـک استفاده می‌شود. به این صورت که اطلاعات سرورها و لینک‌هـا به این الگوریتم داده می‌شود. سپس سروری با کمترین میزان بار انتخاب می‌شود و برای یافتن بهترین مسیر توسط این الگوریتم ، مورچه هایی که کوتاهترین مسیر را دارند فرومون های بیشـتری را آزاد می کنند و در نهایت همه مورچه‌ها به آن مسیر می رونـد و به این ترتیـب کوتـاهترین مسـیر بـه سـرور مـوردنظر یافـت می‌شود.در مقاله [۳۸] مسئله تخصـیص منبـع را بـه‌عنـوان یـک مسئله بهینه‌سازی چند هدفه مدل سازی کرده اسـت. مشـکل بـا استفاده از یک الگوریتم کلونی مورچه اکتشافی حل شده اسـت. ابتدا تعیین کـرده اسـت کـه شـبکه دارای m سـرور بـا n نـوع درخواست محاسباتی و مربوط به n نوع واحد محاسباتی اسـت. تقاضای هـر نـوع واحـد محاسـباتی بایـد بـا گـرم بـودن یـک درخواست جمله تعیین شود و منابع n نوع واحد محاسـباتی بـا استفاده از یک الگوریتم کلونی مورچه بهبودیافته تخصـیص داده می‌شود.

- الگوریتم بهینه‌سازی ازدحام ذرات[3](PSO):

این الگوریتم رفتارهای ازدحام پرنده را شبیه‌سازی می‌کند [۵۰]. در مقاله [۵۱] با درنظرگـرفتن تـأخیر بـین کنترلرهـا و ظرفیـت کنترلـر مشـکل قرارگیـری کنترلـر را بـا اسـتفاده از الگـوریتم بهینه‌سازی ازدحام ذرات را برای حل مسئله پیشنهاد کرده است.

[1] Genetic Algorithm(GA)

[2] Ant Colony Algorithm(ACO)

[3] Particle Swarm Optimization(PSO)

این الگوریتم با تکرار چندین ذره به‌صورت موازی راه‌حل بهینه را جستجو می‌کند و قرارگیری کنترلر به تأخیر کم تعیین می‌شود. در مقاله [۵۲] از الگوریتم ازدحام ذرات برای خوشه‌بندی دامنه کنترلرهای شبکه‌های نرم‌افزار محور استفاده می‌کند. زیرا بخش‌بندی نامعقول دامنه کنترلرهای متفاوت خود باعث عدم تعادل بار کنترلرها می‌شود.

- استفاده ترکیبی از الگوریتم ژنتیک و الگوریتم ازدحام ذرات: در مقاله [۵۳] یک مسئله قرارگیری کنترلر توزیع شده و راه حل برای کاهش تأخیر سوئیچ به کنترلر،تأخیر کنترلر به کنترلر و عدم تعادل بار کنترلر در شبکه‌های نرم‌افزار محور پیشنهاد کرده است. برای حل این مشکلات از الگوریتم ژنتیک چند هدفه استفاده کرده است. اما مشکل این الگوریتم این است که زمان همگرایی طولانی دارد. در نتیجه از الگوریتم ازدحام ذرات برای کاهش زمان همگرایی استفاده کرده است. چون الگوریتم ازدحام ذرات با دادن بهترین موقعیت مکانی برای هدایت هر ذره می‌تواند راه حل بهینه را در زمان همگرایی کوتاه بیابد.

کبیری و همکارانش [۵۴] الگوریتمی برای استفاده از کنترلر و آستانه متغیر با مهاجرت کمتر و بهبود شبکه با کمک الگوریتم ژنتیک و ازدحام ذرات به‌منظور کاهش هزینه مهاجرت و زمان پاسخ پیشنهاد کرده است. در روش پیشنهادی ابتدا با استفاده از یک آستانه متعادل بودن یا نامتعادل بودن کنترلر تشخیص داده می‌شود. سپس از الگوریتم ازدحام ذرات و ژنتیک برای مهاجرت سوئیچ استفاده می‌کند. به این صورت که ابتدا با کمک الگوریتم ژنتیک کنترلرهای هدف، سوئیچ‌های مهاجر و تابع هزینه جمع‌آوری می‌شود. این داده‌ها به‌عنوان ورودی به الگوریتم ازدحام ذرات داده می‌شوند. سپس الگوریتم ازدحام ذرات بهترین و بهینه‌ترین راه‌حل را برای انتقال سوئیچ به کنترلر مناسب پیدا می‌کند.

در جدول ۷ خلاصه‌ای از مقالات تعادل بار شبکه‌های نرم‌افزار محور مبتنی بر بهینه‌سازی آورده شده است. همان‌طور که مشاهده می‌کنید در سال‌های اخیر از روش‌های بهینه‌سازی هوش مصنوعی بیشتر در تعادل بار شبکه‌های نرم‌افزار محور استفاده شده است. علاوه بر این برای عملیات تعادل بار کنترلرها در صفحه کنترل و عملیات مهاجرت سوئیچ از ترکیب دو الگوریتم استفاده کرده‌اند. در مقاله [۲۵] به‌صورت کاملاً جداگانه از دو الگوریتم یادگیری تقویتی و شبکه عصبی مصنوعی استفاده کرده است. اما در مقالات [۵۳،۵۴] از ترکیب دو الگوریتم ژنتیک و ازدحام ذرات برای تعادل بار کنترلر و مهاجرت سوئیچ استفاده کرده‌اند.

مراجع	سال	الگوریتم بهینه‌سازی	توضیحات
[۴۱]	۲۰۲۰	یادگیری ماشین	تعادل بار کنترلر
[۲۵]	۲۰۲۲	یادگیری تقویتی	تعادل بار لینک
[۴۳]	۲۰۲۰	یادگیری تقویتی	تعادل بار کنترلر
[۳۳]	۲۰۲۲	یادگیری عمیق	مهاجرت سوئیچ
[۲۵]	۲۰۲۲	شبکه عصبی مصنوعی	تعادل بار سرور
[۴۵]	۲۰۲۱	شبکه عصبی مصنوعی	طبقه بندی ترافیک
[۴۸]	۲۰۱۶	ژنتیک	تعادل بار کنترلر
[۲۲]	۲۰۲۳	ژنتیک	تعادل بار لینک

پذیرفته شده در مجله محاسبات نرم

[۳۸]	۲۰۲۲	کلونی مورچه‌ها	تعادل بار سرور
[۲۷]	۲۰۲۳	کلونی مورچه‌ها	تعادل بار سرور و لینک
[۵۱]	۲۰۱۵	بهینه‌سازی ازدحام ذرات	تعادل بار کنترلر
[۵۳]	۲۰۱۹	بهینه‌سازی ازدحام ذرات	خوشه بندی کنترلر
[۵۳]	۲۰۱۷	ژنتیک و بهینه‌سازی ازدحام ذرات	تعادل بار کنترلر
[۵۴]	۲۰۲۲	ژنتیک و بهینه‌سازی ازدحام ذرات	تعادل بار کنترلر

جدول ۷ : مروری بر مقالات تعادل بار شبکه‌های نرم‌افزار محور مبتنی بر بهینه‌سازی

۵.پیش‌بینی در تعادل بار شبکه‌های نرم‌افزار محور

در تعادل بار شبکه‌های نرم‌افزار محور از روش‌های پیش‌بینی، برای پیش‌بینی بار کنترلر و بار سرور و ترافیک استفاده می‌شود. در نتیجه باعث ایجاد تعادل بار می‌شود و هم در مصرف انرژی صرفه‌جویی می‌شود. روش‌های متفاوتی برای پیش‌بینی وجود دارد که در تعادل بار شبکه‌های نرم‌افزار محور از روش‌های پیش‌بینی‌های مبتنی بر سری‌های زمانی بیشتر استفاده می‌شود. زیرا پیش‌بینی سری زمانی برای مدیریت منابع یا متعادل‌سازی بار، یک معیار خاص به‌صورت دوره‌ای در فواصل زمانی ثابت نمونه‌برداری می‌شود. نتیجه یک‌سری زمانی خواهد بود که شامل دنباله‌ای از آخرین مشاهدات است. روش‌های سری زمانی این دنباله را برای پیش‌بینی مقادیر آینده استفاده می‌کنند. روش‌های پیش‌بینی سری مانند: میانگین متحرک یکپارچه اتو رگرسیون (ARIMA) ، حداقل میانگین مربعات نرمال شده [1](NLMS)و حافظه بلندمدت کوتاه‌مدت [2] (LSTM).

- **حداقل میانگین مربعات نرمال شده**: پیش‌بینی حداقل میانگین مربع نرمال شده است که بر اساس حجم کاری گذشته، آینده را پیش‌بینی می‌کند [۵۵].در مقاله [۲۴] برای ایجاد تعادل بار سرورها ابتدا ترافیک را به سه دسته تقسیم می‌کند. سپس بر اساس حداقل میانگین مربعات نرمال شده بار سرورها را پیش‌بینی کرده است. بر اساس این پیش‌بینی ترافیک را بین سرورها تقسیم می‌کند. در نتیجه تعادل بار بین سرورها ایجاد می‌شود و سروری دچار اضافه‌بار نمی‌شود. در مقاله [۲۶] برای پیش‌بینی میزان بار سرورها از حداقل میانگین مربع نرمال شده استفاده شده است تا در مراحل بعد با استفاده از این پیش‌بینی بار را بین سرورها متعادل کند و میزان مصرف انرژی را کاهش دهد. بدین صورت که با پیش‌بینی میزان بار سرورها، سرورهای با بارکم خاموش می‌شوند و مصرف انرژی کاهش می‌یابد. در صورتی ایجاد اضافه‌بار برای ایجاد تعادل بار سرورهایی اضافه می‌شود.
- **میانگین متحرک یکپارچه اتو رگرسیون**: در مدل میانگین تحرک یکپارچه اتو رگرسیون مقادیر آینده یک متغیر قرار است ترکیبی خطی از حجم کاری گذشته و خطاهای تصادفی باشد [۵۶]. در مقاله [۴] از میانگین متحرک یکپارچه اتو رگرسیون برای پیش‌بینی میزان بار سوئیچ و جلوگیری از اضافه‌بار کنترلر استفاده شده است. به این صورت که یک الگوریتم زمان‌بندی مهاجرت سوئیچ را برای انتقال برخی از سوئیچ‌ها از کنترلر اضافه‌بار به کنترلر کم‌بار استفاده می‌کند.
- **حافظه بلندمدت کوتاه‌مدت**: یک فرم شبکه عصبی بازگشتی است و توانایی قدرتمندی برای به‌خاطر سپردن پیشینه به‌منظور پیش‌بینی دارد [۵۷]. در مقاله [۱۷] از حافظه بلندمدت کوتاه‌مدت برای پیش‌بینی ترافیک برای ایجاد تعادل بار در لینک‌ها و بهینه‌سازی مصرف انرژی شبکه استفاده می‌کند. به این صورت که با پیش‌بینی ترافیک دستگاه‌های با بارکم خاموش می‌شوند تا مصرف انرژی کاهش یابد و در صورت ایجاد اضافه‌بار در آینده

[1] Normalised Least Connection Mean Squares

[2] Long Short-Term Memory

دستگاه‌هایی برای ایجاد تعادل بار روشن می‌شود. در مقاله [۳۵] حافظه بلندمدت کوتاه‌مدت برای پیش‌بینی ترافیک به‌منظور متعادل‌کردن کنترلرها استفاده می‌کند. به این صورت که اگر امکان اضافه‌بار در آینده وجود داشته باشد عملیات تعادل بار انجام شود.

در مقاله [۴۳] از دو روش میانگین متحرک یکپارچه اتو رگرسیون و حافظه بلندمدت کوتاه‌مدت برای پیش‌بینی بار کنترلر به‌منظور جلوگیری از عدم تعادل بار استفاده شده است. سپس یک مقایسه بین دو رویکرد انجام شده است و نشان داده است که دقت حافظه بلندمدت کوتاه‌مدت از میانگین متحرک یکپارچه اتو رگرسیون بیشتر است.

همان‌طور که در جدول ۸ مشاهده می‌کنید خلاصه‌ای از مقالات پیش‌بینی در تعادل بار شبکه‌های نرم‌افزار محور آورده شده است.در سال های اخیر از روش های پیش بینی برای ایجاد تعادل بار در شبکه‌های نرم‌افزار محور بسیار استفاده شده است و همان‌طور که مشاهده می کنید از پیش بینی در تمام زمینه های تعادل بار شبکه‌های نرم‌افزار محور می توان استفاده کرد. بیشتر مقالاتی که از روش پیش بینی برای تعادل برای شبکه‌های نرم‌افزار محور استفاده کرده‌اند در مصرف انرژی نیز صرفه جویی کرده‌اند در نتیجه پیش بینی بار بر میزان مصرف انرژی بسیار تاثیر گذار است.

جدول ۸ : مروری بر مقالات پیش بینی در تعادل بار شبکه‌های نرم‌افزار محور

مراجع	سال	NLMS	ARIMA	LSTM	پیش بینی	توضیحات
[۲۴]	۲۰۲۰	✓			پیش بینی بار سرور	تعادل بار سرور
[۲۷]	۲۰۲۱	✓			پیش بینی بار سرور	تعادل بار سرور
[۴]	۲۰۱۹		✓		پیش بینی بار سوئیچ	تعادل بار کنترلر
[۱۷]	۲۰۲۰			✓	پیش بینی ترافیک	تعادل بار لینک
[۳۵]	۲۰۲۲			✓	پیش بینی ترافیک	تعادل بار سرور
[۴۳]	۲۰۲۰		✓	✓	پیش بینی بار کنترلر	تعادل بار کنترلر

۶. نتیجه‌گیری

تکنیک‌های بسیاری برای تعادل بار شبکه‌های نرم‌افزار محور وجود دارد. در این مقاله دسته‌بندی دقیق‌تری از تکنیک‌های تعادل بار شبکه‌های نرم‌افزار محور ارائه شده است. به‌صورت کلی این تکنیک‌ها به سه دسته تعادل بار صفحه داده، تعادل بار صفحه کنترل و تعادل بار هم‌زمان صفحه داده و صفحه کنترل تقسیم می شوند. هرکدام از این موارد موردبحث قرار گرفته است. سپس تکنیک‌های بهینه‌سازی مبتنی بر هوش مصنوعی تعادل بار شبکه‌های نرم‌افزار محور ارائه شده است. در نهایت روش‌های پیش‌بینی تعادل بار شبکه‌های نرم‌افزار محور بیان شده است.

مراجع

[1] M.Priyadarsini, and P.Bera, “Software defined networking architecture, traffic management, security, and placement: A survey.” *Computer Networks*.vol.192, 108047.2021. doi: https://doi.org/10.1016/j.comnet.2021.108047

[2] L.Zhu, et al.,” SDN controllers: Benchmarking & performance evaluation.” *arXiv preprint arXiv*:1902.04491, 2019. doi: https://doi.org/10.48550/arXiv.1902.04491

[3] M.Hamdan, et al., "A comprehensive survey of load balancing techniques in software-defined network." *Journal of Network and Computer Applications*, vol. 174. 102856. 2021. doi: https://doi.org/10.1016/j.jnca.2020.102856

[4] A. Filali, , S. Cherkaoui, and A. Kobbane. "Prediction-based switch migration scheduling for SDN load balancing." *International Conference on Communications (ICC)*. 1-6. 2019. doi: https://doi.org/10.1109/ICC.2019.8761469

[5] M. Latah, and L. Toker, "Artificial intelligence enabled software- defined networking: a comprehensive overview." *IET networks*, vol.8. 79-99. 2019. doi: https://doi.org/10.1049/iet-net.2018.5082

[**6**] B.Han, et al., "Network function virtualization: Challenges and opportunities for innovations." *IEEE communications magazine*, vol. 53 . 90-97 . 2015. doi: https://doi.org/10.1109/MCOM.2015.7045396

[**7**] Y.Li, and M. Chen, "Software-defined network function virtualization: A survey." *IEEE Access*, vol. 3 . 2542-2553. 2015. doi: https://doi.org/10.1109/ACCESS.2015.2499271

[**8**] F.Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation." *IEEE Communications Surveys.* vol.16 . 2181-2206 .2014. doi: https://doi.org/10.1109/COMST.2014.2326417

[9] B.Yi, et al., "A comprehensive survey of network function virtualization." *Computer Networks*, vol.133 .212-262 .2018 . doi: https://doi.org/10.1016/j.comnet.2018.01.021

[10] Z. Shang, W. Chen, Q. Ma, and B. Wu, "Design and implementation of server cluster dynamic load balancing based on OpenFlow." *International Joint Conference on Awareness Science and Technology & Ubi-Media Computing* . 691-697 . 2013 . doi: https://doi.org/10.1109/ICAwST.2013.6765526

[11] R. Leland, and B. Hendrickson." An empirical study of static load balancing algorithms." *Proceedings of IEEE Scalable High Performance Computing Conference*. 682-685. 1994. doi: https://doi.org/10.1109/SHPCC.1994.296707

[12] D.LD, B. and P.V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments." *Applied soft computing*, vol.13. 2292-2303. 2013 . doi: https://doi.org/10.1016/j.asoc.2013.01.025

[13] S.Kaur, et al. "Round-robin based load balancing in Software Defined Networking." *international conference on computing for sustainable global development (INDIACom)*. pp. 2136-2139. 2015.

[14] H.Zhong , Y. Fang, and J. Cui, *Reprint of "LBBSRT: An efficient SDN load balancing scheme based on server response time".* Future Generation Computer Systems, vol.80 . 409-416. 2018. doi: https://doi.org/10.1016/j.future.2017.11.012

[15] منتظرالقائم, ا., تحلیل و مدل‌سازی سرورهای VoIP: رویکرد برنامه‌ریزی خطی. محاسبات نرم, ۲۰۲۱. ۷ .p (2): 2-23.

[16] منتظرالقائم, ا .م.ح. یغمایی مقدم, بهبود کارایی پروتکل SIP در شرایط اضافه بار با استفاده از قابلیت مبتنی بر پنجره. محاسبات نرم, ۲۰۲۱. ۲.25-16 .p (2):

[17] X.Chen , et al., "Deep learning-based traffic prediction for energy efficiency optimization in software-defined networking." *IEEE Systems Journal*, vol.15 . 5583-5594. 2020. doi: https://doi.org/10.1109/JSYST.2020.3009315

[18]. T Duong.V.T., "Load balancing routing under constraints of quality of transmission in mesh wireless network based on software defined networking." *Journal of Communications and Networks*, vol. 23.12-22 .2021 . doi: https://doi.org/10.23919/JCN.2021.000004

[19] Y.Wang, C. and S.-Y. "An efficient route management framework for load balance and overhead reduction in SDN-based data center networks." *IEEE Transactions on Network and Service Management*, vol.15 . 1422-1434. 2018. doi: https://doi.org/10.1109/TNSM.2018.2872054

[20] X.Han, , et al.," A service function chain deployment method based on network flow theory for load balance in operator networks." *IEEE Access*, vol. .8 93187-93199. 2020. doi: https://doi.org/10.1109/ACCESS.2020.2994912

[21] T.Andjamba, S. and G.-A.L. Zodi. "A Load Balancing Protocol for Improved Video on Demand in SDN-Based Clouds." *17th International Conference on Ubiquitous Information Management and Communication (IMCOM)*.1-3.2023.doi: https://doi.org/10.1109/IMCOM56909.2023.10035591

[22] J.Galan-Jimenez, et al., "Joint energy efficiency and load balancing optimization in hybrid IP/SDN networks." *Annals of Telecommunications*, vol.78 . pp. 13-31. 2023.

[23] خراسانی فردوانی, م., م. رمضان پور and ر. خورسند, ارائه یک الگوریتم مسیریابی چند مسیره انرژی کارا در سیستم های اینترنت اشیاء. محاسبات نرم, ۲۰۲۱. ۷ .p (1): 34-49.

[24] A.Montazerolghaem, and M.H. Yaghmaee, "Load-balanced and QoS-aware Software-

108749. 2022. doi: https://doi.org/10.1016/j.comnet.2021.108749

[36] S.Zafar , et al., "DSMLB: Dynamic switch-migration based load balancing for software-defined IoT network." *Computer Networks*, vol. 214. 109145. 2022. doi: https://doi.org/10.1016/j.comnet.2022.109145

[37] Z.Guo, et al.," Improving the performance of load balancing in software-defined networks through load variance-based synchronization." *Computer Networks*, vol. 68. 95-109. 2014. doi: https://doi.org/10.1016/j.comnet.2013.12.004

[38] C.Li , K. Jiang ,and Y. Luo, "Dynamic placement of multiple controllers based on SDN and allocation of computational resources based on heuristic ant colony algorithm." *Knowledge-Based Systems*, vol. 241. 108330. 2022. doi: https://doi.org/10.1016/j.knosys.2022.108330

[39] H.Babbar, et al., "Lbsmt: Load balancing switch migration algorithm for cooperative communication intelligent transportation systems." *IEEE Transactions on Green Communications and Networking*, vol. 6.1386-1395 . 2022. doi: https://doi.org/10.1109/TGCN.2022.3162237

[40] H.Wang , et al., "Load-balancing routing in software defined networks with multiple controllers." *Computer Networks*, vol. 141. 82-91. 2018 . doi: https://doi.org/10.1016/j.comnet.2018.05.012

[41] P.Sun, et al., "MARVEL: Enabling controller load balancing in software-defined networks with multi-agent reinforcement learning." *Computer Networks*, vol. 177. 107230. 2020. doi: https://doi.org/10.1016/j.comnet.2020.107230

[42] N.C.Luong , et al., "Applications of deep reinforcement learning in communications and networking: A survey." *IEEE Communications Surveys & Tutorials*, vol. 21. 3133-3174. 2019. doi: https://doi.org/10.1109/COMST.2019.2916583

[43] A.Filali, et al., "Preemptive SDN load balancing with machine learning for delay sensitive applications." *IEEE Transactions on Vehicular Technology*, vol. 69. 15947-15963. 2020. doi: https://doi.org/10.1109/TVT.2020.3038918

[44] N.Michael, "Artificial intelligence a guide to intelligent systems." Addison Wesley. 2005,

[45] C.Fancy, and M. Pushpalatha, "Proactive load balancing strategy towards intelligence-enabled software-defined network." *Arabian Journal for Science and Engineering*, pp. 1-8 . 2021. doi:

[46] X.Yang, "Metaheuristic optimization: algorithm analysis and open problems." *Experimental Algorithms: 10th International Symposium, SEA* defined Internet of Things." *IEEE Internet of Things Journal*, vol. **7**. 3323-3337. 2020 . doi: https://doi.org/10.1109/JIOT.2020.2967081

[25] P.Abhishek , et al., "Load Balancing for Network Resource Management in Software-Defined Networks." *Advances in Distributed Computing and Machine Learning*. pp. 193-203. 2022.

[26] A.Montazerolghaem , "Software-defined Internet of Multimedia Things: Energy-efficient and Load-balanced Resource Management." *IEEE Internet of Things Journal*, vol. 9. 2432-2442. 2021. doi: https://doi.org/10.1109/JIOT.2021.3095237

[27] H.Zheng, et al., "Application of improved ant colony algorithm in load balancing of software-defined networks." *The Journal of Supercomputing*, vol.79. 7438-7460. 2023. doi: https://doi.org/10.1007/s11227-022-04957-8

[28] S.Hassas Yeganeh, and Y. Ganjali. "Kandoo: a framework for efficient and scalable offloading of control applications." in *Proceedings of the first workshop on Hot topics in software defined networks*.pp. 19-24 . 2012.

[29] S.Ejaz , et al., "Traffic load balancing using software defined networking (SDN) controller as virtualized network function." *IEEE Access*, vol. **7**. 46646-46658. 2019. doi: https://doi.org/10.1109/ACCESS.2019.2909356

[30] L.Zhang , et al. "Deployment of intrusion prevention system based on software defined networking." *15th IEEE International Conference on Communication Technology*.26-31. 2013. doi: https://doi.org/10.1109/ICCT.2013.6820345

[31] J.Cui , et al., "A load-balancing mechanism for distributed SDN control plane using response time." *IEEE transactions on network and service management*, vol. 15. 1197-1206. 2018. doi: https://doi.org/10.1109/TNSM.2018.2876369

[32] H.Mokhtar , et al., "Multiple-level threshold load balancing in distributed SDN controllers." *Computer Networks*, vol. 198. 108369. 2021. doi: https://doi.org/10.1016/j.comnet.2021.108369

[33] M.Xiang, et al., "Deep Reinforcement Learning-based load balancing strategy for multiple controllers in SDN." *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 2. 100038. 2022. doi: https://doi.org/10.1016/j.prime.2022.100038

[34] K.S.Sahoo , et al., "ESMLB: Efficient switch migration-based load balancing for multicontroller SDN in IoT." *IEEE Internet of Things Journal*, vol. **7**. 5852-5860. 2019. doi: https://doi.org/10.1109/JIOT.2019.2952527

[35] H.Zhong, et al., "Prediction-based dual-weight switch migration scheme for SDN load balancing." *Computer Networks*, vol.205 .

2011, Kolimpari, Chania, Crete, Greece, pp. 21-32. 2011.

[47] M.A.El-Baz, "A genetic algorithm for facility layout problems of different manufacturing environments." *Computers & Industrial Engineering*, vol. 47. 233-246. 2004. doi: https://doi.org/10.1016/j.cie.2004.07.001

[48] S J.anner, et al. "Evolutionary algorithms for optimized SDN controllers & NVFs' placement in SDN networks." *SDN Day*. 2016 .

[49] M.Dorigo, "Optimization, learning and natural algorithms." *Ph. D. Thesis, Politecnico di Milano*,1992.

[50] R.Poli, J. Kennedy, and T. Blackwell, *Particle swarm optimization.* Swarm intelligence, vol. 1. pp. 33-57. 2007.

[51] C.Gao , et al. "A particle swarm optimization algorithm for controller placement problem in software defined network." in *Algorithms and Architectures for Parallel Processing: 15th International Conference, ICA3PP 2015, Zhangjiajie, China, November 18-20, 2015,*

[52] G.Li, Wang, X., Z.Zhang, "SDN-based load balancing scheme for multi-controller deployment." *IEEE Access*, vol. *7*. 39612-39622. 2019. doi: https://doi.org/10.1109/ACCESS.2019.2906683

[53] L.Liao, and V.C. Leung. "Genetic algorithms with particle swarm optimization based mutation for distributed controller placement in SDNs." *IEEE conference on network function virtualization and software defined networks (NFV-SDN).* 1-6 . 2017. doi: https://doi.org/10.1109/NFV-SDN.2017.8169836

[54] 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 .pp. 2533-2552. 2022. doi:

[55] R.Garroppo, G., Giordano, S., Pagano, M., Procissi, G. "On traffic prediction for resource allocation: A Chebyshev bound based allocation scheme. " Computer Communications, vol.31 . 3741-3751. 2008. doi: https://doi.org/10.1016/j.comcom.2008.05.019

[56] G. M. Jenkins, " Autoregressive–Integrated Moving Average (ARIMA) Models. " *Encyclopedia of Statistical Sciences*, 2004.

[57] S.Hochreiter, & J.Schmidhuber. *Long short-term memory. Neural computation,* vol. *9* . 1735-1780. 1997. doi: https://doi.org/10.1162/neco.1997.9.8.1735