مجله محاسبات نرم، جلد ۱۱، شماره ۲ (پاییز و زمستان ۱۴۰۱)، صفحه ۱۰۶ـ۱۲۹
https://doi.org/10.22052/scj.2023.252610.1134

مجله محاسبات نرم
SOFT COMPUTING JOURNAL
تارنمای مجله: scj.kashanu.ac.ir

دانشگاه کاشان
University of Kashan

طبقه‌بندی و بررسی جامع روش‌های بهینه‌سازی تعادل بار شبکه‌های نرم‌افزار محور✧

سمیه ایمان‌پور[۱]، دانشجوی کارشناسی ارشد، احمدرضا منتظرالقائم[۱*]، استادیار

[۱] دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران.

اطلاعات مقاله

تاریخچه مقاله:
دریافت ۱۴ اسفند ماه ۱۴۰۱
پذیرش ۲ مرداد ماه ۱۴۰۲

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

چکیده

ترافیک شبکه‌ها روزبه‌روز در حال افزایش است؛ به همین دلیل برای مدیریت شبکه‌ها از فناوری شبکه‌های نرم‌افزار محور استفاده می‌شود؛ زیرا این فناوری یک نمای کلی از شبکه ارائه داده و مدیریت پیشرفته را امکان‌پذیر می‌کند. در ضمن برای بهبود عملکرد در شبکه‌های نرم‌افزار محور به تعادل بار نیز نیاز است. برای ایجاد تعادل بار در شبکه‌های نرم‌افزار محور رویکردهای بسیاری پیشنهاد شده است. این رویکردها می‌توانند طبقه‌بندی باشند؛ با این حال طبقه‌بندی‌های ارائه شده تاکنون دقیق نیستند. در این مقاله، طبقه‌بندی دقیقی برای تعادل بار شبکه‌های نرم‌افزار محور ارائه شده و در ادامه رویکردهایی که از الگوریتم‌های بهینه‌سازی مبتنی بر هوش مصنوعی برای حل مساله تعادل بار استفاده کرده‌اند، مورد تجزیه و تحلیل قرار گرفته‌اند. در نهایت روش‌های پیش‌بینی تعادل بار در شبکه‌های نرم‌افزار محور شرح داده شده و چگونگی کمک آنها به کاهش مصرف انرژی مورد بررسی قرار گرفته است.

۱. مقدمه

با گسترش شبکه‌ها و افزایش تعداد کاربران و در نهایت افزایش ترافیک موجود، شبکه‌های سنتی از مشکلات مدیریت شبکه رنج می‌برند، زیرا دستگاه‌های سخت‌افزاری شبکه، مانند سوییچ‌ها، روترها و متعادل‌کننده‌های بار، همگی خاص فروشنده هستند. در چنین شرایطی مدیریت شبکه دشوار می‌شود [۱]، [۲]. به همین دلیل از شبکه‌های نرم‌افزار محور (SDN) [1] استفاده می‌شود که می‌توانند به صورت 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، رویکردهای تعادل بار به صراحت 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 سوییچ‌ها در حال حرکت است که باعث ازدحام و اضافه بار در شبکه می‌شود. حال کنترل‌کننده باید با استفاده از رویکردهای تعادل بار، بار را بین سرورهای مختلف توزیع کند. اما با یک کنترل‌کننده نمی‌توان

✧ نوع مقاله: مروری

* نویسنده مسئول
پست(های) الکترونیک: s.imanpour@eng.ui.ac.ir (ایمان‌پور)
a.montazerolghaem@comp.ui.ac.ir (منتظرالقائم)

[1] Software Defined Network (SDN)

نحوه ارجاع به مقاله: سمیه ایمان‌پور و احمدرضا منتظرالقائم، «طبقه‌بندی و بررسی جامع روش‌های بهینه‌سازی تعادل بار شبکه‌های نرم‌افزار محور»، مجله محاسبات نرم، جلد ۱۱، شماره ۲، صفحات ۱۰۶ـ۱۲۹، پاییز و زمستان ۱۴۰۱.

کل شبکه را مدیریت کرد؛ چون مقیاس‌پذیری و در دسترس بودن را کاهش می‌دهد. در نتیجه از معماری کنترل‌کننده توزیع شده، استفاده شده است. در معماری کنترل‌کننده توزیع شده هر کنترل‌کننده بر اساس ظرفیتی که دارد تعدادی سوییچ و سرور می‌تواند مدیریت کند. سوییچ‌ها و سرورهایی که هر کنترل‌کننده مدیریت می‌کند، در دامنه آن کنترل‌کننده قرار می‌گیرد. در نتیجه دامنه هر کنترل‌کننده شامل چند سوییچ و سرور که توسط آن کنترل‌کننده مدیریت می‌شود. شبکه‌های چند کنترلری نیز ممکن است دچار اضافه بار شوند، در نتیجه از الگوریتم مهاجرت سوییچ استفاده می‌کنیم. مهاجرت سوییچ به معنای انتقال یک سوییچ از دامنه کنترل‌کننده اضافه بار به دامنه کنترل‌کننده کم‌بار برای ایجاد تعادل بار کنترل‌کننده‌ها در شبکه نرم‌افزار محور است [۱]- [۳].

به صورت کلی تعادل بار شبکه‌های نرم‌افزار محور به تعادل بار سرور، تعادل بار پیوند و تعادل بار کنترل‌کننده تقسیم می‌شود. برای حل مشکل تعادل بار شبکه‌های نرم‌افزار محور برخی از مقالات از روش بهینه‌سازی استفاده کرده‌اند. در روش بهینه‌سازی، مساله تعادل بار شبکه‌های نرم‌افزار محور از نظر پیچیدگی محاسبات NP-Hard است [۴]. به همین دلیل در سال‌های اخیر برای حل مساله 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 اضافه کردن 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. هر کدام 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 امر باعث می‌شود مدیریت بهتری روی منابع داشته باشم و هم باعث کاهش مصرف انرژی می‌شود.

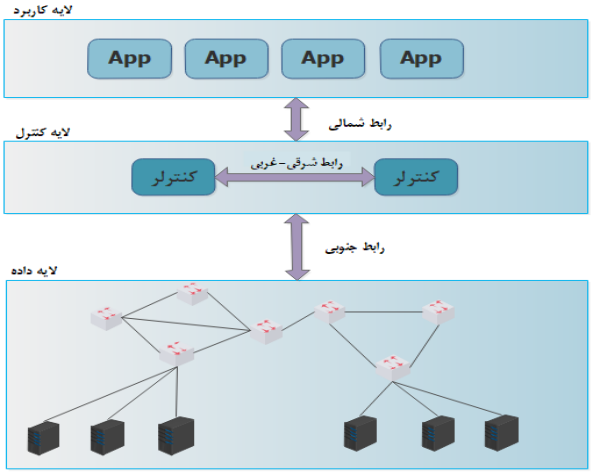
ادامه مقاله به صورت زیر سازماندهی شده است. در بخش ۲ ادبیات موضوع و مفاهیم مورد استفاده در مقاله توضیح داده شده است. در بخش ۳، طبقه‌بندی دقیق‌تری از تعادل بار شبکه‌های نرم‌افزار محور ارائه شده است. در بخش ۴، تعادل بار شبکه‌های نرم‌افزار محور مبتنی بر بهینه‌سازی که از الگوریتم‌های هوش مصنوعی استفاده کرده‌اند 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 فناوری جدید برای مدیریت پیشرفته شبکه‌های امروزی است [۱]. معماری شبکه‌های نرم‌افزار محور همان‌طور که در شکل (۱) نشان داده شده است، شامل سه لایه، لایه کاربرد، لایه کنترل ولایه داده است.

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]. پروتکل OpenFlow متداول‌ترین رابط جنوبی برای ارتباط بین کنترل‌کننده و سوییچ‌ها است و می‌تواند مدیریت شبکه را تسریع کند [۸]. رابط شرقی-غربی برای ارتباط بین کنترل‌کننده‌ها در لایه کنترل است [۲]. کنترل‌کننده مهم‌ترین جز شبکه‌های نرم‌افزار محور است که مدیریت ترافیک لایه داده را به عهده دارد.



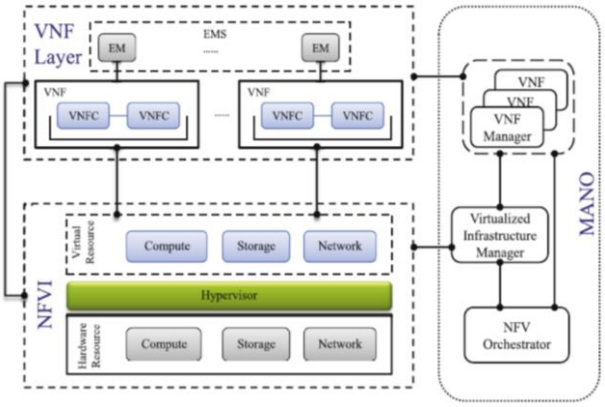
شکل (۱): معماری شبکه نرم‌افزار محور

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) نشان داده شده است که شامل سه بخش است [۹].

- زیرساخت مجازی‌سازی توابع شبکه (NFVI)[2]: 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 را فراهم می‌کند.
- توابع شبکه مجازی (VNF)[3]: 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 (MANO)[4]: مسئول مدیریت و تنظیم زیرساخت مجازی‌سازی توابع شبکه و مدیریت و تهیه توابع شبکه مجازی است.

[1] Network Function Virtualization
[2] Network Function Virtualization Infrastructure
[3] Virtual Network Function
[4] Management and orchestration

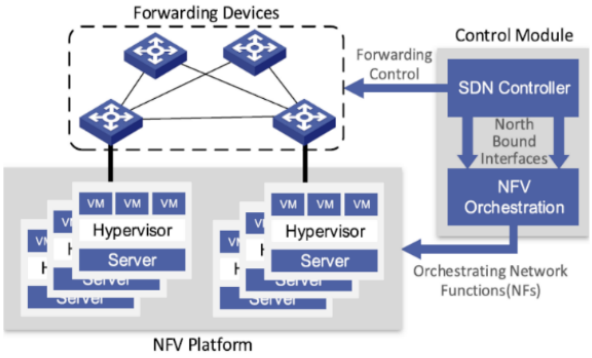


شکل (۲): معماری SDN [۹]

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. هر کدام به صورت مستقل کار خود را به خوبی انجام می‌دهد؛ اما این دو مکمل یکدیگرند. مجازی‌سازی توابع شبکه نرم‌افزار محور همان‌طور که در شکل (۳) نشان داده شده است شامل سه بخش است [۷].

- بستر 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 کنترل‌کننده شبکه نرم‌افزار محور انجام می‌شود و از پروتکل OpenFlow برای ارتباط بین کنترل‌کننده و دستگاه انتقال استفاده شده است.
- ماژول کنترل‌کننده: این ماژول برای مدیریت کل شبکه است که شامل دو بخش کنترل‌کننده شبکه‌های نرم‌افزار محور و ارکستراسیون مجازی‌سازی توابع شبکه می‌باشد. ارکستراسیون مجازی‌سازی توابع شبکه وظیفه مدیریت شبکه مجازی را به عهده دارد و توسط کنترل‌کننده شبکه‌های نرم‌افزار محور از طریق رابط‌هایی استاندارد کنترل می‌شود.



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

۴.۲. تعادل بار شبکه‌های نرم‌افزار محور

تعادل بار روشی برای توزیع ترافیک در شبکه‌ها است. 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) نشان داده شده است، در نمودار درختی قسمت‌هایی که سبز رنگ هستند، نمودار درختی مقاله [۳] هستند.

اما مقالاتی وجود دارند که به صورت 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 نشان داده شده است، قسمت‌های نارنجی رنگ در این مقاله اضافه شده‌اند. این مقاله ساختار دقیق‌تری از تعادل بار شبکه‌های نرم‌افزار محور ارائه داده است.

۱.۳. تعادل بار صفحه داده

تعادل بار صفحه داد برای کشف بهترین سرورها و پیوندها در شبکه برای دریافت و عبور ترافیک است. تعادل بار در صفحه داده به سه دسته تقسیم می‌شود که در ادامه به شرح آنها پرداخته می‌شود.

شکل (۴): نمودار درختی طبقه‌بندی تعادل بار شبکه‌های نرم‌افزار محور

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 به درستی انجام نشود و سروری دچار اضافه بار شود. به همین دلیل، به تعادل بار بین سرورها نیاز داریم [۳]. تعادل بار سرورها خود به دو دسته 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 کاملا 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 رویکرد Round Robin، بار را بین سرورهای مختلف به صورت مساوی تقسیم می‌کنند تا سروری دچار اضافه بار نشود. معماری تعادل بار که شامل یک سوییچ Open Flow و یک کنترل‌کننده POX و چندین سرور متصل به سوییچ با آدرس IP ثابت و کنترل‌کننده است، لیستی از سرورها را دارد. زمانی که بارهای ورودی وارد می‌شوند، کنترل‌کننده طبق ترتیب بار را بین سرورها تقسیم می‌کند. نتایج نشان می‌دهند، 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 به دست آمده 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 شده است شامل سه بخش، سرور و مشتری، سوییچ OpenFlow و کنترل‌کننده 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 محاسبه می‌کند. این مقدار اختلاف اگر کم باشد نشان‌دهنده این است که سرورها بار مشابه دارند و سروری با حداقل زمان پاسخ انتخاب می‌شود. در غیراین‌صورت 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، اگرچه این طرح به طور موثر بر بسیاری از مسائل تعادل بار غلبه می‌کند؛ اما صرفه‌جویی در انرژی سرور را در نظر نمی‌گیرد.

منتظرالقائم [۱۵]، مساله توزیع حالت بین چندین گره را تعریف کرده و هدف از حل این مساله افزایش گذردهی کل تماس‌ها و دسترسی‌پذیری سرورها است. در این مقاله، ابتدا میزان مصرف CPU در یک سرور SIP بررسی شده که نتایج نشان می‌دهد میزان مصرف CPU توسط سرور حالتمند با افزایش نرخ تماس با سرعت بیشتری نسبت به سرور بدون حالت افزایش می‌یابد، زیرا در سرورهای SIP که حالتمند هستند میزان مصرف CPU به دلیل نگه داشتن حالت در زمانی که تماس افزایش می‌یابد بسیار زیاد است، در نتیجه در این مقاله هر سرور تنها برای بخشی از درخواست‌ها حالت را نگه می‌دارد و برای بقیه فاقد حالت است. در مرحله بعد که طراحی چارچوب مبتنی بر شبکه نرم‌افزار محور است، برای توزیع حالت در شبکه SIP در لایه کنترل سه ماژول اضافه شده است و در نهایت با توجه به ارزیابی‌های انجام گرفته، این مدل به افزایش ۱۵ درصد تا ۲۰ درصدی بیشینه گذردهی تماس دست یافته است.

منتظرالقائم و همکاران [۱۶]، یک الگوریتم تطبیقی توزیع شده و انتها به انتهای مبتنی بر پنجره را برای کنترل اضافه بار پیشنهاد کرده‌اند. این مقاله یک الگوریتم توزیع شده کنترل‌کننده اضافه

بار SIP که مبتنی بر بازخورد ضمنی است و البته مبتنی بر شبکه‌های نرم‌افزار محور نیست، پیشنهاد کرده است. دلیل اصلی اضافه بار سرور SIP پیام‌های درخواست است و برای محاسبه بیشینه جدید اندازه پنجره با تحلیل نسبت پیام‌های درخواست به پیام‌های تایید است. نتایج به دست آمده نشان می‌دهد که گذردهی بهبود یافته، تاخیر بسیار کمتر شده و نرخ ارسال مجدد کاهش یافته است.

۲.۱.۳. تعادل بار پیوندها

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

شیانگی و همکاران [۱۷]، یک چارچوب بهینه‌سازی کارایی انرژی مبتنی بر پیش‌بینی ترافیک (TPEO)[1] در شبکه‌های نرم‌افزار محور پیشنهاد می‌کنند. ایده اصلی بهینه‌سازی کارایی انرژی مبتنی بر پیش‌بینی ترافیک بهبود بهره‌وری انرژی شبکه بر اساس پیش‌بینی ترافیک، با در نظر گرفتن موازنه بین بار شبکه و مصرف انرژی است. ابتدا یک رویکرد نظارت بر ترافیک تطبیقی را طراحی می‌کند که می‌تواند دوره نظارت ترافیک را به صورت پویا تنظیم کند و بار ترافیک شبکه‌های نرم‌افزار محور را در زمان واقعی به دست آورد. در این روش، هزینه نظارت بر شبکه کاهش می‌یابد در حالی که 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 غیربرخط ایجاد می‌کنند، سپس ترافیک شبکه را در زمان واقعی توسط پنجره کشویی سری‌های زمانی ثبت و پیش‌بینی می‌کنند. سپس یک الگوریتم اکتشافی برخط جدید برای بهینه‌سازی بازده انرژی شبکه نرم‌افزار محور بر اساس پیش‌بینی ترافیک پیشنهاد می‌کنند که منابع پیوند 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، به صرفه‌جویی انرژی دست یابد و مصرف انرژی کاهش یابد. با این حال، محدودیت این الگوریتم این است که رویکردهای صرفه‌جویی در انرژی منجر به تمرکز نسبی ترافیک شبکه می‌شود که مستعد خرابی پیوندها است.

دانگ و همکاران [۱۸]، یک الگوریتم مسیریابی متعادل‌کننده بار جدید را تحت محدودیت‌های کیفیت انتقال برای شبکه‌های مش بی‌سیم پیشنهاد کرده‌اند. الگوریتم پیشنهادی مبتنی بر شبکه‌سازی نرم‌افزار محور است که مسیریابی تعادل بار تحت محدودیت‌های کیفیت انتقال (LBRCQT)[2] نامیده می‌شود. الگوریتم مسیریابی تعادل بار تحت محدودیت‌های کیفیت انتقال بر اساس اصل مسیریابی متمرکز کار می‌کند و تابع مسیریابی در کنترل‌کننده شبکه نرم‌افزار محور پیاده‌سازی می‌شود. الگوریتم پیشنهادی می‌تواند عملکرد شبکه را از نظر نسبت سیگنال به نویز (SNR)[3]، نسبت تحویل بسته (PDR)[4] و توان بهبود بخشد. وانگ و همکاران [۱۹]، یک چارچوب مدیریت مسیر متعادل برای بهبود عملکرد شبکه مرکز داده‌های درخت چاق[5] با بهره‌برداری از شبکه‌های نرم‌افزار محور با هزینه کم پیشنهاد کرده‌اند. این مقاله ابتدا وضعیت شبکه را با استفاده از سه جدول ذخیره‌سازی اطلاعات مسیرها، بارها، سوییچ‌ها نظارت کرده و سپس در صورت وجود تراکم احتمالی، روش اصلاح مسیر تطبیقی را راه‌اندازی می‌کند. برای صرفه‌جویی در هزینه‌های سربار کنترل‌کننده رویکرد پویایی اطلاعات توسعه داده شده است تا در صورت تطبیق اطلاعات سوییچ‌ها را با پیام‌های کمتر جستجو کند. رویکرد اصلاح تطبیقی دارای دو ویژگی است: (۱) تنها در صورت لزوم فراخوانی می‌شود که در نتیجه در هزینه و مصرف انرژی صرفه‌جویی می‌شود و (۲) ورودی‌های قدیمی را از جدول سوییچ برای جلوگیری از سرریز بافر حذف می‌کند. هان و همکاران [۲۰]، یک روش استقرار زنجیره تابع سرویس

[1] Traffic Prediction-based Energy efficiency Optimization

[2] Load Balancing Routing under Constraints of Quality of Transmission

[3] Signal to Noise Ratio

[4] Package Delivery Ratio

[5] Fat Tree

بر اساس نظریه بدهی شبکه پیشنهاد کرده‌اند. در روش آنها، ابتدا یک مجموعه گره کلید کاندید برای تشکیل شبکه هدایت شده تعیین می‌شود. در مرحله دوم، مساله استقرار به وظیفه‌ای برای یافتن مسیر بهینه در شبکه هدایت شده تبدیل می‌شود. در مرحله سوم، با در نظر گرفتن هزینه انتقال پیوند به عنوان هزینه و تقاضای پهنای باند به عنوان محدودیت، یک الگوریتم حداقل هزینه–حداکثر جریان برای تعیین مسیر بهینه استقرار و تکمیل استقرار زنجیره تابع سرویس استفاده می‌شود.

اندجامبا و همکاران [۲۱]، پروتکل مسیریابی متعادل‌کننده بار (LBRP)[1] مبتنی بر شبکه‌های نرم‌افزار محور را برای تضمین کیفیت ویدئو هنگام پخش در نرم‌افزار تعریف شده، معرفی می‌کنند. این پروتکل از رویکرد مهندسی ترافیک برای دستیابی به کیفیت سرویس استفاده می‌کند. به این صورت که جریان‌های ترافیکی را به دو دسته، جریان‌های حساس به تاخیر (جریان واقعی) و جریان‌هایی که به تاخیر حساس نیستند (جریان‌های غیرواقعی) تقسیم می‌کند. جریان‌های واقعی اولویت بیشتری نسبت به جریان‌های غیرواقعی دارند و تلاش می‌کنند جریان‌های واقعی را از پیوندهایی که پهنای باند تضمین شده دارند، عبور دهند و جریان‌های غیرواقعی را از بقیه پیوندهای جایگزین عبور دهد. این کار باعث بهبود کیفیت تجربه کاربر از برنامه‌های پخش ویدئوی ذخیره شده، می‌شود.

گالان و همکاران [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 شوند. اما چنین مهاجرتی برای شبکه‌های بزرگ ساده یا سریع نیست، در نتیجه باید به صورت تدریجی مهاجرت کرد. چنین مهاجرت‌هایی باعث ایجاد شبکه ترکیبی IP/SDN می‌شود. این مقاله برای تعادل بار پیوندها از الگوی شبکه نرم‌افزار محور برای تقسیم ترافیک بین گره‌های شبکه نرم‌افزار محور استفاده می‌کند. همچنین از الگوریتم ژنتیک برای بهبود مصرف انرژی استفاده می‌کند. به این صورت که با استفاده از الگوریتم ژنتیک بهترین پیکربندی شبکه را که شامل کمترین تعداد پیوندهای فعال باشد با در نظر گرفتن ظرفیت پیوندها و ترافیک موجود، می‌یابد. این رویکرد باعث کاهش ۵۰ درصدی حداکثر استفاده از پیوند و ۶۰ درصدی مصرف انرژی می‌شود.

خراسانی فردوانی و همکاران [۲۳]، یک مسیریابی انرژی کارا مبتنی بر اینترنت اشیا که مبتنی بر شبکه‌های نرم‌افزار محور نیست، ارائه کرده‌اند. این روش برای مسیریابی از متغیرهای تعداد گام، فاصله و انرژی گره‌های میانی برای انتخاب مسیر استفاده می‌کند. روش پیشنهادی از نظر متغیرهای نرخ گم شدن بسته و انرژی باقیمانده بهبود یافته است.

۳.۱.۳. تعادل بار همزمان سرورها و پیوندها

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

منتظرالقائم [۲۴]، یک چارچوب جدید مبتنی بر شبکه‌های مبتنی بر نرم‌افزار برای برآورده کردن الزامات کیفیت سرویس، سرویس‌های مختلف اینترنت اشیا و ایجاد تعادل ترافیک بین سرورهای اینترنت اشیا به طور همزمان پیشنهاد کرده است. ابتدا ثابت کرده است که این دو مساله به طور همزمان NP-hard هستند و برای کاهش پیچیدگی زمانی، مساله را به دو مشکل فرعی شامل (۱) انتخاب سرور و (۲) انتخاب مسیر تقسیم کرده‌اند. در این راستا یک چارچوب مبتنی بر شبکه‌های نرم‌افزار محور طراحی شده است. این چارچوب از سه لایه تشکیل شده است که از طریق یکسری رابط با یکدیگر تعامل دارند. کنترل‌کننده ابتدا آمار منابع مصرفی (CPU و حافظه) سرورهای اینترنت اشیا را جمع‌آوری کرده و سپس با استفاده از الگوریتم حداقل میانگین مربعات نرمال‌شده (NLMS)[2] میزان مصرف منابع را در آینده پیش‌بینی می‌کند و با توجه به خروجی این الگوریتم و با استفاده از سیستم فازی ظرفیت سرورها برای دریافت ترافیک مشخص می‌شود. از طرف دیگر ترافیک اینترنت

[1] Load Balancing Routing Protocol

[2] Normalised Least Connection Mean Squares

اشیا در سه کلاس دسته‌بندی شده است که هر کدام از این کلاس‌ها 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 [۲۵]، تعادل بار مبتنی شبکه‌های نرم‌افزار محور را ارائه کرده است. این مقاله از رویکردهای یادگیری ماشینی مانند شبکه‌های عصبی مصنوعی (ANN)[1] 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 Weighted Round ،Random Least Connection Robin و Round Robin روی متعادل‌کننده بار برای به دست آوردن متغیرهای سرور مانند (۱) استفاده از پردازنده، (۲) استفاده از حافظه (۳)، نوع فایل درخواستی، (۴) بایت ارسال شده به مشتری در صورت درخواست و (۵) بایت‌های دریافت شده، استفاده می‌شود. سپس بر اساس همین متغیرها و الگوریتم‌ها، الگوهایی را یاد می‌گیرد و از طریق آن مدل ساخته می‌شود. برای متعادل‌سازی بار مبتنی بر مسیر از توپولوژی Abilene استفاده می‌شود که برای متعادل‌سازی بار میان مسیرهای با تاخیر کم برای یافتن بهترین مسیرهای مناسب از مشتری به سرور، پیاده‌سازی شده است.

منتظرالقائم [۲۶]، راه‌حلی برای حل بحران مدیریت منابع شبکه‌های اینترنت اشیا چندرسانه‌ای (IoMT)[2] طراحی کرده‌اند. زیرساخت اینترنت اشیا چندرسانه‌ای از دو جنبه با بحران مدیریت منابع مواجه است: تعادل بار و بهینه‌سازی انرژی که محدودیت‌های انرژی و بار به طور همزمان یک مساله NP-hard است و پیچیدگی زمانی بالایی دارد و به طور دقیق‌تر یک مساله برنامه‌ریزی غیرخطی عدد صحیح مختلط (MINLP)[3] است. چون این مساله از پیچیدگی بالایی برخوردار است، به سه مساله فرعی تقسیم شده است: (۱) اندازه شبکه، (۲) انتخاب سرور مناسب و (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 دو لایه ابر لبه (Edge cloud) و ابر هسته (Core cloud) تشکیل شده است که این دو لایه‌ای بودن از منطق شبکه‌های نرم‌افزار محور سرچشمه می‌گیرد. لایه Core cloud شامل طراح کنترل‌کننده منابع کم‌مصرف و بار متعادل (ELRC)[4] است و ابر لبه هم شامل سرورها و سوییچ‌های مجازی است. سرورها همگی مبتنی بر فناوری مجازی‌سازی توابع شبکه هستند که کنترل‌کننده بتواند اندازه شبکه را تغییر دهد. سپس با استفاده از روش پیش‌بینی حداقل میانگین مربعات نرمال شده میزان مصرف سرورها را با توجه به تاریخچه مصرف منابع سرورها (یعنی میزان مصرف CPU و حافظه و دیسک سخت) 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 در صورتی که شبکه در حالت کم‌بار باشد، اندازه شبکه کوچک‌تر می‌شود. در نهایت، با استفاده از الگوریتم حداقل بار، بهترین سرور انتخاب و با استفاده از الگوریتم کوتاه‌ترین مسیر بهترین مسیر مشخص می‌شود.

ژانگ و همکاران [۲۷]، الگوریتم تعادل بار شبکه‌های نرم‌افزار محور بر اساس بهینه‌سازی کلونی مورچه‌ها (IACO-LB)[5] را پیشنهاد کرده‌اند. این مقاله ابتدا اطلاعات کامل سرورها و پیوندها را جمع‌آوری کرده و این اطلاعات را به عنوان ورودی به

[1] Artifical Neural Network
[2] Internet of Multimedia Things
[3] Mixed-Integer Nonlinear Programming
[4] Energy-efficient and Load-balanced Resource Controller
[5] Improved Ant Colony Optimization-Load Balancing

الگوریتم کلونی مورچه‌ها می‌دهد. الگوریتم کلونی مورچه‌ها کم‌بارترین سرور و مسیر را انتخاب می‌کند. این رویکرد سرعت همگرایی پایدار و قابل‌توجهی دارد، اما عیب این رویکرد این است که هزینه پردازش کنترل‌کننده در نظر گرفته نشده است.

در جدول (۱) مروری بر مقالات تعادل بار صفحه داده ذکر شده است و در جدول (۲) مقایسه‌ای بین متغیرهای مقالات صورت پذیرفته است.

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

مرجع	سال	ویژگی	متغیر ورودی	کنترل‌کننده	محیط شبیه‌سازی	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 پیام‌ها	-	محیط واقعی	سرور پویا	×	×
[۱۷]	۲۰۲۰	بهینه‌سازی کارایی انرژی مبتنی بر پیش بینی ترافیک	پهنای باند و تاخیر پیوند	Ryu	Mininet	پیوند	×	✓
[۱۸]	۲۰۲۱	مسیریابی تعادل بار تحت محدودیت‌های کیفیت انتقال (LBRCQT)	پیام Packet-In	-	OMNet ++	پیوند	×	×
[۱۹]	۲۰۱۸	مدیریت مسیر متعادل برای بهبود عملکرد شبکه	بار ورودی سوییچ ها	Ryu	Mininet	پیوند	×	✓
[۲۰]	۲۰۲۰	روش استقرار زنجیره تابع سرویس بر اساس نظریه بدهی شبکه	استقرار نسبت موقعیت، متوسط بلند مدت درآمد نسبت به هزینه، ضریب گسترش طول پیوند، میانگین تاخیر انتقال، نسبت گره/پیوند گلوگاه	-	Matlab	پیوند	✓	×
[۲۱]	۲۰۲۳	پروتکل مسیریابی متعادل کننده بار (LBRP)	تاخیر، پهنای باند، توان عملیاتی	Ryu	Mininet	پیوند	×	×
[۲۲]	۲۰۲۳	تعادل بار ترافیک و به حداقل رساندن مصرف انرژی در شبکه IP/SDN	استفاده از پیوند و میزان مصرف انرژی	-	-	پیوند	✓	×
[۲۳]	۲۰۲۱	مسیریابی انرژی کارا مبتنی بر اینترنت اشیا	تعداد گام ، فاصله و انرژی گره میانی	-	NS-2	پیوند	×	✓
[۲۴]	۲۰۲۰	برآورده کردن الزامات QoS سرویس‌های مختلف و ایجاد تعادل بار بین سرورهای اینترنت اشیا	QoS (تاخیر و توان عملیاتی) میزان مصرف CPU و حافظه	FloodLight	محیط واقعی	سرور و پیوند	×	✓
[۲۵]	۲۰۲۲	تعادل بار سرور و مسیر در SDN با استفاده از روش‌های یادگیری ماشین	میزان مصرف CPU و حافظه، توان عملیاتی، زمان پاسخ	POX	محیط واقعی	سرور و پیوند	×	×
[۲۶]	۲۰۲۲	حل بحران تعادل بار و میزان مصرف انرژی سرورها و سوییچ‌های شبکه IoMT	میزان مصرف CPU و حافظه و دیسک سخت	FloodLight	محیط واقعی	سرور و پیوند	✓	✓
[۲۷]	۲۰۲۳	الگوریتم تعادل بار شبکه‌های نرم‌افزار محور	پهنای باند پیوند و میزان مصرف	-	Matlab	سرور و	×	×

بر اساس کلونی مورچه‌ها (IACO-LB)				CPU و حافظه سرور				پیوند		

جدول (۲): متغیرهای مقالات مربوط به تعادل بار صفحه داده

مراجع	از دست دادن بسته	گذردهی	زمان پاسخ	تاخیر	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]	×	✓	×	×	✓	×	×	×	×	×

در جدول (۲) و بقیه جداول مشابه، سلول‌های با علامت «✓» به این معنا است که مقاله مورد نظر آن متغیر خاص را در نظر گرفته است و سلول‌های «×» به این معنا هستند که مقاله مورد نظر آن متغیرها را در نظر نگرفته است. مقالاتی که برای تعادل بار سرور هستند، تعداد کمی دارند و مصرف انرژی نادیده گرفته شده است و متغیر زمان پاسخ برای این مقالات بسیار حائز اهمیت است. مقالاتی که برای تعادل بار پیوند هستند، کنترل‌کننده منتخب آنها Ryu است زیرا در این مقالات طبق جدول (۲) دو متغیر گذردهی و تاخیر اهمیت دارد و این کنترل‌کننده برای این دو متغیر به نسبت بقیه کنترل‌کننده‌ها بهتر عمل می‌کند. مقالاتی که برای تعادل بار همزمان سرور و پیوند هستند، بیشتر در محیط واقعی انجام‌گرفته‌اند و در آنها متغیرهای گذردهی و نسبت استفاده از منابع دارای اهمیت هستند.

به صورت کلی در مقالات صفحه داده به صرفه‌جویی در مصرف انرژی به نسبت خوبی پرداخته شده است و متغیرهای گذردهی و تاخیر بسیار مهم بوده‌اند و به متغیرهای RMSE و زمان اتمام جریان پرداخته نشده است.

۲.۳. تعادل بار صفحه کنترل

تعادل بار در صفحه کنترل به سه دسته تعادل بار کنترل‌کننده با تمرکز منطقی/توزیع فیزیکی، تعادل بار کنترل‌کننده توزیع شده و تعادل بار کنترل‌کننده مجازی‌سازی تقسیم می‌شود. تعادل بار کنترل‌کننده توزیع شده خود به دو دسته سلسله‌مراتبی عمودی و سلسله‌مراتبی افقی تقسیم می‌شود.

الف. تعادل بار کنترل‌کننده توزیع شده سلسله‌مراتبی عمودی:

در کنترل‌کننده سلسله‌مراتبی عمودی، هر کنترل‌کننده در لایه‌های مختلف قرار می‌گیرد و ارتباط کنترل‌کننده‌های درون لایه‌ها غیر ممکن است و همه کنترل‌کننده‌ها توسط یک کنترل‌کننده ریشه مدیریت می‌شوند [۲۸]. ایجاز و همکاران [۲۹]، به تعادل بار ترافیکی با استفاده از یک کنترل‌کننده شبکه مجازی نرم‌افزار محور (vSDN) و به عنوان یک تابع شبکه مجازی دست یافته‌اند. مجازی‌سازی توابع شبکه معادله موثر تابع شبکه مجازی و

خدمات مرتبط در زیرساخت شبکه پویا را ارائه می‌دهد. هنگام استفاده از کنترل‌کننده مجازی نرم‌افزار محور به عنوان یک تابع شبکه مجازی این امکان هست که می‌توان توابع شبکه مجازی مشابه بیشتری برای آن اضافه کرد و در صورت افزایش بار ترافیکی می‌توان شبکه‌های مجازی نرم‌افزار محور ثانویه را برای به اشتراک‌گذاری بار اضافه کرد. از جایی که تمام منابع مجازی می‌شوند، بنابراین می‌توان منابع سخت‌افزاری را بنا به نیاز اختصاص داد یا اضافه کرد. هنگامی که نیاز به یک کنترل‌کننده مجازی نرم‌افزار محور ثانویه تعیین می‌شود و یک رونوشت از کنترل‌کننده شبکه مجازی نرم‌افزار محور با پیکربندی مشابه کنترل‌کننده شبکه مجازی نرم‌افزار محور اصلی ایجاد می‌شود که به دقت کار می‌کند و بار ترافیک را به اشتراک می‌گذارد. هر دو کنترل‌کننده شبکه مجازی نرم‌افزار محور به طور مستقل در فضای ابری با شفافیت قرار می‌گیرند و اطمینان می‌دهند که هر مشتری در شبکه با وجود کنترل‌کننده 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 دهد.

ب. تعادل بار کنترل‌کننده توزیع شده سلسله‌مراتبی افقی: در کنترل‌کننده سلسله‌مراتبی افقی، همه کنترل‌کننده‌ها روی یک لایه در کنترل‌کننده افقی قرار می‌گیرند؛ لذا اگرچه همه کنترل‌کننده‌ها دامنه‌های متنوعی پیدا می‌کنند؛ اما همه آنها به عنوان کنترل‌کننده ریشه در نظر گرفته می‌شوند که با کل وضعیت شبکه آشنا هستند [۳۰].

مرجع [۳۱]، یک روش متعادل‌سازی بار کنترل‌کننده چندگانه نرم‌افزار محور بر اساس زمان پاسخ (SMCLBRT)[1] پیشنهاد کرده است. این روش تعادل بار شبکه‌های نرم‌افزار محور، بر اساس زمان پاسخ است که با در نظر گرفتن ویژگی‌های متغیر زمان‌های پاسخ بلادرنگ در مقابل بارهای کنترلی با انتخاب آستانه زمان پاسخ مناسب و برخورد همزمان با چندین کنترل‌کننده کار می‌کند. یعنی ابتدا زمان پاسخ تمام کنترل‌کننده‌ها را به دست می‌آورد و از آنها برای به دست آوردن آستانه استفاده می‌کند. برای این منظور، ابتدا زمان پاسخ را تحت بارهای مختلف آزمایش کرده و سپس آستانه زمان پاسخ را با توجه به ویژگی‌های متنوع آنها انتخاب می‌کند. با افزایش بار کنترل‌کننده از حالت عادی به اضافه بار زمان پاسخ سریع‌تر افزایش می‌یابد؛ ولی قبل از این مرحله اگرچه زمان پاسخ با افزایش بار، افزایش می‌یابد؛ ولی به آرامی تغییر می‌کند. بر اساس همین افزایش ناگهانی و مداوم زمان پاسخ آستانه انتخاب شده و سپس زمان پاسخ تمام کنترل‌کننده‌ها با آستانه مناسب مقایسه می‌شوند. بر اساس این مقایسه سرورها را در دو دسته اضافه‌بار و کم‌بار تقسیم می‌کنند. حال سوییچ کنترل‌کننده‌ای که بیشترین حجم کاری را دارد به کنترل‌کننده‌ای با کمترین حجم کاری مهاجرت می‌کند. این طرح می‌تواند توزیع بار ناهموار را متعادل کند و از زمان کمتری برای مقابله با چندین کنترل‌کننده اضافه‌بار استفاده کند. از دیدگاه کاربر، این طرح می‌تواند زمان پاسخگویی را کاهش دهد. مزایای این روش این است که می‌تواند تعادل بار چندین کنترل‌کننده نرم‌افزار محور را به طور موثر و سریع به دست آورد، در حالی که محدودیت این روش، رویکرد متعادل‌سازی بهتر کنترل‌کننده‌های بارگذاری شده متعدد هنگام در نظر گرفتن هزینه مهاجرت است.

مختار و همکاران [۳۲]، طرح مهاجرت سوییچ تعادل بار آستانه چندگانه (MTLB)[2] را برای دستیابی به بار پیوسته در بین کنترل‌کننده‌ها پیشنهاد کرده‌اند. این طرح به طور مستقیم وضعیت بار را به طور دقیق متمایز می‌کند و به صورت پویا مقدار آستانه را هنگامی که سطح بار افزایش یا کاهش می‌یابد، تنظیم می‌کند. این روش بار را به چندین سطح تدریجی طبقه‌بندی می‌کند که نشان‌دهنده مبنایی برای مهاجرت سوییچ است. سپس یک روش همگام‌سازی اطلاعات بار بر اساس مفهوم آستانه چندسطحی طراحی می‌کند. هنگامی که بار یک کنترل‌کننده از این آستانه فراتر رفت یا نزدیک به آن شد، به کنترل‌کننده‌های دیگر اطلاع می‌دهد تا اطلاعات بار بر اساس وضعیت جدید بروز شود که به طور قابل ملاحظه‌ای سربار ناشی از همگام‌سازی اطلاعات بار

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

[2] Multiple-level Threshold Load Balancing

ناخواسته را کاهش می‌دهد. در نهایت با مطالعه دقیق رفتار سوییچ مهاجر و کنترل‌کننده هدف، روشی برای عملیات مهاجرت ارائه کرده است که تعداد دفعات مهاجرت را کاهش 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 همکاران [۳۳]، یک معماری یادگیری تقویتی عمیق (DRL)[1] برای شبکه‌های نرم‌افزار محور برای حل مساله مهاجرت سوییچ (SMP)[2] معرفی کرده‌اند که در آن حالت شبکه به عنوان یک ماتریس دوبعدی رسمیت می‌یابد. یادگیری تقویتی عمیق، ماتریس دوبعدی را به عنوان ورودی می‌گیرد و تصمیمات مهاجرت را به عنوان 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 رویکرد بازپخش تجربه آموزش داده می‌شود. پس از مرحله آموزش، مدل DDQN می‌تواند به سرعت و با دقت تصمیم بگیرد که چگونه سوییچ‌ها را بین کنترل‌کننده‌ها منتقل کند. نتایج نشان می‌دهد که راه‌حل مبتنی بر DDQN می‌تواند به طور موثر اثر تعادل بار را بهبود بخشد و زمان تعادل را کاهش دهد.

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

ژانگ و همکاران [۳۵]، برای چندین کنترل‌کننده توزیع شده، یک طرح مهاجرت سوییچ دو وزنه متعادل‌کننده بار نرم‌افزار محور مبتنی بر پیش‌بینی پیشنهاد کرده‌اند. این طرح بار ترافیک گذشته را به عنوان داده‌های تاریخی برای پیش‌بینی بار ترافیک آینده در نظر می‌گیرد. از طریق فناوری پیش‌بینی، زمان اضافه‌بار کنترل‌کننده به دست می‌آید، به طوری که عملیات مهاجرت سوییچ می‌تواند از قبل انجام شود. همچنین یک الگوریتم اطلاعات بار راه‌اندازی برای حل پردازش اضافی و سربار ارتباطی صفحه کنترل مورد نیاز برای اطلاعات بار فعال دوره‌ای بین کنترل‌کننده‌های توزیع شده، پیشنهاد شده است. با توجه به اطلاعات گذشته، این طرح پیشنهاد می‌کند که مدیریت سوییچ‌های خاص بین کنترل‌کننده‌ها منتقل شود. یک طرح مهاجرت سوییچ با وزن دوگانه نیز برای در نظر گرفتن وضعیت بار سوییچ در آینده و جلوگیری از مهاجرت مکرر سوییچ پیشنهاد شده است که فرکانس مهاجرت سوییچ را کاهش می‌دهد. آزمایش‌ها ثابت کرده‌اند که این طرح می‌تواند به سرعت بار را بین کنترل‌کننده‌ها متعادل کند و تعداد مهاجرت سوییچ‌ها را کاهش دهد.

ظفر و همکاران [۳۶]، یک رویکرد موازنه بار مبتنی بر مهاجرت سوییچ پویا (DSMLB)[6] را برای جلوگیری از سربار کنترل و توزیع کارآمد ترافیک با در نظر گرفتن دستگاه‌های ناهمگن اینترنت اشیا پیشنهاد کرده‌اند. طرح موازنه بار مبتنی بر مهاجرت سوییچ پویا پیشنهادی نسبت بار بلادرنگ به میانگین بار هر کنترل‌کننده را با توصیف معیارهای بار مختلف و انتخاب کنترل‌کننده هدف با بهینه‌سازی بازده مهاجرت با استفاده از منابع

[1] Deep Reinforcement Learning
[2] Switch Migration Problem
[3] Double Deep Q-Network
[4] Efficient Switch Migration based Load Balancing
[5] Technique for Order Preference by Similarity to an Ideal Solution
[6] Dynamic Switch Migration-based Load Balancing

کنترل‌کننده باقیمانده اندازه‌گیری می‌کند. هنگام انتخاب سوییچ‌های کاندید از کنترل‌کننده اضافه‌بار مرتبط، موازنه بار مبتنی بر مهاجرت سوییچ 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 سوییچ پویا از روش‌های مرسوم برای عملکرد متعادل بار کارآمد در یک صفحه کنترل توزیع‌شده با کاهش زمان پاسخ کنترل‌کننده 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 سوییچ پویا همچنین نرخ تعادل بار کنترل‌کننده را بهبود می‌بخشد و سربار ارتباط را کاهش می‌دهد.

در جدول (۳) مروری بر مقالات صفحه کنترل آورده شده است. مقالاتی که مربوط به تعادل بار کنترل‌کننده سلسله‌مراتبی افقی هستند خیلی بیشتر هستند و این مقالات همگی از کنترل‌کننده Floodlight یا Ryu و از محیط شبیه‌سازی Mininet استفاده کرده‌اند و هیچ‌کدام از فناوری مجازی‌سازی توابع شبکه استفاده نکرده و مصرف انرژی را نادیده گرفته‌اند. در جدول (۴) جمع‌بندی مقالات صفحه کنترل را مشاهده می‌کنید که در این مقالات متغیر زمان پاسخ بسیار حائز اهمیت است.

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

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

جدول (۴): متغیرهای مقالات مربوط به تعادل بار صفحه داده

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

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 کنترل‌کننده واحد را در نظر گرفته‌اند درصورتی‌که این تک کنترل‌کننده بودن خود باعث ایجاد گلوگاه می‌شود. از طرف دیگر، شبکه‌ها روزبه‌روز در حال گسترش هستند. پس با یک کنترل‌کننده نمی‌توان کل شبکه را مدیریت کرد؛ چون مقیاس‌پذیری و در دسترس بودن را کاهش می‌دهد. در نتیجه از معماری‌کننده توزیع شده، استفاده شده است. شبکه‌هایی که در صفحه کنترل‌کننده آنها، چندین کنترل‌کننده وجود دارد نیز ممکن است تعادل بار بین کنترل‌کننده‌ها استفاده شود. مهاجرت سوییچ باعث تعادل بار در لایه کنترل و عدم تعادل بار در سرورها می‌شود. به همین دلیل مقالاتی نوشته شده است که تلاش کرده‌اند تعادل بار صفحه داده و صفحه کنترل‌کننده را به صورت همزمان انجام دهند. تعادل بار همزمان صفحه داده و صفحه کنترل به دو دسته تعادل بار سرور و کنترل‌کننده و تعادل بار پیوند و کنترل‌کننده تقسیم می‌شود.

۱.۳.۳. تعادل بار سرور و کنترل‌کننده

الف. تعادل بار سرور پویا و کنترل‌کننده با تمرکز منطقی/ توزیع فیزیکی: گاوو و همکاران [۳۷]، یک روش همگام‌سازی حالت کنترل‌کننده به نام همگام‌سازی مبتنی بر واریانس بار (LVS)[1] را پیشنهاد کرده‌اند تا عملکرد تعادل بار را در شبکه SDN چند دامنه‌ای چند کنترل‌کننده افزایش دهند. ابتدا مشکل حلقه ارسال به دلیل اطلاعات متناقض سرور با کمترین بارگذاری بین کنترل‌کننده‌ها ایجاد می‌شود که می‌تواند منجر به از دست دادن بسته‌ها و اختلالات قابل توجهی در ارتباط شود. سپس مشکل سربار همگام‌سازی کنترل‌کننده ناشی از همگام‌سازی‌های مکرر ایجاد می‌شود. این روش، این دو مشکل را با دو طرح خاص مبتنی بر LVS حل می‌کند که شامل (۱) همگام‌سازی تغییرات سرور با حداقل بار (LSVS)[2] برای حل مشکل حلقه ارسال و (۲)همگام‌سازی تنوع دامنه با حداقل بار (LDVS)[3] برای حل مشکل سربار همگام‌سازی کنترل‌کننده، است. در مقایسه با روش‌های مبتنی بر همگام‌سازی دوره‌ای، روش‌های مبتنی بر LVS همگام‌سازی حالت واقعی را در بین کنترل‌کننده‌ها انجام می‌دهند در حالی که بار یک سرور از یک آستانه مشخص فراتر می‌رود که به طور قابل‌توجهی سربار همگام‌سازی کنترل‌کننده‌ها 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. با این حال، دو روش پیشنهادی مبتنی بر LVS در یک بستر آزمایشی واقعی ارزیابی نشده‌اند. همچنین مصرف

[1] Load Variance-based Synchronization
[2] Least loaded server variation synchronization
[3] Least loaded domain variation synchronization

انرژی و تاخیر این روش مورد ارزیابی قرار نگرفته است.

ب. تعادل بار سرور پویا و کنترل‌کننده توزیع شده: این تعادل بار در دو بخش سلسله مراتبی عمودی و افقی بررسی می‌شود.

- تعادل بار سرور پویا و کنترل‌کننده توزیع شده سلسله‌مراتبی عمودی

شائو و همکاران [۳۸]، یک روش قراردادن کنترل‌کننده قابل اعتماد بر اساس بهینه‌سازی تاخیر و بار پیشنهاد کرده‌اند. روش بهینه‌سازی چند هدفه آنها تاخیر بین کنترل‌کننده، تاخیر کنترل‌کننده به سوییچ و متعادل‌سازی بار را در نظر می‌گیرد. ابتدا هدف بهینه‌سازی معرفی می‌شود و با استفاده از الگوریتم LOCP[1] یک مدل چیدمان کنترل‌کننده قابل اعتماد بر اساس تاخیر و متعادل‌سازی بار با هدف بهینه‌سازی ساخته می‌شود. به منظور تخصیص منطقی منابع محاسباتی، الگوریتم HACA[2] به عنوان یک روش تخصیص منبع بر اساس محدودیت‌های تاخیر کار و قابلیت اطمینان و یک روش محدودیت چندهدفه مبتنی بر تاخیر انتقال و قابلیت اطمینان در نظر گرفته شده است. ابتدا هدف محدودیت معرفی می‌شود و یک مدل تخصیص منبع بر اساس محدودیت‌های تاخیر کار و قابلیت اطمینان با هدف محدودیت ساخته می‌شود. نتایج تجربی نشان می‌دهد که الگوریتم LOCP در مقایسه با سایر الگوریتم ها، تعادل بار کنترل‌کننده را ۱۸/۳۶٪ افزایش می‌دهد و در عین حال تاخیر انتشار کمتر و تاخیر در صف را تضمین می‌کند. الگوریتم HACA می‌تواند به طور موثر مشکل تخصیص بار منبع کار را حل کند و زمان پاسخ منبع کاربر و همچنین متوسط تاخیر تکمیل را کاهش دهد. معایب این روش عبارتند از (۱) هنگام در نظر گرفتن معیارهای بهینه‌سازی برای قراردادن کنترل‌کننده، تعداد معیارهای بهینه‌سازی در نظر گرفته شده به طور جامع کم است، (۲) در مشکل قرار دادن کنترل‌کننده قابل‌اعتماد، خرابی اجزای شبکه را می‌توان بیش از خرابی‌های تک پیوند در نظر گرفت.

- تعادل بار سرور پویا و کنترل‌کننده توزیع شده سلسله‌مراتبی افقی

در پژوهش دیگر یک الگوریتم مهاجرت سوییچ متعادل‌کننده بار آگاه از QoS پویا (LBSMT) پیشنهاد شده است. هنگامی که بین سوییچ‌ها و دامنه‌های مختلف ارتباط درونی وجود دارد، مهاجرت سوییچ‌ها صورت می‌گیرد و بار روی سرورها نامتعادل می‌شود. عدم تعادل بار باعث افزایش زمان پاسخ و کاهش توان عملیاتی می‌شود. هر سوییچ در معماری کنترل‌کننده توزیع شده توسط یک کنترل‌کننده در دامنه محلی مدیریت می‌شود. کنترل‌کننده‌ها جریان‌های ترافیک را مدیریت می‌کنند و با افزایش ترافیک اینترنت، جریان بین کنترل‌کننده و سوییچ overload می‌شود، در حالی که تعداد کمی از کنترل‌کننده‌ها underload می‌شوند. به دلیل کاهش توان و زمان پاسخ طولانی، چنین عدم تعادلی بر اثربخشی شبکه‌های نرم‌افزار محور تاثیر می‌گذارد. مهاجرت سوییچ به طور معمول برای تغییر تخصیص بارهای کنترل‌کننده با جابه‌جایی سوییچ از یک کنترل‌کننده overload به یک کنترل‌کننده underload انجام می‌شود. در شبکه چندین دامنه وجود دارد که هر دامنه از یک کنترل‌کننده و چند سوییچ تشکیل شده است. به بار تمام دامنه‌ها مقادیر آستانه در KB/sec اختصاص داده می‌شود و اگر تعداد بازدیدها در یک دامنه از یک سوییچ خاص از مقدار آستانه عبور کرده باشد، بار به سوییچ دیگری از آن دامنه منتقل می‌شود و اگر یک دامنه دچار اضافه‌بار شود؛ یعنی تمام سوییچ‌های دامنه از مقدار آستانه عبور کرده باشند، در این صورت مهاجرت سوییچ از یک دامنه به دامنه دیگر صورت می‌گیرد. هدف این طرح نشان دادن مهاجرت سوییچ برای یافتن هزینه مهاجرت و نرخ تعادل بار است. این محاسبات به دو صورت (۱) درون دامنه و (۲) بین دامنه انجام می‌شود. رویکرد پیشنهادی زمان پاسخگویی را ۲ برابر نسبت به Round Robin و LBBSRT، توان عملیاتی را ۰/۴ برابر نسبت به Round Robin و ۰/۵ برابر از LBBSRT، استفاده از CPU را ۰/۶ برابر از Round Robin و LBBSRT و استفاده از حافظه را ۰/۲ برابر از Round Robin و LBBSRT بهبود بخشیده است. اما در این طرح سرورهای مختلف در دامنه‌ها به صورت همگن در نظر گرفته شده‌اند [۳۹].

[1] Multi-controller placement
[2] Heuristic ant colony algorithm

۲.۳.۳. تعادل بار پیوند و کنترل‌کننده

وانگ و همکاران [۴۰]، مسیریابی تعادل بار را برای مشکل پیوندها و کنترل‌کننده‌ها (LBR-LC)[1] در یک شبکه‌های مبتنی بر نرم‌افزار فرموله کرده و NP-hard بودن آن را ثابت کرده‌اند. آنها یک الگوریتم مبتنی بر گرد کردن[2] برای حل این مشکل پیشنهاد و عملکرد تقریبی آن را تحلیل کرده‌اند. علاوه بر این، آنها در مورد رویکرد کارآمد برای نگهداری وضعیت شبکه در میان کنترل‌کننده‌های توزیع شده، بحثی صورت داده‌اند. نتایج شبیه‌سازی گسترده نشان می‌دهد که الگوریتم پیشنهادی آنها می‌تواند حداکثر زمان پاسخ کنترل‌کننده را تا ۷۰ درصد در مقایسه با راه‌حل قبلی کاهش دهد، در حالی که تنها حداکثر بار پیوند را تا ۳ درصد افزایش می‌دهد. مزایای این روش تعادل بار بهتر و کیفیت سرویس بهتر است. محدودیت این روش هم این است که بار فراتر از آستانه افزایش می‌یابد. در جدول (۵) و (۶) جمع‌بندی مقالات تعادل بار همزمان صفحه داده و صفحه کنترل آورده شده است. این مقالات مصرف انرژی را نادیده گرفته‌اند و همگی از کنترل‌کننده‌های POX و NOX و Ryu استفاده کرده‌اند که همگی آنها معماری متمرکز یا توزیع شده دارند.

۴. تعادل بار شبکه‌های نرم‌افزار محور مبتنی بر بهینه‌سازی

در تعادل بار شبکه‌های نرم‌افزار محور هدف حل مساله‌های تعادل بار سرور، پیوند و کنترل‌کننده است. در برخی از مقالات سعی شده است که این مسائل را با کمک بهینه‌سازی حل کنند، زیرا بهینه‌سازی فرآیندی است که به کمک آن بهترین پاسخ برای یک مساله به دست می‌آید. الگوریتم‌های بهینه‌سازی به دو دسته کلی تقسیم می‌شوند: (۱) الگوریتم‌های هوش مصنوعی و (۲) الگوریتم‌های ریاضی. در این مقاله تاکید بر روی الگوریتم‌های هوش مصنوعی در تعادل بار شبکه‌های نرم‌افزار محور است، لذا در ادامه روی این بخش تمرکز می‌شود.

۱.۴. الگوریتم‌های هوش مصنوعی

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 برای حل مساله تعادل بار در شبکه‌های نرم‌افزار محور استفاده می‌شود جایی که مساله تعادل بار خود NP-Hard نیز است. از طرفی، الگوریتم‌های هوش مصنوعی به دو دسته، یادگیری ماشین و فرااابتکاری تقسیم می‌شود [۵].

۱.۱.۴. یادگیری ماشین

الگوریتم‌های یادگیری ماشین، برنامه‌هایی هستند که بر اساس داده‌های موجود در محیط و تجربیات گذشته خود آموزش داده می‌شوند [۵].

سان و همکاران [۴۱]، یک طرح متعادل‌کننده کنترل‌کننده پویا با استفاده از رویکرد یادگیری ماشین پیشنهاد کرده‌اند. هدف الگوریتم یادگیری ماشین در این مقاله، مقابله با مهاجرت سوییچ است، به این صورت که ورودی این الگوریتم صفحه کنترل و خروجی آن تصمیم‌گیری در مورد مهاجرت سوییچ است. این الگوریتم در دو مرحله کار می‌کند. در مرحله اول یاد می‌گیرد چگونه سوییچ‌ها را منتقل کند و در مرحله دوم در مورد مهاجرت سوییچ‌ها تصمیم‌گیری می‌کند.

الف. یادگیری تقویتی: یادگیری تقویتی یک رویکرد 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 پیوندها استفاده شده است، به این صورت که ابتدا پیوندها در جدول یادگیری قرار می‌گیرند. در ادامه بهترین پیوند برای جریان جدید بر اساس متغیر تاخیر انتخاب می‌شود، یعنی مسیری که تاخیر کمتری داشته باشد، در اولویت بالاتری برای انتخاب قرار دارد.

[1] Load-Balancing Routing for both Links and Controllers
[2] Rounding-based

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

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

جدول (۶): متغیرهای مقالات تعادل بار صفحه داده و صفحه کنترل

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

فیلالی و همکاران [۴۳]، یک الگوریتم یادگیری تقویتی برای حل مشکل تعادل بار بین کنترل‌کننده‌ها و هزینه‌های عملیات مهاجرت که بر اساس زمان پاسخ کنترل‌کننده است را 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 سوییچ است و عمل برنده دوم تعادل بارکم و انجام مهاجرت سوییچ است. اولویت عمل برنده اول بیشتر از عمل برنده دوم بوده و عمل بازنده تعادل بار زیاد و انجام مهاجرت زیاد است که در این صورت تعادل بار کنترل‌کننده در شبکه نرم‌افزار محور وجود ندارد؛ اما هزینه مهاجرت بسیار زیادی وجود دارد. این الگوریتم بر اساس اعمال ذکر شده در بالا تصمیماتی را می‌گیرد که بر اساس آنها یا پاداش می‌گیرد، یا جریمه می‌شود. بر اساس همین پاداش‌ها و جریمه‌ها تجربه کسب می‌کند.

ب. یادگیری عمیق: یادگیری عمیق زیرشاخه‌ای از یادگیری ماشین است که هدف آن ابداع ماشین‌هایی است که حتی در موقعیت‌های غیرمنتظره با محیط تعامل داشته باشد [۵]. در مقاله [۳۳]، از یادگیری عمیق برای حل مساله مهاجرت سوییچ استفاده شده است. به این صورت که حالت شبکه به عنوان ورودی به الگوریتم یادگیری عمیق داده شده و تصمیمات مهاجرت به عنوان خروجی تولید می‌شوند.

ج. شبکه عصبی مصنوعی: شبکه عصبی مصنوعی به طور عمده از مغز انسان برای حل مسائل پیچیده الهام می‌گیرد [۴۴]. تعادل بار شبکه‌های نرم‌افزار محور نیز از همین ویژگی شبکه‌های عصبی برای حل مسائل استفاده می‌کند. محبوب‌ترین رویکرد

یادگیری عمیق، شبکه عصبی مصنوعی است [۵]. در مقاله [۲۵]، برای تعادل بار سرورها در شبکه‌های نرم‌افزار محور از الگوریتم شبکه عصبی مصنوعی استفاده شده است، به این صورت که داده‌های سرورها جمع‌آوری شده و سپس الگوریتم‌های Random، Round Robin، Weighted Round Robin و Least Connection روی متعادل‌کننده بار به منظور به دست آوردن متغیرهای سرور و برچسب به سرور اجرا می‌شوند. در ادامه همه متغیرها به عنوان ورودی به الگوریتم داده می‌شوند تا الگویی را بر اساس داده‌ها یاد بگیرد و از طریق آن مدل را بسازد. در مقاله [۴۵]، تعادل بار هوشمند پیشنهاد شده است که شامل تعادل بار پیوند، تعادل‌سازی بار سرور و طبقه‌بندی ترافیک است. در تعادل بار پیشنهادی، برای طبقه‌بندی ترافیک از شبکه عصبی مصنوعی استفاده می‌شود.

۲.۱.۴. الگوریتم‌های فراابتکاری

الگوریتم‌های فراابتکاری برای حل مسائل بهینه‌سازی پیچیده که در زمان معقول قابل‌حل نیستند، مناسب هستند [۴۶]. به همین دلیل برای حل مسائل تعادل بار شبکه‌های نرم‌افزار محور که پیچیده هستند، مناسب می‌باشند. الگوریتم فراابتکاری انواع مختلفی دارد که در ادامه به شرح برخی از آنها پرداخته می‌شود.

الگوریتم ژنتیک (GA)[1]: الگوریتم ژنتیک یک روش برنامه‌نویسی است که می‌تواند فضای راه‌حل را با استفاده از مفهوم برگرفته از ژنتیک طبیعی و نظریه تکامل کاوش کند [۵]، [۴۷]. سانر از الگوریتم ژنتیک در مرجع [۴۸] برای یافتن بهترین موقعیت کنترل‌کننده برای به حداقل رساندن تاخیر سوییچ برای کنترل‌کننده استفاده می‌کند. در مقاله [۲۲]، برای صرفه‌جویی در مصرف انرژی از الگوریتم ژنتیک استفاده شده است، به این صورت که پیکربندی شبکه در یک کروموزوم نگاشت شده و محدودیت‌ها شامل ظرفیت پیوندها و ترافیک موجود است. هدف به حداقل رساندن تعداد پیوندهای فعال برای اطمینان از صرفه‌جویی در انرژی است.

الگوریتم کلونی مورچه‌ها (ACO)[2]: این الگوریتم به شبیه‌سازی رفتار مورچه‌ها برای حل مسائل می‌پردازد [۴۹]. در مقاله [۲۷]، برای تعادل بار سرور و پیوند از این روش استفاده شده است، به این صورت که اطلاعات سرورها و پیوندها به این الگوریتم داده شده و سپس سروری با کمترین میزان بار انتخاب می‌شود و برای یافتن بهترین مسیر توسط این الگوریتم، مورچه‌هایی که کوتاهترین مسیر را دارند، فرومونهای بیشتری را آزاد می‌کنند و در نهایت همه مورچه‌ها به آن مسیر می‌روند. به این ترتیب کوتاهترین مسیر به سرور مورد نظر یافت می‌شود. در مقاله [۳۸]، مساله تخصیص منبع به عنوان یک مساله بهینه‌سازی چند هدفه مدل‌سازی و مساله با استفاده از یک الگوریتم کلونی مورچه اکتشافی حل شده است. ابتدا تعیین شده است که شبکه دارای m سرور با n نوع درخواست محاسباتی مربوط به n نوع واحد محاسباتی است. تقاضای هر نوع واحد محاسباتی باید با گرم بودن یک درخواست جمله تعیین شود و منابع n نوع واحد محاسباتی با استفاده از یک الگوریتم کلونی مورچه بهبود یافته تخصیص داده می‌شود.

الگوریتم بهینه‌سازی ازدحام ذرات(PSO)[3]: این الگوریتم رفتارهای ازدحام پرنده را شبیه‌سازی می‌کند [۵۰]. در مقاله [۵۱]، با در نظر گرفتن تاخیر بین کنترل‌کننده‌ها و ظرفیت کنترل‌کننده مشکل قرارگیری کنترل‌کننده را با استفاده از الگوریتم بهینه‌سازی ازدحام ذرات بررسی کرده‌اند. این الگوریتم با تکرار چندین ذره به صورت موازی راه‌حل بهینه را جستجو می‌کند و قرارگیری کنترل‌کننده با تاخیر کم تعیین می‌شود. در مقاله [۵۲] از الگوریتم ازدحام ذرات برای خوشه‌بندی دامنه کنترل‌کننده‌های شبکه‌های نرم‌افزار محور استفاده شده است، زیرا بخش‌بندی نامعقول دامنه کنترل‌کننده‌های متفاوت خود باعث عدم تعادل بار کنترل‌کننده‌ها می‌شود.

استفاده ترکیبی از الگوریتم ژنتیک و الگوریتم ازدحام ذرات: در مقاله [۵۳]، مساله قرارگیری کنترل‌کننده توزیع شده و راه‌حل برای کاهش تاخیر سوییچ به کنترل‌کننده، تاخیر کنترل‌کننده به

[1] Genetic Algorithm

[2] Ant Colony Algorithm

[3] Particle Swarm Optimization

کنترل‌کننده و عدم تعادل بار کنترل‌کننده در شبکه‌های نرم‌افزار محور پیشنهاد شده است. برای حل این مشکلات از الگوریتم ژنتیک چندهدفه استفاده شده است. اما مشکل این الگوریتم این است که زمان همگرایی طولانی دارد. در نتیجه، از الگوریتم ازدحام ذرات برای کاهش زمان همگرایی استفاده شده است، چون الگوریتم ازدحام ذرات با دادن بهترین موقعیت مکانی برای هدایت هر ذره می‌تواند راه‌حل بهینه را در زمان همگرایی کوتاه بیابد.

کبیری و همکاران [۵۴]، الگوریتمی برای استفاده از کنترل‌کننده و آستانه متغیر با مهاجرت کمتر و بهبود شبکه با کمک الگوریتم ژنتیک و ازدحام ذرات به منظور کاهش هزینه مهاجرت و زمان پاسخ پیشنهاد کرده‌اند. در روش پیشنهادی آنها، ابتدا با استفاده از یک آستانه متعادل بودن یا نامتعادل بودن کنترل‌کننده تشخیص داده شده و سپس از الگوریتم ازدحام ذرات و ژنتیک برای مهاجرت سوییچ استفاده می‌شود، به این صورت که ابتدا با کمک الگوریتم ژنتیک کنترل‌کننده‌های هدف، سوییچ‌های مهاجر و تابع هزینه جمع‌آوری می‌شوند. این داده‌ها به عنوان ورودی به الگوریتم ازدحام ذرات داده شده و سپس الگوریتم ازدحام ذرات بهترین و بهینه‌ترین راه‌حل را برای انتقال سوییچ به کنترل‌کننده 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 کنترل‌کننده‌ها در صفحه کنترل و عملیات مهاجرت سوییچ از ترکیب دو الگوریتم استفاده کرده‌اند. در مقاله [۲۵]، به صورت کاملا جداگانه از دو الگوریتم یادگیری تقویتی و شبکه عصبی مصنوعی استفاده شده است. اما در مقالات [۵۳] و [۵۴] از ترکیب دو الگوریتم ژنتیک و ازدحام ذرات برای تعادل بار کنترل‌کننده و مهاجرت سوییچ استفاده کرده‌اند.

۵. پیش‌بینی در تعادل بار شبکه‌های نرم‌افزار محور

در تعادل بار شبکه‌های نرم‌افزار محور از روش‌های پیش‌بینی، برای پیش‌بینی بار کنترل‌کننده، بار سرور و ترافیک استفاده می‌شود. در نتیجه باعث ایجاد تعادل بار شده و هم در مصرف انرژی صرفه‌جویی می‌شود. روش‌های متفاوتی برای پیش‌بینی وجود دارد اما در تعادل بار شبکه‌های نرم‌افزار محور از روش‌های پیش‌بینی‌های مبتنی بر سری‌های زمانی بیشتر استفاده می‌شود، زیرا در پیش‌بینی سری زمانی برای مدیریت منابع یا متعادل‌سازی بار، به صورت 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. از روش‌های پیش‌بینی سری می‌توان به میانگین متحرک یکپارچه اتورگرسیون (ARIMA)، حداقل میانگین مربعات نرمال‌شده (NLMS)[1] و حافظه بلندمدت کوتاه‌مدت (LSTM)[2] اشاره کرد.

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

مرجع	سال	الگوریتم بهینه‌سازی	توضیحات
[۴۱]	۲۰۲۰	یادگیری ماشین	تعادل بار کنترل‌کننده
[۲۵]	۲۰۲۲	یادگیری تقویتی	تعادل بار پیوند
[۴۳]	۲۰۲۰	یادگیری تقویتی	تعادل بار کنترل‌کننده
[۳۳]	۲۰۲۲	یادگیری عمیق	مهاجرت سوییچ
[۲۵]	۲۰۲۲	شبکه عصبی مصنوعی	تعادل بار سرور
[۴۵]	۲۰۲۱	شبکه عصبی مصنوعی	طبقه‌بندی ترافیک
[۴۸]	۲۰۱۶	ژنتیک	تعادل بار کنترل‌کننده
[۲۲]	۲۰۲۳	ژنتیک	تعادل بار پیوند
[۳۸]	۲۰۲۲	کلونی مورچه‌ها	تعادل بار سرور
[۲۷]	۲۰۲۳	کلونی مورچه‌ها	تعادل بار سرور و پیوند
[۵۱]	۲۰۱۵	ازدحام ذرات	تعادل بار کنترل‌کننده
[۵۳]	۲۰۱۹	ازدحام ذرات	خوشه‌بندی کنترل‌کننده
[۵۳]	۲۰۱۷	ژنتیک و ازدحام ذرات	تعادل بار کنترل‌کننده
[۵۴]	۲۰۲۲	ژنتیک و ازدحام ذرات	تعادل بار کنترل‌کننده

[1] Normalised Least Connection Mean Squares
[2] Long Short-Term Memory

حداقل میانگین مربعات نرمال‌شده: در پیش‌بینی حداقل میانگین مربع نرمال‌شده بر اساس حجم کاری گذشته، آینده را پیش‌بینی می‌کنند [۵۵]. در مقاله [۲۴]، نویسندگان برای ایجاد تعادل بار سرورها ابتدا ترافیک را به سه دسته تقسیم کرده و سپس بر اساس حداقل میانگین مربعات نرمال‌شده بار سرورها را پیش‌بینی کرده‌اند. آنها بر اساس این پیش‌بینی ترافیک را بین سرورها تقسیم کردند. در نتیجه تعادل بار بین سرورها ایجاد شده و سروری دچار اضافه‌بار نمی‌شود. در مقاله [۲۶]، برای پیش‌بینی میزان بار سرورها از حداقل میانگین مربع نرمال‌شده استفاده شده است تا در مراحل بعد با استفاده از این پیش‌بینی بار بین سرورها متعادل شود و میزان مصرف انرژی کاهش یابد، به این صورت که با پیش‌بینی میزان بار سرورها، سرورهای با بار کم خاموش می‌شوند و مصرف انرژی کاهش می‌یابد. در صورت ایجاد اضافه‌بار، برای ایجاد تعادل بار، سرورهایی اضافه می‌شوند.

میانگین متحرک یکپارچه اتورگرسیون: در مدل میانگین متحرک یکپارچه اتورگرسیون مقادیر آینده یک متغیر قرار است ترکیبی خطی از حجم کاری گذشته و خطاهای تصادفی باشد [۵۶]. در مقاله [۴]، از میانگین متحرک یکپارچه اتورگرسیون برای پیش‌بینی میزان بار سوییچ و جلوگیری از اضافه‌بار کنترل‌کننده استفاده شده است، به این صورت که یک الگوریتم زمان‌بندی مهاجرت سوییچ برای انتقال برخی از سوییچ‌ها از کنترل‌کننده اضافه‌بار به کنترل‌کننده کم‌بار استفاده شده است.

حافظه بلندمدت کوتاه‌مدت: یک فرم شبکه عصبی بازگشتی است و توانایی قدرتمندی برای به خاطر سپردن پیشینه به منظور پیش‌بینی را دارد [۵۷]. در مقاله [۱۷]، از حافظه بلندمدت کوتاه‌مدت برای پیش‌بینی ترافیک برای ایجاد تعادل بار در پیوندها و بهینه‌سازی مصرف انرژی شبکه استفاده شده است، به این صورت که با پیش‌بینی ترافیک دستگاه‌های با بار کم خاموش می‌شوند تا مصرف انرژی کاهش یابد. در صورت ایجاد اضافه‌بار در آینده دستگاه‌هایی برای ایجاد تعادل بار روشن می‌شوند. در مقاله [۳۵]، حافظه بلندمدت کوتاه‌مدت برای پیش‌بینی ترافیک به منظور متعادل‌کردن کنترل‌کننده‌ها استفاده شده است، به این صورت که اگر امکان اضافه‌بار در آینده وجود داشته باشد، عملیات تعادل بار انجام می‌شود.

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

در جدول (۸) خلاصه‌ای از مقالات پیش‌بینی در تعادل بار شبکه‌های نرم‌افزار محور آورده شده است. در سال‌های اخیر از روش‌های پیش‌بینی برای ایجاد تعادل بار در شبکه‌های نرم‌افزار محور بسیار استفاده شده است و همانطور که مشاهده می‌کنید از پیش‌بینی در تمام زمینه‌های تعادل بار شبکه‌های نرم‌افزار محور می‌توان استفاده کرد. بیشتر مقالاتی که از روش پیش‌بینی برای تعادل برای شبکه‌های نرم‌افزار محور استفاده کرده‌اند، در مصرف انرژی نیز صرفه‌جویی کرده‌اند، در نتیجه پیش‌بینی بار بر میزان مصرف انرژی بسیار تاثیرگذار است.

جدول (۸) : مروری بر مقالات پیش‌بینی در تعادل بار شبکه‌های نرم‌افزار محور

مرجع	سال	NLMS	ARIMA	LSTM	پیش بینی	توضیحات
[۲۴]	۲۰۲۰	✓	×	×	پیش‌بینی بار سرور	تعادل بار سرور
[۲۷]	۲۰۲۱	✓	×	×	پیش‌بینی بار سرور	تعادل بار سرور
[۴]	۲۰۱۹	×	✓	×	پیش‌بینی بار سوییچ	تعادل بار کنترل‌کننده
[۱۷]	۲۰۲۰	×	×	✓	پیش‌بینی ترافیک	تعادل بار پیوند
[۳۵]	۲۰۲۲	×	×	✓	پیش‌بینی ترافیک	تعادل بار سرور
[۴۳]	۲۰۲۰	×	✓	✓	پیش‌بینی بار کنترل‌کننده	تعادل بار کنترل‌کننده

۶. نتیجه‌گیری

رویکردهای بسیاری برای تعادل بار شبکه‌های نرم‌افزار محور وجود دارد. در این مقاله دسته‌بندی دقیق‌تری از این رویکردها ارائه شده است. به صورت کلی این رویکردها به سه دسته تعادل بار صفحه داده، تعادل بار صفحه کنترل و تعادل بار همزمان صفحه داده و صفحه کنترل تقسیم می‌شوند. هر کدام از این در این مقاله موارد مورد بحث قرار گرفته است. همچنین روش‌های بهینه‌سازی مبتنی بر هوش مصنوعی تعادل بار شبکه‌های نرم‌افزار محور مورد بررسی قرار گرفته‌اند. در نهایت، روش‌های پیش‌بینی تعادل بار شبکه‌های نرم‌افزار محور بیان شده‌اند.

تعارض منافع: نویسندگان اعلام می‌کنند که هیچ تعارض منافعی ندارند.

مراجع

[1] M. Priyadarsini and P. Bera, “Software defined networking architecture, traffic management, security, and placement: A survey,” Comput. Networks, vol. 192, p. 108047, 2021, doi: 10.1016/j.comnet.2021.108047.

[2] L. Zhu, M.M. Karim, K. Sharif, F. Li, X. Du, and M. Guizani, “SDN controllers: Benchmarking & performance evaluation,” arXiv preprint arXiv: 1902.04491, 2019, doi: 10.48550/arXiv.1902.04491.

[3] M. Hamdan, et al., “A comprehensive survey of load balancing techniques in software-defined network,” J. Netw. Comput. Appl., vol. 174, p. 102856, 2021, doi: 10.1016/j.jnca.2020.102856.

[4] A. Filali, S. Cherkaoui, and A. Kobbane. “Prediction-based switch migration scheduling for SDN load balancing,” in Int. Conf. Commun. (ICC), Shanghai, China, 2019, pp. 1-6, doi: 10.1109/ICC.2019.8761469.

[5] M. Latah and L. Toker, “Artificial intelligence enabled software-defined networking: a comprehensive overview,” IET Networks, vol. 8, pp. 79-99, 2019, doi: 10.1049/iet-net.2018.5082.

[6] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, “Network function virtualization: Challenges and opportunities for innovations,” IEEE Commun. Mag., vol. 53, no. 2, pp. 90-97, 2015, doi: 10.1109/MCOM.2015.7045396.

[7] Y. Li and M. Chen, “Software-defined network function virtualization: A survey,” IEEE Access, vol. 3, pp. 2542-2553, 2015, doi: 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 Commun. Surv. Tutorials, vol. 16, no. 4, pp. 2181-2206, 2014, doi: 10.1109/COMST.2014.2326417.

[9] B. Yi, X. Wang, K. Li, S.K. Das, and M. Huang, “A comprehensive survey of network function virtualization,” Comput. Networks, vol. 133, pp. 212-262, 2018, doi: 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,” in Int. Joint Conf. Awareness Sci. Technol. Ubi-Media Comput. (iCAST UMEDIA), Aizuwakamatsu, Japan, 2013, pp. 691-697, doi: 10.1109/ICAwST.2013.6765526.

[11] R. Leland and B. Hendrickson, “An empirical study of static load balancing algorithms,” in Proc. IEEE Scal. High Perform. Comput. Conf., Knoxville, TN, USA, 1994, pp. 682-685, doi: 10.1109/SHPCC.1994.296707.

[12] L.D.D. Babu and P.V. Krishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments,” Appl. Soft Comput., vol. 13, no. 5, pp. 2292-2303, 2013, doi: 10.1016/j.asoc.2013.01.025.

[13] S. Kaur, K. Kumar, J. Singh, and N.S. Ghumman, “Round-robin based load balancing in Software Defined Networking,” in 2nd Int. Conf. Comput. Sustai. Global Dev. (INDIACom), New Delhi, India, 2015, pp. 2136-2139.

[14] H. Zhong, Y. Fang, and J. Cui, “Reprint of LBBSRT: An efficient SDN load balancing scheme based on server response time,” Future Gener. Comput. Syst., vol. 80, pp. 409-416, 2018, doi: 10.1016/j.future.2017.11.012.

[15] A. Montazerolghaem, “Analysis and Modeling of VoIP Servers: A Linear Programming Approach,” Soft Comput. J., vol. 7, no. 2, pp. 2-23, 2019, dor:

20.1001.1.23223707.1397.7.2.1.2 [In Persian].

[16] A. Montazerolghaem and M.H.Y. Moghadam, "Improving efficiency of SIP protocol using window-based overload conditions," Soft Comput. J., vol. 2, no. 2, pp. 16-25, 2014, dor: 20.1001.1.23223707.1392.2.2.54.0 [In Persian].

[17] X. Chen, X. Wang, B. Yi, Q. He, and M. Huang, "Deep learning-based traffic prediction for energy efficiency optimization in software-defined networking," IEEE Syst. J., vol. 15, no. 4, pp. 5583-5594, 2021, doi: 10.1109/JSYST.2020.3009315.

[18] L.H. Binh and T.-V.T. Duong, "Load balancing routing under constraints of quality of transmission in mesh wireless network based on software defined networking," J. Commun. Networks, vol. 23, no. 1, pp. 12-22, 2021, doi: 10.23919/JCN.2021.000004.

[19] Y.-C. Wang and S.-Y. You, "An efficient route management framework for load balance and overhead reduction in SDN-based data center networks," IEEE Trans. Netw. Serv. Manag., vol. 15, no. 4, pp. 1422-1434, 2018, doi: 10.1109/TNSM.2018.2872054.

[20] X. Han, X. Meng, Z. Yu, Q.-Y. Kang, and Y. Zhao, "A service function chain deployment method based on network flow theory for load balance in operator networks," IEEE Access, vol. 8, pp. 93187-93199, 2020, doi: 10.1109/ACCESS.2020.2994912.

[21] T.S. Andjamba and G.-A. Lusilao-Zodi, "A Load Balancing Protocol for Improved Video on Demand in SDN-Based Clouds," in 17th Int. Conf. Ubiquitous Info. Manag. Commun. (IMCOM), Seoul, Korea, Republic of, 2023, pp. 1-6, doi: 10.1109/IMCOM56909.2023.10035591.

[22] J. Galan-Jimenez, M. Polverini, F.G. Lavacca, J.L. Herrera, and J. Berrocal, "Joint energy efficiency and load balancing optimization in hybrid IP/SDN networks." Ann. Telecommunications, vol. 78, no. 1-2, pp. 13-31, 2023, doi: 10.1007/s12243-022-00921-y.

[23] M. Khorasani, M. Ramezanpour, and R. Khorsand, "Energy Efficient Multi Path Routing Protocol in Internet of Things," Soft Comput. J., vol. 7, no. 1, pp. 34-49, 2018, doi: 10.22052/7.1.34 [In Persian].

[24] A. Montazerolghaem and M.H.Y. Moghaddam, "Load-balanced and QoS-aware Software-defined Internet of Things," IEEE Internet Things J., vol. 7, no. 4, pp. 3323-3337, 2020, doi: 10.1109/JIOT.2020.2967081.

[25] P.M. Abhishek, A. Naik, P. Doddannavar, R. Patil, M.M. Raikar, and S.M. Meena, "Load Balancing for Network Resource Management in Software-Defined Networks," in Advances in Distributed Computing and Machine Learning. Lecture Notes in Networks and Systems, vol 427, Springer, Singapore, 2022, doi: 10.1007/978-981-19-1018-0_17.

[26] A. Montazerolghaem, "Software-defined Internet of Multimedia Things: Energy-efficient and Load-balanced Resource Management," IEEE Internet Things J., vol. 9, no. 3, pp. 2432-2442, 2022, doi: 10.1109/JIOT.2021.3095237.

[27] H. Zheng, J. Guo, Q. Zhou, Y. Peng, and Y. Chen, "Application of improved ant colony algorithm in load balancing of software-defined networks," J. Supercomput., vol. 79, no. 7, pp. 7438-7460, 2023, doi: 10.1007/s11227-022-04957-8.

[28] S.H. Yeganeh and Y. Ganjali, "Kandoo: a framework for efficient and scalable offloading of control applications," in Proc. 1st Workshop Hot Topics Softw. Defined Networks (HotSDN), Helsinki, Finland, 2012, pp. 19-24, doi: 10.1145/2342441.2342446.

[29] 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, doi: 10.1109/ACCESS.2019.2909356.

[30] L. Zhang, G. Shou, Y. Hu, and Z. Guo, "Deployment of intrusion prevention system based on software defined networking," in 15th IEEE Int. Conf. Commun. Technol. (ICCT), Guilin, China, 2013, pp. 26-31, doi: 10.1109/ICCT.2013.6820345.

[31] J. Cui, Q. Lu, H. Zhong, M. Tian, and L. Liu, "A load-balancing mechanism for distributed SDN control plane using response time," IEEE Trans. Netw. Serv. Manag., vol. 15, no. 4, pp. 1197-1206, 2018, doi: 10.1109/TNSM.2018.2876369.

[32] H. Mokhtar, X. Di, Y. Zhou, A. Hassan, Z. Ma, and S. Musa, "Multiple-level threshold load balancing in distributed SDN controllers," Comput. Networks, vol. 198, p. 108369, 2021, doi: 10.1016/j.comnet.2021.108369.

[33] M. Xiang, M. Chen, D. Wang, and Z. Luo, "Deep Reinforcement Learning-based load balancing strategy for multiple controllers in SDN," e-Prime-Adv. Electr. Eng. Electron. Energy, vol. 2, p. 100038, 2022, doi: 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 Things J., vol. 7, no. 7, pp.

5852-5860, 2020, doi: 10.1109/JIOT.2019.2952527.

[35] H. Zhong, J. Xu, J. Cui, X. Sun, C. Gu, and L. Liu, "Prediction-based dual-weight switch migration scheme for SDN load balancing," Comput. Networks, vol. 205, p. 108749, 2022, doi: 10.1016/j.comnet.2021.108749.

[36] S. Zafar, Z. Lv, N.H. Zaydi, M. Ibrar, and X. Hu, "DSMLB: Dynamic switch-migration based load balancing for software-defined IoT network," Comput. Networks, vol. 214, p. 109145, 2022, doi: 10.1016/j.comnet.2022.109145.

[37] Z. Guo, M. Su, Y. Xu, Z. Duan, L. Wang, S. Hui, and H.J. Chao, "Improving the performance of load balancing in software-defined networks through load variance-based synchronization," Comput. Networks, vol. 68, pp. 95-109, 2014, doi: 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," Knowl. Based Syst., vol. 241, p. 108330, 2022, doi: 10.1016/j.knosys.2022.108330.

[39] H. Babbar, S. Rani, A.K. Bashir, and R. Nawaz, "Lbsmt: Load balancing switch migration algorithm for cooperative communication intelligent transportation systems," IEEE Trans. Green Commun. Netw., vol. 6, no. 3, pp. 1386-1395, 2022, doi: 10.1109/TGCN.2022.3162237.

[40] H. Wang, H. Xu, L. Huang, J. Wang, and X. Yang, "Load-balancing routing in software defined networks with multiple controllers," Comput. Networks, vol. 141, pp. 82-91, 2018, doi: 10.1016/j.comnet.2018.05.012.

[41] P. Sun, Z. Guo, G. Wang, J. Lan, and Y. Hu, "MARVEL: Enabling controller load balancing in software-defined networks with multi-agent reinforcement learning," Comput. Networks, vol. 177, p. 107230, 2020, doi: 10.1016/j.comnet.2020.107230.

[42] N.C. Luong, D.T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D.I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," IEEE Commun. Surv. Tutorials, vol. 21, no. 4, pp. 3133-3174, 2019, doi: 10.1109/COMST.2019.2916583.

[43] A. Filali, Z. Mlika, S. Cherkaoui, and A. Kobbane, "Preemptive SDN load balancing with machine learning for delay sensitive applications," IEEE Trans. Veh. Technol., vol. 69, no. 12, pp. 15947-15963, 2020, doi: 10.1109/TVT.2020.3038918.

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

[45] C. Fancy and M. Pushpalatha, "RETRACTED ARTICLE: Proactive Load Balancing Strategy Towards Intelligence-Enabled Software-Defined Network," Arab. J. Sci. Eng., vol. 48, p. 2577, 2023, doi: 10.1007/s13369-021-05621-8.

[46] X. Yang, "Metaheuristic optimization: algorithm analysis and open problems," in Exp. Algorithms: 10th Int. Symp. (SEA), Kolimpari, Chania, Crete, Greece, 2011, pp. 21-32, doi: 10.1007/978-3-642-20662-7_2.

[47] M.A. El-Baz, "A genetic algorithm for facility layout problems of different manufacturing environments," Comput. Ind. Eng., vol. 47, no. 2-3, pp. 233-246, 2004, doi: 10.1016/j.cie.2004.07.001.

[48] J.-M. Sanner, M. Ouzzif, Y. Hadjadj-Aoul, and J.-E. Dartois, "Evolutionary algorithms for optimized SDN controllers & NVFs' placement in SDN networks," in SDN Day. 2016, Paris, France, 2016.

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

[50] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," Swarm. Intell., vol. 1, pp. 33-57, 2007, doi: 10.1007/s11721-007-0002-0.

[51] C. Gao, H. Wang, F. Zhu, L. Zhai, and S. Yi, "A particle swarm optimization algorithm for controller placement problem in software defined network," in Algor. Archit. Parallel Process. 15th Int. Conf. (ICA3PP), Zhangjiajie, China, 2015, pp. 44-54, doi: 10.1007/978-3-319-27137-8_4.

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

[53] L. Liao and V.C.M. Leung, "Genetic algorithms with particle swarm optimization based mutation for distributed controller placement in SDNs," in IEEE Conf. Netw. Func. Virtualiz. Softw. Defined Networks (NFV-SDN), Berlin, Germany, 2017, pp. 1-6, doi: 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," Wirel. Networks, vol. 28, no. 6, pp. 2533-2552, 2022, doi: 10.1007/s11276-022-02990-2.

[55] R.G. Garroppo, S. Giordano, M. Pagano, and G. Procissi, “On traffic prediction for resource allocation: A Chebyshev bound based allocation scheme,” Comput. Commun., vol. 31, no. 16, pp. 3741-3751, 2008, doi: 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 and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, pp. 1735-1780, 1997, doi: 10.1162/neco.1997.9.8.1735.