



Tasks migration-based load-balancing to improve QoS in SDN-based IoT

Mohammad Rostami¹, Afsaneh Fatemi^{2*}, Salman Goli Bidgoli³

^{1, 2} Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran.

³ Faculty of Computer Engineering, University of Kashan, Kashan, Iran.

ARTICLE INFO

Article Type:

Original Research

Received: 2025.08.03

Revised: 2025.10.26

Accepted: 2026.05.17

Keyword:

Internet of Things
Software-Defined Networking
Load-balancing
Task migration
Quality of Service

*Corresponding Author:

Afsaneh Fatemi

Email:

a_fatemi@eng.ui.ac.ir

ABSTRACT

The development of Internet of Things (IoT) devices leads to increased tasks and resource workload imbalance, which leads to a decrease in Quality of Service (QoS). In order to control resource workload, the use of Software-Defined Networking (SDN) provides efficient solutions for load-balancing with the aim of improving the QoS in IoT applications. Task growth can lead to overloading of some SDN controllers and reduced resource efficiency, necessitating the use of effective load-balancing approaches. In this research, a task migration-based load-balancing approach is presented, in which overloaded controller tasks are selected based on the least internal dependency and assigned to underloaded controllers with the most external dependency and the least communication cost.

The proposed method is implemented in the Mininet environment with the RYU controller and compared with the conventional SMSC, DLBM, DHA, and CAMD approaches. The evaluation criteria include response time, migration cost, migration count, energy consumption, and processor resource utilization in ARN, Atlanta, and NSF topologies. Simulation results show that the proposed method reduces the controller response time by 36.3%, migration cost by 43%, migration count by 26%, energy consumption by 21%, and resource utilization by 19.7% on average compared to the compared approaches. These results indicate that dependency-based task migration for load-balancing can effectively improve QoS parameters and increase user satisfaction in real IoT networks.



EXTENDED ABSTRACT

Introduction

Today, IoT devices generate a huge volume of tasks, which poses challenges in overloading network resources and achieving improved QoS. To achieve optimal management of IoT resources, SDN can be used to provide improved QoS. In the SDN, multiple large-scale controllers with a distributed control scheme are proposed to address the issues related to availability, flexibility, and scalability of the IoT network. Therefore, as IoT tasks increase, the processing capacity of some controllers will exceed the threshold, which will lead to overload of some controllers and affect the QoS and network performance. To meet the QoS given the controller capacity limitations and the growth of user tasks, the load-balancing issue must be addressed effectively.

This research aims to manage the workload of controllers in the SDN-based IoT architecture and presents an effective approach called "load-balancing based on switch task migration". In the proposed approach, response time from the users' direction and energy consumption and resource efficiency from the service providers' direction are examined.

Methodology

In the SDN-based IoT, to deal with limited controller resources, the task migration technique from an overloaded controller to an underloaded controller is used with the aim of load-balancing in the network. Network migration techniques are also used for other purposes such as reliability, energy management, response time reduction, and overall QoS improvement. Migration concepts may be expressed with or without considering the dependencies of tasks on switches. Storing information about these dependencies in controllers can improve task migration performance and reduce related costs, such as time lost during migration and the number of migrations. In this situation, the controller can improve the QoS in the network by deciding on the migration of tasks between controllers, taking into account the dependency of tasks on switches, such as internal and external dependencies, and resource sharing.

In the problem of assigning switch tasks to controllers, there are three key sub-problems whose solution is essential to optimize network performance and maintain QoS. The first sub-problem is when the task migration should occur. The second sub-problem is which task should migrate. The third sub-problem is which controller the task should migrate to. To solve the first sub-problem, the controller's energy consumption is considered to exceed the threshold in order to overload the controller. To solve the second sub-problem, tasks are prioritized for migration based on the internal dependencies of tasks on switches in the controller. To solve the third sub-problem, the task migrates to the controller containing the highest external dependency and also the lowest communication cost. As a result, by placing the tasks of related switches in the controllers, the energy consumption parameter is reduced and some network parameters are improved.

In general, the proposed method consists of three steps.

Step 1: Prioritize tasks based on dependency matrix

Step 2: Find tasks based on internal dependencies for migration

Step 3: Select appropriate controller for task mapping

Results and discussion

The experiment was conducted on the Mininet platform with RYU controllers based on the Python programming language under the Apache 2.0 license. RYU is the most suitable controller available for flexibility and reliability in a large-scale SDN-based IoT environment. Also, the Java-based Mininet simulator is designed for SDN, which creates a topology consisting of servers, switches, controllers, and virtual links.

This research has conducted simulation experiments using a PC with Intel Core i7 CPU@3.40GHz and 12GB memory running Windows 10 and Virtualbox Ubuntu 16.04 operating system with Linux kernel version 4.4. The input task streams are generated by the Iperf tool and for each task stream, the switch sends the packet_in event to the controller.

To evaluate the performance of the proposed method, a comparison is made with four related approaches. SMSC, This approach considers static mapping between switch and controller and does not follow load-balancing technique. DLBM: A load-balancing approach that selects the switch with the highest transfer rate from the overloaded controller and randomly migrates to the nearest underloaded controller. DHA: This approach randomly selects the switch with the maximum load deviation from the overload controller and migrates to the underload controller with the highest remaining capacity. CAMD: Selecting the switch with the lightest workload and migrating to the under-loaded controller from among the set of under-loaded controllers is suggested.

The performance metrics used to evaluate the performance of the proposed method include response time, migration cost, migration number, energy consumption, and resource efficiency. The results show that the controller response time decreases with increasing workload compared to the proposed approach compared to the other four approaches. The proposed method reduces the migration cost by being more accurate in selecting the destination controller. In fact, the proposed method selects the task based on the least internal dependency on the overload controller switches for migration and considers the minimum migration cost during destination controller selection. Also, the proposed method migrates tasks based on priority and assigns them to controllers with high external dependencies on switches, which reduces the number of migrations, resulting in lower energy consumption than other approaches. Due to task migration based on dependency analysis and communication cost, it has led to more targeted use of controllers' processing resources, which brings the highest resource efficiency.

Conclusion

As IoT becomes increasingly popular in various applications, the issue of managing input tasks is crucial for optimizing the performance of IoT applications. The IoT network with congested tasks requires load-balancing techniques to ensure QoS. SDN is proposed to meet the QoS requirements of IoT applications and simultaneously balance traffic between controllers based on available capacity. This research focuses

on task migration-based load-balancing in SDN control level controllers to manage and improve network performance, which aims to effectively allocate switch tasks from overloaded controllers to underloaded controllers.

The load-balancing technique based on task migration is implemented on each controller and is activated when the controller load is unbalanced. In the proposed method, the energy consumption is used for the controller's workload status in the network. Response time, migration cost, migration number, energy consumption, and resource efficiency were used as performance comparison criteria for load-balancing controller approaches. Simulation results show the effectiveness of the proposed method.

تعالادل بار مبتنی بر مهاجرت وظایف برای بهبود کیفیت سرویس در اینترنت اشیا^۱ مبتنی بر شبکه نرم افزار محور

محمد رستمی^۱، افسانه فاطمی^{۱*}، سلمان گلی بیدگلی^۲

۱- دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران.

۲- دانشکده مهندسی کامپیوتر، دانشگاه کاشان، کاشان، ایران.

چکیده

اطلاعات مقاله

نوع مقاله: مقاله پژوهشی

دریافت مقاله: ۱۴۰۴/۰۵/۱۲

بازنگری مقاله: ۱۴۰۴/۰۸/۰۴

پذیرش مقاله: ۱۴۰۵/۰۲/۲۷

کلید واژگان:

اینترنت اشیا
شبکه نرم افزار محور
تعالادل بار
مهاجرت وظیفه
کیفیت سرویس

*نویسنده مسئول:

افسانه فاطمی

پست الکترونیکی:

a_fatemi@eng.ui.ac.ir

توسعه دستگاه‌های اینترنت اشیا^۱ موجب افزایش وظایف و عدم تعادل بارکاری منابع می‌شود که کاهش کیفیت سرویس^۲ را به همراه دارد. در راستای کنترل بارکاری منابع، بکارگیری شبکه نرم افزار محور^۳، راهکارهای کارآمدی را جهت برقراری تعادل بار^۴ با هدف بهبود کیفیت سرویس در کاربردهای اینترنت اشیا فراهم می‌سازد. رشد وظایف می‌تواند به اضافه بار برخی از کنترلرهای شبکه نرم افزار محور و کاهش بهره‌وری منابع منجر شود که استفاده از رویکردهای موثر تعادل بار ضروری می‌گردد. در این تحقیق، رویکرد تعادل بار مبتنی بر مهاجرت وظیفه ارائه شده است که وظایف کنترلر اضافه بار^۵ بر اساس کمترین وابستگی داخلی انتخاب و به کنترلرهای کم‌بار با بیشترین وابستگی خارجی و کمترین هزینه ارتباطی تخصیص داده می‌شوند. روش پیشنهادی در محیط Mininet با کنترلر RYU پیاده‌سازی و با رویکردهای مرسوم DLBM، SMSC، DHA و CAMD مقایسه شده است. معیارهای ارزیابی شامل زمان پاسخ، هزینه مهاجرت، تعداد مهاجرت، مصرف انرژی و بهره‌وری منبع پردازنده بر روی توپولوژی‌های NSF و Atlanta، ARN می‌باشند. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی به طور متوسط موجب کاهش ۳۶،۳ درصدی زمان پاسخ کنترلر، ۴۳ درصدی هزینه مهاجرت، ۲۶ درصدی تعداد مهاجرت، ۲۱ درصدی مصرف انرژی و افزایش ۱۹،۷ درصدی بهره‌وری منبع نسبت به رویکردهای مورد مقایسه می‌شود. این نتایج بیانگر آن است که مهاجرت وظیفه مبتنی بر وابستگی در جهت تعادل بار می‌تواند به‌طور موثر موجب بهبود پارامترهای کیفیت سرویس گردد و رضایت کاربران را در شبکه‌های واقعی اینترنت اشیا افزایش دهد.

^۱ Internet of Things (IoT)

^۲ Quality of Service (QoS)

^۳ Software-Defined Networking (SDN)

^۴ Load-balancing

^۵ Overload



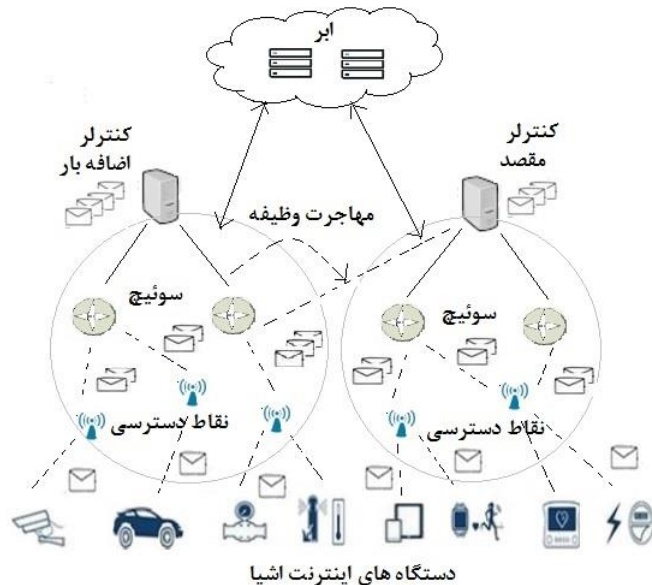
مقدمه

امروزه دستگاه‌های اینترنت اشیا حجم عظیمی از وظایف را تولید می‌کنند که منجر به اضافه بار منابع شبکه و کاهش کیفیت سرویس می‌شود [۱]. برای مدیریت بهینه منابع، شبکه نرم‌افزار محور به عنوان راهکاری موثر برای بهبود کیفیت سرویس مطرح است. در شبکه نرم‌افزار محور چندین کنترلر در مقیاس وسیع با طرح کنترل توزیع شده برای افزایش دسترس پذیری، انعطاف پذیری و مقیاس پذیری شبکه اینترنت اشیا پیشنهاد می‌شوند [۲]. تخصیص مطلوب بارکاری (وظایف) به منابع از طریق کنترلر موجب برآورده شدن نیازمندی‌های کیفیت سرویس می‌گردد [۳]. از این رو، با افزایش وظایف اینترنت اشیا، ظرفیت پردازش برخی کنترلرها از حد آستانه فراتر خواهد رفت که منجر به اضافه بار برخی کنترلرها شده و بر روی کیفیت سرویس و عملکرد شبکه تاثیر منفی می‌گذارد [۲].

سطح کنترل شبکه نرم‌افزار محور می‌تواند به عنوان تصمیم‌گیرنده در انتقال وظایف به کنترلرها با هدف کنترل بارکاری و نظارت بر شبکه عمل کند. همچنین، با قابلیت برنامه‌ریزی شبکه موجب بهبود بهره‌وری منابع و عملکرد شبکه می‌شود و برآورده نمودن نیازمندی‌های کیفیت سرویس را با توزیع بار مناسب بهبود می‌بخشد [۵].

با توجه به نداشت ایستا بین سوئیچ و کنترلر و همچنین تولید مستمر وظایف توسط دستگاه‌های ناهمگن اینترنت اشیا موجب افزایش زمان پاسخ برخی کنترلرها و عدم تعادل بار می‌گردد [۴]. تکنیک‌های تعادل بار با هدف تخصیص یا انتقال بهینه وظایف و بهبود عملکرد شبکه، مورد توجه محققان قرار گرفته و منجر به ارائه رویکردهای متعدد جهت بهبود کیفیت سرویس شده است. بنابراین، مدیریت تعادل بار بین کنترلرها با توجه به محدودیت ظرفیت برای حفظ کیفیت سرویس ضروری است.

رویکرد مهاجرت وظایف سوئیچ با انتقال وظایف از کنترلرهای دارای اضافه بار به کنترلرهای کم بار، به حفظ تعادل بار و بهبود کیفیت سرویس دست می‌یابد. با این حال، رویکردهای پیشین در مورد مهاجرت وظایف کنترلر عمدتاً بر بهینه‌سازی تعادل بار بین کنترلرها تمرکز دارند و به دلیل محدودیت در کارایی مهاجرت و استفاده ناکارآمد از منابع کنترلر، توانایی پاسخ به نوسانات ترافیک در زمان واقعی در اینترنت اشیا مبتنی بر شبکه نرم‌افزار محور را ندارند. این محدودیت‌ها منجر به کاهش عملکرد شبکه و تشخیص عدم تعادل بار جدید بین کنترلرها پس از مهاجرت وظیفه می‌شود. بنابراین، این تحقیق با هدف مدیریت حجم وظایف کنترلرها در معماری اینترنت اشیا مبتنی بر شبکه نرم‌افزار محور انجام شده و رویکردی مؤثر با عنوان "تعادل بار مبتنی بر مهاجرت وظایف سوئیچ" ارائه شده است. شکل ۱ تصویری از معماری شبکه نرم‌افزار محور چندکنترلی برای اینترنت اشیا را نشان می‌دهد.



شکل ۱. معماری اینترنت اشیا مبتنی بر شبکه نرم افزار محور چندکنترلی.

هدف این تحقیق ارائه رویکردی نوین برای تعادل بار کنترلرهای شبکه نرم افزار محور در اینترنت اشیا است که با مهاجرت وظایف سوئیچ ها مبتنی بر وابستگی داخلی و خارجی از کنترلر اضافه بار به کنترلر کم بار انجام می شود. این رویکرد علاوه بر حفظ تعادل بار، بهبود همزمان زمان پاسخ، مصرف انرژی و بهره‌وری منابع کنترلر را تضمین می نماید. از مهم ترین نوآوری های این تحقیق می توان به موارد زیر اشاره نمود:

- تشخیص اضافه بار کنترلر بر اساس مصرف انرژی.
- اولویت بندی وظایف برای مهاجرت بر اساس ماتریس وابستگی.
- انتخاب کنترلر مقصد با بیشترین وابستگی خارجی و کمترین هزینه ارتباطی.
- تجمیع وظایف وابسته برای بهبود بهره‌وری منابع و پارامترهای کیفیت سرویس نسبت به رویکردهای موجود.

در ادامه در بخش دوم به مرور تحقیقات مرتبط با موضوع پرداخته خواهد شد و سپس در بخش سوم طرح پیشنهادی تشریح می شود و نحوه ارزیابی رویکرد پیشنهادی با رویکردهای موجود در بخش چهارم بیان می گردد و در نهایت در بخش پنجم نتیجه گیری به همراه پیشنهادهای آینده توضیح داده می شود.

پیشینه پژوهش

در این بخش مروری بر مهم ترین تحقیقات پیشین در حوزه اینترنت اشیا مبتنی بر شبکه نرم افزار محور با دیدگاه تعادل بار در راستای بهبود کیفیت سرویس ارائه می شود. اینترنت اشیا به دلیل گستردگی دستگاه ها و همچنین محدودیت منابع، نیازمند رویکردهای تعادل بار جهت بهبود کیفیت سرویس می باشد [۶]. تاکنون رویکردهای مختلفی جهت تعادل بار در راستای بهبود کیفیت سرویس در اینترنت اشیا مبتنی بر شبکه نرم افزار

محور شامل تکنیک‌های مهاجرت سوئیچ، زمانبندی، مسیریابی، منطق فازی، سیاست تخصیص، پیش‌بینی، قرارگیری کنترلر و گرادیان سیاست مورد بررسی قرار گرفته که در این بخش به آنها اشاره شده است.

سان و همکاران در [۷] طرح تبادل بار مبتنی بر یادگیری تقویتی چندعاملی^۱ در سطح کنترل شبکه نرم‌افزار محور بر اساس مهاجرت سوئیچ را معرفی می‌کنند. عامل‌ها در نقش کنترلر عمل نموده و با تصمیم‌گیری در مورد مهاجرت سوئیچ، توزیع بار بین کنترلرها را مدیریت می‌کنند. هر عامل با مشاهده ویژگی‌های بارکاری و وضعیت شبکه، سیاستی یاد می‌گیرد تا سوئیچ‌ها را به گونه‌ای منتقل کند که از اضافه بار کنترلرها جلوگیری شده و زمان پاسخ کنترلرها بهبود یابد. ساهو و همکاران در [۸] رویکرد تبادل بار مبتنی بر مهاجرت پویای سوئیچ را برای جلوگیری از سربار سطح کنترل و توزیع کارآمد ترافیک با در نظر گرفتن دستگاه‌های نا همگن اینترنت اشیا پیشنهاد کردند. کنترلر هدف بر اساس کارایی مهاجرت کنترلرها و حداکثر بهره‌وری منابع باقیمانده برای بهبود عملکرد تبادل بار کنترلرها انتخاب می‌گردد.

شی و همکاران در [۹] رویکرد تبادل بار چندکنترلی مبتنی بر مهاجرت سوئیچ را برای بهبود عملکرد شبکه پیشنهاد نمودند. با دیدگاه کلی ارائه شده توسط کنترلرهای شبکه نرم‌افزار محور، پروتکل مسیریابی آگاه از کیفیت سرویس بر اساس یادگیری تقویتی انجام شده است. شهزاد و همکاران در [۱۰] الگوریتم انتخاب منصفانه سوئیچ برای شبکه‌های نرم‌افزار محور را در مقیاس بزرگ با هدف کاهش سربار سطح کنترل طراحی نموده‌اند. وظایف با توجه به فاصله و مقصدشان اولویت‌بندی شده و تضمین می‌گردد که سوئیچ‌ها وظایف با بالاترین اولویت برای حفظ کیفیت سرویس را انتخاب کنند.

آدکویا و همکاران در [۱۱] الگوریتم تصمیم‌گیری مهاجرت سوئیچ برای شناسایی همزمان هزینه مهاجرت و تغییرات تبادل بار برای انتخاب بهینه کنترلر ارائه کردند. همچنین، از واریانس کنترلر و وضعیت بار متوسط کنترلر برای تعیین مجموعه کنترلرهای کم‌بار در شبکه استفاده گردید. زارع سلطان و همکاران در [۱۲] تکنیک تبادل بار از طریق گرادیان سیاست^۲ مبتنی بر یادگیری تقویتی جهت جلوگیری از ازدحام و بهبود کارایی با توزیع ترافیک وظایف در سرورها پیشنهاد نمودند.

شیرانی و همکاران در [۱۳] به ارائه معماری کنترلر در شبکه نرم‌افزار محور جهت بهبود تبادل بار پرداخته‌اند. ساختار کنترلر پیشنهادی از تحلیل و پیش‌بینی مبتنی بر حافظه کوتاه‌مدت دو طرفه استفاده می‌کند که می‌تواند دقت پیش‌بینی بار آینده را از طریق تحلیل‌های آینده-گذشته و گذشته-آینده بهبود بخشد. بابار و همکاران در [۱۴] رویکرد تبادل بار بین کنترلرهای توزیع شده در اینترنت اشیا مبتنی بر شبکه نرم‌افزار محور با مقیاس بزرگ برای کاربرد شهر هوشمند معرفی می‌کنند. با افزایش بارکاری کنترلرها از حد آستانه، سوئیچ‌های دارای بارکاری سنگین به کنترلرهای با بار کمتر منتقل می‌شوند.

منتظرالقائم و یغمایی در [۴] چارچوبی جهت مدیریت ترافیک مقیاس‌پذیر اینترنت اشیا مبتنی بر شبکه نرم‌افزار محور پیشنهاد می‌دهند که نیازهای کیفیت سرویس را برآورده نموده و به طور همزمان تبادل ترافیک را بین سرورهای اینترنت اشیا بر اساس منطق فازی و سیستم پیش‌بینی بارکاری آینده بر اساس بارکاری گذشته

^۱ Multi-agent reinforcement learning

^۲ Policy gradient

ایجاد می‌نماید. کائو و همکاران در [۱۵] معماری اینترنت اشیا وسایل نقلیه^۱ مبتنی بر محاسبات ابر-مه و شبکه نرم‌افزار محور را پیشنهاد می‌دهند که استفاده موثر از منابع محاسباتی بر اساس خوشه‌های مه برای تضمین کیفیت سرویس بکار می‌رود. هنگامی که منبع محاسباتی در خوشه مه ناکافی است، کنترلر می‌تواند منبع سرویس را از ابر یا خوشه مه مجاور درخواست نماید.

خو و همکاران در [۱۶] رویکرد قرارگیری کنترلرها به عنوان مسأله بهینه‌سازی چندهدفه را با استفاده از معماری دو لایه سطح کنترل برای مدیریت تغییرات مداوم در ترافیک شبکه در محیط‌های اینترنت اشیا مبتنی بر شبکه نرم‌افزار محور معرفی می‌کنند که قابلیت‌های مدیریت سطح کنترل را افزایش می‌دهد. علی و همکاران در [۲] به رویکرد تعادل بار مبتنی بر تخصیص کنترلر پیرو^۲ کارآمد برای شبکه اینترنت اشیا چند دامنه‌ای مبتنی بر شبکه نرم‌افزار محور می‌پردازند که هدف آن انتقال مؤثر سوئیچ‌ها به کنترلر با منابع آزاد است. در طرح کنترل توزیع شده، رتبه‌بندی کنترلرها و انتخاب کنترلر هدف بر اساس فرآیند تصمیم‌گیری چندمعیاره با حفظ کیفیت سرویس استفاده شده است. در جدول ۱ کاربردهای تعادل بار جهت کنترلر بار سمت سرور و کنترلر به همراه مشخص نمودن تکنیک، پارامترهای کیفیت سرویس و نهاد و محل تعادل بار، بیان شده است.

شهربابکی و همکاران در [۲۸] رویکرد تعادل بار بر اساس کنترلر شبکه نرم‌افزار محور انجام می‌شود که به طور دوره‌ای بارکاری و تأخیر سرورهای لبه و ابر را پایش می‌کند. الگوریتم تعادل بار جریان‌های ویدئویی اینترنت اشیا را بر اساس تأخیر پیش‌بینی شده بین سرورها تخصیص می‌دهد. این فرآیند باعث توزیع یکنواخت بار پردازشی و کاهش میانگین تأخیر می‌شود. کومار و مالیک در [۲۹] رویکرد تعادل بار کنترلر و لینک برای اینترنت اشیا صنعتی مبتنی بر شبکه نرم‌افزار محور ارائه می‌دهند. با پایش وضعیت کنترلرها و لینک‌ها، جریان داده صنعتی بین کنترلرها و مسیرهای شبکه تخصیص می‌یابد که بار پردازشی و ترافیک شبکه متعادل شود. این رویکرد از ترکیب کنترل جریان و مهاجرت جریان استفاده می‌کند تا از اضافه بار کنترلرها جلوگیری نماید.

جدول ۱. مقایسه یافته‌های مرتبط‌ترین کاربردهای تعادل بار موجود با پژوهش حاضر.

کاربرد	تکنیک	پارامتر کیفیت سرویس	نهاد تعادل بار		محل تعادل بار		مرجع	سال
			کنترلرها	سرورها	سطح کاربرد	سطح کنترل		
توزیع بار	مهاجرت سوئیچ	توان عملیاتی، گم شدن بسته، زمان پاسخ	✓	-	-	✓	[۱۱]	۲۰۲۰
توزیع بار	مهاجرت سوئیچ	زمان پردازش، تاخیر، بهره‌وری منبع	✓	-	-	✓	[۷]	۲۰۲۰
معماری	پیش‌بینی- منطق فازی	توان عملیاتی، تاخیر	-	✓	✓	-	[۴]	۲۰۲۰

^۱ Internet of Vehicles (IoV)

^۲ Slave

سیاست معماری	مهاجرت سوئیچ	بهره‌وری منبع، زمان پاسخ، تاخیر	✓	-	-	✓	[۱۴]	۲۰۲۱
معماری	تخصیص- خوشه‌بندی	تاخیر، مصرف انرژی، پایداری، تعالل بار	-	✓	✓	✓	[۱۵]	۲۰۲۱
توزیع بار	مهاجرت سوئیچ	زمان پاسخ	✓	-	-	✓	[۸]	۲۰۲۲
مسیریابی	مهاجرت سوئیچ	زمان پاسخ، تاخیر، مصرف انرژی	✓	-	-	✓	[۹]	۲۰۲۲
معماری	قرارگیری کنترلر	تاخیر، قابلیت اطمینان، تعادل بار	-	✓	✓	-	[۱۶]	۲۰۲۳
سیاست تخصیص	مهاجرت سوئیچ	تاخیر، زمان پاسخ، بهره‌وری منبع	✓	-	-	✓	[۲]	۲۰۲۳
زمان‌بندی	مهاجرت سوئیچ	توان عملیاتی، گم شدن بسته، هزینه	-	-	-	✓	[۱۰]	۲۰۲۴
توزیع بار	سیاست معماری	تاخیر	✓	-	-	✓	[۲۸]	۲۰۲۴
توزیع بار	گرایان سیاست	زمان پاسخ، مصرف انرژی	-	✓	✓	-	[۱۲]	۲۰۲۵
سیاست تخصیص	کنترل جریان و مهاجرت	تاخیر، نرخ گم شدن بسته، بهره‌وری منبع	✓	-	-	✓	[۲۹]	۲۰۲۵
سیاست معماری	پیش‌بینی	زمان پاسخ، سربار پردازش	✓	-	-	✓	[۱۳]	۲۰۲۵
توزیع بار	مهاجرت وظیفه	زمان پاسخ، مصرف انرژی، بهره‌وری منبع	✓	-	-	✓	روش پیشنهادی	۲۰۲۶

مفهوم تعادل بار می‌تواند روی سروورها و یا کنترلرها صورت گیرد و پیاده‌سازی تکنیک‌های تعادل بار در سطوح کاربرد و کنترل قابل انجام است [۱۷] و نیاز به تعادل بار مبتنی بر مهاجرت وظیفه بر اساس وظایف ورودی در سطوح زیرساخت احساس می‌شود. مهاجرت وظیفه در بین کنترلرهای شبکه، منجر به حفظ پایداری اجرای وظیفه و همچنین توزیع کل وظایف شبکه بین کنترلرها می‌شود [۱۸]. مدل ارائه شده از لحاظ بهبود پارامترهای کیفیت سرویس در مقایسه با رویکردهای ارائه شده قبلی راندمان بهتری داشته است.

در اینترنت اشیاء مبتنی بر شبکه نرم افزار محور، کیفیت سرویس از دیدگاه کاربران و ارائه دهندگان سرویس قابل بررسی است. کاربران می توانند با ارزیابی پارامترهای کیفیت سرویس، نیازهای خود را از کاربردهای اینترنت اشیاء مشخص کنند و ارائه دهندگان سرویس نیز با ارزیابی پارامترهای کیفیت سرویس، عملکرد و کیفیت ارائه سرویس های خود را با تاکید بر افزایش رضایت و جذب کاربران بیشتر مدیریت می نمایند. بهینه سازی کیفیت سرویس به طور همزمان برای موجودیت های فوق الذکر، می تواند تضمین کننده منافع دو طرف بوده و کارآیی شبکه را نیز بالا ببرد [۱۹] و [۲۰].

تاکنون، پارامترهای مختلفی در جهت بهبود کیفیت سرویس توسط محققان در پیشینه تحقیق معرفی شده که بیشتر از دیدگاه کاربر نهایی یا ارائه دهنده سرویس به این موضوع نگاه شده است که کمبود مطالعات و رویکردهای موثر در مورد مسأله بهینه سازی کیفیت سرویس از دیدگاه کاربر و ارائه دهنده سرویس به طور همزمان وجود دارد. در طول تحقیق موثرترین پارامترها از دیدگاه کاربر و ارائه دهنده سرویس جهت بهبود کیفیت سرویس بیان شده است. در رویکرد پیشنهادی، زمان پاسخ از دید کاربران و مصرف انرژی و بهره وری منبع از دید ارائه دهندگان سرویس بررسی شده است. در واقع کاربران علاقه مند هستند که وظایف خود را در کوتاه ترین زمان ممکن تکمیل کنند. از سوی دیگر، ارائه دهندگان سرویس در تلاش هستند تا بهبود مصرف انرژی و استفاده از منابع خود را به حداکثر برسانند تا سود خود را افزایش دهند [۱] و [۲۰].

همچنین، مشاهده می شود که بیشتر از کنترلرهای توزیع شده در لایه های مختلف شبکه جهت بهبود پارامترهای کیفیت سرویس استفاده شده است. با توجه به مزایای کنترلرهای توزیع شده شبکه نرم افزار محور و بکارگیری آن در شبکه اینترنت اشیاء می تواند با کنترل و توزیع وظایف ورودی بین گره ها (سوئیچ، کنترلر و سرور) بار هر گره را در زمان واقعی حفظ و امکان تحویل به موقع سرویس را برای کاربران فراهم نماید. این امر منجر به بهبود عملکرد شبکه و پارامترهای کیفیت سرویس می شود. بنابراین، با توجه به رشد دستگاه های نهایی و همچنین منابع محدود در کنترلر ها در مقابل وظایف عظیم دستگاه ها، روش تعادل بار مبتنی بر مهاجرت وظیفه از کنترلر های اضافه بار به کنترلر های کم بار جهت بهبود پارامترهای کیفیت سرویس از دیدگاه مختلف انجام می گیرد.

طرح پیشنهادی

در شبکه نرم افزار محور، کنترلر ها به صورت دوره ای اطلاعات کنترلی شبکه را با یکدیگر رد و بدل می کنند اما به دلیل اتصال ایستای کنترلر و سوئیچ و همچنین تغییرات بار ترافیکی سوئیچ ها، ممکن است توزیع نامطلوب بار کاری در بین کنترلر ها ایجاد شده و بر روی پارامترهای کیفیت سرویس به عنوان مثال زمان پاسخ، مصرف انرژی و بهره وری منابع شبکه تأثیر منفی بگذارد. با توجه به اینکه در شرایط واقعی، بار کاری و تعادل بار روی کنترلر ها متأثر از شرایط شبکه و پویا می باشد. در نظر گرفتن مفاهیم پویای شبکه در ورود بار کاری و توزیع وظایف یا مهاجرت بین کنترلر ها از طریق انتقال وظایف به کنترلر های کم بار ضروری بوده و می تواند موجب تعادل بار و بهبود کیفیت سرویس در شبکه اینترنت اشیاء گردد.

در اینترنت اشیاء مبتنی بر شبکه نرم افزار محور، برای مقابله با منابع محدود کنترلر ها، از تکنیک مهاجرت وظیفه از کنترلر اضافه بار به کنترلر کم بار با هدف تعادل بار در شبکه یاد می شود. تکنیک مهاجرت در شبکه با

اهداف دیگری نظیر قابلیت اطمینان، مدیریت انرژی، کاهش زمان پاسخ و به طور کلی بهبود کیفیت سرویس نیز مورد استفاده قرار می‌گیرد. مفاهیم مهاجرت ممکن است با یا بدون در نظر گرفتن وابستگی وظایف به سوئیچ‌ها بیان شود. ذخیره اطلاعات مربوط به این وابستگی‌ها در کنترلرها، می‌تواند منجر به بهبود عملکرد مهاجرت وظیفه و کاهش هزینه‌های مربوطه نظیر، زمان از دست رفته در هنگام مهاجرت و همچنین تعداد مهاجرت‌ها شود. با بررسی تحقیقات انجام شده، تاکنون تکنیک مهاجرت وظیفه بر اساس وابستگی جهت بهبود کیفیت سرویس مورد توجه قرار نگرفته است. در این شرایط کنترلر می‌تواند با تصمیم‌گیری در مورد مهاجرت وظایف بین کنترلرها با در نظر گرفتن وابستگی وظایف به سوئیچ‌ها تحت عنوان وابستگی داخلی و خارجی و اشتراک‌گذاری منابع موجب بهبود کیفیت سرویس در شبکه گردد.

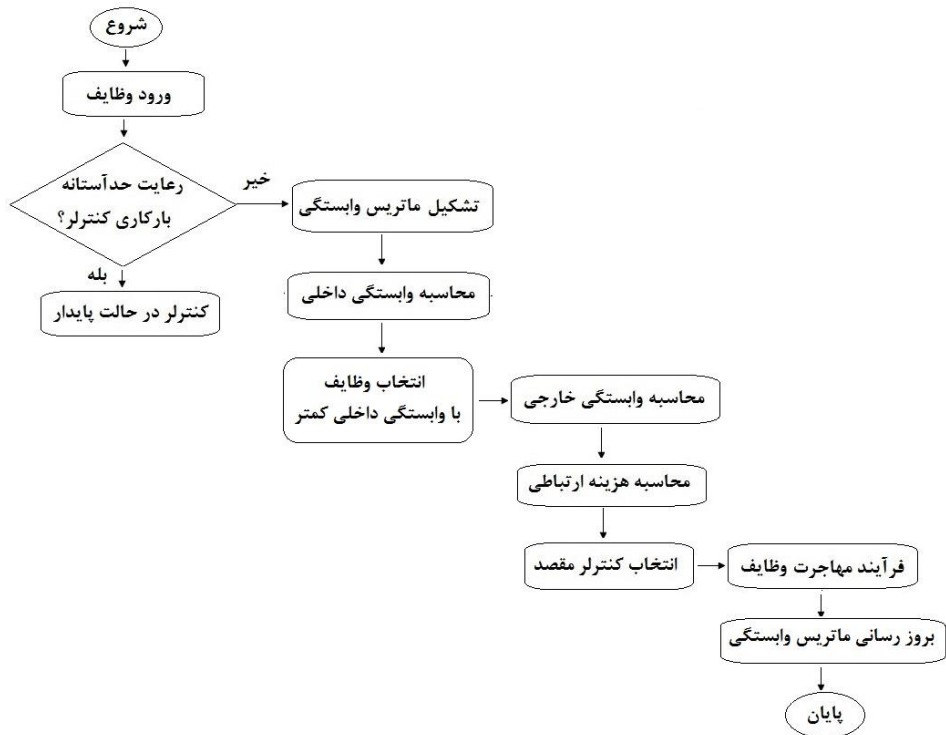
در مسأله تخصیص وظایف سوئیچ‌ها به کنترلرها، سه زیرمسأله اساسی وجود دارد که حل موثر آن‌ها برای بهینه‌سازی عملکرد شبکه و حفظ کیفیت سرویس ضروری است. زیرمسأله اول، چه زمانی باید مهاجرت وظیفه صورت گیرد. زیر مسأله دوم، کدام وظیفه مهاجرت نماید. زیر مسأله سوم، وظیفه به کدام کنترلر مهاجرت کند. برای حل زیر مسأله اول، فراتر رفتن انرژی مصرفی کنترلر از حد آستانه که موجب اضافه بار شده و نیاز به مهاجرت وظیفه است. برای حل زیر مسأله دوم، بر اساس وابستگی داخلی وظایف به سوئیچ‌ها در کنترلر، اولویت‌بندی وظایف جهت مهاجرت انجام می‌شود. برای حل زیر مسأله سوم، وظیفه به کنترلر حاوی بالاترین وابستگی خارجی و همچنین کمترین هزینه ارتباطی مهاجرت می‌نماید. در نتیجه، با قرارگیری وظایف سوئیچ‌های مرتبط در کنترلرها، پارامتر انرژی مصرفی کاهش یافته و بهبود برخی پارامترهای شبکه را به همراه دارد. به طور کلی روش پیشنهادی شامل سه مرحله است.

مرحله اول: تشکیل ماتریس وابستگی جهت اولویت‌بندی وظایف کنترلر

مرحله دوم: انتخاب وظایف برای مهاجرت بر اساس وابستگی داخلی

مرحله سوم: انتخاب کنترلر مقصد جهت نگاشت وظایف

طرح تعادل بار مبتنی بر مهاجرت وظایف از طریق کنترلرها بر اساس وابستگی وظایف به سوئیچ‌ها در شکل ۲ تحت عنوان ساختار روش پیشنهادی نمایش داده می‌شود.



شکل ۲. روند نمای روش پیشنهادی.

فرضیات

- در ادامه برخی از فرضیات مرتبط با ارائه روش پیشنهادی بیان شده است.
- فرض ۱: در شبکه، وظایف سوئیچ‌ها می‌توانند از یک کنترلر به کنترلر دیگر دسترسی و مهاجرت داشته باشند تا بار سطح کنترل متعادل گردد.
- فرض ۲: هنگامی که وظیفه برای مهاجرت انتخاب شده باشد، مجاز به بازگشت به کنترلر قبلی تا زمانی که عملیات مهاجرت به پایان نرسیده باشد، نیست.
- فرض ۳: یک کنترلر به عنوان کنترلر اصلی به هر سوئیچ اختصاص داده می‌شود.
- فرض ۴: اضافه بار همه کنترلرها در زمان خاص t غیرممکن است.

بررسی کنترلر دارای اضافه بار بر اساس مصرف انرژی

موقعی که انرژی مصرفی یک کنترلر از حد آستانه تعریف شده فراتر رود، کنترلر در وضعیت اضافه بار قرار می‌گیرد و فرآیند مهاجرت وظایف فعال می‌شود. در این شرایط، برخی از وظایف به کنترلر مقصد تخصیص داده می‌شوند که باعث کاهش تعداد مهاجرت‌های غیر ضروری وظایف نیز می‌شود. می‌توان مصرف انرژی کنترلر را به صورت مجموع انرژی مصرف شده توسط هر وظیفه مطابق رابطه ۱ در نظر گرفت.

$$E_j = \sum_{i=1}^n t_i \times W_j \quad (1)$$

که n تعداد وظایف، t_i زمان پردازش وظیفه i ام (بر حسب ثانیه)، W_j توان مصرفی کنترلر j برای انجام وظایف (بر حسب وات) و E_j انرژی مصرفی کل کنترلر j بیان می‌گردد. برای محاسبه زمان پردازش وظیفه i ام، از رابطه ۲ استفاده می‌شود.

$$PT_i = \frac{\text{DataSize}[i]}{\text{Processing rate}_j} \quad (2)$$

که $\text{DataSize}[i]$ حجم داده مربوط به وظیفه i ام و پارامتر Processing rate_j نرخ پردازش کنترلر j (بایت در ثانیه) را نشان می‌دهند. شبه کد محاسبه کل انرژی مصرفی کنترلر و بررسی حداستانه انرژی مصرفی در الگوریتم ۱ بیان شده است.

Pseudocode 1 shows the energy consumption of the controller algorithm.
 Input: $T = [t_1, t_2, \dots, t_n]$, $\text{DataSize}[i]$, Processing rate_j , Threshold, W_j
 Output: E_total , Overloaded = True or False
 Algorithm:
 $E_total \leftarrow 0$
 for each task t_i in T do:
 $PT_i \leftarrow \text{DataSize}[i] / \text{Processing rate}_j$
 $E_i \leftarrow PT_i \times W_j$
 $E_total \leftarrow E_total + E_i$
 if $E_total > \text{Threshold}$ then
 Overloaded $\leftarrow$ True
 else
 Overloaded $\leftarrow$ False
 return (E_total , Overloaded)

مرحله اول: تشکیل ماتریس وابستگی جهت اولویت‌بندی وظایف کنترلر

برای محاسبه وابستگی وظایف به سوئیچ‌های کنترلرها، ماتریس وابستگی تشکیل می‌گردد. ماتریس وابستگی شامل سطرهایی از وظایف کنترلر و ستون‌هایی از سوئیچ‌های متصل به کنترلرها است. مقدار وابستگی هر وظیفه به هر سوئیچ متصل به کنترلرها بر اساس میزان تعامل وظایف با سوئیچ‌ها و در بازه صفر و یک محاسبه می‌شود. محاسبه مقدار وابستگی در رابطه ۳ بیان شده است.

$$D(t, S_i) = \frac{I(t, S_i)}{\sum_{j=1}^n I(t, S_j)} \quad (3)$$

که $D(t, S_i)$ درصد وابستگی وظیفه t به سوئیچ S_i ، $I(t, S_i)$ تعداد تعاملات وظیفه t با سوئیچ S_i و $\sum_{j=1}^n I(t, S_j)$ تعداد کل تعاملات وظیفه t با تمام سوئیچ‌ها را نشان می‌دهد. به عنوان مثال فرض کنید در جدول ۲ وظیفه t_1 از سوئیچ S_1 در طول اجرا دارای تعاملات با سوئیچ‌های S_1 ، S_2 و S_3 می‌باشد.

جدول ۲. مثالی از تعاملات وظیفه t_1 از سوئیچ S_1 .

سوئیچ	تعداد تعامل با t_1
S_1	۱۰ تعامل
S_2	۶ تعامل
S_3	۴ تعامل

مقدار وابستگی به صورت زیر محاسبه می‌شود.

$$S_1: 10/20 = 0.5, \quad S_2: 6/20 = 0.3, \quad S_3: 4/20 = 0.2$$

شبه کد ایجاد ماتریس وابستگی در الگوریتم ۲ آمده است. T لیستی از وظایف، S لیستی از سوئیچ‌ها و interactions[n][m] ماتریس دو بعدی که تعداد تعاملات وظایف با سوئیچ‌ها را نشان می‌دهد.

Pseudocode 2 shows the dependency matrix algorithm.
 Input: $T = [t_1, t_2, \dots, t_n]$, $S = [s_1, s_2, \dots, s_m]$, interactions[n][m]
 Output: D[n][m]
 Algorithm:
 for each task t_i in T do:
 total_interactions = 0
 for each switch s_j in S do:
 total_interactions += interactions [t_i][s_j]
 for each switch s_j in S do:
 if total_interactions == 0:
 D [t_i][s_j] = 0
 else
 D [t_i][s_j] = interactions [t_i][s_j] / total_interactions
 Return D

مرحله دوم: انتخاب وظایف برای مهاجرت بر اساس وابستگی داخلی

در این گام از ماتریس وابستگی برای اولویت‌بندی وظایف کنترلر اضافه بار جهت مهاجرت استفاده می‌شود و وظایفی انتخاب می‌شوند که کمترین وابستگی داخلی (وابستگی به سوئیچ‌های محلی کنترلر اضافه بار) را داشته باشند. فرض نمایید کنترلر اضافه بار دارای وظایف t_1 ، t_2 و t_3 و سوئیچ‌های محلی متصل به این کنترلر S_1 ، S_2 و S_3 می‌باشند. سوئیچ‌های S_4 و S_5 به کنترلرهای خارجی متصل‌اند، اما وظایف کنترلر اضافه بار ممکن است به آن‌ها نیز وابسته باشند (وابستگی خارجی). تشکیل ماتریس وابستگی بر اساس تعداد تعاملات وظایف با سوئیچ‌های کنترلرها در جدول ۳ نشان داده شده است.

جدول ۳. نمونه‌ای از ماتریس وابستگی.

سوئیچ وظیفه	S1 (محلی)	S2 (محلی)	S3 (محلی)	S4 (خارجی)	S5 (خارجی)
t_1	۰,۵	۰,۳	۰,۱	۰,۰۵	۰,۰۵
t_2	۰,۱	۰,۶	۰,۰	۰,۱	۰,۲
t_3	۰,۲	۰,۲	۰,۲	۰,۳	۰,۱

محاسبه وابستگی داخلی هر وظیفه به صورت زیر محاسبه می‌شود.

$$I_Dependency(t_1) = 0.5 + 0.3 + 0.1 = 0.9$$

$$I_Dependency(t_2) = 0.1 + 0.6 + 0.0 = 0.7$$

$$I_Dependency(t_3) = 0.2 + 0.2 + 0.2 = 0.6$$

بنابراین وظیفه t_3 کمترین وابستگی داخلی را داشته و در اولویت برای مهاجرت قرار می‌گیرد. شبه کد مربوط به انتخاب وظیفه با کمترین وابستگی داخلی جهت مهاجرت در الگوریتم ۳ بیان شده است.

Pseudocode 3. Task selection algorithm with minimum internal dependencies.

Input: $D[n][m]$

Output: SelectedTask

Algorithm:

```

min_internal_dependency = + ∞
SelectedTasks = None
for i ← 0 to n-1 do
    internal_sum = 0
    for j ← 0 to m-1 do
        if Switches[j] ∈ ControllerSwitches then
            internal_sum += D[i][j]
        if internal_sum < min_internal_dependency then
            min_internal_dependency = internal_sum
            SelectedTask = T[i]
    return SelectedTask

```

ControllerSwitches، لیست سوئیچ‌هایی متصل به کنترلر اضافه بار (محلی) و $D[n][m]$ ، ماتریس وابستگی که مقدار $D[i][j]$ بیانگر میزان وابستگی وظیفه t_i به سوئیچ S_j را نشان می‌دهند که مقادیر بین صفر و یک می‌باشند.

مرحله سوم: انتخاب کنترلر مقصد جهت نگاشت وظایف

در این گام هدف آن است که کنترلر مناسبی برای نگاشت وظایف مهاجرتی انتخاب شود. در روش پیشنهادی ابتدا کنترلرهایی که مصرف انرژی آنها از حد آستانه تعریف شده فراتر رفته از لیست کاندید حذف می‌شوند. از لیست کاندید، کنترلری انتخاب می‌شود که همزمان بیشترین وابستگی خارجی وظیفه به سوئیچ‌های آن و کمترین هزینه ارتباطی (تاخیر) با کنترلر اضافه بار داشته باشد. در صورتی که چند کنترلر مقادیر یکسانی داشته باشند، کنترلری برگزیده می‌شود که وظایف نگاشت شده به آن، وابستگی داخلی کمتری به سوئیچ‌های خودش داشته باشد. چنانچه شرایط فوق برآورده نشوند، وظیفه به اولین کنترلر موجود در لیست کاندید نگاشت می‌شود. رتبه‌بندی کنترلرها جهت انتخاب کنترلر مقصد جهت نگاشت وظایف مهاجرتی بر اساس رابطه ۴ محاسبه می‌شود. مقادیر وابستگی خارجی و هزینه ارتباطی باید نرمال شده و در بازه صفر و یک قرار گیرند.

$$\text{Score}(C) = E_Dependency(t, C) \times \alpha + (1 - (\text{CommCost}(\text{Coverload}, C) / \text{MaxComm})) \times \beta \quad (۴)$$

در رابطه $E_Dependency(t, C)$ میزان وابستگی خارجی وظیفه t به کنترلر C ، $\text{CommCost}(\text{Coverload}, C)$ هزینه ارتباطی میان کنترلر اضافه بار و کنترلر مقصد، MaxComm بیشترین مقدار هزینه ارتباطی میان کنترلر اضافه بار و سایر کنترلرها و α و β ضرایب وزنی هستند که مجموع آن‌ها برابر با ۱ است. وزن‌دهی پارامترهای α و β به صورت تجربی تعیین شده است. با توجه به اهمیت بالاتر وابستگی خارجی در بهبود زمان پاسخ و کاهش هزینه مهاجرت مقدار α بزرگتر از β در نظر گرفته شد. در تنظیمات آزمایش‌ها، مقادیر $\alpha=0.6$ و $\beta=0.4$ استفاده شدند که باعث ایجاد تعادل مناسب میان وابستگی خارجی و هزینه ارتباطی گردید. کنترلری که بالاترین

مقداری (C) Score را داشته باشد، به عنوان کنترلر مقصد مهاجرت انتخاب می‌شود. شبه کد فرآیند انتخاب کنترلر مقصد در الگوریتم ۴ آورده شده است.

```
Pseudocode 4 shows the select target controller algorithm.
Input: t, Overload_Controller, CandidateControllers, E_Dependency[t][C],
CommCost[Ci][Cj],  $\alpha$ ,  $\beta$ 
Output: TargetController
Algorithm:
maxComm  $\leftarrow$  0
for each C in CandidateControllers do
    if CommCost[Overload_Controller][C] > maxComm then
        maxComm  $\leftarrow$  CommCost[Overload_Controller][C]
bestScore  $\leftarrow$   $-\infty$ 
TargetController  $\leftarrow$  None
for each C in CandidateControllers do
    dep  $\leftarrow$  E_Dependency[t][C]
    normCost  $\leftarrow$  CommCost[Overload_Controller][C] / maxComm
    score  $\leftarrow$   $\alpha * \text{dep} + \beta * (1 - \text{normCost})$ 
    if score > bestScore then
        bestScore  $\leftarrow$  score
        TargetController  $\leftarrow$  C
return TargetController
```

به طور خلاصه، این پژوهش، رویکرد مهاجرت وظایف در راستای تعادل بار در اینترنت اشیا مبتنی بر شبکه نرم‌افزار محور پیشنهاد می‌کند که هدف اصلی بهبود پارامترهای کیفیت سرویس است. در این رویکرد، کنترلر متعادل کننده بار می‌تواند در مورد انجام وظایف ورودی و یا هدایت وظایف به سایر کنترلرها تصمیم‌گیری نماید. رویکرد پیشنهادی با انتخاب بهینه وظایف مهاجرتی و تخصیص به مناسب‌ترین کنترلر مقصد بر اساس وابستگی‌های داخلی و خارجی، از بروز اضافه بار جلوگیری می‌کند. نتایج نشان می‌دهد که این رویکرد منجر به کاهش زمان پاسخ، تعداد مهاجرت‌ها، هزینه مهاجرت و مصرف انرژی شده و در عین حال بهره‌وری منابع را افزایش می‌دهد. مقایسه روش پیشنهادی با چهار رویکرد مرتبط با موضوع پژوهش، نشان می‌دهد که در روش پیشنهادی، وظایف کمتری نیاز به مهاجرت پیدا می‌کنند و با انتقال وظایف به کنترلرهای با وابستگی بالاتر، مهاجرت‌های غیر ضروری کاهش یافته و مصرف انرژی بهینه می‌شود. پس از هر فرآیند مهاجرت وظیفه، ماتریس وابستگی بین کنترلر اضافه بار و مقصد به‌روزرسانی می‌گردد تا تصمیم‌گیری‌های بعدی دقیق‌تر انجام شود.

شبیه‌سازی و نتایج

برای ارزیابی روش پیشنهادی، آزمایش‌ها بر روی پلتفرم Mininet [۲۱] (نسخه 2.3.0d6) انجام شد. کنترلرها با استفاده از RYU [۲۲] (نسخه 4.34) مبتنی بر زبان برنامه‌نویسی پایتون^۱ تحت مجوز Apache 2.0 پیاده‌سازی گردید. RYU به دلیل انعطاف‌پذیری، پشتیبانی از پروتکل OpenFlow و سازگاری با محیط‌های مقیاس‌پذیر اینترنت اشیا انتخاب شد. توپولوژی‌های آزمایشی مطابق جدول ۴ (NSF, Atlanta, ARN) با تعداد

^۱ Python

مختلف گره‌ها و یال‌ها ایجاد شدند. هر توپولوژی شامل چندین سوئیچ، سرور تولید ترافیک و کنترلر می‌باشد. سوئیچ‌ها از پروتکل OpenFlow 1.3 برای مدیریت وظایف و تبادل پیام‌های کنترلر با کنترلر استفاده نمودند. برای تولید وظایف، از ابزار iperf3 (نسخه 3.1.3) استفاده شد [۲۳] و وظایف با اندازه بسته ۵۱۲ بایت و نرخ ارسال متغیر (۱۰ تا ۵۰ Mbps) ایجاد شدند. هر آزمایش به مدت ۳۰۰ ثانیه اجرا و با ۱۰ بار تکرار، میانگین نتایج گزارش گردید. آزمایش شبیه‌سازی بر روی رایانه شخصی با پردازنده Intel Core i7 CPU@3.40GHz و ۱۲ گیگابایت حافظه که ویندوز ۱۰ و سیستم عامل Virtualbox Ubuntu 16.04 با هسته لینوکس نسخه ۴،۴ انجام گردید. در الگوریتم پیشنهادی، سه گام اساسی پیاده‌سازی شد. (۱) فاز انتخاب وظیفه: رتبه‌بندی وظایف بر اساس ماتریس وابستگی داخلی و انتخاب وظیفه با کمترین وابستگی. (۲) فاز یافتن وظایف همکار: شناسایی وابستگی‌های خارجی هر وظیفه به سایر کنترلرها. (۳) فاز نگاشت وظیفه: انتخاب کنترلر مقصد بر اساس بیشترین وابستگی خارجی و کمترین هزینه ارتباطی.

انتخاب توپولوژی

عملکرد روش پیشنهادی با استفاده از توپولوژی‌های شبکه، شامل Atlanta، ARN و NSF ارزیابی شد. ویژگی‌های توپولوژی‌های شبکه در جدول ۴ خلاصه شده است. تعداد سوئیچ‌های مدیریت شده توسط هر کنترلر RYU بر اساس مقیاس توپولوژی از ۵ تا ۲۰ تنظیم گردید.

جدول ۴. ویژگی توپولوژی‌های شبکه.

توپولوژی	ARN	Atlanta	NSF
تعداد گره	۳۰	۱۵	۱۳
تعداد یال	۲۹	۲۲	۱۵
تعداد کنترلر	۴	۳	۳
ظرفیت کنترلرها (واحد)	۱۰۰	۱۰۰	۱۰۰
حد آستانه کنترلرها (واحد)	۸۵	۸۰	۸۰

رویکردهای موجود

برای ارزیابی عملکرد روش پیشنهادی (TMLB^۱)، به مقایسه با چهار رویکرد مرتبط پرداخته می‌شود. (۱) SMSC^۲: این رویکرد نگاشت ایستا بین سوئیچ و کنترلر را در نظر می‌گیرد و از تکنیک تعادل بار پیروی نمی‌کند [۲۴].

(۲) DLBM^۳: رویکرد تعادل بار که سوئیچ با بالاترین نرخ انتقال را از کنترلر اضافه بار انتخاب نموده و به طور تصادفی به نزدیکترین کنترلر کم بار مهاجرت می‌دهد [۲۵]. این می‌تواند منجر به زمان اجرای سریع و کاهش

^۱ Task Migration-based Load Balancing

^۲ Static Mapping between Switch and Controller

^۳ Dynamic Load Balancing Mechanism

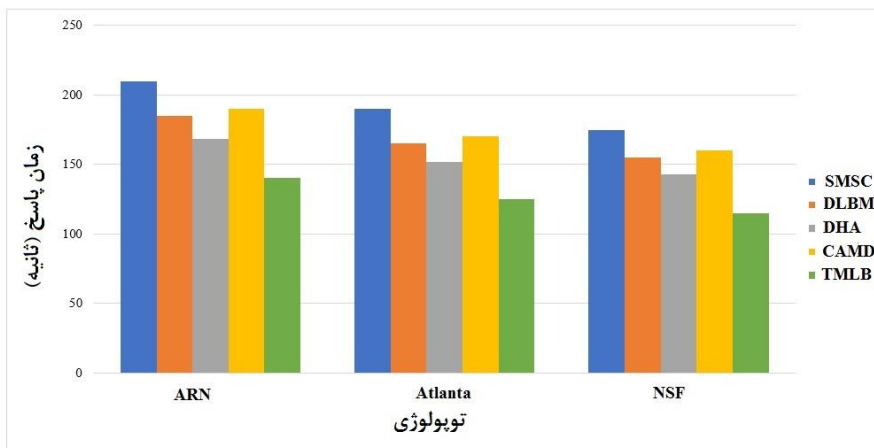
هزینه مهاجرت شود. با این حال، این فرآیند موجب افزایش ازدحام در هنگام انتخاب کنترلر مقصد جهت پذیرش وظایف می‌شود.

۳) DHA^۱: این رویکرد به طور تصادفی سوئیچ با حداکثر انحراف بار را از کنترلر اضافه بار انتخاب می‌کند و به کنترلر کم بار با بالاترین ظرفیت باقیمانده مهاجرت می‌دهد [۲۶].

۴) CAMD^۲: انتخاب سوئیچ با سبک‌ترین بار کاری و مهاجرت به کنترلر کم بار از میان مجموعه کنترلرهای کم بار پیشنهاد شده است. این رویکرد منجر به نرخ تعادل بار پایین می‌شود و می‌تواند باعث کاهش راندمان شبکه، توان عملیاتی کم، زمان پاسخ بالا و از دست دادن بسته زیاد شود [۲۷].

ارزیابی زمان پاسخ

زمان پاسخ کنترلر هنگام وقوع عدم تعادل بار به شدت افزایش می‌یابد. نرخ انتقال پیام‌های PACKET-IN برای اضافه بار برخی از کنترلرها تغییر می‌یابد تا تفاوت در زمان پاسخ رویکردهای مختلف تحت شرایط ترافیکی مختلف مشاهده گردد. از شکل ۳، می‌توان درک نمود که زمان پاسخ کنترلر در چهار رویکرد دیگر با افزایش بار کاری در مقایسه با رویکرد پیشنهادی (TMLB) که زیر ۱۴۰ میلی‌ثانیه می‌باشد، افزایش می‌یابد.



شکل ۳. نتایج میانگین زمان پاسخ کنترلر.

این نشان می‌دهد که رویکرد پیشنهادی TMLB عملکرد تعادل بار کنترلر را از طریق مهاجرت وظیفه و انتخاب کنترلر مقصد با در نظر گرفتن وابستگی خارجی و هزینه ارتباطی، بهبود می‌بخشد. زمان پاسخ رویکرد TMLB تحت شرایط ترافیکی توپولوژی ARN به طور متوسط حدود ۴۶٫۶ درصد، ۳۶٫۳ درصد، ۲۹٫۸ درصد و ۳۵٫۵ درصد در مقایسه با SMSC، DLBM، DHA و CAMD کاهش می‌یابد. همچنین مشاهده می‌شود که

^۱ Distributed Hopping Algorithm

^۲ Controller Adaptation and Migration Decision

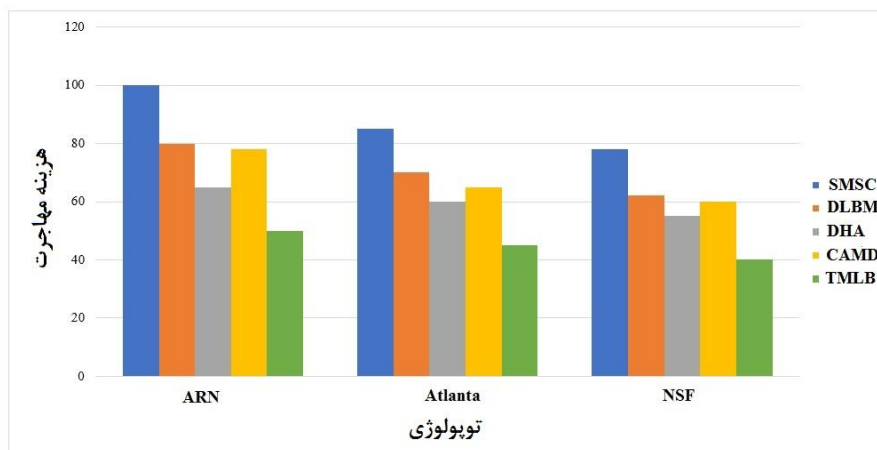
رویکردهای DHA و DLBM نتایج بهتری را ارائه می‌دهند و زمان پاسخ کنترلر در مقایسه با SMSC و CAMD کاهش می‌یابد.

ارزیابی هزینه مهاجرت

برای محاسبه هزینه مهاجرت وظیفه از کنترلر اضافه بار به کنترلر مقصد (هدف)، از رابطه ۵ استفاده می‌شود.

$$\text{MigrationCost}(t_i, C_o, C_d) = \text{CommCost}(C_o, C_d) + \text{DataSize}[i] + \text{StateSync}(C_o, C_d) \quad (5)$$

که t_i وظیفه C_o کنترلر اضافه بار، C_d کنترلر مقصد، $\text{CommCost}(C_o, C_d)$ تأخیر ارتباطی میان کنترلر اضافه بار و مقصد (بر حسب ثانیه)، $\text{DataSize}[i]$ حجم داده مربوط به وظیفه C_o (بر حسب بایت) و $\text{StateSync}(C_o, C_d)$ هزینه همگام‌سازی جدول جریان و وضعیت بین کنترلرها می‌باشند. در شکل ۴ میزان هزینه مهاجرت TMLB در مقایسه با سایر رویکردهای مطرح شده را نشان می‌دهد که کاهش قابل توجهی داشته است. روش پیشنهادی با دقت بیشتر در انتخاب کنترلر مقصد، هزینه مهاجرت را کاهش می‌دهد.



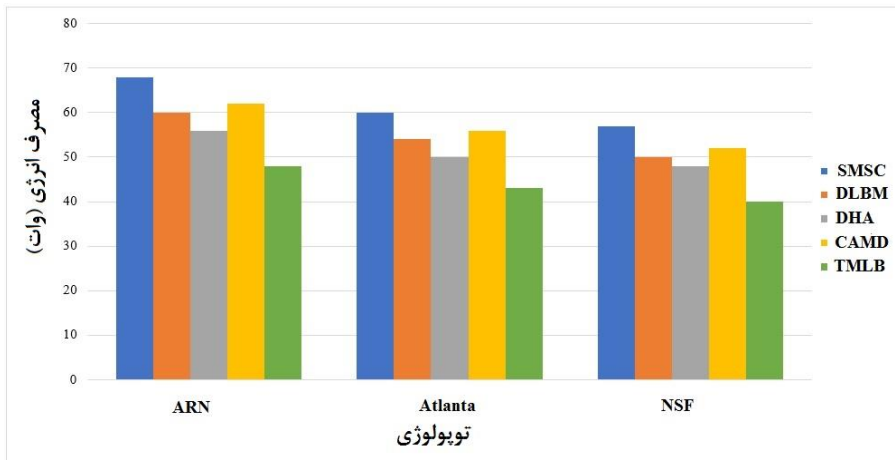
شکل ۴. نتایج عملکرد میانگین هزینه مهاجرت.

هزینه مهاجرت روش پیشنهادی تحت توپولوژی Atlanta به طور متوسط حدود ۴۵ درصد نسبت به DLBM و ۳۷ درصد نسبت به DHA و ۴۸ درصد نسبت به CAMD کاهش می‌یابد. در مجموع، هزینه مهاجرت TMLB به طور متوسط ۴۳ درصد بهبود می‌یابد. TMLB وظیفه را بر اساس کمترین وابستگی داخلی به سوئیچ‌های کنترلر اضافه بار برای مهاجرت انتخاب می‌کند و حداقل هزینه مهاجرت را در طول انتخاب کنترلر مقصد در نظر می‌گیرد.

ارزیابی مصرف انرژی

همچنین روش پیشنهادی، مهاجرت وظیفه بر اساس اولویت و اختصاص به کنترلر با وابستگی خارجی بالا به سوئیچ‌ها می‌باشد که این عمل باعث کاهش تعداد مهاجرت‌ها می‌گردد، در نتیجه میزان مصرف انرژی کمتر از

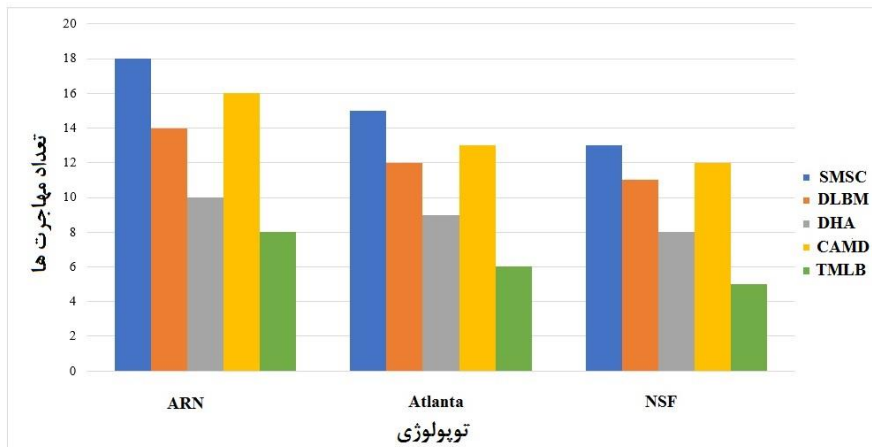
رویکردهای دیگر است. لذا می‌توان بیان نمود روش پیشنهادی دارای انرژی مصرفی کمتری نسبت به سایر رویکردها می‌باشد. در شکل ۵، میزان مصرف انرژی رویکردهای مختلف نشان داده شده است. بنابراین، با مهاجرت هدفمند وظایف از کنترلرهای اضافه بار به کنترلرهای بهینه، مصرف انرژی کل شبکه کاهش یافته است. انرژی مصرفی پایین‌تر به معنای مدیریت بارکاری مؤثر در شبکه است.



شکل ۵. میزان انرژی مصرفی.

ارزیابی تعداد مهاجرت

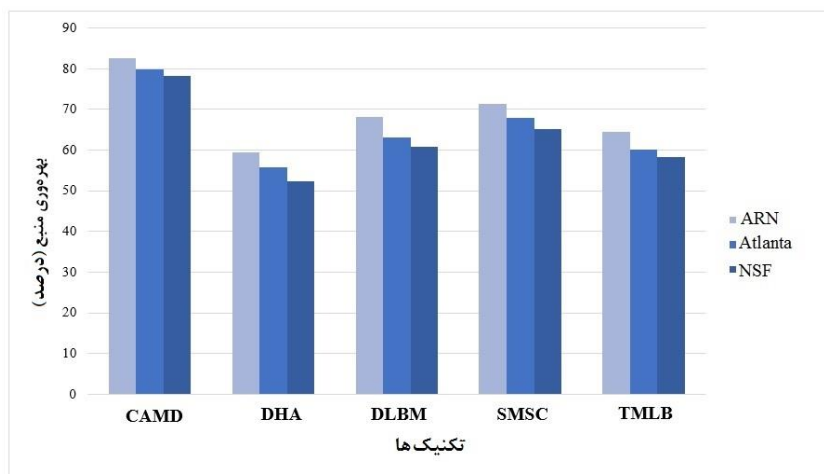
در شکل ۶ تعداد وظایف مورد مهاجرت در طول شبیه‌سازی را نشان می‌دهد که در روش پیشنهادی کمترین تعداد مهاجرت دیده می‌شود و این نشان می‌دهد که در روش پیشنهادی وظایف کمتری نیاز به مهاجرت پیدا می‌کنند. لذا با ارزیابی وابستگی وظایف، نسبتاً مهاجرت‌های ضروری انجام می‌شود.



شکل ۶. تعداد مهاجرت وظایف.

ارزیابی بهره‌وری منبع

در راستای بهبود کیفیت سرویس، میزان بهره‌وری منابع (مشغولی پردازنده) در رویکردهای مختلف بر اساس توپولوژی‌های مورد بررسی در شکل ۷ نشان داده شده است. نتایج نشان می‌دهد که روش پیشنهادی بالاترین میانگین بهره‌وری منبع پردازنده را در توپولوژی‌ها نسبت به سایر رویکردها ارائه می‌کند.



شکل ۷. میزان بهره‌وری منابع.

روش پیشنهادی به دلیل مهاجرت وظایف بر اساس تحلیل وابستگی و هزینه ارتباطی، باعث استفاده هدمندتر از منابع پردازشی کنترلرها شده است. در هر سه توپولوژی، روش TMLB نزدیک به بیشینه آستانه بهره‌وری عمل نموده و موجب افزایش عملکرد شبکه شده است. SMSC با رویکرد نگاشت ایستا و بدون تکنیک تعادل بار، کمترین بهره‌وری را داشته است. DLBM و DHA در بهره‌وری منبع عملکرد بهتری نسبت به SMSC و CAMD دارند، اما به دلیل تصمیم‌گیری‌های تصادفی یا بدون در نظر گرفتن وابستگی وظایف، همچنان منابع به صورت بهینه استفاده نشده‌اند.

به منظور بررسی و تحلیل پارامترهای کیفیت سرویس در رویکردهای مختلف، جدول ۵ مقایسه‌ای بین رویکردها بر اساس میانگین مقادیر در هر توپولوژی ارائه می‌دهد. در این جدول، هر رویکرد بر اساس پارامترهای زمان پاسخ، هزینه مهاجرت، تعداد مهاجرت، مصرف انرژی و بهره‌وری منبع مورد ارزیابی قرار گرفته است.

جدول ۵. مقایسه رویکردها بر اساس مقادیر پارامترهای کیفیت سرویس.

پارامتر (میانگین)	SMSC	DLBM	DHA	CAMD	TMLB
زمان پاسخ	۱۹۱/۶۷	۱۶۸/۳۳	۱۵۴/۳۳	۱۷۳/۳۴	۱۲۶/۶۷
هزینه مهاجرت	۸۷/۶۷	۷۰/۶۷	۶۰/۰	۶۷/۶۷	۴۵/۰
تعداد مهاجرت	۱۵/۳	۱۲/۳	۹	۱۳/۶	۶/۳
مصرف انرژی	۶۱/۸	۵۴/۷	۵۱/۳	۵۶/۶	۴۳/۶

بهره‌وری منبع	۵۵/۸	۶۴/۱	۶۸/۲	۶۱	۸۰/۲
---------------	------	------	------	----	------

همچنین، به منظور روشن‌تر ساختن برتری‌های روش پیشنهادی، در جدول ۶ مقایسه‌ای میان روش‌های SMSC، DHA، DLBM و CAMD و روش پیشنهادی (TMLB) از نظر ایده اصلی، معیار انتخاب سوئیچ، معیار انتخاب کنترلر مقصد، پارامتر ارزیابی و محدودیت‌ها ارائه گردیده است.

جدول ۶. مقایسه رویکرد پیشنهادی و رویکردهای موجود مهاجرت سوئیچ.

مرجع	ایده اصلی	معیار انتخاب سوئیچ	معیار انتخاب کنترلر مقصد	کیفیت سرویس	محدودیت‌ها
SMSC [24]	نگاشت ایستایین سوئیچ و کنترلر	ثابت (بدون انتخاب پویا)	ثابت	زمان پاسخ	عدم تعادل بار، گلوگاه ^۱ در کنترلرها
DLBM [25]	تعادل بار پویا با مهاجرت سوئیچ پربار	سوئیچ با بیشترین نرخ انتقال	نزدیک‌ترین کنترلر کم بار (تصادفی)	زمان پاسخ، هزینه مهاجرت	احتمال ازدحام در کنترلر مقصد، عدم توجه به وابستگی وظایف
DHA [26]	مهاجرت مبتنی بر انحراف بار کنترلرها	سوئیچ با بیشترین انحراف بار از کنترلر اضافه بار	کنترلر کم بار با بیشترین ظرفیت باقیمانده	زمان پاسخ، مقیاس‌پذیری	عدم در نظر گرفتن هزینه ارتباطی و وابستگی وظایف
CAMD [27]	مهاجرت مبتنی بر انتخاب سبک‌ترین سوئیچ	سبک‌ترین سوئیچ	کنترلر کم بار (تصادفی)	زمان پاسخ، توان عملیاتی	تعادل بار ضعیف، توان عملیاتی پایین
TMLB روش پیشنهادی	مهاجرت بر اساس ماتریس وابستگی	وظایف سوئیچ با وابستگی داخلی کم از کنترلر اضافه بار	کنترلر کم بار با بیشترین وابستگی خارجی و کمترین هزینه ارتباطی	زمان پاسخ، هزینه مهاجرت، مصرف انرژی، بهره‌وری منبع	نیاز به محاسبه ماتریس وابستگی (پیچیدگی محاسباتی)

در رویکردهای موجود تمرکز بر روی نرخ ترافیک وظایف و یا ظرفیت باقیمانده کنترلرها می‌باشد، روش پیشنهادی با در نظر گرفتن وابستگی وظایف و هزینه ارتباطی قادر است مهاجرت وظایف را بهینه‌سازی نماید که منجر به کاهش زمان پاسخ، هزینه مهاجرت و مصرف انرژی و افزایش بهره‌وری منابع شده است. جدول ۷ مقایسه آماری روش پیشنهادی (TMLB) با چهار رویکرد موجود را بر اساس ۱۰ بار تکرار شبیه‌سازی و معیارهای زمان پاسخ، هزینه مهاجرت، تعداد مهاجرت، مصرف انرژی و بهره‌وری منبع نشان می‌دهد.

جدول ۷. مقایسه آماری رویکرد پیشنهادی با رویکردهای موجود.

پارامتر	رویکردهای موجود	SD $\pm$ میانگین	TMLB $\pm$ میانگین SD	t-value	p-value
---------	-----------------	------------------	-----------------------	---------	---------

¹ Bottleneck

زمان پاسخ	SMSC	191.7 ± 15.3	126.7 ± 10.1	-11.2	<0.001
	DLBM	168.3 ± 13.5	126.7 ± 10.1	-7.8	<0.001
	DHA	154.3 ± 12.3	126.7 ± 10.1	-5.5	<0.001
	CAMD	173.3 ± 13.9	126.7 ± 10.1	-8.6	<0.001
هزینه مهاجرت	SMSC	87.7 ± 7.0	45.0 ± 3.6	-17.1	<0.001
	DLBM	70.7 ± 5.7	45.0 ± 3.6	-12.1	<0.001
	DHA	60.0 ± 4.8	45.0 ± 3.6	-7.9	<0.001
	CAMD	67.7 ± 5.4	45.0 ± 3.6	-11.0	<0.001
تعداد مهاجرت	SMSC	15.3 ± 2.3	6.3 ± 0.9	-11.5	<0.001
	DLBM	12.0 ± 1.8	6.3 ± 0.9	-8.7	<0.001
	DHA	10.0 ± 1.5	6.3 ± 0.9	-6.0	<0.001
	CAMD	13.0 ± 2.0	6.3 ± 0.9	-9.1	<0.001
مصرف انرژی	SMSC	61.8 ± 4.9	43.6 ± 3.5	-9.5	<0.001
	DLBM	54.0 ± 4.3	43.6 ± 3.5	-6.3	<0.001
	DHA	51.0 ± 4.1	43.6 ± 3.5	-4.5	<0.001
	CAMD	56.0 ± 4.5	43.6 ± 3.5	-6.9	<0.001
بهره‌وری منبع	SMSC	55.8 ± 3.3	80.2 ± 4.8	+13.2	<0.001
	DLBM	64.0 ± 3.8	80.2 ± 4.8	+8.0	<0.001
	DHA	68.0 ± 4.1	80.2 ± 4.8	+6.3	<0.001
	CAMD	61.0 ± 3.7	80.2 ± 4.8	+7.2	<0.001

همان‌طور که مشاهده می‌شود، روش TMLB در تمامی معیارها نسبت به رویکردهای موجود عملکرد بهتری دارد. تمامی تفاوت‌ها از نظر آماری معنادار هستند ($p < 0.001$)، که برتری روش TMLB را تأیید می‌کند. این نتایج نشان می‌دهد که مهاجرت وظایف مبتنی بر وابستگی، تعادل بار بین کنترلرها را به‌طور مؤثر برقرار کرده، بهره‌وری منابع را بهبود می‌بخشد و کیفیت سرویس شبکه‌های اینترنت اشیا را افزایش می‌دهد.

پیچیدگی زمانی رویکرد پیشنهادی

به‌منظور تحلیل کارایی محاسباتی رویکرد پیشنهادی، سه مرحله الگوریتم از نظر پیچیدگی زمانی مورد بررسی قرار می‌گیرد. در مرحله اول، تشکیل ماتریس وابستگی شامل محاسبه میزان تعامل هر وظیفه با سوئیچ‌های متصل به کنترلر است. از آنجا که برای هر n وظیفه، وابستگی به m سوئیچ محاسبه می‌شود، پیچیدگی زمانی این مرحله برابر با $O(n \times m)$ است. به‌روزرسانی ماتریس نیز در صورت تغییر وضعیت شبکه یا مهاجرت وظیفه، با همان مرتبه زمانی انجام می‌شود. در مرحله دوم، انتخاب وظیفه با کمترین وابستگی داخلی نیازمند پیمایش سطوح ماتریس وابستگی برای n وظیفه است که دارای زمان $O(n)$ می‌باشد. در مرحله سوم، فرآیند انتخاب کنترلر مقصد بر اساس وابستگی خارجی و هزینه ارتباطی میان k کنترلر کاندید انجام می‌شود که دارای زمان $O(k)$ است. در نهایت، پیچیدگی زمانی کلی رویکرد پیشنهادی $O(n \times m) + O(n) + O(k)$ خواهد بود.

نتیجه‌گیری

با توجه به گسترش سریع کاربردهای اینترنت اشیا، مدیریت بهینه وظایف ورودی برای ارتقای عملکرد و تضمین کیفیت سرویس شبکه‌های اینترنت اشیا حیاتی است. کنترل ازدحام وظایف در شبکه نیازمند به‌کارگیری تکنیک‌های مؤثر تعادل بار است. شبکه نرم‌افزارمحور با قابلیت مدیریت و کنترل سراسری شبکه، امکان توزیع مؤثر وظایف بین کنترلرها بر اساس ظرفیت موجود را فراهم می‌کند. این تحقیق بر تعادل بار مبتنی بر مهاجرت

وظایف از کنترلرهای دارای اضافه بار به کنترلرهای کم بار در شبکه نرم افزار محور تمرکز دارد. نوآوری استفاده شده در این پژوهش شامل: انتخاب وظایف برای مهاجرت در کنترلر اضافه بار بر اساس وابستگی داخلی سوئیچها، استفاده از مصرف انرژی به عنوان معیار وضعیت بارکاری کنترلر، تخصیص مؤثر وظایف به کنترلرهای کم بار با بیشترین وابستگی خارجی و کمترین هزینه ارتباطی، که منجر به بهبود همزمان زمان پاسخ، بهره‌وری منابع و مصرف انرژی می‌شود.

نتایج شبیه سازی در سه توپولوژی (NSF, Atlanta, ARN) نشان داد که رویکرد پی‌شنهادی (TMLB) نسبت به رویکردهای SMSC, DLBM, DHA و CAMD، عملکرد شبکه را بهبود بخشیده و کیفیت سرویس را افزایش می‌دهد. به طور میانگین، زمان پاسخ کنترلر ۳۶/۳ در صد، هزینه مهاجرت ۴۳ در صد، تعداد مهاجرت ۲۶ درصد، مصرف انرژی کنترلرها ۲۱ درصد کاهش یافته و بهره‌وری منابع ۱۹/۷ درصد افزایش یافته است. روش پیشنهادی در کاربردهای واقعی نظیر اینترنت اشیا صنعتی (جلوگیری از تأخیر در خطوط تولید هوشمند)، شبکه‌های هوشمند انرژی (تضمین پاسخ سریع کنترلرها در مدیریت بار شبکه برق)، شهر هوشمند (مدیریت وظایف حسگرهای ترافیکی و امنیتی برای کاهش ازدحام داده‌ها) و سلامت هوشمند (حفظ کیفیت سرویس در انتقال داده‌های حیاتی بیماران) مورد اهمیت است.

محدودیت‌های پژوهش موارد زیر در نظر گرفته شده است:

- آزمایش‌ها در محیط شبیه ساز Mininet و بر روی سه توپولوژی انجام شدند و قابلیت تعمیم به شرایط واقعی نیازمند بررسی‌های بیشتر است.

- پارامترهای ارزیابی محدود به زمان پاسخ، هزینه مهاجرت، تعداد مهاجرت، مصرف انرژی و بهره‌وری منبع بوده و سایر پارامترهای کیفیت سرویس نظیر نرخ ازدست‌دادن بسته‌ها یا تأخیر انتهابه‌انتها بررسی نشدند.

- کنترلرها در این تحقیق به صورت همگن در نظر گرفته شدند، در حالی که در محیط‌های واقعی، ناهمگنی منابع و سخت‌افزار کنترلرها می‌تواند بر کارایی مهاجرت وظایف اثرگذار باشد.

پیشنهادهای زیر برای تحقیقات آینده مطرح می‌گردد:

- تأثیر معماری شبکه و ارتباط کنترلرها: نوع معماری و نحوه ارتباط کنترلرها می‌تواند عملکرد مهاجرت وظایف را تحت تأثیر قرار دهد. معماری سلسله‌مراتبی درختی کنترلرها می‌تواند در بهبود پارامترهای کیفیت سرویس موثرتر باشد.

- پیش‌بینی کیفیت سرویس قبل از تصمیم‌گیری: با توجه به تغییرات پویا در شبکه، پیش‌بینی و ارزیابی مقادیر پارامترهای کیفیت سرویس بر اساس توافقنامه سطح سرویس^۱ می‌تواند تصمیمات مهاجرت وظایف را بهینه سازد.

پیوست

به منظور تکرارپذیری آزمایش‌های انجام‌شده و امکان بازتولید نتایج در محیط‌های مشابه، جزئیات کامل مربوط به تنظیمات محیط و پارامترهای شبیه‌سازی در جدول ۸ ارائه شده است.

جدول ۸. تنظیمات محیط و پارامترهای شبیه‌سازی.

^۱ Service Level Agreement (SLA)

نام پارامتر	مقدار
محیط شبیه‌سازی	Mininet 2.3.0d6
کنترلر	RYU 4.34
زبان برنامه‌نویسی	Python 3.8
سیستم‌عامل میزبان	Ubuntu 16.04 (VirtualBox)
ابزار تولید ترافیک	iperf3 (نسخه 3.1.3)
پردازنده	Intel Core i7 @ 3.4GHz
حافظه	12 GB
نسخه OpenFlow	1.3
پهنای باند لینک‌ها	Mbps ۱۰۰
تاخیر لینک‌ها	۱۰-۲۰ میلی ثانیه
نوع لینک بین کنترلرها	لینک TCP مجازی در Mininet
آستانه انرژی کنترلرها	۸۰ تا ۸۵ واحد
نرخ پردازش کنترلر	10^6 بایت بر ثانیه
اندازه داده وظایف	۵۱۲ بایت
نرخ ارسال وظایف	۱۰ تا ۵۰ Mbps
سیاست زمانبندی وظایف	FIFO (First In, First Out)
تعداد کنترلر	۳ الی ۴
ضرایب وزنی انتخاب کنترلر مقصد	$\beta = 0.4, \alpha = 0.6$
تعداد تکرار هر آزمایش	۱۰ بار
مدت زمان هر آزمایش	۳۰۰ ثانیه
نوع توپولوژی	NSF، Atlanta، ARN
پارامترهای مورد ارزیابی	زمان پاسخ، هزینه مهاجرت، تعداد مهاجرت، مصرف انرژی، بهره‌وری منبع

فهرست منابع

- [1] Pourbahrami, S., Emadi, M. (2024). Cluster Head Selection Algorithm for Internet of Things in Wireless Networks based on the Density Peak Clustering. Karafan Journal, 21(1), 91-107. <https://doi.org/10.48301/kssa.2024.415637.2702>. (in persian)
- [2] Ali, J., Jhaveri, R. H., Alswailim, M., Roh, B. H. (2023). ESCALB: An effective slave controller allocation-based load balancing scheme for multi-domain SDN-enabled-IoT networks. Journal of King Saud University-Computer and Information Sciences, 35(6), 101566. <https://doi.org/10.1016/j.jksuci.2023.101566>.
- [3] Mahmoudian, M., Khorsand, R., Ramezanzpour, M. (2022). A Prediction based Proactive Resource Provisioning Strategy for Multi-objective Workflow Scheduling in Cloud Computing. Karafan Journal, 19(3), 529-551. <https://doi.org/10.48301/kssa.2022.341879.2104>. (in persian)
- [4] Montazerolghaem, A., Yaghmaee, M. H. (2020). Load-balanced and QoS-aware software-defined Internet of Things. IEEE Internet of Things Journal, 7(4), 3323-3337. <https://doi.org/10.1109/JIOT.2020.2967081>.

- [5] Belgaum, M. R., Musa, S., Alam, M. M., Su'ud, M. M. (2020). A systematic review of load balancing techniques in software-defined networking. *Ieee Access*, 8, 98612-98636. <https://doi.org/10.1109/ACCESS.2020.2995849>.
- [6] Sabzevari, K., Soleymani, A. (2024). Providing a Routing Method in Internet of Things for Better Load Distribution Using the Improved Gray Wolf Optimization Algorithm. *Karafen Journal*, 21(1), 109-131. <https://doi.org/10.48301/kssa.2024.421630.2740>. (in persian)
- [7] Sun, P., Guo, Z., Wang, G., Lan, J., Hu, Y. (2020). MARVEL: Enabling controller load balancing in software-defined networks with multi-agent reinforcement learning. *Computer Networks*, 177, 107230. <https://doi.org/10.1016/j.comnet.2020.107230>.
- [8] Zafar, S., Lv, Z., Zaydi, N. H., Ibrar, M., Hu, X. (2022). DSMLB: Dynamic switch-migration based load balancing for software-defined IoT network. *Computer Networks*, 214, 109145. <https://doi.org/10.1016/j.comnet.2022.109145>.
- [9] Shi, Y., Yang, Q., Huang, X., Li, D., Huang, X. (2022). An sdn-enabled framework for a load-balanced and qos-aware internet of underwater things. *IEEE Internet of Things Journal*, 10(9), 7824-7834. <https://doi.org/10.1109/JIOT.2022.3231329>.
- [10] Shahzad, M., Liu, L., Kaushik, A., Bibi, I., Belkout, N. E. (2025). Fair switch selection for large scale software defined networks in next generation internet of things. *Telecommunication Systems*, 88(2), 1-14. <https://doi.org/10.1007/s11235-025-01290-2>.
- [11] Adekoya, O., Aneiba, A., Patwary, M. (2020). An improved switch migration decision algorithm for SDN load balancing. *IEEE Open Journal of the Communications Society*, 1, 1602-1613. <https://doi.org/10.1109/OJCOMS.2020.302897>.
- [12] Zare Soltani, M. A., Hosseini Seno, S. A., Mohajerzadeh, A. (2025). Efficient dynamic load balancing in software-defined networks using policy gradient: a strategy for enhanced QoS and reduced energy consumption. *Computing*, 107(7), 1-78. <https://doi.org/10.1007/s00607-025-01498-9>.
- [13] Shirani, L., Movahednejad, H., Sharifi, M., Mahmoudi, M. (2025). Improving load balancing in distributed software-defined networks with a bidirectional long short-term memory-based controller design. *The Journal of Supercomputing*, 81(10), 1137. <https://doi.org/10.1007/s11227-025-07595-y>.
- [14] Babbar, H., Rani, S., Gupta, D., Aljahdali, H. M., Singh, A., Al-Turjman, F. (2021). Load balancing algorithm on the immense scale of internet of things in SDN for smart cities. *Sustainability*, 13(17), 9587. <https://doi.org/10.3390/su13179587>.
- [15] Cao, B., Sun, Z., Zhang, J., Gu, Y. (2021). Resource allocation in 5G IoV architecture based on SDN and fog-cloud computing. *IEEE transactions on intelligent transportation systems*, 22(6), 3832-3840. <https://doi.org/10.1109/TITS.2020.3048844>.
- [16] Xu, C., Xu, C., Li, B., Li, S., Li, T. (2023). Load-aware dynamic controller placement based on deep reinforcement learning in SDN-enabled mobile cloud-edge computing networks. *Computer Networks*, 234, 109900. <https://doi.org/10.1016/j.comnet.2023.109900>.
- [17] Huang, V., Fu, Q., Chen, G., Wen, E., Hart, J. (2017, October). BLAC: A bindingless architecture for distributed SDN controllers. In 2017 IEEE 42nd Conference on Local Computer Networks (LCN) (146-154). IEEE. <https://doi.org/10.1109/LCN.2017.62>.

- [18] Li, J., Shen, X., Chen, L., Ou, J., Wosinska, L., Chen, J. (2018). Delay-aware bandwidth slicing for service migration in mobile backhaul networks. *Journal of Optical Communications and Networking*, 11(4), B1-B9. <https://doi.org/10.1364/JOCN.11.0000B1>.
- [19] Punitha, V., Mala, C. (2021). Traffic classification for efficient load balancing in server cluster using deep learning technique. *The Journal of Supercomputing*, 77(8), 8038-8062. <https://doi.org/10.1007/s11227-020-03613-3>.
- [20] Rostami, M., Goli-Bidgoli, S. (2024). An overview of QoS-aware load balancing techniques in SDN-based IoT networks. *Journal of cloud computing*, 13(1), 89. <https://doi.org/10.1186/s13677-024-00651-7>.
- [21] Mininet Emulator. Accessed: Oct. 20, 2018. [Online]. Available: <http://mininet.org/>
- [22] Ryu SDN Framework, <http://osrg.github.io/ryu/>.
- [23] iPerf, <https://iperf.fr/>.
- [24] Heller, B., Sherwood, R., McKeown, N. (2012). The controller placement problem. *ACM SIGCOMM Computer Communication Review*, 42(4), 473-478. <https://doi.org/10.1145/2377677.2377767>.
- [25] Lan, W., Li, F., Liu, X., Qiu, Y. (2018, February). A dynamic load balancing mechanism for distributed controllers in software-defined networking. In 2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) (pp. 259-262). IEEE. <https://doi.org/10.1109/ICMTMA.2018.00069>.
- [26] Cheng, G., Chen, H., Wang, Z., Chen, S. (2015, May). DHA: Distributed decisions on the switch migration toward a scalable SDN control plane. In 2015 IFIP Networking Conference (IFIP Networking) (1-9). IEEE. <https://doi.org/10.1109/IFIPNetworking.2015.7145319>.
- [27] Sahoo, K. S., Sahoo, B. (2019). CAMD: a switch migration based load balancing framework for software defined networks. *IET Networks*, 8(4), 264-271. <https://doi.org/10.1049/iet-net.2018.5166>.
- [28] Shahrabaki, P. P., Coutinho, R. W., Shayan, Y. R. (2024). SDN-LB: A novel server workload balancing algorithm for IoT video analytics. *Ad Hoc Networks*, 155, 103398. <https://doi.org/10.1016/j.adhoc.2024.103398>.
- [29] Kumar, S., Malik, A. (2025). EFLB-IIoT: Enhanced Flow Control and Load Balancing Approach for SDN-Enabled Industrial Internet-of-Things. *International Journal of Communication Systems*, 38(6), e70043. <https://doi.org/10.1002/dac.70043>.