Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: jscit.nit.ac.ir
Volume 10, Number 3, Autumn 2021, pp. 11-24
Received: 20/11/2020, Revised: 12/04/2021, Accepted: 01/06/2021

Improving Quality of Service in Software Defined–Internet of Things

Hussaindad Saadat[1], Seyedakbar Mostafavi[2*]
1- Computer Engineering Department, Yazd University, Yazd, Iran.
2*- Computer Engineering Department, Yazd University, Yazd, Iran.
[1] hussaindad.saadat@stu.yazd.ac.ir, [2*] a.mostafavi@yazd.ac.ir

Corresponding author's address: Seyedakbar Mostafavi, Computer Engineering Department, Yazd University, Yazd, Iran.

Abstract-The Internet of Things is a growing and emerging technology made up millions of interconnected smart objects which use Internet as the platform for connecting to each other and transferring information in heterogeneous environments. The multiplicity of protocols, the amount of packets sent, the sensitivity of packets to latency and loss require the various levels of quality of services. In order to be able to deliver quality of service effectively in heterogeneous environments, a centralized control module in the network is required. Software-defined networks with the ability to program and centralize the network can help us achieve this goal. By separating the data plane from the control plane, these networks provide us with mechanisms and facilities that increase the level of the QoS in the network. In this paper, we improve the quality of services in Software-defined Internet of Things using a fuzzy multi-objective approach which is applied to the routing process based on fuzzy goal programming algorithm. The results of this study can be used in the development of models and frameworks to improve the quality of services in the software defined-internet of things networks. The proposed method has better conditions than competitive algorithms in terms of end-to-end delay, number of violated flows and activated links.

Keywords- Internet of Things, Software defined network, Quality of service, Fuzzy goal programming algorithm.

مجله علمی پژوهشی رایانش نرم و فناوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: jscit.nit.ac.ir
جلد ۱۰، شماره ۳، پاییز ۱۴۰۰، صفحه ۱۱-۲۴
دریافت: ۱۳۹۹/۰۸/۳۰، بازنگری: ۱۴۰۰/۰۱/۲۳، پذیرش: ۱۴۰۰/۰۳/۱۱

ارائه روشی برای بهبود کیفیت خدمات در شبکه اینترنت اشیا-نرم افزار محور

حسین داد سعادت[۱]، سید اکبر مصطفوی*[۲]
۱- دانشکده مهندسی کامپیوتر-شبکه های کامپیوتری، دانشگاه یزد، یزد، ایران.
*۲- دانشکده مهندسی کامپیوتر-شبکه های کامپیوتری، دانشگاه یزد، یزد، ایران.
[1]Hussaindad.saadat@stu.yazd.ac.ir, [2*]a.mostafavi@yazd.ac.ir

* نشانی نویسنده مسئول: سید اکبر مصطفوی، یزد، صفائیه، بلوار دانشگاه، دانشگاه یزد، دانشکده مهندسی کامپیوتر

چکیده- اشیای هوشمند بر بستر اینترنت اشیا به یکدیگر متصل شده و اطلاعات را در محیط‌های ناهمگن بین یکدیگر مبادله می کنند. داده‌های متنوع در اینترنت ا شیا از محیط‌های ناهمگن به سمت شبکه فر ستاده می شود. تعدد پروتکل‌ها، میزان ار سال ب سته‌ها، حسا سیت بسته‌ها نسبت به تأخیر و اتلاف نیازمند کیفیت خدمات متنوع می‌با شند. برای اینکه بتوان کیفیت خدمات را به صورت مؤثر در محیط‌های ناهمگن ارائه نمود، نیاز به یک دید جامع از شرایط شبکه است. شبکه‌های نرم‌افزار محور با دارا بودن قابلیت برنامه‌ریزی و متمرکز سازی شبکه می‌توانند ما را جهت تحقق این هدف یاری نمایند. این شبکه‌ها با جداسازی بخش داده از بخش کنترل، سازوکارها و امکاناتی را در اختیار ما قرار می‌دهد که موجب افزایش سطح کیفیت خدمات در شبکه می‌گردد. در این مقاله، رو شی جهت بهبود کیفیت خدمات در اینترنت اشیا مبتنی بر شبکه‌های نرم‌افزار محور با بهره‌گیری از رویکرد چندهدفه فازی ارائه شده و فرآیند مسیریابی بر اساس الگوریتم برنامه‌ریزی آرمانی فازی انجام می شود. نتایج این بررسی می‌تواند در توسعه مدل‌ها و چارچوب‌ها جهت بهبود کیفیت خدمات ناهمگن در شبکه اینترنت ا شیای نرم‌افزار محور موردا ستفاده قرار گیرد. روش پی شنهادی ازنظر تأخیر انتها به انتها، تعداد جریان‌های نقص شده کیفیت خدمات و پیوندهای فعال‌شده نسبت به الگوریتم های رقیب عملکرد بهتری از خود نشان می دهد.

واژه‌های کلیدی: اینترنت اشیا، شبکه نرم‌افزار محور، کیفیت خدمات ناهمگن، الگوریتم برنامه‌ریزی آرمانی فازی.

۱- مقدمه

اینترنت اشیا متشکل از محیط‌های ناهمگن و دارای ترافیک غیرقابل‌پیش‌بینی می‌باشد. محیط‌های ناهمگن در اینترنت اشیا به شکل واکنشی عمل می‌کنند. این خاصیت واکنشی بودن باعث می‌شود تا پیش‌بینی ترافیک در شرایط گوناگون مشکل شود. عدم پیش‌بینی ترافیک در محیط‌های ناهمگن و افزایش اتصال وسایل متعدد توام با ترافیک متنوع، باعث بروز ازدحام و صف در شبکه خواهد شد. بروز صف و ازدحام در شرایطی که پاسخگویی به‌صورت درست و دقیق صورت نگیرد، باعث اتلاف و تأخیر بسته‌ها می‌شود. برای اینکه بتوان تا حدی از تأخیر و اتلاف بسته‌ها جلوگیری کرد، می‌توان از توازن بار در شبکه استفاده کرد و در موارد مختلف ترافیک ورودی را روی گره‌های متفاوت تقسیم کرد تا از صف و ازدحام جلوگیری شود [۱].

به دلیل مشکلات مدیریت نامتمرکز و تنوع پروتکل های شبکه، تامین نیازمندی های برنامه های کاربردی در بخش فراهم سازی کیفیت خدمات سخت و طاقت فرسا است. برای اینکه بتوان از قابلیت کنترل متمرکز روی کل شبکه بهره گرفت، از شبکه نرم‌افزار محور[۱] (SDN) استفاده می شود. شبکه نرم افزار محور یک معماری جدید در شبکه است که لایه کنترل را از لایه داده جدا می‌سازد [۲]. در SDN هوشمندی شبکه به‌صورت متمرکز است و زیرساخت شبکه از برنامه‌ها جدا می‌باشد. در لایه کنترل یک کنترلر شبکه نرم‌افزار محور وجود دارد. از آنجا که ممکن است کنترلرهای زیادی در شبکه وجود داشته باشد، حالت‌های متنوعی در شبکه توسط کنترلر به وجود می‌آید که ممکن است توسط یک کنترلر تأمین شود یا اینکه

توسط کنترلرهای توزیعی انجام شوند [۳].

با ظهور اینترنت اشیا اشیای هوشمند قادر هستند تا از طریق بستر جهانی اینترنت به یکدیگر متصل شوند. در یک شبکه اینترنت اشیای فراگیر، تعداد زیادی از اشیای هوشمند و متنوع با یکدیگر ارتباط دارند. به‌عنوان نمونه می‌توان به تگ‌های شناسه فرکانس رادیویی[۲]، حسگرهای بی‌سیم[۳]، محرک کننده‌های بی‌سیم، وسایل ارتباطی ماشین به ماشین و غیره اشاره کرد [۵].

اشیای هوشمند در بستر اینترنت اشیا نیازمند خدمات باکیفیت[۴] شبکه هستند تا بتوانند از محیط‌های مختلف مثل کاربردهای خانگی، دوربین‌های امنیتی، حسگرهای نمایشی[۵] و محرک کننده‌ها[۶]، اطلاعات را جمع‌آوری نمایند. با توجه به اینکه این اشیا در محیط‌های گوناگون مورداستفاده قرار می‌گیرند، نیازمند کیفیت خدمات متنوع می‌باشند [۵].

امروزه از محیط‌های نامتجانس داده‌های متنوع وارد شبکه اینترنت اشیا می‌شوند و انتظار می رود که در آینده این مقدار داده به چندین برابر افزایش یابد. همچنین این محیط‌ها نیازمند کیفیت خدمات متفاوت هستند. با بکارگیری شبکه های نرم افزارمحور در اینترنت اشیا، می توان با اختیار داشتن کنترلر متمرکز شبکه در قسمت‌های مختلف شبکه تنظیمات متنوعی را انجام داد.

درخواست‌های ورودی از محیط‌های مختلف به شبکه اینترنت اشیا متنوع و ناهمگن است. به عنوان مثال، برخی کاربردها حساس به تأخیر بسته‌ها هستند و برخی دیگر حساس به اتلاف بسته‌ها در زمان ارسال بسته‌ها از سمت فرستنده به سمت گیرنده می‌باشند. لذا برای اینکه بتوانیم کیفیت خدمات موردانتظار را برای محیط‌های ناهمگن ارائه دهیم، طوری که کاربران واقع در شرایط مختلف از خدمات ارائه‌شده توسط اینترنت اشیا رضایت داشته باشند، باید سازوکار تنظیم کیفیت خدمات در شبکه را مورد بررسی قرار دهیم [۶]. اینکه چه سازوکاری داشته باشیم که کیفیت خدمات متنوع را برای درخواست‌های مختلف در لایه دسترسی برای مشتریان فراهم کنیم مسئله اصلی این مقاله است.

با توجه به موارد زیر، تحقیق در این حوزه ضروری به نظر می رسد:

- به دلیل عدم توازن بار در شبکه اینترنت اشیا ناهمگن، عملکرد شبکه برای تمامی کاربران مطلوب نیست. درنتیجه رضایت مشتریان و کیفیت خدمات کاهش پیدا می‌کند.
- انتخاب بهترین سرویس‌دهنده با حداقل زمان پاسخگویی توأم با استفاده بهینه از پهنای باند جهت افزایش کارایی شبکه ضروری است.
- وجود تأخیر و جریان‌های دارای نقص کیفیت خدمات و تعداد لینک‌های فعال در کیفیت خدمات تأثیرگذار است و باید در حل مسئله مدنظر قرار گیرد.

روش های موجود، مسئله تفکیک ترافیک و تامین کیفیت سرویس را در بستر اینترنت اشیا نرم افزارمحور به صورت مجزا مورد بررسی قرار داده اند [۵]. هیچ یک از کارهای موجود، بهینه سازی همزمان تاخیر، اتلاف بسته و پهنای باند را مورد بررسی قرار نداده اند. هدف اصلی این مقاله تمرکز روی توزیع و تقسیم عادلانه ترافیک در بخش های مختلف شبکه و تخصیص منابع به صورت موثر می باشد. بنابراین جهت بهبود کیفیت خدمات از الگوریتم بهینه‌سازی چندهدفه برنامه‌ریزی آرمانی فازی در شبکه‌های SDN استفاده شده است. مدلسازی مسئله به صورت فرآیند مسیریابی چندهدفه و ترکیب آن با رویکرد فازی و ارائه الگوریتم برنامه‌ریزی آرمانی فازی مبتنی بر متغیرهای تحقیق از جمله جنبه های نوآوری این پژوهش می‌باشد. در این مقاله روی تأخیر انتها به انتها، تعداد جریان‌های دارای نقص QoS و پیوندهای فعال‌شده بررسی کامل صورت گرفته است. نتایج ارزیابی ها نشان می دهد که روش پیشنهادی در قیاس با الگوریتم های موجود مانند SWAY، عملکرد بهتری را از خود نشان داده است.

ادامه این مقاله به صورت زیر سازماندهی شده است. بخش ۲ پیشینه تحقیق را در خود جای داده است. در بخش ۳ مدل سیستم مورد نظر پیشنهاد شده و مورد بررسی قرار گرفته است. در بخش ۴ رویکرد و نتایج ارزیابی روش پیشنهادی از طریق شبیه سازی ارائه شده است و نتایج بدست آمده تحلیل شده و با روش قبلی مقایسه گردیده است. در بخش ۵ مباحث مطرح شده در مقاله جمع بندی شده است.

۲- پیشینه تحقیق

در بخش پیشینه تحقیق، سابقه فعالیت های تحقیقاتی از قبیل شبکه نرم‌افزار محور، چالش‌های موجود در شبکه نرم‌افزار محور، کیفیت خدمات در شبکه اینترنت اشیا-نرم افزار محور و توازن بار مورد بررسی قرار گرفته است.

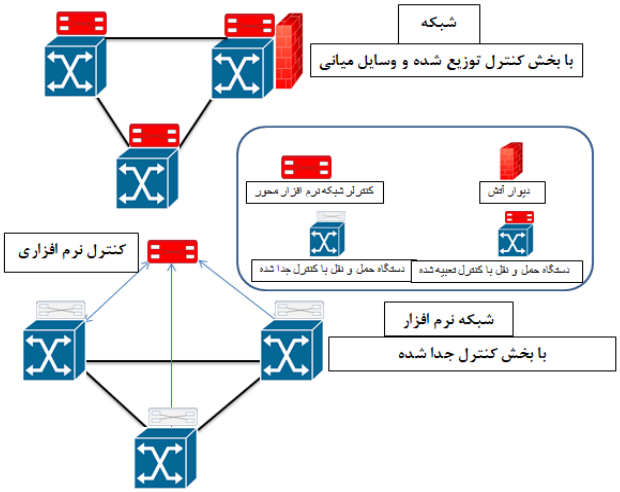
۲-۱- شبکه نرم‌افزار

با توجه به تعریف، شبکه‌های SDN مدیریت شبکه را آسان‌تر کرده است و فرصت های مناسبی را برای مراکزی با داده‌های بزرگ و عموماً مبتنی بر محاسبات ابری و زیرساخت‌های اینترنتی و همچنین شبکه‌های سیار مهیا نموده است. در یک مرکز داده‌ای بزرگ ممکن است ده‌ها هزار کنترل‌کننده و هزاران سوئیچ وجود

داشته باشد. لذا بدیهی است که شبکه زیرساختی باید زیربنای همه سرویس‌های لایه‌های بالاتر بوده و دارای قابلیت دسترسی و مقیاس‌پذیری بالایی باشد. شبکه‌های SDN نیازمند برنامه‌نویسی سطح بالا در کنترلر متمرکز جهت راه‌اندازی و انتزاع خدمات مختلف ارسال داده است، اما در مقابل انعطاف‌پذیری و قابلیت کنترل بالایی نیز دارند [۷].

با توجه به‌اندازه، ناهمگن بودن و پیچیدگی‌های شبکه‌های فعلی، روش‌های متداول و مرسوم برای پیکربندی، بهینه‌سازی و عیب‌یابی ممکن است غیر مؤثر باشد و در برخی موارد، حتی ممکن است باعث ایجاد خطا شود. بهینه‌سازی تک‌تک دستگاه‌های شبکه، بدون در نظر گرفتن کل شبکه و دستگاه‌های دیگر مربوط به آن، ممکن است منجر به عملکرد ناسازگار شبکه با خروجی‌های نامطلوب شود. فناوری شبکه‌های نرم‌افزار محور، برای طراحی بهتر شبکه‌های نسل آینده ارائه‌شده است که ایده اصلی آن، جداسازی سطح کنترل از سطح داده و ایجاد امکان مدیریت مؤثر و انعطاف‌پذیری بیشتر و همچنین کنترل عملکرد شبکه با برنامه‌های نرم‌افزاری است [۸-۱۰].

معماری شبکه نرم‌افزار محور دارای ویژگی‌های پویا، مدیریت پذیر، مقرون‌به‌صرفه و انطباق پذیر می‌باشد. شبکه‌های نرم‌افزار محور دارای یک معماری چندلایه هستند. توسعه در این شبکه‌ها با انتشار پروتکل OpenFlow آغاز شد که جداسازی کنترل از سوئیچینگ سخت‌افزاری را امکان پذیر کرد [۱۱]. معماری SDN در نمای ساده در شکل ۱ نشان داده‌شده است [۱۲]. جداسازی سطوح داده و کنترل با تعریف یک واسط برنامه‌نویسی کاربردی معنا پیدا می‌کند که کنترلر از طریق این واسط تجهیزات شبکه را برنامه‌ریزی می‌کند.



شکل ۱: معماری شبکه‌های نرم‌افزار محور توزیع‌شده

۲-۲- کیفیت خدمات در اینترنت اشیا نرم افزار محور

نیلوی سها و سمارش بیرا طرح مسیریابی کیفیت خدمات آگاه از ترافیک را برای شبکه اینترنت اشیا نرم‌افزار محور پیشنهاد کردند [۵]. در این مقاله دو نوع استراتژی مسیریابی کیفیت خدمات حساس به تأخیر و حساس به اتلاف بسته‌ها برای بسته‌هایی که از سمت وسایل انتهایی به شبکه وارد می‌شود، در نظر گرفته‌شده است تا عملکرد شبکه را بهبود بخشد. برای انتخاب مسیر بهینه از الگوریتم کوتاه‌ترین مسیر Yen's k استفاده نمودند تا مسیر بهینه را انتخاب نمایند. نتایج بدست آمده از شبیه‌سازی در این مقاله نشان داده است که میزان تأخیر انتها به انتها[۷] نسبت به معیارهای موردنظر در مطالعه به‌صورت قابل‌توجهی کاهش‌یافته و میزان اتلاف بسته‌ها نیز تقلیل پیداکرده است [۵].

جیافو وان و همکارانش روی فناوری مهندسی ترافیک بر اساس شبکه نرم‌افزار محور تمرکز کردند. آنها یک چارچوب مرجع برای مهندسی ترافیک در شبکه‌های نرم‌افزار محور ارائه داده‌اند که شامل دو بخش اندازه‌گیری ترافیک و مدیریت ترافیک می‌باشد. در مدیریت شبکه، هدف اصلی تهیه استراتژی زمان‌بندی ترافیک طبق پارامترهای شبکه هست که با استفاده از فنّاوری اندازه‌گیری جریان در شبکه حاصل می‌شود. نتیجه این اندازه‌گیری برای قناعت دادن به نیازهای خاص در شبکه می‌باشد. در این مقاله چهار دیدگاه توازن بار ترافیک، تضمین کیفیت خدمات، زمان‌بندی ذخیره انرژی و مدیریت ترافیک در شبکه‌های دوگانه IP/SDN موردبررسی قرارگرفته است [۱۳].

مکی و همکارانش مسئله انتخاب فنّاوری دسترسی رادیویی[۸] در شبکه‌های ناهمگن[۹] را موردتوجه قرار داده اند. رویکردهای فعلی به معیارهای مربوط به سیگنال نظیر نسبت سیگنال به تداخل به‌اضافه نویز[۱۰] جهت انتخاب بهترین شبکه برای کاربر بی‌سیم متکی هستند. بااین‌وجود، چنین رویکردهایی به کیفیت خدمات نیازهای کاربران بی‌سیم توجه ندارند؛ بنابراین اغلب آن‌ها را به مناسب‌ترین شبکه متصل نمی‌کند. در این مقاله چارچوب انتخاب فناوری دسترسی رادیویی در شبکه‌های نامتجانس بر اساس شبکه‌های نرم‌افزار محور پیشنهادشده است. چارچوب پیشنهادی یک استراتژی انتخاب فناوری دسترسی رادیویی که نشان‌دهنده الزامات کیفیت خدمات از جریان‌های ارتباطی پایین‌رو[۱۱] است را با استفاده از یک مشخصه به نام عامل مناسب[۱۲] نشان می‌دهد. این چارچوب به انعطاف‌پذیری و متمرکز بودن شبکه نرم‌افزار محور برای اجرای نظارت و مکانیسم ارزیابی ظرفیت فناوری دسترسی رادیویی کمک می‌کند تا در تحقق استراتژی انتخاب متکی باشد. نتیجه شبیه‌سازی نشان‌دهنده دستاوردهای مهم به‌دست‌آمده از چارچوب انتخاب فناوری دسترسی

رادیویی با توجه به میزان داده‌های اختصاص داده‌شده به کاربران بی‌سیم، رضایت آن‌ها و کیفیت تجربه[۱۳] آن‌ها در مقایسه با روش‌های دیگر انتخاب فناوری دسترسی رادیویی است [۱۴].

در مرجع [۱۵] یک معماری برای استقرار خدمات مبتنی بر کیفیت سرویس در شبکه های اینترنت اشیا نرم افزارمحور ارائه شده است. در این معماری، کنترلر SDN شبکه های مختلف IoT را برنامه دهی می کند تا سرویس مدنظر را بر اساس کیفیت سرویس درخواستی را دریافت نمایند. نویسندگان در مرجع [۱۶] یک معماری ترکیبی از SDN و Fog را برای اینترنت اشیا پیشنهاد می کنند که در آن الگوریتمی مبتنی بر قطعه بندی مجازی برای بهبود کیفیت سرویس پیشنهاد می شود. گو و همکاران در مرجع [۱۷] یک پروتکل مسیریابی آگاه از کیفیت سرویس مبتنی بر یادگیری عمیق ارائه کرده اند که مشخصه های ترافیک را از سابقه ترافیک شبکه استخراج می کند و بر اساس آن سیاست های شبکه را بهینه می کند. در مرجع [۱۸] از رویکرد یادگیری تقویتی چندعاملی برای یادگیری توزیع بار و تخصیص منابع شبکه استفاده می کنند.

۲-۳- سیاست های مدیریت ترافیک از طریق مهاجرت سوئیچ

در سال‌های اخیر تحقیقات زیادی در مورد مهاجرت سوییچ در شبکه‌های SDN انجام‌شده است. چند سیاست مختلف برای توازن بار با مهاجرت سوییچ وجود دارد، به‌طور مختصر مهم‌ترین آن‌ها را مرور می‌کنیم.

۲-۳-۱- سیاست نوبت گردشی

در سال ۲۰۱۶ روشی توسط استبان و همکارانش بیان گردیده است که در این روش سه متعادل‌کننده بار را با استفاده از شبکه SDN پیاده سازی می‌کند. ابتدا اصول سه الگوریتم صف FIFO[14]، نوبت گردشی (RR)[۱۵] و نوبت گردشی کسری (DRR)[۱۶] برر سی شده است. سپس جنبه‌های اصلی طراحی و پیاده‌سازی ماژول‌های تعادل بار بر اساس هریک از الگوریتم‌های ذکر شده بیان شده ا ست. این روش کاربردی قادر است درخواست یک کلاینت را بر اساس معیار مرتبط با الگوریتم صف داده‌شده به یک سرور انتساب دهد. الگوریتم DRR مبتنی بر الگوریتم RR است اما پارامترهای بیشتری دارد و شکلی از صف را ارائه می‌دهد که بر اساس تعداد بایت‌هایی که هر سرور می‌تولند در یک دوره بپذیرد. پارامترهایی که در این الگوریتم ملاحظه می شود عبارت‌اند از: کوانتوم، حداکثر اندازه یک بسته که می‌توا ند در دوره فعلی توسط یک سرور دریافت شود، شمارنده کسری، که اگر اندازه بسته بزرگ‌تر از کوانتوم باشد افزایش می‌یلبد یعنی هنگامی‌که بسته نمی‌تواند در دوره فعلی پردازش شود [۱۹].

هر سرور صف خود را دارد و فقط بسته‌هایی را که در آن هستند پردازش می‌کند. به‌علاوه فرض شده است که مقدار کوانتوم برای هر سرور ثابت است. برای شروع توزیع بسته‌ها، مقادیر کوانتوم باید به یک مقدار ایستا و مقدار شمارنده به صفر تنظیم شود. پردازش DRR به این صورت است که شمارنده با مقدار خودش به‌علاوه مقدار کوانتوم به روز می‌شود. اندازه بسته با مقدار شمارنده مقایسه می‌شود، اگر از آن کوچک‌تر بود بسته به سرور ارسال‌شده و شمارنده مقداردهی اولیه می‌شود. اگر از آن بزرگ‌تر بود، بسته در صف سرور باقی می‌ماند و منتظر ارسال شدن در دور بعدی می‌شود و مقدار شمارنده برابر با مقدار کوانتوم خواهد شد. الگوریتم DRR تنها می‌تواند به‌طور کامل در سطح بسته پیاده‌سازی شود که مطلوب نیست، بنابراین تنها جنبه‌های خاصی از DRR مانند کنترل تعداد بایت‌ها و ترتیب پردازش کلاینت برای انتخاب، استفاده می‌شود. در سازوکار مبتنی بر DRR سه بخش را می‌توان متمایز کرد. ترتیب رسیدن کاربران؛ تعامل بین کنترل‌کننده و سوئیچ که از یک بازخورد ثابت میان آن‌ها ساخته‌شده است؛ و پرداختن به هر کلاینت در زمان درخواست به سروری با میزان بار کمتر. در این مورد اگر تعداد زیادی درخواست از کاربران برسد، یک تأخیر رخ خواهد داد، چون یک پردازش اضافه برای هر کلاینت به‌منظور بررسی تعداد بایت‌های پردازش‌شده توسط هر سرور انجام خواهد شد. اما این تأخیر تنها در پاسخ اولین درخواست هر کلاینت آشکار خواهد شد، چون هر بار که یک قانون تولید و راه‌اندازی می‌شود، هیچ تأخیر بیشتری در درخواست‌های بعدی کلاینت وجود نخواهد داشت.

الگوریتم FIFO سریع‌ترین الگوریتم است زیرا یک سرور را به‌صورت تصادفی انتخاب می‌کند. الگوریتم DRR نیز کندترین است زیرا ضروری است که در آن از بایت‌های پردازش‌شده توسط سرور مطلع بود. کارایی الگوریتم RR هم مابین دو الگوریتم دیگر است چون برای انتساب باید یک لیست ایجاد شود. با استفاده از این روش اولین درخواست برای یک سرور همیشه دیرتر سرویس داده می‌شود، اما هر بار که یک قانون برای بسته‌هایی که بخشی از یک اتصال هستند؛ در سوئیچ اعمال می‌شود، به دلیل روش کار SDN، زمان پاسخ آن سرور کوتاه‌تر خواهد شد. شبکه نرم‌افزار محور انعطاف‌پذیری را در این سناریو تأمین می‌کند به‌علاوه با اضافه کردن یک الگوریتم جدید صف، تغییراتی در برنامه‌نویسی ایجاد می‌شود بدون اینکه نیازی به خرید یک محصول جدید و مسائل بالقوه آن مانند نصب، نگهداری و عملیات تجهیزات جدید باشد.

۲-۳-۲- سیاست مبتنی بر زمان پاسخ

ژونگ و همکاران در سال ۲۰۱۸ روشی به نام LBBSRT[۱۷] را ارائه دادند که طی آن کنترل‌کننده زمان پاسخ سرورها را به‌صورت

بلادرنگ اندازه‌گیری کرده و بر اساس آن سروری را با حداقل و یا پایدارترین زمان پاسخ برای پردازش درخواست کاربر انتخاب می‌کند. این روش در مقایسه با الگوریتم‌های معمول تعادل بار مانند نوبت گردشی و یا تصادفی، تعادل بار بهتری را فراهم می‌کند. روش‌های سنتی توازن بار به شرایط سخت‌افزاری محدود می‌شوند و از زمان‌های پاسخ سرور به‌درستی استفاده نمی‌کنند. برای حل این مسئله یک روش توازن بار بر اساس زمان‌های پاسخ سرور و با بهره‌گیری از انعطاف‌پذیری SDN در این مقاله پیشنهادشده است. با استفاده از زمان پاسخ هر سرور که به‌صورت بلادرنگ توسط کنترل‌کننده اندازه‌گیری شده است، درخواست کاربران به‌صورت یک‌بار متعادل‌شده مساوی میان سرورها، پردازش می‌کند. آزمایش‌ها شبیه‌سازی نشان داده است که این روش اثر توازن بار بهتری دارد و با حداقل متوسط زمان‌های پاسخ درخواست‌ها را پردازش می‌کند. به‌علاوه این روش به‌آسانی پیاده‌سازی شده و کم‌هزینه و مقیاس‌پذیر است. در روش‌های سنتی توازن بار با استفاده از پیام ICMP[18] زمان پاسخ سرور به دست می‌آمد. بنابراین زمان به‌دست‌آمده از دقت کافی برخوردار نبود. به همین منظور، در این روش دو ماژول به کنترل‌کننده SDN اضافه‌شده است. ماژولی برای اندازه‌گیری زمان پاسخ سرور و ماژول دیگر برای فرآیند انتخاب سرور و از تنظیم جدول جریان استفاده می‌شود، ایرادی که این مقاله دارد این است که هزینه مهاجرت را در نظر نگرفته است و دیگری اینکه فقط با توجه به معیار زمان پاسخ تصمیم به مهاجرت می‌گیرد [۲۰]. عبدالرحیم فیلالی و همکارانش با استفاده از مدل چندمرحله‌ای پیش‌بینی کننده ARIA[19]، بار کنترل‌کننده‌ها را پیش‌بینی می‌کند و اگر تشخیص دهد بار کنترل‌کننده‌ای بیشتر از توان محاسباتی آن باشد با به دست آوردن زمان پاسخ کنترل‌کننده، نتیجه را ارزیابی می‌کند. در این مقاله نیز به هزینه مهاجرت توجهی نشده است [۱].

۲-۳-۳- سیاست مبتنی بر تعداد اتصالات

سلو و همکارانش دو الگوریتم BalCon[20] و BalConPlus را جهت تشخیص و کاهش عدم توازن بار با مهاجرت سوییچ‌ها ارائه داده‌اند [۲۱-۲۲].

BalCon شبکه را مانیتور می‌کند و بار ورودی سوییچ‌ها و بار بین سوییچ‌ها را محاسبه می‌کند. بار کنترل‌کننده از سه قسمت اصلی محاسبه می‌شود:

- هزینه بار مسیر بیرون از SDN
- هزینه بار مسیر از دانه‌های دیگر SDN
- هزینه بار نصب قانون جدید در هر سوییچ که توسط کنترل‌کننده، کنترل می‌شود.

الگوریتم BalCon در سناریوهایی که درخواست مهاجرت به‌صورت سریالی باشد مناسب نیست. به همین دلیل الگوریتم BalConPlus ارائه شد که دو تفاوت با الگوریتم BalCon دارد:

- زمانی که سوییچ برای مهاجرت انتخاب می شود،BalConPlus همیشه سوییچ‌های جایگزین انتخاب می‌کند.
- زمانی که مسیر مهاجرت انتخاب می‌شود، BalConPlus همیشه مسیرهای جایگزین دیگری انتخاب می‌کند.

۲-۳-۴- سیاست مبتنی بر ظرفیت پردازش

در تحقیقی کائور در سال ۲۰۱۷ برای انتخاب سرور از میان خوشه سرورها، روشی ارائه کرد که در آن کنترل‌کننده سروری با حداقل تعداد بسته را انتخاب می‌کند. در این استراتژی بعد از هر دو ثانیه، کنترل‌کننده پیام flow_statistics_request را به سوییچ ارسال می‌کند. زمانی که سوییچ پیام پاسخ را به کنترل‌کننده ارسال می‌کند، داخل کنترل‌کننده رویداد flow_statistics_received ایجاد می‌شود. تابع مدیریت این رویداد فراخوانی شده و تعداد کل بسته‌های پاسخ داده‌شده را برای هر سرور استخراج می‌کند. بعد از شمارش، سرور با حداقل تعداد بسته توسط متعادل‌کننده بار به‌عنوان سرور مقصد انتخاب می‌شود. دو روش برای ارتباط بین کلاینت و سرور وجود دارد: در اولین روش زمانی که درخواست کلاینت به متعادل‌کننده بار می‌رسد، متعادل‌کننده بار درخواست کلاینت را به ماشین سروری که بر اساس سیاست تعادل بار انتخاب‌شده، ارسال می‌کند. زمانی که پاسخ بسته به متعادل‌کننده بار برمی‌گردد، متعادل‌کننده بار بسته پاسخ را به ماشین کلاینت ارسال می‌کند. درواقع ارتباط بین کلاینت و سرور هر بار طی چهار گام متوالی شکل می‌گیرد. درگیر شدن متعادل‌کننده بار در هردو مسیر رفت‌وبرگشت منجر به صرف زمان اضافی خواهد شد.

در روش دوم متعادل‌کننده بار تنها در فرایند ارسال بسته درخواست شرکت داده می‌شود. ماشین سرور مستقیماً بسته پاسخ را به سمت ماشین کلاینت ارسال می‌کند و تأخیر اضافی به این صورت حذف می‌شود و ارتباط بین کلاینت و سرور هر بار در طول سه‌گام متوالی شکل می‌گیرد. در این مقاله از روش دوم برای ارتباط استفاده‌شده است. لذا تأخیر ناشی از مداخله کنترل‌کننده در مسیر برگشت بسته از روال ارتباطی سرور و کلاینت حذف می‌شود [۲۳].

مین و همکارانش یک الگوریتم بر اساس Q-learning را ارائه داده‌اند که به‌صورت خودآموز شبکه را بررسی کرده و مهاجرت را انجام می‌دهد اما به نتایج مناسبی نرسیده است [۲۴]. آل تامو همکارانش روش جدیدی پیشنهاد کرده است که جای مهاجرت کامل سوییچ از یک کنترل‌کننده به کنترل‌کننده دیگر فقط بخشی از بار سوییچ

را به کنترل‌کننده‌ای که بار کمتری دارد مهاجرت دهیم، اما متأسفانه هنوز واسطه‌ای در Open Flow برای پیاده‌سازی وجود ندارد [۲۵]. وو و همکارانش از تئوری بازی برای بهبود توازن بار با استفاده از مهاجرت سوییچ استفاده کرده‌اند. ابتدا با مانیتور کردن شبکه، عدم توازن بار را تشخیص می‌دهند و سپس از عدم همکاری بازی[۲۱] بهترین کنترل‌کننده را برای مقصد مهاجرت انتخاب می‌کند. با استفاده از این روش توازن بار و استفاده مفید از منابع افزایش پیداکرده اما هزینه‌های مهاجرت در نظر گرفته نشده است [۲۶]. ژاآوه و همکارانش یک شما برای مهاجرت سوییچ‌ها به نام DMSM[۲۲] را ارائه کرده‌اند. در این مقاله ابتدا مطمئن می‌شوند ارتباط بین سوییچ و کنترل‌کننده در زمان مهاجرت حفظ می‌شود سپس با ایجاد مصالحه‌ای بین نرخ تعادل بار و تأخیر انتقال[۲۳] کنترل‌کننده، باعث کاهش تأخیر محاسبات می‌شود. تعادل بار اندازه‌گیری نشده است و هزینه‌های مهاجرت نیز در نظر گرفته نشده است [۲۷]. ساهو و همکارانش چارچوب ESB[۲۴] را برای مهاجرت سوییچ‌ها در کنترل‌کننده‌های IOT[۲۵] ارائه کرده‌اند. برای پیدا کردن مؤلفه‌های مهاجرت از سه معیار CPU، حافظه و پهنای باند[۲۶] کنترل‌کننده استفاده کرده‌اند. هزینه مهاجرت را نیز حساب کرده‌اند. در آخر پیاده‌سازی را در شبکه‌ی کوچکی انجام دادند و می‌تواند نتایج متفاوتی با دنیای واقعی داشته باشد، زیرا ترافیک در IoT بسیار بالا است [۲۸].

اکبر نقابی و همکارانش الگوریتم‌های فرا اکتشافی برای بهبود توازن بار در SDN را مرور کرده‌اند و بیان نموده‌اند که: ازآنجاکه منابع شبکه محدود هستند، بهینه‌سازی استفاده از منابع در شبکه‌ها مسئله مهمی است. توازن بار توزیع متعادل بارها را در چندین منبع بهبود می‌بخشد تا به حداکثر میزان اطمینان و کارایی منابع شبکه برسد. کنترل‌کننده‌های SDN می‌توانند در مقایسه با شبکه‌های سنتی تعادل بار بهینه‌ای ایجاد کنند زیرا دارای دید جهانی شبکه هستند. مسئله تعادل بار را می‌توان با استفاده از فن‌های متفاوتی اکتشافی الهام گرفته از طبیعت مختلف

حل کرد زیرا دارای ماهیت کامل NP است. ازاین‌رو، برای حل مسئله متعادل‌سازی بار در SDN روش‌های فرا-ابتکاری الهام گرفته از طبیعت روش‌های مهمی هستند. در ادامه نویسندگان به‌طور سامانمند فن‌های ابتکاری طبیعت الهام گرفته از طبیعت را بررسی کردند. همچنین، مزایا و معایبی را در نظر گرفته است که با توجه به فن‌های فرا ابتکاری الهام گرفته از طبیعت، معیارهای الگوریتم‌های آن‌ها را در نظر گرفته است [۲۹]. خلاصه کارهای انجام شده در زمینه کیفیت خدمات شبکه های SDN و IoT در جدول ۱ ارائه شده است.

جدول ۱: خلاصه پیشینه تحقیق

سال انتشار	روش	مزایا	معایب
سلو و همکارانش ۲۰۱۷	ارائه الگوریتم BalCon برای مهاجرت پویا سوییچ‌ها	در شبکه‌هایی که نیاز به تعداد مهاجرت‌های خیلی زیادی نیست کارایی بسیار خوبی دارد	در شبکه‌هایی که نیاز به مهاجرت‌های سریالی دارد دچار مشکل می‌شود
کائور ۲۰۱۷	استفاده از زمان پاسخ سرور به‌جای بار سرور برای تشخیص انجام مهاجرت	استفاده از زمان پاسخ	در نظر نگرفتن هزینه مهاجرت
آل تامو و همکارانش ۲۰۱۹	به‌جای مهاجرت کامل یک سوییچ به کنترل‌کننده دیگر، مقداری از بار سوییچ را انتقال می‌دهد	۱-کم شدن بار ناشی از انجام مهاجرت برای نصب مسیرهای جدید ۲-کم شدن تعداد مهاجرت‌ها	۱-شبکه را پیچیده و کنترل سوییچ را سخت می‌کند ۲-واسط مناسبی برای پیاده‌سازی ندارد
مین و همکارانش ۲۰۱۹	با پیش‌بینی بار کنترل‌کننده‌ها، اضافه‌بار را تشخیص می‌دهد	قبل از اضافه‌بار، اضافه‌بار را تشخیص داده و برای اطمینان زمان پاسخ کنترل‌کننده را چک می‌کند	آموزش شبکه در ابتدای کار زمان‌بر است
ساهو و همکارانش ۲۰۱۹	از سه مولفه برای مهاجرت CPU، حافظه و پهنای باند کنترل‌کننده استفاده نموده	با در نظر گرفتن چند معیار و ایجاد توازن بین آن‌ها مؤلفه‌های مهاجرت را تشخیص می‌دهد	برای شبکه کوچکی آزمایش شده است
ژاآوه و همکارانش ۲۰۱۹	با ایجاد مصالحه‌ای بین نرخ تعادل بار و تأخیر انتقال کنترل‌کننده اقدام به مهاجرت می‌کند	۱- ارتباط بین سوییچ و کنترلر در زمان مهاجرت حفظ می‌شود ۲-از دو مؤلفه مهم برای اقدام به مهاجرت استفاده می‌کند	۱-پس از مهاجرت تعادل بار اندازه‌گیری نشده است ۲-هزینه‌ای مهاجرت در نظر گرفته نشده است

بررسی پیشینه تحقیق نشان داد که مقالات متعددی به بررسی پارامترهای بهبود کیفیت خدمات در شبکه اینترنت اشیا پرداخته است. به طور خلاصه، بررسی مقالاتی که از طریق توازن بار برای بهبود کیفیت خدمات تلاش نموده اند نشان می دهد که روش های مبتنی بر توازن بار:

- از مهاجرت سوییچ ا ستفاده کردند، اما برای شبکه‌های بزرگ قابل‌استفاده نیستند.
- زمان پا سخ را در نظر گرفته اند اما هزینه مهاجرت را در نظر نگرفته اند.

- مهاجرت کامل انجام نداده لند و در نتیجه پیچیدگی افزایش یافته است.

بررسی مقالاتی که تأخیر و اتلاف را در نظر گرفتند نشان می دهد که این مقالات:

- الگوریتم م سیریابی پی شنهاد کرده اند، اما توازن بار را در نظر نگرفته اند و در نتیجه کوتاه‌ترین م سیر باعث ازدحام شده است.
- تفکیک و اولویت‌بندی ترافیک را انجام داده اند.
- الگوریتم زمان‌بندی ترافیک را پیشنهاد کرده اند.

در موارد مختلفی که بررسی‌شده است، هر روش دارای مزایا و معایب منحصربه‌فرد خودش می‌باشد، اما به‌صورت کلی استفاده از توازن بار در اینترنت اشیا با استفاده از شبکه نرم‌افزار محور می‌تواند به توزیع عادلانه ترافیک کمک کند و از این طریق میزان عملکرد کلی شبکه را افزایش دهد. به دلیل ناهمگن بودن محیط‌های اینترنت اشیا، توازن بار یک امر ضروری است. با توجه به اینکه شبکه نرم‌افزار محور دارای دید جامع نسبت به شبکه می‌باشد، ازاین‌رو این ویژگی می‌تواند در بخش توازن بار کمک کند تا عملکرد کلی شبکه افزایش یابد.

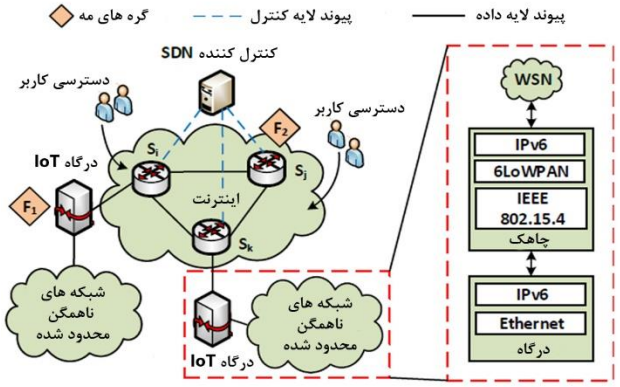
۳- مدل سیستم

در این پژوهش یک مدل اتصال گرا برای اینترنت اشیاء در نظر گرفته می‌شود که در مرجع [۲۶] مورد بحث قرارگرفته است. در این مدل، شبکه‌های محدودشده ناهمگن [۲۷] از طریق IP با استفاده از دروازه‌های IoT به اینترنت متصل می‌شوند [۲۸]. همچنین جهت ارتباطات با تأخیر کم و فناوری بی‌سیم ناهمگن از فناوری ترکیبی فیبر نوری استفاده می‌شود [۲۹-۳۰].

دروازه IoT و سوئیچ‌ها با استفاده از SDN به‌عنوان گره‌های مه [۳۱] کار می‌کنند تا پردازش را به لبه نزدیک‌تر کرده و بار را در شبکه اصلی کاهش دهند. شکل ۲ معماری SDIoT در نظر گرفته‌شده در این مقاله را نمایش می‌دهد.

کنترل‌کننده SDN از طریق پروتکل OpenFlow با سوئیچ‌ها ارتباط برقرار می‌کند و ارتباط بین کنترلر و لایه برنامه از طریق رابط برنامه‌نویسی برنامه کاربردی شمالی SDN (Northbound API) حاصل می‌شود.

برای شبکه داده‌شده، معیارهای QoS مربوط به هر لینک $(i,j)\epsilon\mathcal{L}$ و تأخیر $d(i,j)$ و احتمال از دست دادن بسته $l(i,j)$ و پهنای باند در دسترس $c(i,j)$ می باشد.



شکل ۲: معماری SDIoT [۵]

تأخیر یک معیار افزودنی است، درحالی‌که پهنای باند مقعر است. ترکیب قوانین برای احتمال از دست دادن بسته پیچیده‌تر است و ازاین‌رو، لگاریتم احتمال موفقیت (که خود چند برابر است) در نظر گرفته می‌شود.

درنتیجه طبق رابطه (۱-۳) مقدار $\hat{l}(i,j)$ محاسبه می شود.

$$\hat{l}(i,j) = \log\left(1 - l(i,j)\right) \text{ [5]} \qquad (۱-۳)$$

ازاین‌رو برای هر مسیر $\mathcal{P} = \{i,j,k,\dots,s,t\}$ می باشد.

همچنین رابطه‌های (۲-۳)، (۳-۳)، (۴-۳) و (۵-۳) مقادیر تأخیر، پهنای باند و احتمال از دست دادن بسته را محاسبه می‌کند.

$$D_{\mathcal{P}} = d(i,j) + d(j,k) + \cdots + d(s,t) \text{ [5]} \qquad (۲-۳)$$

$$C_{\mathcal{P}} = \min\{c(i,j), d(j,k), \dots, d(s,t)\} \qquad \text{[5]} \qquad (۳-۳)$$

$$\hat{L}_{\mathcal{P}} = \hat{l}(i,j) + \hat{l}(j,k) + \cdots + \hat{l}(s,t) \text{ [5]} \qquad (۴-۳)$$

$$L_{\mathcal{P}} = 1 - \exp\left(\hat{L}_{\mathcal{P}}\right) \text{ [5]} \qquad (۵-۳)$$

همچنین متغیر $x_k(i,j)$ یک تابع دوحالته مانند رابطه (۶-۳) می‌باشد.

$$x_k(i,j) = \begin{cases} 1 \\ 0 \end{cases} \qquad (۶-۳)$$

زمانی مقدار رابطه (۶-۳) برابر یک می‌شود که فرآیند مسیریابی بین گره‌های i و j انجام‌شده باشد در غیر این صورت مقدار آن صفر خواهد بود. این فرآیند برای کلیه گره‌های همسایه گره جاری مانند شکل ۳-۲ آزمایش و بر اساس آن فرآیند مسیریابی پیاده‌سازی می‌گردد.

همچنین جهت محاسبه مقدار R(i) از رابطه (۳-۷) استفاده می‌گردد.

$$R(i) = \sum_{f_k \in \mathcal{F}} R_k(i) \text{ [5]} \qquad (۳-۷)$$

همچنین $R_k(i)$ نیز بر اساس رابطه (۳-۸) محاسبه می‌شود.

$$R_k(i) = \sum_{j \in \mathcal{N}(i)} x_k(i,j) \text{ [5]} \qquad (۳-۸)$$

همچنین جهت محاسبه تأخیر و بسته‌های ازدست‌رفته از رابطه‌های (۳-۹) و (۳-۱۰) استفاده می‌شود.

$$D(f_k) = \sum_{(i,j)\in\mathcal{L}} d(i,j) x_k(i,j) \text{ [5]} \qquad (۳-۹)$$

$$\hat{L}(f_k) = \sum_{(i,j)\in\mathcal{L}} \hat{I}(i,j) x_k(i,j) \text{ [5]} \qquad (۳-۱۰)$$

از قانون ترکیب پهنای باند، ظرفیت یک مسیر که یک جریان از آن عبور می‌کند، ازنظر ریاضی بر اساس رابطه (۳-۱۱) تعریف می‌شود.

$$C(f_k) = \min_{(i,j)\in\mathcal{L}} (c(i,j) x_k(i,j)) \text{ [5]} \qquad (۳-۱۱)$$

درنهایت جهت محاسبه پهنای باند (ظرفیت) از رابطه (۳-۱۲) استفاده می‌شود.

$$C_{res}(i,j) = c(i,j) - \sum_{f_k \in \mathcal{F}} q_k^{bandwith} x_k(i,j) \text{ [5]} \qquad (۳-۱۲)$$

۳-۱- روش پیشنهادی

اهداف تأخیر، پهنای باند (ظرفیت) و احتمال از دست دادن بسته در مرجع [۵] در نظر گرفته شده است. برای بهبود عملکرد و ایجاد تغییرات در راهکار قبلی، در این روش جهت بهینه‌سازی مدل از یک رویکرد چندهدفه فازی تحت عنوان برنامه‌ریزی آرمانی فازی استفاده می‌گردد. ابتدا به تشریح الگوریتم برنامه‌ریزی آرمانی پرداخته می شود و سپس فرآیند بهینه‌سازی مدل‌سازی می‌شود.

بهینه‌سازی مدل بدین‌صورت است که در بین پارامترهای یک تابع به دنبال مقادیری باشیم که تابع را کمینه یا بیشینه نماید. ازجمله الگوریتم‌های بهینه‌سازی الگوریتم‌های فرا مکاشفه‌ای می‌باشد. با توجه به پیچیدگی و دشواری حل مسئله بهینه سازی، استفاده از الگوریتم‌های فرا مکاشفه‌ای می تواند منجر به نتایج بهتری نسبت به الگوریتم‌های دقیق شود [۳۲].

برنامه‌ریزی آرمانی یکی از تکنیک های کارآمد حل مسائل تصمیم‌گیری چندهدفه است. این تکنیک در مورد حل انواع مختلفی از مسائل برنامه‌ریزی که دارای اهداف متعارض و چندگانه‌اند کاربرد دارد. اولویت اهداف توسط تصمیم‌گیرنده تعیین می‌شود. برخلاف برنامه‌ریزی خطی که مستقیماً به بهینه‌سازی تابع هدف می‌پردازد، برنامه‌ریزی آرمانی به مینیمم کردن انحراف بین اهداف و راه‌حل بهینه می‌پردازد.

تابع هدف مسئله اصلی به‌صورت محدودیت جدید همراه با متغیر کمکی و مقدار بهینه موردنظر دوباره فرمول‌بندی می‌شود. به متغیرهای کمکی متغیرهای انحراف از آرمان گفته می‌شود. روش‌های برنامه‌ریزی آرمانی دارای ساختار مشابه و مشترک هستند. هدف الگوریتم برنامه‌ریزی آرمانی حداقل نمودن انحراف نامساعد هر هدف نسبت به آرمان‌هایش است. به این شکل که در تابع هدف مسئله مجموع انحرافات نامطلوب از هر آرمان که توسط تصمیم‌گیرنده تعیین می‌شود را باید به حداقل مقدار ممکن رساند. متغیرهای انحرافی دلالت بر مقادیری دارند که نشان می‌دهند کدام‌یک از آرمان‌های متعدد، نسبت به سطوح مشخص خود بیشتر یا کمتر برآورده شده‌اند. برونداد این فن، مشخص خواهد کرد که کدام‌یک از آرمان‌ها برآورده شده، کدام‌یک به دست نیامده و به چه میزان برآورده نشده است. در بسیاری از موقعیت‌های واقعی، ممکن است انحراف از آرمان‌های خاصی بسیار مهم‌تر از انحراف از سایر آرمان‌ها باشد.

فرض می‌کنیم که $x_1, x_2, x_3, \dots, x_n$ متغیرهای تصمیم مسئله و Kتعداد هدف‌های موردنظر باشد. اگر C_{jk} ضریب g_k آن (j =1,2,...,n) در تابع هدف شماره K (k=1, 2… K) و همچنین g_k آرمان تعیین‌شده برای این هدف باشد روش پیشنهادی به دنبال جوابی است که تا حد امکان دستیابی به کلیه آرمان‌های مسئله را میسر سازد. رابطه (۳-۱۳) ساختار اهداف را در یک شمای کلی نمایش می‌دهد.

$$\sum_{j=1}^{n} c_j x_j = g_1 \quad \text{هدف } \mathbf{1} \qquad (۳-۱۳)$$

$$\sum_{j=2}^{n} c_{j2} x_j = g_2 \quad \text{هدف } \mathbf{2}$$

$$\sum_{j=3}^{n} c_{j3} x_j = g_3 \quad \text{هدف } \mathbf{3}$$

$$\sum_{j=k}^{n} c_{jK} x_j = g_K \quad \text{هدف } \boldsymbol{K}$$

ازآنجاکه تحقق هم‌زمان همه این آرمان‌ها مقدور نیست، لذا ضروری است که تا حد امکان دستیابی به هدف‌ها تعریف شود. در ساده‌ترین

حالت، زمانی که انحراف از آرمان‌های تعیین‌شده در هر دو جهت دارای اهمیت یکسانی باشد، می‌توان تابع هدف تلفیقی برنامه‌ریزی آرمانی را به شرح زیر فرموله کرد.

رابطه (۱۴-۳) مجموع انحراف از آرمان‌ها را حداقل می‌کند.

$$Z = \sum_{k=1}^{K} \left| \left(\sum_{j=1}^{n} c_{jK} x_j - g_k \right) \right| \qquad (۱۴-۳)$$

اگر کارایی را به‌صورت رابطه (۱۵-۳) نمایش داده شود.

$$Z \propto D(f_k) + L(f_k) + C_{res}(i,j) \qquad (۱۵-۳)$$

برای آن‌که رابطه (۱۵-۳) به یک رابطه تساوی تبدیل گردد باید ضرایبی را به فاکتورهای موردبررسی اضافه گردد این ضرایب به‌صورت w_1، w_2 وw_3 می‌باشد. در رابطه (۱۶-۳) رابطه نهایی ارائه می‌گردد.

$$Z = w_1 D(f_k) + w_2 L(f_k) + w_3 C_{res}(i,j) \qquad (۱۶-۳)$$

ضرایب w_1، w_2 وw_3 را با توجه به شرایط و میزان اهمیت هر یک از متغیرها در مسئله به‌صورت اختصاصی برای هر مسئله به‌صورت اختصاصی مقداردهی می‌شود.

با توجه به اینکه مجموع وزن‌های w_1 و w_2 و w_3 باید ۱ باشد لذا فرآیند وزن دهی مبتنی بر منطق فازی بر روی هدف w_1 و w_2 و w_3 انجام می‌گیرد. درواقع در هر مرحله یک وزن ثابت به هر یک از w_1، w_2 و w_3 اعطا می‌شود. سپس در هر مرحله به یکی از این وزن‌ها مثلاً w_1 به‌اندازه یک مقدار ثابت اضافه و به‌اندازه نصف همین مقدار از وزن w_2 و w_3 کم می‌شود. این کار برای هر یک از سه وزن تا رسیدن به بهترین جواب ادامه پیدا می‌کند. مثلاً در یک مرحله مقادیر وزن‌ها با افزایش وزن هدف اول به شکل رابطه (۱۷-۳) است:

در روابط فوق* $w_1 *, w_2 *, w_3$ عبارت از وزن‌های قبلی w_1, w_2, w_3 می‌باشد که در دوره قبل از اجرای الگوریتم یک مقدار را اختیار کرده و در این مرحله، به خاطر اضافه شدن یا کم شدن یک مقدار جهت ایجاد توازن بار * $w_1 *, w_2 *, w_3$ نام نهاده شده است. زمانی که توازن بار با اجرای الگوریتم صورت می‌گیرد، مقادیر فعلی که برای هریکی از وزن‌های w_1, w_2, w_3 به دست می‌آید، در اجرای بعدی الگوریتم جهت توازن بار، همین مقدارها با * $w_1 *, w_2 *, w_3$ نمایش داده می شود.

$$w_1 = w_1 * + Value_Rang \qquad (۱۷-۳)$$
$$w_2 = w_2 * - \frac{Value_Rang}{2}$$
$$w_3 = w_3 * - \frac{Value_Rang}{2}$$

همان‌طور که در رابطه (۱۷-۳) مشاهده می‌شود هدف اول جهت افزایش وزن فازی انتخاب‌شده و به‌اندازه Value_Rang به مقدار آن اضافه می‌شود. به دلیل اینکه در منطق فازی اگر به یک هدف مقداری اضافه شود باید به همان اندازه از سایر اهداف کم شود. به همین دلیل نصف مقدار وزن اضافه‌شده از سایر هدف‌های کم می‌شود.

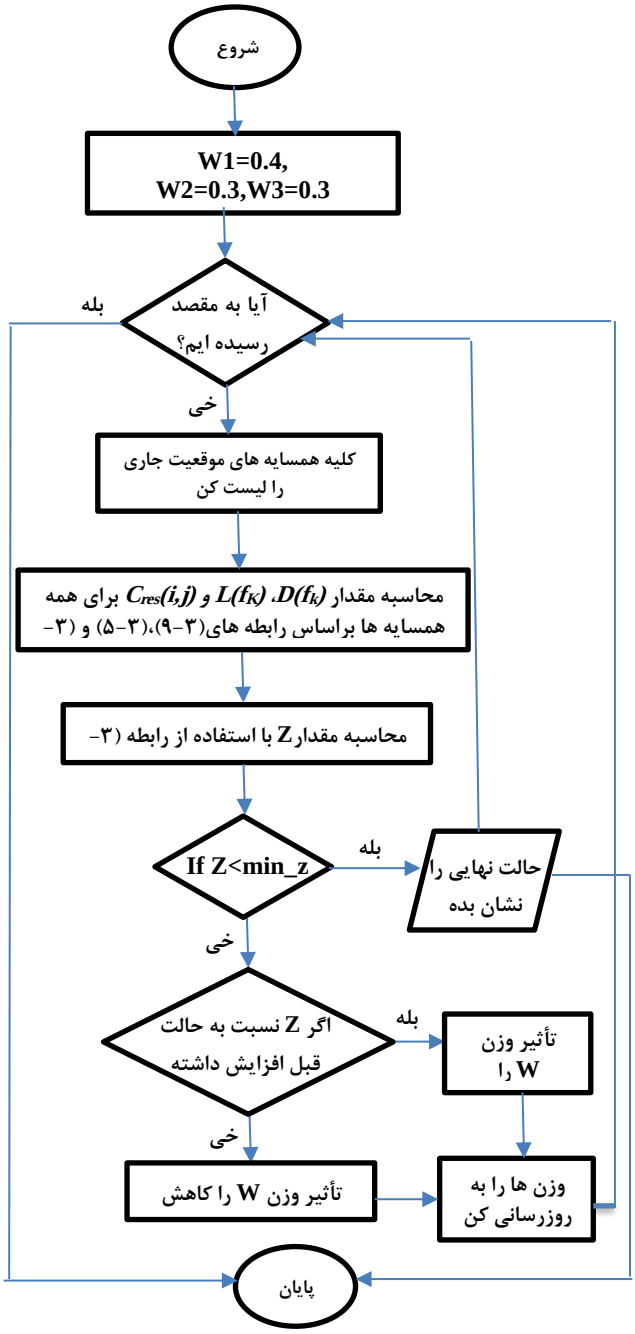
برای ایجاد توازن، این روال برای w_3، w_2 و w_1 اتفاق می‌افتد. مقادیر نهایی محاسبه و سپس در تمامی این حالات درمجموع بهترین جواب انتخاب می‌شود. در کلیه این حالات مجموع سه ضریب ۱ می‌شود. هدف از روش پیشنهادی کاهش مقدار Z می‌باشد.

در ادامه فلوچارت روش پیشنهادی در شکل ۳ ارائه شده است. زمانی که بخواهیم از یک وزن کم نموده و به وزن دیگر اضافه کنیم، می‌توانیم به‌اندازه نصف Value_Range از دو وزن دیگر بکاهیم و به وزن موردنظر اضافه نماییم. دلیل اینکه مقدار Value_Range را یک عدد مشخص و کوچک انتخاب کردیم این است که این عدد بسیار کوچک است و درنتیجه تغییرات قابل‌توجهی به وجود نمی‌آورد. بعد از آزمایش مقادیر مختلف، یکی از مقادیر که با تکرار کمتر و میزان انحراف کمتر سیستم را به نتیجه نزدیک‌تر می‌ساخت، به عنوان مقدار ثابت و مشخص برای Value_Range در نظر گرفتیم. ناگفته نباید گذاشت که این مقدار بعد از تعدد اجرا و آزمایش اعداد مختلف بدست آمده و به‌صورت تصادفی انتخاب نشده است.

الگوریتم‌های فرا اکتشافی برخلاف الگوریتم‌های دقیق روی کاهش انحرافات تمرکز می‌کند. زمانی که تعدد اجرا زیاد گردد، ۰.۰۳ یکی از مقادیری است که نتایج بهتری به دست می‌دهد و در توازن بار به‌صورت مؤثر عمل می‌کند.

زمانی که تأخیر، اتلاف و ظرفیت باقیمانده محاسبه شد، با مقادیری که در مرحله قبل بدست آمده مقایسه می‌شود. در صورتی که یکی از این پارامترها نسبت به محاسبه دوره قبلی، شرایط نامناسب‌تری داشته باشد، آن‌وقت تأثیر وزن‌ها در نظر گرفته می‌شود طوری که موازنه بین تمام وزن‌ها در نظر گرفته‌شده و در قسمت بدست آوردن Z نتیجه مطلوب را به نمایش بگذارد. ناگفته نماند که در بعضی موارد مثل هجوم بسته‌ها به شبکه و با توجه به وجود توازن بار، عملکرد شبکه پایین می‌آید. در این‌گونه موارد خاص Z می‌تواند

نسبت به مرحله قبل شرایط نامساعدتری داشته باشد، پس این‌گونه نیست که Z همیشه نسبت به اجرای قبلی شرایط بهتری داشته باشد.



شکل ۳: فلوچارت روش پیشنهادی

تغییر وزن‌ها بر اساس بیشترین و کمترین مقدار بدست آمده انتخاب می‌شود. بعد از اینکه تأخیر مجموع، اتلاف مجموع و ظرفیت باقی‌مانده بدست آمد، از بیشترین مقادیر بدست آمده روی کمترین مقدار اضافه می‌شود. بعدازاینکه مقادیر هر سه پارامتر بدست آمد، بین مقادیر بدست آمده مقایسه صورت می‌گیرد که کدام پارامتر بیشترین مقدار و کدام پارامتر کمترین مقدار را داشته است. هر پارامتری که کمترین مقدار را داشته باشد، تأثیر وزن روی آن افزایش‌یافته و از مقادیری با مقدارهای بالاتر کاسته شده و وزن‌ها به‌روزرسانی می‌شود.

۴- نتایج شبیه سازی

جهت ایجاد یک محیط اجرای آزمایش با شرایط یکسان، ابتدا به معرفی محیط شبیه‌سازی شبکه و سخت‌افزار آزمایش پرداخته می‌شود.

۴-۱- شبیه‌ساز OMNET++

جهت شبیه‌سازی و پیاده‌سازی الگوریتم پیشنهادی از نرم‌افزار برنامه‌نویسی مبتنی بر زبان C++ تحت عنوان OMNET++ استفاده می‌گردد. بخش‌های اصلی شبیه‌سازی به شرح زیر است:

- طراحی معماری شبکه.
- تعریف متغیرهای تحقیق در زمان شبیه‌سازی
- انجام فرآیند ارتباط متغیرهای تحقیق با معماری شبکه

۴-۲- سخت‌افزار و شرایط تعریف‌شده برای شبیه‌سازی

محیط انجام آزمایش دارای مشخصات زیر است:

- CPU سیستم اجراکننده شبیه سازی یک پردازنده Core I7 با فرکانس تا ۲.۲ مگاهرتز
- حافظه RAM ۱۶ گیگابایت
- نسخه نرم‌افزار OMNET++ نسخه ۵.۶.۱
- در سیستم‌عامل شبیه‌سازی Windows 10

۴-۳- پیاده‌سازی

جهت پیاده‌سازی شبکه در دو ساختار بررسی می‌گردد:

- ساختار گره‌های شبکه
- ساختار و توپولوژی کلی شبکه

۴-۳-۱- ساختار یک گره

هر گره عضو شبکه، شامل پارامترها و ارتباطات با سایر گره‌ها می‌باشد. در بخش کد نویسی برای هر گره بر اساس پارامترها و ارتباطات با سایر گره‌ها مبتنی بر توپولوژی شبکه رفتار طراحی می‌شود. ابتدا به بررسی ساختار و توپولوژی شبکه پرداخته می‌شود.

۴-۳-۲- ساختار و توپولوژی شبکه

توپولوژی شبکه در دو معماری زیر ارائه می‌گردد:

- توپولوژی AttMpls
- توپولوژی Goodnet

۴-۴- نتایج ثبت‌شده

روش پیشنهادی ازنظر پارامترهای تأخیر انتها به انتها، جریان‌های نقض شده QoS و پیوندهای فعال با مرجع [۵] و الگوریتم SWAY مقایسه می‌شود.

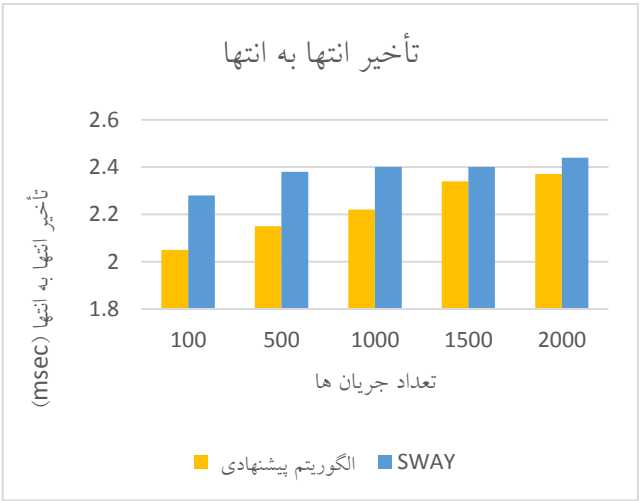
۴-۴-۱- تأخیر انتها به انتها

شکل ۴ تأخیر انتها به انتها روش پیشنهادی را با الگوریتم SWAY مقایسه می‌کند. دلیل کاهش تأخیر الگوریتم پیشنهادی نسبت به الگوریتم SWAY در بهره‌گیری روش پیشنهادی از وزن‌دهی مناسب مبتنی بر بهینه‌سازی چندهدفه می‌باشد. با بهره‌گیری از بهینه‌سازی چندهدفه و ترکیب مناسب با منطق فازی به علت تسریع در فرآیند انتخاب جریان تأخیر انتها به انتها نیز کاهش می‌یابد. همچنین با افزایش تعداد جریان‌ها به علت وجود محاسبات پیچیده‌تر تأخیر افزایش می‌یابد. بهره‌گیری از بهینه‌سازی چندهدفه و الگوریتم برنامه‌ریزی آرمانی فازی در تحقق هم‌زمان اهداف بهبود حاصل می‌نماید.

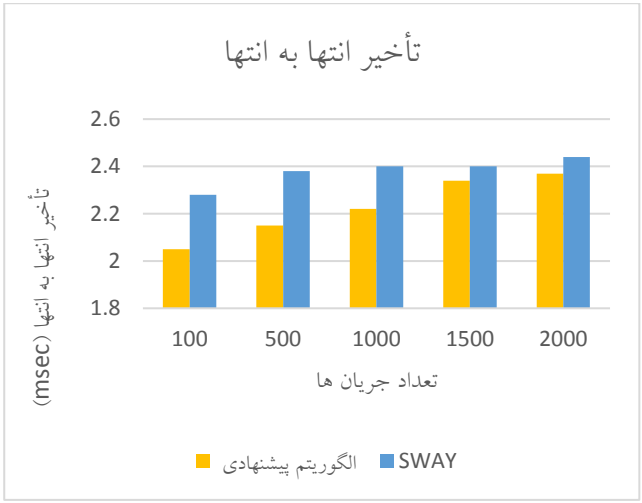
شکل ۵ تأخیر انتها به انتها در توپولوژی Goodnet را نمایش می‌دهد. دلیل کاهش تأخیر انتها به انتها در توپولوژی Goodnet نسبت به توپولوژی AttMpls در میزان اتصالات مستقیم مبتنی بر همبندی شبکه است. در توپولوژی Goodnet میزان تأخیر کمتری رخ می‌دهد.

۴-۴-۲- جریان‌های نقض شده QoS

شکل ۶ تعداد جریان‌های نقض شده QoS نسبت به تعداد جریان‌ها را نمایش می‌دهد. در روش پیشنهادی با بهره‌گیری از الگوریتم برنامه‌ریزی آرمانی فازی به‌عنوان یک رویکرد مبتنی بر افزایش کیفیت اهداف با بهینه‌سازی چند هدف به شکلی که نزدیک‌ترین



شکل ۴: تأخیر انتها به انتها در توپولوژی AttMpls

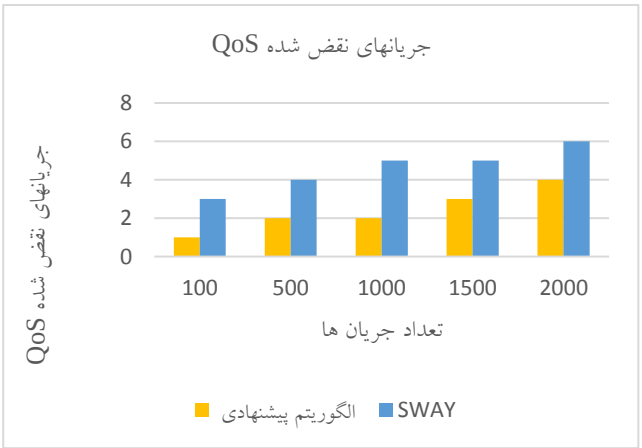


شکل ۵: تأخیر انتها به انتها در توپولوژی Goodnet

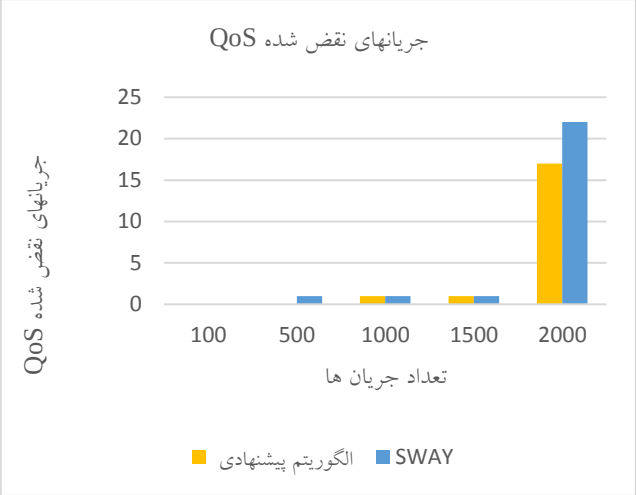
وضعیت به آرمانی‌ترین حالت باشد تعداد جریان‌های نقض شده روش پیشنهادی نسبت به الگوریتم SWAY کاهش‌یافته است. وقتی منطق فازی با یک رویکرد چندهدفه ترکیب می‌شود ماحصل آن ارائه مدلی با بهبود کیفیت و بهتر است. در الگوریتم پیشنهادی بهبود کیفیت و بهینه بودن اهداف نمود پیداکرده است.

شکل ۷ تعداد جریان‌های نقض شده QoS در توپولوژی Goodnet را نمایش می‌دهد. همچنان که مشاهده می شود، میزان جریان‌هایی که کیفیت خدمات را نقض می‌کند نسبت به روش قبلی کاهش‌یافته و این باعث می‌شود تا عملکرد بهتر و مناسب‌تری را داشته باشیم و از طریق بهبود عملکرد، رضایت کاربران را بدست بیاوریم.

در توپولوژی Goodnet به علت ساختار آن تعداد جریان‌های نقض شده QoS نسبت به توپولوژی AttMpls کاهش می‌یابد.



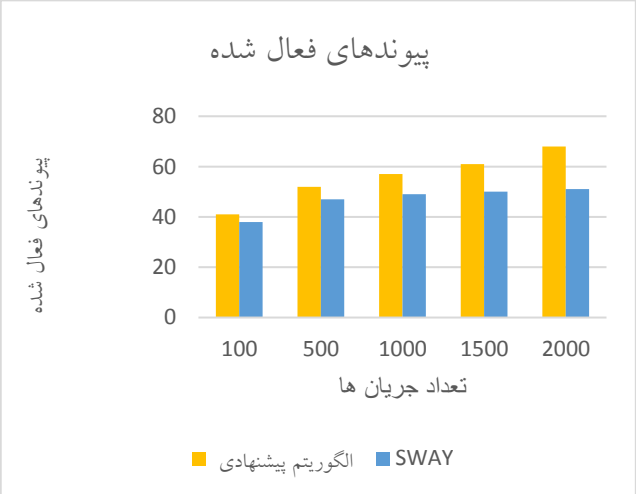
شکل ۶: تعداد جریان‌های نقض شده QoS در توپولوژی AttMpls



شکل ۷: تعداد جریان‌های نقض شده QoS در توپولوژی Goodnet

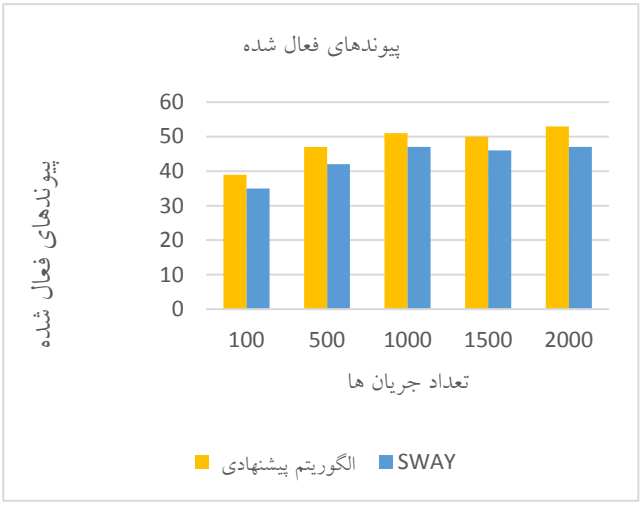
۴-۴-۳- پیوندهای فعال

شکل ۸ پیوندهای فعال‌شده در شبکه را در توپولوژی AttMpls نمایش می‌دهد. به علت ساختار محیط برنامه‌نویسی OMNET++ پیوندهای فعال نسبت به الگوریتم SWAT افزایش می‌یابد.



شکل ۸: پیوندهای فعال‌شده در توپولوژی AttMpls

شکل ۹ پیوندهای فعال‌شده در توپولوژی Goodnet نسبت به توپولوژی AttMpls افزایش‌یافته است. زمانی که تعداد پیوندهای فعال بیشتر باشد، تعدد مسیر بیشتر و با استفاده از توازن بار عملکرد کلی شبکه در حالتی که هجوم داده از محیط‌های ناهمگن صورت می‌گیرد، بهتر خواهد شد. گرچه با افزایش تعداد مسیر، هزینه افزایش می‌یابد، اما باید این را در نظر بگیریم که هجوم داده‌ها روی شبکه ممکن است شبکه را با مشکل مواجه سازد. ما نمی‌توانیم همزمان هم هزینه را در نظر بگیریم و هم آمادگی برای مقابله با هجوم داده‌ها داشته باشیم. محیط‌های ناهمگن در اینترنت اشیا غیرقابل‌پیش‌بینی است و این باعث می‌شود که اگر توازن بار به‌صورت درست صورت نگیرد، عملکرد کلی شبکه کاهش پیداکرده و درنتیجه کیفیت خدمات تقلیل پیدا کند.



شکل ۹: پیوندهای فعال‌شده در توپولوژی Goodnet

۵- نتیجه‌گیری

با توجه به ناهمگن بودن محیط‌های اینترنت اشیا، هجوم داده‌ها در شبکه‌های سنتی باعث می‌شود تا اینکه شبکه از عملکرد خوبی برخوردار نباشد و کیفیت خدمات کاهش پیدا کند. به دلیل اینکه شبکه نرم‌افزار محور می‌تواند به‌صورت مجزا هر جریان را بررسی و قانون متناسب را برای همان جریان تعیین نماید، این ویژگی توجه محققین را به خود جلب نموده و باعث شده تا از خاصیت شبکه نرم‌افزار محور در محیط‌های ناهمگن اینترنت اشیا بهره‌مند شوند.

با توجه به دید جامع SDN نسبت به کل شبکه می توان برای توازن بار از شبکه نرم‌افزار محور استفاده کرد. در شبکه‌های کامپیوتری سنتی، توزیع بار ناهمگون، غیریکنواخت و مبتنی بر روش‌های تصادفی در جریان‌های ورودی منجر به سیستم فاقد توازن بار می‌شود. کاهش QoS، از بین رفتن تعدادی از جریان‌های ورودی، سربار شدن، اشغال شدن ظرفیت شبکه، ناپایداری و درنهایت کاهش کارایی شبکه از نتایج عدم توازن بار است. لذا، در این مقاله به توزیع عادلانه ترافیک در قسمت‌های گوناگون شبکه و فرآیند تخصیص کارآمد منابع موجود پرداخته‌شده و از الگوریتم بهینه‌سازی چندهدفه برنامه‌ریزی آرمانی فازی در شبکه‌های SDN استفاده می‌شود.

ارزیابی های انجام شده نشان می دهد که روش پیشنهادی ازنظر تأخیر انتها به انتها، تعداد جریان‌های نقض شده QoS و پیوندهای فعال‌شده از الگوریتم SWAY شرایط بهتری را دارا می‌باشد.

۶- مراجع

[1] A. Filali, S. Cherkaoui and A. Kobbane, "Prediction-Based Switch Migration Scheduling for SDN Load Balancing," in *in ICC 2019-2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, China, 20-24 May 2019.

[2] L. Xiaoming, Y. Jinyao and R. Hui, "Software defined traffic engineering for improving Quality of Service," *China Communications*, vol. 14, no. 10, pp. 12-25, Oct 2017.

[3] A. Ian F, L. Ahyoung, W. PU, L. Min and C. Wu, "Research challenges for traffic engineering in software defined networks," *IEEE Network*, vol. 30, no. 3, pp. 52-58, May-June 2016.

[4] G. White, A. Palade, C. Cabrera and S. Clarke, "Quantitative Evaluation of QoS Prediction in IoT," in *in 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops*, Denver, 2017.

[5] N. Saha, S. Bera and S. Misra, "Sway: Traffic-Aware QoS Routing in Software-Defined IoT," *IEEE Transactions on Emerging Topics in Computing*, pp. 1-12, 2018.

[6] H. Elhammouti, E. Sabir, M. Benjillali, L. Echabbi and H. Tembine, "Self-Organized Connected Objects: Rethinking QoS Provisioning for IoT Services," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 41-47, Sept. 2017.

[7] K. Murat and D. Arjan, "A survey: Control plane scalability issues and approaches in software-defined networking (SDN)," *Computer Networks*, vol. 112, pp. 279-293, 2017.

[8] K. MathewsModieginyane, B. BettyLetswamotse, RezaMalekian and A. M. Abu-Mahfouz, "Software defined wireless sensor networks application opportunities for efficient network management: A survey," *Computers & Electrical Engineering*, vol. 66, pp. 274-287, 2018.

[9] M. Chahal, S. Harit, K. K. Mishra, A. K. Sangaiah and Z. Zheng, "A survey on software-defined networking in vehicular ad hoc networks: Challenges, applications and use cases," *Sustainable cities and society*, vol. 35, pp. 830-840, 2017.

[10] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393-430, 2018.

[11] S. Schaller and D. Hood, "Software defined networking architecture standardization," *Computer standards & interfaces*, vol. 54, pp. 197-202, 2017.

[12] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617-1634, 2014.

[13] Z. Shu, J. Wan, J. Lin, S. Wang, D. Li, S. Rho and C. Yang, "Traffic engineering in software-defined networking: Measurement and management," *IEEE Access*, vol. 4, pp. 3246-3256, 2016.

[14] A. Raschellà, F. Bouhafs, G. C. Deepak and M. Mackay, "QoS aware radio access technology selection framework in heterogeneous networks using SDN," *Journal of Communications and Networks*, vol. 19, no. 6, pp. 577-586, Dec 2017.

[15] S. Tomovic et al., "An architecture for QoS-aware service deployment in software-defined IoT networks," 2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC), Bali, Indonesia, 2017, pp. 561-567

[16] Ahammad, I., Khan, M. A., Salehin, Z. U., Uddin, M., & Soheli, S. J. Improvement of QoS in an IoT Ecosystem by Integrating Fog Computing and SDN. International Journal of Cloud Applications and Computing (IJCAC), vol. 11, no. 2, pp. 48-66, 2021.

[17] X. Guo, H. Lin, Z. Li and M. Peng, "Deep-Reinforcement-Learning-Based QoS-Aware Secure Routing for SDN-IoT," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6242-6251, July 2020

[18] Manel Majdoub, Ali El Kamel and Habib Youssef, " DQR: An Efficient Deep Q-Based Routing Approach in Multi-Controller Software Defined WAN (SD-WAN)", Journal of Interconnection Networks, vol. 20, no. 04, 2020

[19] M. E. Olaya, I. Bernal and D. Mejía, "Application for load balancing in SDN," in *in 2016 8th Euro American Conference on Telematics and Information Systems (EATIS)*, Cartagena, Colombia, 28-29 April 2016.

[20] Z. Hong, F. Yaming and C. Jie, "Reprint of "LBBSRT: An efficient SDN load balancing scheme based on server response time," *Future Generation Computer Systems*, vol. 80, pp. 409-416, 2018.

[21] M. Cello, Y. Xu, A. Walid, G. Wilfong, H. J. Chao and M. Marchese, "BalCon: A distributed elastic SDN control via efficient switch migration," in *in 2017 IEEE International Conference on Cloud Engineering (IC2E)*, Vancouver, BC, Canada, 4-7 April 2017.

[22] Y. Xu, M. Cello, I.-C. Wang, A. Walid, G. Wilfong, C. H.-P. Wen, M. Marchese and H. J. Chao, "Dynamic switch migration in distributed software-defined networks to achieve controller load balance," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 515-529, 05 February 2019 .

[23] H. Kaur and N. Jyoti, "Traffic Based Load Balancing in Software Defined Networking," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 9, no. 06, pp. 1-6, 2017.

[24] Z. Min, Q. Hua and Z. Jihong, "Dynamic switch migration algorithm with Q-learning towards scalable SDN control plane," in *in 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, Nanjing, China, 2017.

[25] F. AL-Tam and N. Correia, "Fractional switch migration in multi-controller software-defined networking," *Computer Networks*, vol. 157, no. 5, pp. 1-10, 2019.

[26] G. Wu, J. Wang, M. S. Obaidat, L. Yao and K.-F. Hsiao, "Dynamic switch migration with noncooperative game towards control plane scalability in SDN," *International Journal of Communication Systems*, vol. 32, no. 7, p. e3927, 2019.

[27] H. Xiao, B. Hu, L. Zhou and F. Wang, "DMSSM: A Decision-Making Scheme of Switch Migration for SDN Control Plane," in *in 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, Dalian, China, China, 2019.

[28] K. S. Sahoo, D. Puthal, M. Tiwary, M. Usman, B. Sahoo, Z. Wen, B. P. S. Sahoo and R. Ranjan, "ESMLB: Efficient Switch Migration-based Load Balancing for Multi-Controller SDN in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5852 - 5860, 2019.

[29] N. J. N. M. H. a. A. R. A. Akbar Neghabi, "Nature-inspired meta-heuristic algorithms for solving the load balancing problem in the software-defined network," *International Journal of Communication Systems*, vol. 32, no. 4, p. e3875, 2019.

[30] J. Jin, J. Gubbi, T. Luo and M. Palaniswami, "Network architecture and QoS issues in the internet of things for a smart city," in *in 2012 International Symposium on Communications and Information Technologies (ISCIT)*, Gold Coast, QLD, Australia, 2012.

[31] M. E. a. A. K. C. Bormann, "Terminology for constrained-node networks," in *Internet Engineering Task Force (IETF): Fremont, CA*, USA, 2014.

[32] A. Hakiri, P. Berthou, A. Gokhale and S. Abdellatif, "Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications," *IEEE communications magazine*, vol. 53, no. 9, pp. 48-54, 2015.

[33] H. Guo, J. Liu, Z. M. Fadlullah and N. Kato, "On minimizing energy consumption in FiWi enhanced LTE-A HetNets," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 4, pp. 579-591, 2016.

[34] H. Guo, J. Liu and L. Zhao, "Big data acquisition under failures in FiWi enhanced smart grid," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 3, pp. 420 - 432, 01 March 2017.

[35] S. Tomovic, K. Yoshigoe, I. Maljevic and I. Radusinovic, "Software-defined fog network architecture for IoT," *Wireless Personal Communications*, vol. 92, no. 1, pp. 181-196, 2017.

[36] K. Hussain, M. N. M. Salleh, S. Cheng and Y. Shi, "Metaheuristic Research: A Comprehensive Survey," *Universiti Tun Hussein Onn Malaysia*, vol. 52, no. 12, pp. 1-35, 13 January 2018.

پاورقی‌ها:

[1] Software Defined Network
[2] Radio Frequency Identification
[3] Wireless sensor
[4] Quality of service
[5] Display Sensor
[6] Actuator
[7] End to End Delay
[8] Radio Access Technology
[9] Heterogeneous Network
[10] Signal to Interference Plus Noise Ratio
[11] Downlink
[12] Factor Fitting
[13] Quality of Experience
[14] First In First Out
[15] Round Robin
[16] Deficit Round Robin
[17] Load Balancing Scheme Based On Server Response Time
[18] Internet Control Message Protocol
[19] Autoregressive Integrated Moving Average
[20] Balanced Controller
[21] Non cooperative Game
[22] Decision-Making Scheme of Switch Migration
[23] Transmission Delay
[24] Efficient Switch Migration-based Load Balancing
[25] Internet of Things (IoT)
[26] Bandwidth