فصلنامه سیستم‌های پردازشی و ارتباطی چندرسانه‌ای هوشمند
Intelligent Multimedia Processing and Communication Systems (IMPCS)

Winter 2023, 3 (4), 15-27
DOR: 20.1001.1.27832570.1401.3.4.2.6

Received: 24 Sep 2022
Accepted: 3 Nov 2022

مقاله پژوهشی

Dynamic Migration of SDN-based Switches to Distribute Control Layer Load and Increase Efficiency Using Ryu Controller

Neda Abbasi[1], Shiva Karimi[2*]

1. Mcs Student, Computer Engineering, Faculty of Electrical and Computer Engineering, Zanjan Branch, Islamic Azad University, Zanjan, Iran.
2. Assistant Professor, Faculty of Electrical and Computer Engineering, Zanjan Branch, Islamic Azad University, Zanjan, Iran.*(Corresponding Author)* shkarimi2013@yahoo.com

Abstract

Introduction: Network needs in today's era include comprehensive improvement of communication and connections with more bandwidth, minimum delay and high throughput. Software-defined networking is one of the most promising modes of future Internet development due to their centralized planning and management capabilities. The load balancing problem for distributed SDN controllers is one of the fundamental challenges of these networks, and a single centralized controller may cause reliability and scalability problems, and although several controllers can solve these problems, a flexibility mechanism is required for load balancing.

Method: Therefore, in this research, a load balancing mechanism is proposed based on the load information strategy, so that each controller in the system periodically reports its load information to other controllers, and the overload controllers of the plan, load infomation They do not collect all other controllers before local decisions are made, which in turn reduces controller resource consumption. A load balancing strategy is proposed in the proposed response time-based scheme for multiple SDN controllers, which takes into account the real-time response time characteristics that change with the controller load. Another key part of the proposed scheme is that all these things are done in the Ryu controller by choosing an appropriate response time threshold and processing multiple overload controllers simultaneously, which can solve the load balancing problem of the SDN control panel of the overload controller well. slow This model proposes to transfer the management of specific switches between controllers.

Results: In the simulation, the historical load and the future load of the switches were considered, and a switch migration algorithm with double weight was proposed, which led to a decrease in the frequency of switch migration. Experiments have proven that this scheme can quickly provide load balancing between controllers and reduce the number of switch migrations. The results showed that this design can lead to the migration of the overload controller and quickly reduce the workload of the overload controller. As a result, the proposed scheme can effectively achieve load balancing of multiple SDN controllers with high speed.

Keywords: Dynamic migration of switches, Software-based, Distribution and load balancing, Control layer, Ryu controller.

فصلنامه سیستم‌های پردازشی و ارتباطی چندرسانه‌ای هوشمند
Intelligent Multimedia Processing and Communication Systems (IMPCS)
انجمن علمی تجارت الکترونیکی ایران

مهاجرت پویا سوئیچ‌های مبتنی بر SDN به منظور توزیع بار لایه کنترل و افزایش بهره‌وری با استفاده از کنترلر ریو

تاریخ دریافت: ۱۴۰۱/۰۷/۰۲
تاریخ پذیرش: ۱۴۰۱/۰۸/۱۲

دورهٔ سوم ، زمستان ۱۴۰۱
شمارهٔ چهارم، صص: ۱۵-۲۷

ندا عباسی[۱]، شیوا کریمی[۲*]

۱. دانشجو کارشناسی ارشد، مهندسی کامپیوتر، دانشکده برق و کامپیوتر، واحد زنجان، دانشگاه آزاد اسلامی، زنجان، ایران.
Neda.abbasi24724@gmail.com
۲. استادیار، دانشکده برق و کامپیوتر، واحد زنجان، دانشگاه آزاد اسلامی، زنجان، ایران. (نویسنده مسئول)shkarimi2013@yahoo.com

چکیده: نیازهای شبکه در عصر کنونی شامل ارتقای همه‌جانبه ارتباطات و اتصالات با پنهای باند بیشتر، حداقل تأخیر و دستیابی به توان بالاست. شبکه‌های مبتنی بر نرم‌افزار به دلیل قابلیت‌های برنامه‌ریزی و مدیریت متمرکز، یکی از امیدوارکننده‌ترین حالت‌های توسعه اینترنت آینده به‌شمار می‌آیند. مشکل تعادل بار برای کنترل‌کننده‌های SDN توزیع‌شده یکی از چالش‌های اساسی این شبکه ها به‌حساب‌می‌آید و ممکن است یک کنترل کننده متمرکز منفرد باعث مشکلات قابلیت اطمینان و مقیاس‌پذیری شود و اگرچه چندین کنترلر می‌توانند این مشکلات را حل‌کنند اما یک مکانیسم انعطاف‌پذیر برای تعادل بار مورد نیاز است. از این رو در این در پژوهش یک مکانیسم تعادل بار بر اساس استراتژی اطلاعات بار پیشنهادمی‌شود به طوری‌که هر کنترل‌کننده در سیستم به‌صورت دوره‌ای اطلاعات بار خود را به کنترل‌کننده‌های دیگر گزارش‌می‌دهد و کنترل‌کننده‌های اضافه بار طرح، اطلاعات بار همه کنترل‌کننده‌های دیگر را قبل از تصمیم‌گیری محلی جمع‌آوری‌نمی‌کنند، که این خود باعث کاهش مصرف منابع کنترل‌کننده-می‌شود. استراتژی متعادل‌سازی بار در طرح پیشنهادی مبتنی بر زمان پاسخ برای کنترل‌کننده‌های SDN متعدد پیشنهادشده، که ویژگی‌های زمان پاسخ بلادرنگ را که با بار کنترل‌کننده تغییرمی‌کند، درنظرمی‌گیرد. بخش کلیدی دیگر طرح پیشنهادی این است که تمامی این موارد در کنترل‌کننده Ryu با انتخاب یک آستانه زمان پاسخ مناسب و پردازش چندین کنترل‌کننده اضافه بار به‌طور همزمان انجام‌شده که مشکل تعادل بار صفحه کنترل SDN کنترل‌کننده اضافه بار را می‌تواند به‌خوبی حل‌کند. این مدل پیشنهادمی‌کند که مدیریت سوئیچ‌های خاص بین کنترل‌کننده‌ها منتقل‌شود. در شبیه‌سازی بار تاریخی و بار آینده سوئیچ‌ها درنظرگرفته‌شد و یک الگوریتم مهاجرت سوئیچ با وزن دوگانه پیشنهادشد که منجر به کاهش فرکانس مهاجرت سوئیچ گشت. آزمایش‌ها ثابت کرده‌اند که این طرح می‌تواند به‌سرعت تعادل بار بین کنترل‌کننده‌ها را تأمین و تعداد مهاجرت سوئیچ‌ها را کاهش‌دهد. نتایج نشان‌داد این طرح می‌تواند منجر به مهاجرت کنترل‌کننده اضافه بار شده و به سرعت بار کاری کنترلر اضافه بار را کاهش‌دهد. در نتیجه، طرح پیشنهادی می‌تواند به‌طور مؤثر با سرعت بالا به تعادل بار چندین کنترل‌کننده SDN دست‌یابد.

واژه‌های کلیدی: مهاجرت پویا سوئیچ‌های مبتنی بر نرم‌افزار، توزیع و تعادل بار، لایه کنترل، کنترلر Ryu.

۱- مقدمه

شبکه‌های مبتنی بر نرم‌افزار دیدگاه جدیدی در مورد نحوه مدیریت شبکه‌ها است که به‌سرعت در حال تبدیل‌شدن به راه حلی برای کسانی است که در غلبه بر محدودیت‌های شبکه‌های سنتی مشکل‌دارند، این با جداکردن سخت‌افزار از نرم‌افزار یعنی جداکردن صفحه کنترل (که تعیین‌می‌کند کجا ترافیک ارسال‌شود) است. از صفحه داده (که این تصمیمات را انجام‌می‌دهد و ترافیک را ارسال‌می‌کند)، شبکه مبتنی بر نرم‌افزار سخت‌افزار را قادرمی‌سازد تا از یک برنامه نرم‌افزاری متمرکز که از خود سخت‌افزار جداشده‌است، کنترل/ مدیریت شود[۱]. با در نظر گرفتن سربار شبکه، تکنیک‌های متعادل‌سازی بار از اهمیت قابل‌توجهی برخوردار هستند. تعادل بار مستقیماً بر برنامه و دسترسی به سرویس برای کاربران تأثیرمی‌گذارد. هدف از تعادل بار، همان‌طور که اشاره‌شد بهینه‌سازی استفاده از منبع با به حداکثر رساندن توان عملیاتی، به حداقل رساندن زمان پاسخ و جلوگیری از بارگذاری بیش از حد روی هر منبع واحد است. برای کاهش بار ترافیک و کاهش خطر تبدیل سرور به نقطه حساس به خطا، بسیاری از مراکز داده روش‌های سخت‌افزاری اختصاصی را برای فعال‌کردن تعادل بار برای پشتیبانی از تعداد زیادی از کاربران درنظرمی‌گیرند. با وجود این، تهیه سیستم‌های سخت‌افزاری معمولاً گران هستند، استقرار آن‌ها از نظر فنی چالش برانگیز است و برای کار مداوم نیاز به مداخله انسان دارند[۲]. چند پارامتر برای ارزیابی یک محاسبه متعادل‌کننده بار مورد نیاز است. قابل‌توجه‌ترین پارامترهای ذهنی برای تعادل بار در شبکه‌های مبتنی نرم‌افزار شامل: درجه تعادل بار، تأخیر زمان اجرا، مصرف انرژی، میانگین همگام‌سازی در واحد زمان، ورودی‌های بازارسال، سربار هزینه مهاجرت، نرخ ازدست‌دادن بسته، حجم کار، زمان پاسخگویی، توان عملیاتی، استفاده از منابع و ریشه میانگین مربعات خطا[۳]:

در شبکه مبتنی بر نرم‌افزار، برای ارضای تنظیم بار مورد نیاز، کنترل‌کننده باید یک پیشروی از تعیین‌های حداقل بارگذاری شده سرور را به‌پایان‌برساند. در هر صورت، استفاده از RLS به‌عنوان یک کنترل‌کننده در شبکه مبتنی بر نرم‌افزار نشان‌دهنده چند مشکل است، به‌عنوان مثال، گلوگاه‌های یک کنترل‌کننده منفرد، پاسخ‌دهی، کیفیت تزلزل‌ناپذیر و تطبیق‌پذیری ضعیف. برای رفع مشکلات ارجاع‌شده، استفاده از کنترل‌کننده‌های پراکنده مختلف که با هم همکاری می‌کنند، پاسخی اساسی برای برآورده‌کردن ظرفیت کنترل‌کننده به‌طور منسجم است. چندین تلاش تحقیقاتی برای پیشنهاد یک مسیر صحیح در شبکه مبتنی بر نرم‌افزار انجام شده است. نقش کنترل‌کننده به دو مرحله تقسیم می‌شود. مرحله اول برای نظارت مستمر بار ترافیکی و مرحله دوم برای تصمیم‌گیری صحیح برای یک مسیر بهینه استفاده می‌شود. معماری تعادل بار سه لایه به نام‌های برنامه کنترل کننده، کنترل‌کننده و شبکه های انتقال‌داده‌است. وظایف هر لایه در زیر توضیح داده‌شده‌است[۳]:

برنامه کنترلر: برنامه متعادل‌کننده بار به‌عنوان برنامه کنترلی اجرامی‌شود که میزبان/سرور بهینه موجود در شبکه انتقال داده را شناسایی‌می‌کند.

کنترلر: شامل ماژول‌هایی به‌عنوان مسیر ارسال‌شده، جدول مسیریابی، وضعیت شبکه و بار سرور است. کنترلر اطلاعات مربوط به توپولوژی شبکه، وضعیت پیوند و بار جاری روی سرور متصل به دامنه ها را با پرس‌وجو از دستگاه های ارسال به صورت دوره‌ای دریافت‌می‌کند و این اطلاعات را به برنامه متعادل‌کننده بار که در برنامه کنترلر پیاده‌سازی شده‌است، ارسال‌می‌کند.

شبکه‌های انتقال داده: این شبکه از چند دامنه تشکیل‌شده و هر دامنه توسط لبه دستگاه‌های انتقال داده موجود در دامنه مربوطه به هم متصل‌می‌شود. هر دامنه شامل چند میزبان (یا سرور) است که با دستگاه‌های انتقال داده و دستگاه‌های انتقال داده با یکدیگر مرتبط هستند.

انتخاب کنترل‌کننده در یک شبکه مبتنی بر نرم‌افزار اهمیت بسیار زیادی دارد. به‌طوری که کنترلرکننده با قرارگیری مناسب در شبکه تعداد زیادی میزبان، مسیریاب و سوئیچ را شبیه‌سازی یا تقلیدمی‌کند و می‌تواند تا حد زیادی از بار شبکه کاسته و باعث تأخیر کمتر در شبکه و افزایش امنیت در شبکه شود. با توجه به اینکه کنترلرها نقش اساسی در شبکه‌های مبتنی بر نرم‌افزار دارند و مشخص می‌کنند که سوئیچ‌ها در قبال پکت‌ها چه عملی را باید انجام‌دهند با تعیین کد مناسب و دادن هشدارهایی که توسط کنترلر ریو انجام‌می‌شود بار شبکه کمتر شده و زمان تأخیر به حداقل‌ترین زمان ممکن می‌رسد. همچنین کنترلر Ryu در محیط شبیه‌سازی با مشاهده توان کنترل‌کننده آزمایش‌می‌شود و عملکرد آن در شرایط شبکه پویا مورد بررسی قرارمی‌گیرد. در نتیجه این پژوهش به این سؤال پاسخ داده‌می‌شود که: آیا مهاجرت پویا سوئیچ های مبتنی بر SDN به منظور توزیع بار لایه کنترل و افزایش بهره‌وری با استفاده از کنترلر ریو امکان‌پذیر است؟

۲.پیشینهٔ پژوهش

Deng و همکارانش در سال ۲۰۲۱ مقاله‌ای تحت عنوان "متعادل‌سازی بار کنترل‌کننده SDN کارآمد با استفاده از یادگیری تقویت‌کننده عمیق" ارائه‌دادند. آن‌ها معتقدند که برای حل مسائل مقیاس‌پذیری و قابلیت اطمینان مانند نقطه شکست واحد، خوشه کنترل‌کننده توزیع‌شده به‌طور گسترده در شبکه‌های نرم‌افزاری تعریف‌شده در مقیاس بزرگ استفاده‌می‌شود. در مواجهه با مشکل تخصیص بار نامتعادل خوشه کنترل‌کننده ناشی از نوسان ترافیک شبکه، مهاجرت سوئیچ پویا به‌طور مؤثر آن را حل‌می‌کند. با این حال، مهاجرت سوئیچ پویا یک مشکل NP-hard است، زمان تصمیم‌گیری با افزایش مقیاس شبکه بسیار افزایش‌می‌یابد. با درنظرگرفتن این نکته، این مقاله ELB-PDQN را پیشنهادمی‌کند، یک طرح متعادل‌کننده بار کارآمد بر اساس اولویت‌بندی تجربه تکراری عمیق Q-Learning (PDQN) برای SDN ، که هدف آن حل مؤثر مشکل

مهاجرت سوئیچ پویا در شبکه‌های مقیاس بزرگ است .ELB-PDQN از یادگیری تقویت عمیق چند عاملی و تصمیم‌گیری توزیع‌شده استفاده-می‌کند که یک ماژول متعادل‌کننده بار و یک مدل PDQN را بر روی هر کنترل‌کننده در خوشه کنترل‌کننده مستقرمی‌کند. این روش پیچیدگی مدل PDQN را کاهش‌می‌دهد و عملکرد متعادل‌کننده بار را بهبود می-بخشد. علاوه بر این، این مقاله عوامل مختلفی را در شبکه درنظرمی‌گیرد که بر عملکرد متعادل‌سازی بار تأثیرمی‌گذارند تا مکانیزم متعادل‌سازی بار کم پیچیدگی و عملکرد بالا محقق‌شود و قفل همگام‌سازی را برای حل تضادهای تعادلی ناشی از تصمیم‌گیری توزیع‌شده معرفی‌شود. نتایج شبیه‌سازی نشان‌می‌دهد که عملکرد ELB-PDQN حداقل ۳۷٫۶٪ بهبودیافته‌است و زمان مورد نیاز برای تصمیم‌گیری در مقایسه با طرح موجود به‌طور قابل‌توجهی کاهش‌یافته‌است[۴].

Yeo و همکارانش در سال ۲۰۲۱ مقاله‌ای تحت عنوان "دستیابی به توزیع بار متعادل با انتقال سوئیچ مبتنی بر یادگیری تقویتی در کنترلرهای SDN توزیع‌شده" ارائه دادند. آن‌ها معتقدند که کنترل‌کننده‌های توزیع‌شده در شبکه‌های تعریف‌شده با نرم‌افزار به دلیل استقرار مقیاس‌پذیر و قابل اعتمادشان در محیط‌های SDN فعلی، به یک رویکرد امیدوارکننده تبدیل‌می‌شوند. از آنجاکه ترافیک شبکه با زمان و مکان متفاوت است، یک نگاشت ایستا بین سوئیچ‌ها و کنترل‌کننده‌ها باعث توزیع نابرابر بار بین کنترل‌کننده‌ها می‌شود. روش‌های مهاجرت دینامیکی سوئیچ‌ها می‌تواند توزیع بار متعادلی را بین کنترل‌کننده‌های SDNفراهم‌کند. اخیراً، روش‌های یادگیری تقویتی موجود برای مهاجرت سوئیچ پویا مانند MARVEL در حال مدل‌سازی تعادل بار هر کنترل‌کننده به عنوان بهینه‌سازی خطی است. حتی اگر به‌طور گسترده برای مدل‌سازی جریان شبکه استفاده‌شود، این نوع بهینه‌سازی خطی به‌خوبی با حجم کار دنیای واقعی کنترل‌کننده‌های SDN سازگار نیست، زیرا همبستگی‌های بین انواع منابع به‌طور غیرمنتظره و پیوسته تغییرمی‌کند. در نتیجه، استفاده از مدل خطی برای استفاده از منابع، تشخیص اینکه کدام نوع منابع در حال حاضر بیش از حد بارگذاری شده‌اند را دشوارمی‌کند. علاوه بر این، این کار هزینه زمانی بالایی را به همراه دارد. در این مقاله، یک طرح انتخاب سوئیچ و کنترل‌کننده مبتنی بر یادگیری تقویتی برای مهاجرت سوئیچ، متعادل‌سازی بار یادگیری تقویتی آگاه از سوئیچ (SAR-LB) پیشنهادمی‌شود. SAR-LB از نسبت استفاده از انواع منابع مختلف در کنترلرها و سوئیچ‌ها به‌عنوان ورودی شبکه عصبی استفاده‌می‌کند. همچنین سوئیچ‌ها را به‌عنوان عوامل RL برای کاهش فضای عمل یادگیری درنظرمی‌گیرد، در حالی که همه موارد مهاجرت را در نظر می‌گیرد. نتایج تجربی نشان‌می‌دهد که SAR-LB به دلیل تصمیم‌گیری دقیق مهاجرت سوئیچ، توزیع بار بهتر (نزدیک به یکنواخت) را در بین کنترل‌کننده‌های SDN به‌دست‌آورد. طرح پیشنهادی انحراف استاندارد نرمال‌شده بهتری را در میان کنترل‌کننده‌های SDN توزیع‌شده نسبت به طرح‌های موجود تا ۳۴ درصد به‌دست‌می‌آورد[۵].

Malbašić و همکارانش در سال ۲۰۲۲ مقاله‌ای تحت عنوان " شبکه‌های SDN ترکیبی: یک طرح متعادل‌کننده بار سرور چند پارامتری" ارائه‌دادند. آن‌ها معتقدند که شبکه‌های تعریف‌شده با نرم‌افزار مزایای بسیاری از جمله قابلیت برنامه‌ریزی ترافیک، چابکی و اتوماسیون شبکه را ارائه‌می‌دهند. با این حال، محدودیت‌های بودجه با مشکلات فنی (مانند مقیاس‌پذیری، تحمل خطا، مسائل امنیتی) و گاهی اوقات چالش‌های تجاری (پذیرش کاربر و اطمینان اپراتورهای شبکه) باعث-می‌شود ارائه‌دهندگان برای استقرار کامل SDN تصمیم‌گیری‌نکنند. بنابراین، استقرار تدریجی عملکرد SDN از طریق قراردادن مجموعه محدودی از دستگاه‌های SDN در میان دستگاه‌های سنتی، محیطی منطقی و کارآمد را نشان‌می‌دهد که می‌تواند خدمات مدرن و فشرده‌تری را به مشتریان ارائه‌دهد. یک چالش منحصر به فرد توزیع انعطاف پذیر بارها بر روی سرورهایی است که این خدمات را در محیط‌های شبکه ارائه‌می‌دهند. پژوهش در این مقاله بر توسعه یک طرح متعادل‌کننده بار جدید با استفاده از یک محیط SDN ترکیبی ساخته شده با حداقل مجموعه ای از دستگاه های SDN (کنترل‌کننده و یک سوئیچ) متمرکز است. یک طرح متعادل‌کننده بار جدید را برای نظارت بر شاخص‌های بار سرور فعلی و اعمال معیارهای چند پارامتری برای زمان‌بندی اتصالات برای متعادل‌کردن بار روی سرورها تا حد امکان مؤثر پیشنهادمی‌شود. اساس طرح جدید تعادل بار، نظارت مستمر بر شاخص‌های بار سرور و پیاده‌سازی معیارهای چند پارامتری (بارCPU ، خواندن ورودی/خروجی، نوشتن ورودی/خروجی، آپلود پیوند، دانلود لینک) برای زمان‌بندی اتصالات است. آزمایش انجام‌شده بر روی سرورها با هدف متعادل‌کردن بار سرور تا حد امکان مؤثر است. نتایج به‌دست‌آمده نشان‌می‌دهد که این مکانیسم به نتایج بهتری نسبت به طرح‌های متعادل‌کننده بار موجود در شبکه‌های سنتی و SDN دست‌می‌یابد. علاوه بر این، یک طرح متعادل-کننده بار پیشنهادی را می توان با سرویس‌های مختلف استفاده‌کرد و در هر محیط مشتری-سرور اعمال‌کرد[۶].

Irandoost و همکارانش در سال ۲۰۲۲ مقاله‌ای تحت عنوان "یک رویکرد خودکار یادگیری برای تعادل بار در شبکه‌های تعریف‌شده با نرم‌افزار " ارائه‌دادند. آن‌ها معتقدند از آنجا که شبکه نرم‌افزاری یک فناوری متمرکز منطقی است، اهمیت مقیاس‌پذیری صفحه کنترل با افزایش مقیاس شبکه افزایش‌یافته‌است. بنابراین استفاده از کنترل‌کننده-های متعدد به جای کنترل‌کننده متمرکز پیشنهادشد. اگرچه چندین کنترلر مقیاس‌پذیری را در شبکه‌های مبتنی بر نرم‌افزار حل‌کرده‌اند، اما به دلیل تخصیص استاتیک بین کنترل‌کننده‌ها و سوئیچ‌ها، با عدم تعادل بار روی کنترل‌کننده‌ها مواجه‌شدند. در نتیجه، مهاجرت سوئیچ به‌عنوان یک رویکرد کارآمد برای حل تخصیص استاتیک بین کنترل‌کننده و سوئیچ پیشنهادشده‌است. مهاجرت سوئیچ امکان اتصال پویا بین کنترلرها و سوئیچ‌ها را فراهم‌می‌کند، اما اینکه کدام کنترلر یا سوئیچ برای مهاجرت مناسب است به خودی خود به یک مشکل حیاتی تبدیل-

شده‌است. یک خودکار یادگیری با ساختار متغیر برای انتخاب کنترل-کننده هدف در روش پیشنهادی ارائه‌شده‌است. همه موارد انتخاب و واکنش محیط با خودکار یادگیری ارزیابی‌می‌شوند و بهترین کنترل‌کننده را برای هزینه‌های مهاجرت انتخاب‌می‌کنند. روش پیشنهادی با الگوریتم‌های پیشرفته مقایسه‌شده است. نتایج نشان‌داد که رویکرد پیشنهادی می‌تواند با متعادل‌کردن کنترل‌کننده‌ها با انتخاب بهینه کنترل‌کننده‌های هدف برای مهاجرت سوئیچ، تأخیر ارسال بسته‌ها در شبکه را کاهش‌دهد[۷].

Liu و همکارانش در سال ۲۰۲۲ مقاله‌ای تحت عنوان " CASM:یک استراتژی مهاجرت سوئیچ آگاه از هزینه برای شبکه‌های بین داده‌ای نوری الاستیک" ارائه‌دادند. آن‌ها معتقدند که در شبکه‌های نوری الاستیک بین مرکز داده، استقرار چند کنترلر برای بهبود پایداری و مقیاس پذیری صفحه کنترل اتخاذشده‌است. همان‌طور که مقیاس شبکه افزایش‌می‌یابد، طرح استقرار چند کنترل‌کننده سنتی ویژگی‌های دینامیکی ترافیک را نادیده‌می‌گیرد و در نتیجه بار نامتعادل در بین چندین کنترل‌کننده ایجادمی‌شود. در پاسخ به این مشکل، مکانیسم مهاجرت سوئیچ موجود برای دستیابی به توزیع متعادل بارهای کنترلی پیشنهادشده‌است. با این حال، اکثر کارهای تحقیقاتی موجود هزینه اضافی مهاجرت سوئیچ را درنظرنمی‌گیرند و عملکرد متعادل‌کننده بار کنترل‌کننده به‌طور قابل‌توجهی پس از مهاجرت سوئیچ بهبودنمی‌یابد. در این مقاله، یک استراتژی مهاجرت سوئیچ آگاه از هزینه برای متعادل‌سازی بار کنترل‌کننده پیشنهادمی‌کنیم .استراتژی استراتژی مهاجرت سوئیچ آگاه از هزینه پیشنهادی ابتدا بار کنترل‌کننده را از طریق چند شاخص عملکردی که بر بار کنترل‌کننده تأثیرمی‌گذارند اندازه‌گیری‌می‌کند، و سپس قضاوت‌می‌کند که آیا کنترلر بر اساس زمان پاسخ کنترل‌کننده به پیام درخواست بارگذاری شده یا کم‌بار شده‌است، در نتیجه عملکرد متعادل‌سازی بار کنترلر را بهبودمی‌بخشد. علاوه بر این، هنگام انتخاب سوئیچ برای مهاجرت، استراتژی مهاجرت سوئیچ آگاه از هزینه سوئچ بهینه را برای مهاجرت بر اساس به حداقل رساندن هزینه مهاجرت انتخاب‌می‌کند و در نتیجه هزینه مهاجرت سوئیچ را کاهش-می‌دهد. ارزیابی عملکرد نشان‌می‌دهد که استراتژی مهاجرت سوئیچ آگاه از هزینه به‌طور قابل‌توجهی عملکرد متعادل‌کننده بار کنترلرها را بهبودمی‌بخشد و هزینه مهاجرت را در مقایسه با راه‌حل‌های موجود کاهش‌می‌دهد[۸].

Li و همکارانش در سال ۲۰۲۱ مقاله‌ای تحت عنوان " تعادل بار کنترل‌کننده SDN بر اساس یادگیری تقویتی" ارائه‌دادند. آن‌ها معتقدند با هدف اضافه بار محلی استقرار چند کنترلر در شبکه‌های تعریف شده توسط نرم‌افزار، یک مکانیسم متعادل‌کننده بار کنترل کننده SDNبر اساس یادگیری تقویتی طراحی‌شده‌است. دامنه مهاجرت خروجی جفت‌شده اولیه و مهاجرت در دامنه با محاسبه انحراف نسبت بار بین کنترل‌کننده‌ها به‌دست‌می‌آید، یک سه‌گانه مهاجرت اولیه، شامل دامنه مهاجرت ذکرشده در بالا و گروهی از سوئیچ‌ها که تابع دامنه مهاجرت هستند، باعث‌می‌شود راندمان مهاجرت به حد مطلوب محلی می‌رسد. تحت محدودیت بهترین بازده مهاجرت در کل و بدون تضاد مهاجرت، انتخاب مجموعه های متعدد از سه‌گانه بر اساس یادگیری تقویتی، به‌عنوان مهاجرت نهایی این دور برای دستیابی به تعادل بار کنترلر بهینه جهانی با حداقل هزینه. نتایج تجربی نشان‌می‌دهد که این مکانیسم می‌تواند از منابع کنترل‌کننده‌ها استفاده کامل‌کند، به سرعت بار بین کنترلرها را متعادل‌کند، سربار مهاجرت غیرضروری را کاهش‌دهد و نرخ پاسخ‌دهی سریع‌تری به درخواست‌های بسته ورودی دریافت-کند[۹].

۳- روش پیشنهادی

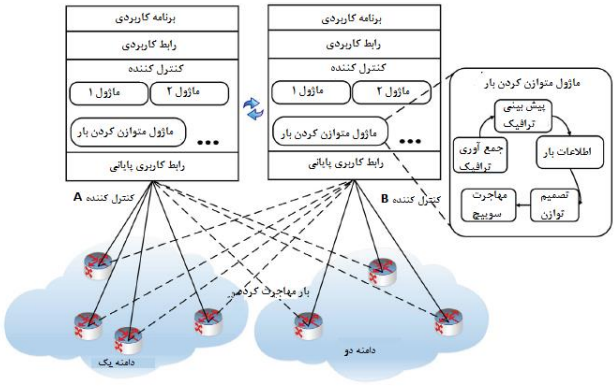
شبکه‌های IP سنتی طیف وسیعی از کاربردها را دارند، اما مدیریت آن‌ها بسیار پیچیده و دشوار است. پیکربندی شبکه بر اساس سیاست‌های از پیش تعریف‌شده و پیکربندی مجدد شبکه برای پاسخگویی به خرابی‌ها، بارها و تغییرات دشوار است. معماری شبکه های تعریف‌شده با نرم‌افزار (SDN) یکی از امیدوارکننده‌ترین معماری های اینترنت آینده درنظرگرفته‌می‌شود. به دلیل سه ویژگی جداسازی کنترل حمل و نقل، مدیریت متمرکز و یک رابط باز، تأثیر قابل‌توجهی بر شبکه سنتی دارد. این بازسازی معماری سنتی شبکه است و نقش مهمی در تحول شبکه آینده دارد. SDN منطق کنترل شبکه فعلی (صفحه کنترل) را تغییرمی‌دهد تا مسیریاب‌ها و سوئیچ‌های زیرین (سطح داده) را که ترافیک را هدایت‌می‌کنند، جداکند، بنابراین یکپارچگی عمودی را شکسته است. علاوه بر این، سوئیچ های شبکه به دلیل جداشدن صفحه کنترل و صفحه داده، به عنوان دستگاه‌های حمل‌ونقل ساده درنظرگرفته‌می‌شوند. منطق کنترل در یک کنترل‌کننده منطقی متمرکز (یا سیستم عامل شبکه) پیاده‌سازی‌می‌شود، در نتیجه عملیاتی مانند اجرای خط‌مشی و پیکربندی شبکه را ساده‌می‌کند[۱۰].

در این مطالعه، تمرکز ما طراحی یک الگوریتم مهاجرت سوئیچ دو وزنه برای حل عملیات مهاجرت مکرر سوئیچ است. این فناوری جدید برای پیش‌بینی داده‌های بار آینده از طریق داده‌های بار تاریخی کنترلر ریو، و دانستن زمانی است که بار بالای کنترل‌کننده از قبل ظاهرمی‌شود. به‌طور خلاصه، کمک‌های اصلی این کار عبارت است از: پیش‌بینی داده‌های بار آینده بر اساس داده‌های بار تاریخی کنترل‌کننده و دانستن زمانی که بار بالای کنترل‌کننده از قبل ظاهرمی‌شود. همچنین از یک استراتژی همگام‌سازی کنترل‌کننده‌های ریو راه‌اندازی استفاده-می‌کنیم تا پردازش اضافی و سربار ارتباطی صفحه کنترل را کاهش‌دهیم که برای اطلاعات بار فعال دوره‌ای لازم است. در نهایت یک طرح مهاجرت سوئیچ با وزن دوگانه برای درنظرگرفتن وضعیت بار سوئیچ در آینده و جلوگیری از مهاجرت مکرر سوئیچ پیشنهادشده‌است. نوآوری این مقاله کار بر روی کنترلر ریو است که در ابتدا مفاهیم کلیدی این کنترلر اشاره‌شده‌است.

کنترلر ریو: کنترلر Ryu یک چارچوب SDN منبع باز و مبتنی بر کامپوننت است که به‌طور کامل در پایتون پیاده‌سازی‌شده

است. Ryu از پروتکل OpenFlow برای ارتباط با سوئیچ‌ها استفاده می‌کند تا نحوه مدیریت شبکه جریان‌های ترافیک را تغییردهد و به یک الگوی برنامه‌نویسی رویداد محور اجازه می‌دهد که در آن جریان برنامه توسط رویدادها تعیین‌شود. ماژولی به نام ryu.controller.ofp_event کلاس‌های رویداد را صادرمی‌کند که دریافت پیام‌ها از سوییچ‌های متصل را توصیف‌می‌کند. Ryu مؤلفه‌های نرم‌افزاری را باAPI های کاملاً تعریف-شده ارائه‌می‌کند که ایجاد برنامه‌های کنترلی و مدیریت شبکه SDN را ساده‌می‌کند. علاوه بر این، Ryu از پروتکل‌های مختلفی برای مدیریت زیرساخت شبکه مانندOpenFlow ، Netconf (RFC 6241)، OF-config و غیره پشتیبانی می‌کند. هدف از این پروتکل‌ها جمع‌آوری اطلاعات شبکه با استفاده از کنترلر، انجام تجزیه و تحلیل و همگام‌سازی قوانین جدید شبکه است. کنترلر از APIهای NBI مانند - REST/RESTful، REST/RPC API های تعریف‌شده توسط کاربر و غیره استفاده‌می‌کند. Ryu مجموعه‌ای از اجزای خاص مانند مجازی-سازی OpenStack/Quantum ، فایروال، OFREST و غیره را برای برنامه‌های SDN فراهم‌می‌کند[۱۱].

معماری پیشنهادی: یک معماری تصمیم‌گیری توزیع‌شده در شکل ۱ پیشنهادشده‌است که در آن یک رابطه چند به چند بین کنترل‌کننده‌ها و سوئیچ ها وجوددارد. OpenFlow V1.3 یا نسخه بالاتر آن از این رابطه پشتیبانی‌می‌کند. در یک ابزار برای ارتباطات چندپخشی قابل‌اعتماد است. ارتباط و هماهنگی کنترل بر اساس JGroup است. یک مکانیسم متعادل‌کننده بار پیشنهادشده که به عنوان یک ماژول از هر کنترل‌کننده SDN در معماری اجرامی‌شود. این ماژول متعادل‌کننده بار نامیده‌می‌شود. ماژول شامل پنج بخش است: (۱) جمع آوری ترافیک برای شمارش ورود پیام های PACKET_IN سوئیچ‌های مدیریت‌شده توسط دامنه کنترل‌کننده SDN در همه زمان‌ها استفاده‌می‌شود. (۲) پیش‌بینی ترافیک استفاده از داده‌های مؤلفه جمع‌آوری ترافیک برای پیش‌بینی بار داده‌های مرحله بعدی از طریق روش LSTM است. (۳) اطلاعات بار مسئول کنترل‌کننده محلی است که اطلاعات بار خود را برای کنترلرهای دیگر ارسال‌کند. (۴) عملکرد مؤلفه تصمیم‌گیری تعادل، اتخاذ تصمیمات متعادل‌کننده بار است. (۵) مهاجرت سوئیچ وظیفه جابجایی سوئیچ‌اهای انتخاب‌شده برای متعادل‌کردن بار بین کنترلرها را بر عهده دارد[۱۰].



شکل ۱: معماری پیشنهادی تصمیم‌گیری توزیع شده کنترل‌کننده ریو[۱۰]

ماژول متعادل‌کننده بار روی هر کنترل کننده با اجرای پنج قسمت بالا برای متعادل‌کردن بار کنترل کننده همکاری می‌کند. اولاً، مؤلفه جمع‌آوری ترافیک تعداد پیام‌های PACKET_IN دریافت‌شده از سوییچ‌های موجود در دامنه را به‌صورت بلادرنگ می‌شمرد و داده‌ها را به مؤلفه پیش‌بینی ترافیک ارسال‌می‌کند. مؤلفه پیش‌بینی ترافیک از داده‌های تاریخی برای پیش‌بینی بار سوئیچ‌ها در حوزه کنترل‌کننده در مرحله بعد استفاده‌می‌کند و نتایج را به مؤلفه بعدی منتقل‌می‌کند. ثانیاً، هنگامی که کنترل‌کننده در مرحله بعدی از آستانه بار فراتر رفت، مؤلفه اطلاعات بار فعال‌می‌شود تا اطلاعات بار کنترل‌کننده را به کنترل-کننده‌های دیگر ارسال‌کند. در غیر این صورت هیچ عملیاتی انجام نمی‌شود. سپس، ماژول تصمیم‌گیری تعادل، کنترل‌کننده‌ها را به مجموعه‌ای از کنترل‌کننده‌های اضافه بار و مجموعه‌ای از کنترل‌کننده‌های بدون اضافه بار تقسیم‌می‌کند و با استفاده از الگوریتم مهاجرت سوئیچ دو وزنی پیشنهادشده در این روش، سوئیچ‌هایی را که نیاز به مهاجرت دارند پیداکرده و هدف مناسب را انتخاب‌می‌کند. کنترل-کننده ریو برای پذیرش سوئیچ‌های مجموعه انتخاب. ثالثاً، کنترلر هدف تصمیم‌می‌گیرد که آیا این سوئیچ‌ها را بپذیرد یا خیر. اگر چنین است، این سوئیچ‌ها به کنترل‌کننده‌های هدف خود مهاجرت می‌کنند. در نهایت، پس از تکمیل مهاجرت، کنترل‌کننده منبع و هدف، اطلاعات بار خود را به‌روز می‌کنند و از طریق مؤلفه اطلاعات بار به سایر کنترلرها گزارش‌می‌دهند[۱۰].

طرح پیشنهادی از پنج بخش جمع‌آوری ترافیک، پیش‌بینی ترافیک، اطلاعات بار، تصمیم‌گیری تعادل و مولفه مهاجرت سوئیچ.تشکیل‌شده‌است:

مؤلفه جمع‌آوری ترافیک: مؤلفه جمع‌آوری ترافیک روی هر کنترل‌کننده اجرامی‌شود تا اطلاعات بار را در زمان واقعی نظارت کند. در این مطالعه، نرخ رسیدن پیام‌های PACKET_IN را به عنوان شاخصی برای تصمیم‌گیری متعادل انتخاب‌می‌کنیم .یک شبکه SDN را تعریف-می‌شود که از N کنترلرها $C = \{C_1, C_2, \dots, C_n\}$ و K سویچ با $S = \{S_1, S_2, \dots, S_K\}$. این کنترلرها دامنه‌های شبکه N را مدیریت می کنند، ما فرض‌می‌کنیم که همه کنترلرها قدرت پردازش یکسانی دارند. در هر

دامنه، ترافیک شبکه و بار کاری کنترلر به صورت پویا تغییرمی‌کند. اجازه‌دهید $Q_{Ci}(S \in Q_{ci})$. فرض‌کنید $M = \max\{|Q_{C1}|, |Q_{C2}|, \dots, |Q_{CN}|\}$ W اندازه پنجره کشویی را نشان‌می‌دهد، سپس ماتریس جمع‌آوری ترافیک $M * W$ (TCM) است[۱۰].

$$TCM = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1w} \\ X_{21} & X_{22} & \cdots & X_{2w} \\ \vdots & \vdots & \ddots & \vdots \\ X_{M1} & X_{M2} & \cdots & X_{MW} \end{bmatrix}$$

مؤلفه پیش‌بینی ترافیک: مؤلفه پیش‌بینی ترافیک روی هر کنترل‌کننده اجرامی‌شود، که از داده‌های مؤلفه جمع‌آوری ترافیک برای پیش‌بینی داده‌های آینده استفاده‌می‌کند. پیش‌بینی ترافیک شبکه استفاده از داده‌های ترافیک شبکه گذشته و فعلی برای پیش‌بینی ترافیک شبکه آینده است. همان‌طور که همه ما می‌دانیم، ترافیک اینترنت مشابه است، بنابراین دقت پیش بینی بالایی دارد[۱۰]. مؤلفه پیش‌بینی ترافیک در این مقاله از شبکه عصبی بازگشتی حافظه کوتاه مدت (LSTM-RNN) استفاده‌می‌کند.

تشخیص کنترل‌کننده اضافه بار: این مؤلفه در داخل کنترلر به عنوان یک ماژول کنترلر اجرامی‌شود، بنابراین می‌تواند بررسی‌کند که آیا کنترلر در زمان واقعی بارگذاری شده‌است یا خیر. از مؤلفه پیش‌بینی بار برای به‌دست‌آوردن اثبات پیش‌بینی ترافیک سوئیچ‌ها در دامنه کنترلر استفاده‌می‌کنیم $TM_{Prediction} = (X_{1t+1}, X_{2t+1}, \dots, X_{Mt+1})$. مقادیر ماتریس پیش‌بینی ترافیک را جمع می‌کنیم تا مقدار پیش‌بینی بار کنترل‌کننده را به‌دست‌آوریم و مقدار پیش‌بینی بار را با یک آستانه معقول مقایسه‌می‌کنیم. هنگامی که به‌دست‌آمد که کنترل‌کننده از مقدار بار پیش‌بینی‌شده فراتر می‌رود، به طور فعال با کنترل‌کننده‌های دیگر ارتباط برقرارمی‌کند و اطلاعات بار را به سایر کنترلرها اطلاع می‌دهد و درخواستی برای عملیات مهاجرت سوئیچ به کنترل‌کننده‌ای که الزامات را برآورده‌می‌کند ارسال‌می‌کند. برای جلوگیری از ارسال درخواست‌های انتقال سوئیچ توسط کنترل‌کننده‌های منبع متعدد به یک کنترل‌کننده هدف است. بنابراین، اطلاعات بار کنترلر هدف را نمی‌توان به‌موقع به‌روز کرد و ممکن است کنترلر هدف بیش از حد بارگذاری‌شود. یک روش متقابل منحصر به فرد را اتخاذمی‌کنیم. هنگامی که کنترل‌کننده منبع درخواست سوئیچ را به کنترل‌کننده هدف ارسال می‌کند، کنترل‌کننده هدف به عنوان کنترل‌کننده هدف کنترل‌کننده منبع دیگر فعال‌نمی‌شود تا زمانی که کنترل‌کننده هدف درخواست را دریافت‌کند. کنترل‌کننده منبع و کنترل‌کننده هدف عملیات اطلاعات بار را راه‌اندازی‌می‌کنند و اطلاعات بار را به‌روز می‌کنند. به منظور اطمینان از همگام‌سازی اطلاعات بار کنترلر، کنترل‌کننده هدف می‌تواند درخواست‌های انتقال سوئیچ جدید را بپذیرد[۱۰].

مؤلفه تصمیم‌گیری تعادل: این مؤلفه داده‌ها را از مؤلفه پیش‌بینی بار دریافت‌می‌کند، که از استراتژی مربوطه برای تعیین بارگذاری بیش از حد کنترل‌کننده استفاده‌می‌کند. استراتژی و طرح سوئیچ پیشنهادشده در این مطالعه برای انتخاب کنترل‌کننده بارگذاری‌شده که نیاز به مهاجرت دارد به تفصیل در ادامه توضیح‌داده‌شده‌است:

یک طرح مهاجرت سوئیچ با وزن دوگانه به‌طور خاص، M حداکثر تعداد سوئیچ‌های مدیریت‌شده توسط کنترلر را نشان‌می‌دهد و W به معنای اندازه پنجره کشویی است. در ماتریس جمع‌آوری ترافیک، Xij نشان‌دهنده ترافیک ارسال‌شده توسط سوئیچ iامین در دامنه کنترل‌کننده به کنترل‌کننده در $[j, j + \Delta t]$ است، یعنی ترافیک ارسال شده توسط سوئیچ به کنترل‌کننده تعداد پیام های PACKET_IN. پیش‌بینی ماتریس ترافیک (TM) با استفاده از مدل LSTMRNN شامل دو مرحله است: مرحله آموزش(ارائه داده‌های آموزشی در لایه ورودی و تنظیم پویا پارامترهای شبکه عصبی (NN) و محقق‌شدن مقدار خروجی مورد انتظار مجموعه ورودی) و پیش‌بینی (آزمایش شبکه عصبی)[۱۰].

انتخاب سوئیچ: هنگامی که کنترل‌کننده قضاوت می‌کند که از آستانه فراتر رفته‌است، به‌طور فعال با کنترلرهای دیگر ارتباط برقرارمی‌کند، سایر کنترلرها را از اطلاعات بار مطلع‌می‌کند و درخواست انتقال سوئیچ را برای کنترل‌کننده ارسال‌می‌کند که شرایط را برآورده می‌کند. با این حال، انتخاب سوئیچ‌های موجود در دامنه به‌عنوان سوئیچ برای مهاجرت یک مشکل دشوار است. برای حل مهاجرت سوئیچ مکرر، این مطالعه روش جدیدی را پیشنهادمی‌کند که یک طرح مهاجرت سوئیچ دو وزن است. در این طرح، داده‌های پیش بینی ترافیک سوئیچ‌های دامنه را می‌توان از ماتریس پیش‌بینی به‌دست‌آورد. ما ترافیک بار سوئیچ فعلی و داده‌های ترافیکی پیش‌بینی آینده را درنظرمی‌گیریم. درنظرگرفتن ترافیک بار سوئیچ در حال حاضر می‌تواند مشکل مهاجرت سوئیچ‌های کمتر و بیشتر را حل‌کند و در نظر گرفتن جریان بار آینده می‌تواند این مشکل را حل کند که سوئیچ مهاجرت به کنترل‌کننده هدف در آینده می‌تواند منجرشود. سوئیچ کمتری وجوددارد. امکان‌پذیری امکان‌سنجی وزن سوئیچ را به صورت زیر تعریف‌می‌کنیم:

$$S_{weight} = w_1 \times X_{it} + w_2 \times X_{it+1}$$

جایی که Xit نشان‌دهنده مقدار پیام های PACKET_IN ارسال‌شده توسط سوئیچ iام در زمان $[t, t + \Delta t]$و X_{it+1} نشان‌دهنده مقدار پیش‌بینی‌شده سوئیچ iام در آن زمان $[t, t + 1 + \Delta t]$ است و w_1 و w_2 ضرایب وزنی هستند و مجموع آن‌ها ۱,۰ است. S_{weigh} استاندارد مهاجرت سوئیچ ها را نشان‌می‌دهد. برای رهاسازی هرچه سریعتر بار کنترل‌کننده اضافه بار، سوئیچ با بیشترین وزن را در اولویت قرارمی‌دهیم. اگر سوئیچ با وزن بیشتر انتخاب‌شود، بار کنترلر را می‌توان به زیر آستانه کاهش‌داد و انتخاب سوئیچ تکمیل‌می‌شود. در غیر این صورت، سوئیچ فوق الذکر و یک سوئیچ دیگر با وزن بالا به عنوان سوئیچ‌های مهاجرت‌شده و غیره گروه‌بندی می‌شوند. بنابراین، سوئیچ‌های کنترل‌کننده اضافه بار را به ترتیب نزولی با وزن دوگانه مرتب‌می‌کنیم.

مؤلفه مهاجرت سوئیچ: عملیات مهاجرت سوئیچ برای تغییر نقش کنترل‌کننده‌ای است که سوئیچ به آن تعلق دارد. در یک ابزار برای ارتباطات چندپخشی قابل‌اعتماد است. می‌تواند تغییر نقش کنترل‌کننده‌ای که سوئیچ به آن تعلق دارد را تکمیل‌کند. با این حال، هنگامی که دو یا چند کنترل‌کننده به طور همزمان اضافه بار می‌شوند، به دلیل استراتژی اطلاعات بار، هر کنترل‌کننده با بار بالا ممکن است

درخواست‌های تعادل بار را به همان کنترل‌کننده هدف ارسال‌کند. این ممکن است باعث تداخل مهاجرت و بارگذاری بیش از حد کنترلر هدف شود. به منظور اجتناب از این وضعیت، ما یک روش منحصر به فرد متقابل را اتخاذمی‌کنیم. هنگامی که کنترلر منبع یک درخواست انتقال سوئیچ را به کنترلر هدف ارسال‌می‌کند، اگر کنترلر هدف با دریافت درخواست موافقت‌کند، بلافاصله عملیات انتقال سوئیچ را تکمیل‌می‌کند و به‌طور فعال استراتژی اعلان بار را برای تکمیل همگام سازی اطلاعات بار راه‌اندازی‌می‌کند. در این زمان، کنترلر هدف می‌تواند درخواست انتقال سوئیچ جدید را بپذیرد[۱۰].

۴- شبیه‌سازی و نتایج

در این مقاله یک مکانیسم متعادل‌کننده بار بر اساس استراتژی اطلاعات بار پیشنهادشده‌است. هر کنترلر در سیستم به‌طور دوره‌ای اطلاعات بار خود را به کنترل‌کننده‌های دیگر گزارش‌می‌دهد. کنترل‌کننده‌های اضافه بار طرح، اطلاعات بار همه کنترل‌کننده‌های دیگر را قبل از تصمیم‌گیری محلی جمع‌آوری‌نمی‌کنند، که مصرف منابع کنترل‌کننده را کاهش-می‌دهد. استراتژی متعادل‌سازی بار مبتنی بر زمان پاسخ برای کنترل‌کننده‌های SDN متعدد پیشنهادشده، که ویژگی‌های زمان پاسخ بلادرنگ را که با بار کنترل‌کننده تغییرمی‌کند، درنظرمی‌گیرد. تمامی این موارد در کنترل‌کننده Ryu با انتخاب یک آستانه زمان پاسخ مناسب و پردازش چندین کنترل‌کننده اضافه بار به‌طور همزمان، مشکل تعادل بار صفحه کنترل SDN کنترل کننده اضافه بار را می‌توان به‌خوبی حل-کرد. در این قسمت به ارائه شبیه‌سازی و نتایج حاصل از آن پرداخته-می‌شود.

اولین شبیه سازی(ارزیابی عملکرد کنترلر): ما فرض می‌کنیم که قابلیت‌های پردازشی همه کنترلرها یکسان است، یعنی قابلیت‌های پردازشی همه کنترلرها را می‌توان با آزمایش قابلیت‌های پردازش یک کنترلر نشان‌داد. ما از کنترلر Ryu استفاده‌می‌کنیم که روی میزبان ۶۴ بیتی اوبونتو LTS۱۶,۰۴ اجرامی‌شود. آن‌ها دارای پیکربندی یکسان با پردازنده Core i5 اینتل ۳,۳ گیگاهرتز و ۸ گیگابایت حافظه DDR3 هستند. در شکل ۲ نمونه کد اولیه شبیه‌سازی برروی کنترلر Ryu برای ساخت توپولوژی نشان‌داده‌شده‌است. شکل ۳ کد تشکیل توپولوژی و شکل ۴ تصویر توپولوژی نشان‌داده‌شده‌است.

```
ip netns add gateway1
ip netns add gateway2
brctl addbr vrrp-br0
brctl addbr vrrp-br1
ip link add veth0 type veth peer name veth0-br0
ip link add veth1 type veth peer name veth1-br0
ip link add veth2 type veth peer name veth2-br0
ip link add veth3 type veth peer name veth3-br1
ip link add veth4 type veth peer name veth4-br1
ip link add veth5 type veth peer name veth5-br1
brctl addif vrrp-br0 veth0-br0
brctl addif vrrp-br0 veth1-br0
brctl addif vrrp-br0 veth2-br0
brctl addif vrrp-br1 veth3-br1
brctl addif vrrp-br1 veth4-br1
brctl addif vrrp-br1 veth5-br1
ip link set vrrp-br0 up
ip link set vrrp-br1 up
```

```
ip link set veth0 up
ip link set veth0-br0 up
ip link set veth1-br0 up
ip link set veth2-br0 up
ip link set veth3-br1 up
ip link set veth4-br1 up
ip link set veth5 up
ip link set veth5-br1 up
ip link set veth1 netns gateway1
ip link set veth2 netns gateway2
ip link set veth3 netns gateway1
ip link set veth4 netns gateway2
ip netns exec gateway1 ip link set veth1 up
ip netns exec gateway2 ip link set veth2 up
ip netns exec gateway1 ip link set veth3 up
ip netns exec gateway2 ip link set veth4 up
ip netns exec gateway1 .ryu-vrrp veth1 '10.0.0.2' 254
ip netns exec gateway2 .ryu-vrrp veth2 '10.0.0.3' 100
```

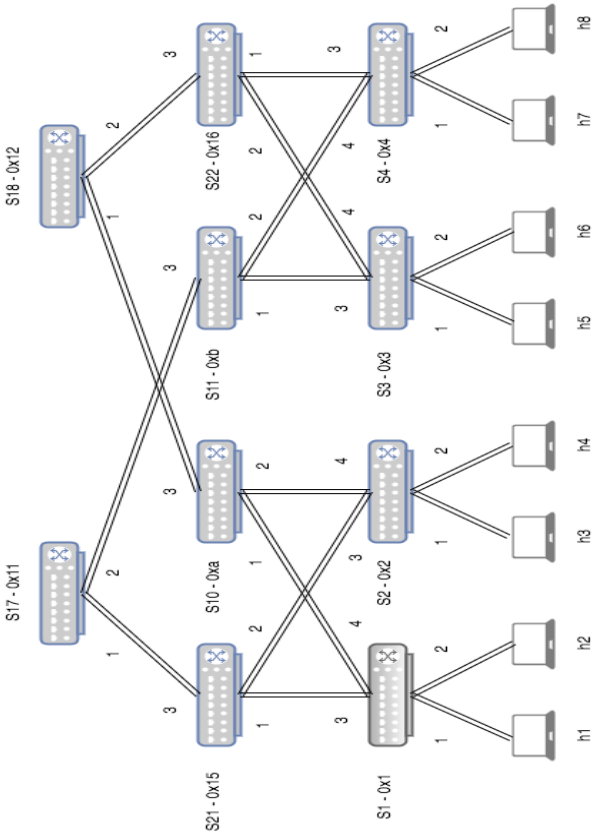
شکل ۲: خطوط فرمان پل‌ها و تنظیم رابط‌های لازم

```
from ryu.lib import hub
hub.patch()
import ryu.contrib
from oslo.config import cfg
import logging
import netaddr
import sys
import time
from ryu import log
log.early_init_log(logging.DEBUG(
from ryu import flags
from ryu import version
from ryu.base import app_manager
from ryu.controller import controller
from ryu.lib import mac as lib_mac
from ryu.lib.packet import vrrp
from ryu.services.protocols.vrrp import api as vrrp_api
from ryu.services.protocols.vrrp import event as vrrp_event
class VRRPTestRouter(app_manager.RyuApp:(
def __init__(self, *args, **kwargs:(
super(VRRPTestRouter, self).__init__(*args, **kwargs(
print args
self.logger.debug('vrrp_config %s', args(
self._ifname = args[0[
self._primary_ip_address = args[1[
self._priority = int(args[2([
def start(self:(
print 'start'
hub.spawn(self._main(
def _main(self:(
print self
interface = vrrp_event.VRRPInterfaceNetworkDevice)
lib_mac.DONTCARE, self._primary_ip_address, None, self._ifname(
self.logger.debug('%s', interface(
ip_addresses = [_IP_ADDRESS[
config = vrrp_event.VRRPConfig)
version=vrrp.VRRP_VERSION_V3, vrid=_VRID, priority=self._priority,
ip_addresses=ip_addresses(
self.logger.debug('%s', config(
rep = vrrp_api.vrrp_config(self, interface, config(
self.logger.debug('%s', rep(
def main:()
vrrp_config = sys.argv[-3[:
sys.argv = sys.argv[:-3[
CONF(project='ryu', version='ryu-vrrp %s' % version(
log.init_log()
 #always enable ofp for now.
app_lists = ['ryu.services.protocols.vrrp.manager,'
'ryu.services.protocols.vrrp.dumper,'
'ryu.services.protocols.vrrp.sample_manager['
app_mgr = app_manager.AppManager.get_instance()
app_mgr.load_apps(app_lists(
contexts = app_mgr.create_contexts()
app_mgr.instantiate_apps(**contexts(
vrrp_router = app_mgr.instantiate(VRRPTestRouter, *vrrp_config, **contexts(
vrrp_router.start()
while True:
time.sleep(999999)
app_mgr.close()
if __name__ == "__main:"__
main()
```

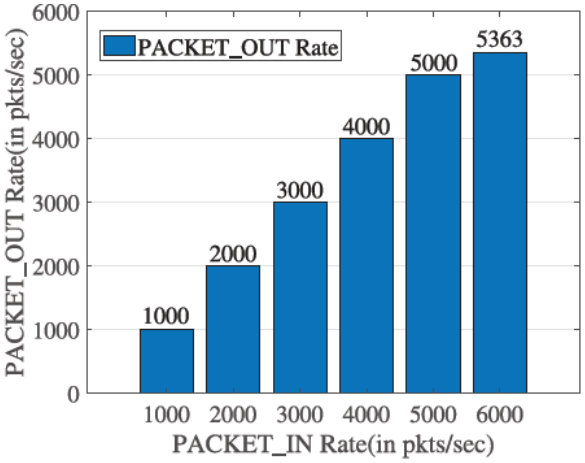
شکل ۳: کد اجرای Ryu خطوط فرمان پل‌ها و تنظیم رابط‌های لازم

ابزار تستی که ما استفاده‌می‌کنیم OFsuite_performance است. به عنوان یک عضو جدید "OFsuite_Performance" در مجموعه ابزار OFsuite که به‌طور مستقل توسط مرکز تست و صدور گواهینامه جهانی SDN توسعه‌یافته‌است، به پشتیبانی از تست عملکرد کنترلر اختصاص‌دارد. OFsuite_Performance می‌تواند تعداد زیادی سوئیچ OpenFlow V1.3 را در یک سرور لینوکس عمومی شبیه‌سازی‌کند و

می‌تواند توپولوژی های مختلف شبکه و همه رویدادهای SDN را شبیه سازی‌کند.

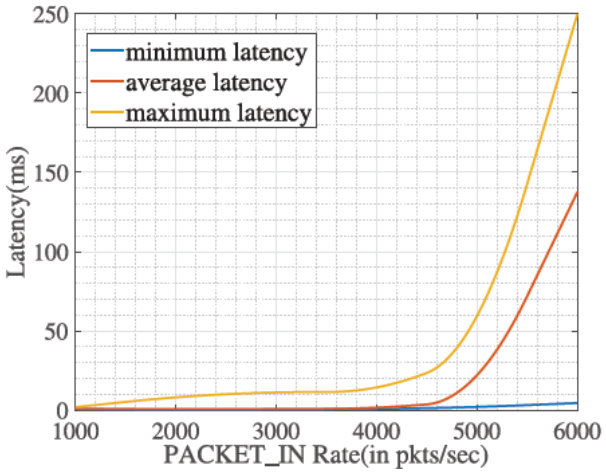


شکل ۴: توپولوژی استفاده‌شده در شبیه‌سازی



شکل ۵: ارتباط پیام PACKET_OUT و ارتباط دریافت پیام PACKET_IN

ما نتیجه را در شکل ۵ با متوسط‌کردن داده‌های رابطه بین نرخ PACKET_IN و نرخ PACKET_OUT برای ۲۰ بار به‌دست می‌آوریم. هنگامی که نرخ بارگذاری PACKET_IN تنظیم‌شده توسط ابزار تست کمتر از ۵۰۰۰ بسته در ثانیه است، کنترل‌کننده تحت آزمایش می‌تواند پیام PACKET_IN ارسال‌شده را کنترل‌کند و سرعت تحویل PACKET_OUT را می‌توان با سرعت PACKET_IN مقایسه‌کرد. اما زمانی که نرخ PACKET_IN بیشتر از ۵۰۰۰ بسته در ثانیه است، مثلاً ۶۰۰۰ بسته در ثانیه، کنترل‌کننده تحت آزمایش نمی‌تواند PACKET_IN دریافتی را به‌طور کامل پردازش‌کند و نرخ صدور PACKET_OUT نیز کمتر از نرخ PACKET_IN است. از این منظر، آزمایش به حداکثر نرخ تحویل PACKET_OUT کنترلر آزمایش‌شده رسیده‌است. این حداکثر ظرفیت پردازش حداکثر کنترلر است.



شکل ۶: تغییر منحنی زمان پاسخ به بار کنترلر

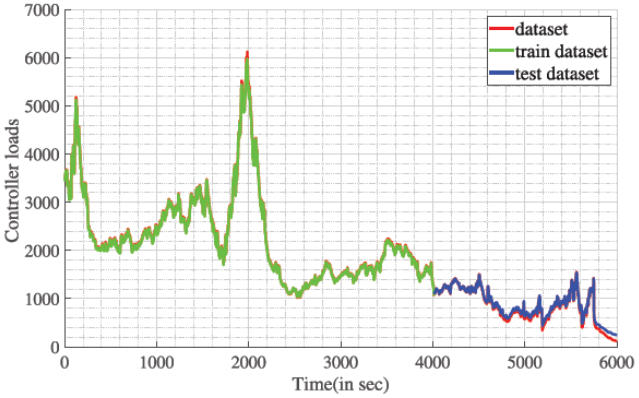
شکل ۶ رابطه بین تأخیر زمانی کنترل‌کننده پردازش PACKET_IN و نرخ رسیدن پیام PACKET_IN را نشان‌می‌دهد. جدول ۱ داده‌های خاص را نشان‌می‌دهد. داده‌ها با میانگین گیری چندین بار برای اطمینان از قابلیت اطمینان داده‌ها و عبور از حداکثر تأخیر به‌دست‌می‌آیند. هنگامی که نرخ رسیدن پیام PACKET_IN 3000 بسته در ثانیه یا ۴۰۰۰ بسته در ثانیه است، تأخیر تقریباً یکسان است، اما زمانی که نرخ رسیدن ۵۰۰۰ بسته در ثانیه باشد، زمان تأخیر غیرقابل-قبول خواهدبود. این مطالعه از روش [۱۲] استفاده‌می‌کند و از زمان پاسخ برای تعیین آستانه معقول استفاده‌می‌کند. از طریق تصویر تابع ورود پیام PACKET_IN و زمان، طرح از مشتق دوم برای یافتن آستانه معقول استفاده‌می‌کند.

جدول ۱: زمان تأخیر نرخ PACKET_IN

PACKET_ In Rate (in pkts/s)	Minimum latency (ms)	Average latency (ms)	Maximum latency (ms)
1000	0.70996	0.73175	1.85693
2000	0.01807	0.60239	8.448
3000	0.35498	0.51537	11.55786
4000	0.63306	0.96216	11.22705
5000	1.83276	6.26612	35.69214
6000	4.44605	137.5306	249.78613

ارزیابی مدل پیش‌بینی ترافیک: آموزش یادگیری ماشینی به داده‌های زیادی نیاز دارد. اگرچه SDN می‌تواند داده‌های آموزشی زیادی ارائه‌دهد، اما برای جمع‌آوری حجم زیادی از داده‌ها به‌عنوان آموزش نیاز به کار زیادی دارد. از آنجاکه حالت ارتباطی سوئیچ/کنترل‌کننده مشابه حالت معماری سنتی Client-Server (C/S) است، این مطالعه استفاده از زبان برنامه نویسی جاوا را برای تحقق معماری C/S پیشنهادمی‌کند، سرور کنترل‌کننده را شبیه‌سازی می‌کند و مشتری سوئیچ ها را شبیه

سازی می کند. . ما چندین سوئیچ را از طریق چندین کلاینت شبیه-سازی می کنیم، پیام‌ها را به‌طور تصادفی از طریق کلاینت‌ها به سرور ارسال‌می‌کنیم و سپس از مهندسی ویژگی برای استخراج داده‌ها برای آموزش استفاده‌می‌کنیم.

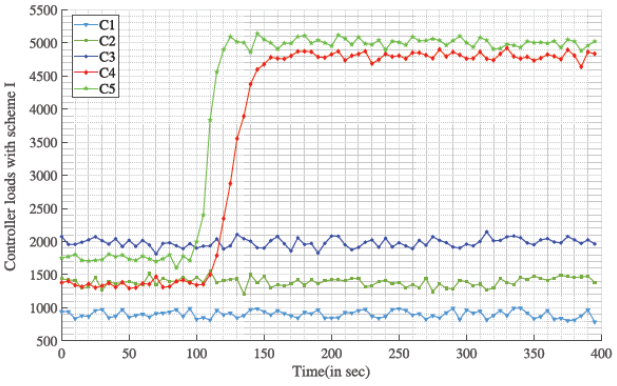


شکل ۷: توزیع بار کنترل‌کننده در طول زمان تغییرمی‌کند

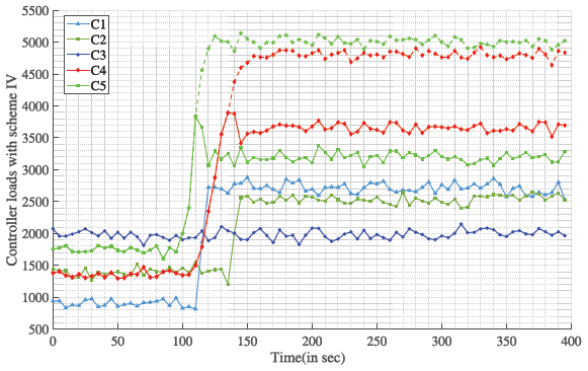
ما از چارچوب TensorFlow استفاده‌می‌کنیم و شکل ۷ نتایج ما را نشان-می‌دهد. ما با ۲/۳ تمری می‌کنیم و با ۱/۳ از داده‌ها تست می‌کنیم. شکل نشان‌می‌دهد که مدل شبکه می‌تواند به‌خوبی با داده‌ها مطابقت داشته-باشد و تابع ضرر تعریف‌شده $mean\_squ$ تلفات را می‌توان به 10^{-5} کاهش‌داد، که نشان‌می‌دهد مدل به‌خوبی می‌تواند با داده‌های شبکه مطابقت داشته‌باشد. ما داده‌ها و کدهای مربوطه را ارائه‌کرده‌ایم.

ارزیابی قابلیت تعادل بار: در ادامه این بخش، ما عمدتاً چهار طرح را با هم مقایسه‌می‌کنیم: طرح I، مدل نگاشت کنترل‌کننده سوئیچ‌های استاتیک. طرح دوم، با استفاده از روش طرح III، SMCLBRT و طرح IV، طرح ارائه‌شد. ما همان محیط آزمایشی را شبیه‌سازی‌می‌کنیم و هر شبیه‌سازی برای ۴۰۰ ثانیه اجرامی‌شود. با انتخاب دو کنترلر و افزایش مداوم درخواست بسته سوئیچ‌ها، تغییرات توزیع بار را تحت طرح های مختلف مشاهده‌می‌کنیم.

ابتدا، ما طرح IV را با طرح I مقایسه‌می‌کنیم تا اثربخشی مطالعه خود را تأییدکنیم. از طریق اجرای آزمایش‌های شبیه‌سازی، تغییرات بار را در دو حالت ثبت‌می‌کنیم. یکی نگاشت سوئیچینگ استاتیک است، دیگری از طرح موجود در روش پیشنهادی استفاده می-کند. شکل ۸- الف و ب به ترتیب توزیع بار دو مورد را نشان‌می‌دهند. ما شروع به افزایش بار کنترلرهای $C4$ و ۵ در ۱۰۰ ثانیه می‌کنیم. $C4$ و ۵ به ترتیب در ۱۱۵ ثانیه و ۱۴۰ ثانیه از ظرفیت پردازش کنترلر فراتر می‌روند و همچنان بارگذاری‌می‌شوند. در مورد تعادل بدون بار در شکل ۸ (الف)، زمانی که $C4$ و ۵ از مقدار آستانه کنترلر تجاوزکنند، زمان پاسخگویی به بیش از ۱۵ میلی‌ثانیه می‌رسد و بارگذاری اضافه‌می شود. با این حال، همان‌طور که در شکل ۸ (ب) نشان‌داده‌شده‌است، روش پیشنهادی طرح پیش‌بینی ترافیک را اتخاذ می‌کند. زمانی که پیش‌بینی می‌شود بار در مرحله بعد از آستانه فراتر رود، عملیات متعادل‌سازی بار از قبل انجام‌می‌شود که می‌تواند بار $C4$ و ۵ را به ۱ و ۲ با بار کم منتقل کند. بار $C4$ و ۵ به‌طور مؤثر کاهش‌می‌یابد و زمان پاسخ به میزان زیادی کاهش‌می‌یابد.



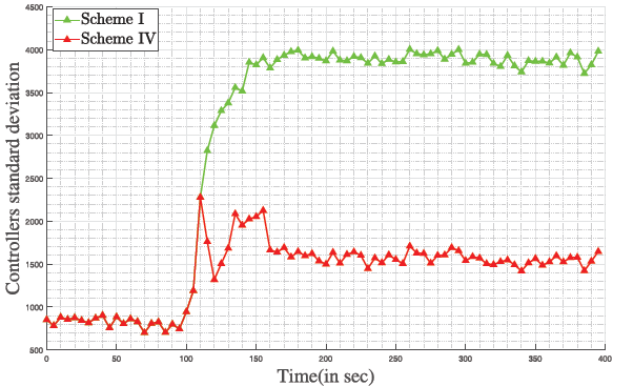
الف



ب

شکل ۸: توزیع بار پنج کنترلر. (الف) تحت نگاشت کنترل‌کننده سوئیچ استاتیک. (ب) تحت این طرح کاغذی

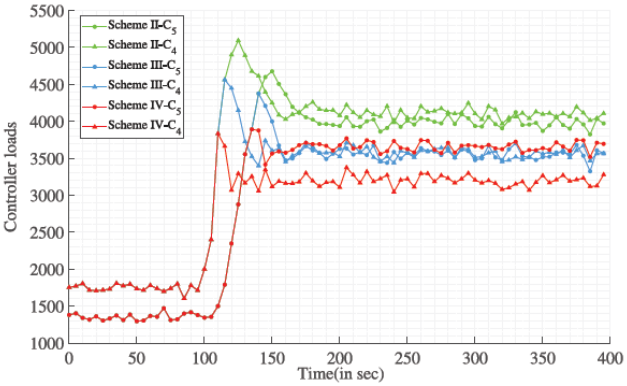
در عین حال، تغییر انحراف استاندارد کنترلرها را در طول زمان در شکل ۹ نشان‌می‌دهیم. مشاهده‌می‌شود که بار کنترل‌کننده به تدریج متعادل-می‌شود.



شکل ۹: انحراف استاندارد کنترل‌کننده ها در طول زمان تغییرمی‌کند

زمان تعادل: شکل ۱۰ توزیع بار کنترل‌کننده اضافه بار را قبل و بعد از عملیات متعادل‌سازی بار تحت سه طرح متعادل‌کننده بار نشان-می‌دهد. تغییر بار کنترلر $C5$ را درنظرمی‌گیریم، زمان شروع متعادل-سازی بار این طرح ۱۱۰ ثانیه و زمان شروع متعادل‌سازی بار طرح II و طرح III به ترتیب ۱۲۵ ثانیه و ۱۱۵ ثانیه است. سپس زمان پایان متعادل‌سازی بار در مطالعه ما ۱۱۸ ثانیه است، درحالی‌که زمان پایان

طرح II و طرح III به ترتیب ۱۴۸ ثانیه و ۱۳۰ ثانیه است. از شکل ۴-۸ می‌توان‌دریافت که کنترلر ۴ نیز نتیجه مشابهی دارد.



شکل ۱۰: تغییرات بار کنترل کننده‌های اضافه بار

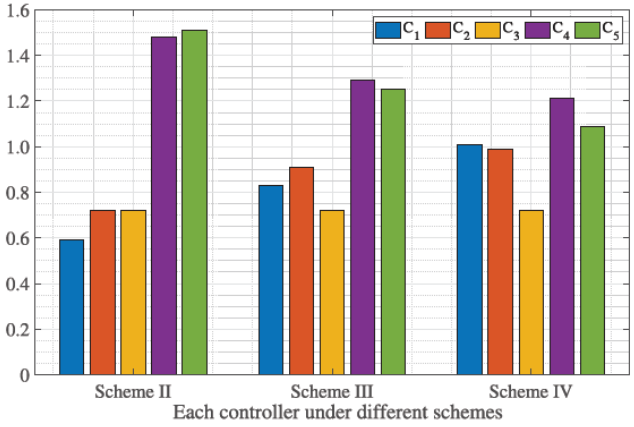
از آنجاکه زمان انتقال سوئیچ تحت تأثیر فشار بار فعلی کنترل-کننده اصلی آن قرارمی‌گیرد، با فرض اطمینان از ضرورت، هر چه زودتر سوئیچ جابجا شود، زمان انتقال کمتر مصرف‌می‌شود و می توان زمان خاموشی سیستم را کوتاه کرد. . از نتایج نشان‌داده‌شده در شکل، طرح ما می‌تواند توزیع بار ناهموار را از قبل متعادل‌کند، و زمان کمتری را برای مقابله با چندین کنترل‌کننده اضافه بار مصرف می‌کنیم. از دیدگاه کاربر، طرح ما می‌تواند سرعت پاسخ دو کنترلر اضافه بار را زودتر و سریعتر کاهش‌دهد، در نتیجه کیفیت تجربه بهتر (QoE) را تضمین‌می کند. طرح ما می‌تواند عملیات متعادل‌سازی بار را از قبل شروع‌کند، زیرا ما از فناوری پیش‌بینی ترافیک برای اطلاع از زمان رسیدن بارهای بالا قبل از زمان‌بندی برای عملیات متعادل‌سازی بار استفاده‌می‌کنیم. از سوی دیگر، در فرآیند متعادل سازی، از الگوریتم مهاجرت سوئیچ دو وزنه استفاده‌می‌کنیم که به‌راحتی می تواند به حالت بار نرمال تمایل پیداکند و فرکانس مهاجرت سوئیچینگ را کاهش‌دهد. سپس، در معماری SDN توزیع‌شده، از استراتژی اطلاعات بار راه‌اندازی استفاده-می‌کنیم. هنگامی که چندین کنترل‌کننده از بار فراتر رود، استراتژی اطلاعات بار به‌طور فعال آغازمی‌شود و از اصل انتخاب متقابل انحصاری کنترل‌کننده هدف استفاده‌می‌شود. برای درک پردازش موازی چندین کنترل‌کننده اضافه بار.

توزیع بار: با افزایش مداوم بسته‌های داده پیام درخواست سوئیچ، مطالعات مختلف می توانند به یک اثر متعادل‌کننده بار بهتر دست‌یابند و به حالت بار پایدار دست‌یابند. مقادیر متوسط بار کنترل کننده‌های مختلف تحت سایر طرح‌ها تحت شرایط بار پایدار اندازه‌گیری می‌شود. اجازه‌دهید نسبت نشان‌دهنده نسبت توزیع بار باشد، همان‌طور که در معادله نشان‌داده‌شده‌است. (۳). می‌تواند به‌طور تقریبی نرخ تعادل را تخمین‌بزند.

$$ratio = \frac{Load_{Ci}}{\overline{Load}} \qquad (۱-۴)$$

که در آن $LoadCi$ نشان‌دهنده حجم کار کنترلر Ci، و $Load$ نشان‌دهنده میانگین حجم کار پنج کنترل‌کننده است.

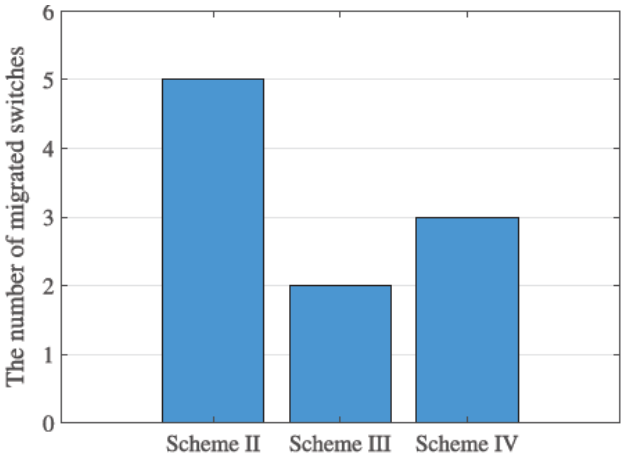
ما نسبت توزیع بار را در طرح‌های مختلف مقایسه‌می‌کنیم، همان‌طور که در شکل ۱۱ نشان‌داده‌شده‌است. طرح ما نرخ بارگذاری بهتری نسبت به سایر طرح‌ها دارد. دلایل اصلی به شرح زیر توضیح‌داده-شده‌است. روش مبتنی بر پیش‌بینی ترافیک می‌تواند به ما کمک کند تا کنترل‌کننده اضافه بار را زودتر از موعد مقرر پیداکرده و با آن مقابله‌کنیم. در فرآیند افزایش پیام‌های درخواست سوئیچ، طرح ما از یک الگوریتم مهاجرت سوئیچ با وزن دوگانه استفاده‌می‌کند، با در نظر گرفتن بار فعلی و بار آینده سوئیچ و انتخاب یک سوئیچ یا گروه سوئیچ‌ها مناسب برای مهاجرت به کنترل‌کننده هدف. هنگام انتخاب کنترل‌کننده هدف، اعمال محدودیت‌های ما همچنین می‌تواند تضمین‌کند که کنترل‌کننده هدف پس از مهاجرت سوئیچ در حالت بار بالا قرارنخواهدگرفت.



شکل ۱۱: نرخ تعادل طرح های مختلف

شماره سوئیچ مهاجرت‌شده: ابزار اصلی برای چندین کنترلر SDN برای دستیابی به تعادل بار، مهاجرت سوئیچ است. اگر سوئیچ های بیشتری در یک دوره زمانی منتقل‌شوند، ممکن است خدمات شبکه عادی به‌طور جدی تحت تأثیر قرار گیرند. ما تعداد سوئیچ‌هایی را که در طول اضافه بار متعادل‌سازی بار در هر حالت مهاجرت‌کرده‌اند، مقایسه-کردیم.

همان‌طور که در شکل ۱۲ نشان‌داده‌شده‌است، در سه طرح متعادل‌کننده بار، تعداد کلیدهایی که پس از متعادل‌سازی بار در طرح III مهاجرت‌کرده‌اند، کمترین میزان را دارد. دلیل اصلی این است که طرح III همیشه سوئیچ‌هایی را انتخاب‌می‌کند که بیشترین تأثیر را بر بار کنترلر دارند، بنابراین تعداد سوئیچ‌های مهاجرت‌شده کمترین است. طرح ما پس از متعادل‌کردن بار، سوئیچ‌های کمتری را منتقل‌می‌کند. در فرآیند تصمیم‌گیری مهاجرت، ما یک الگوریتم مهاجرت سوئیچ دو وزنه را برای محاسبه وزن هر سوئیچ در دامنه کنترل‌کننده اضافه بار اتخاذمی‌کنیم و سپس سوئیچ یا گروه سوئیچ را با بیشترین مقدار وزن برای مهاجرت انتخاب‌می‌کنیم.



شکل ۱۲: تعداد سوئیچ های منتقل-شده

همه سوئیچ ها به‌طور مستقیم به هر کنترلر در این محیط شبکه متصل هستند اما فقط توسط کنترلر اصلی مدیریت‌می‌شوند. فرآیند مهاجرت سوئیچ برای اطمینان از فعالیت، امنیت و سریال‌سازی نیاز به تغییر زمان ارتباط کنترلر و زمان مهاجرت دارد. زمان تبادل پیام در طول هر فرآیند مهاجرت تقریباً یکسان است. ما فرض‌می‌کنیم که سربار ارتباط و زمان مهاجرت یک سوئیچ از یک کنترلر به کنترلر دیگر C و T است. اجازه‌دهید Num تعداد سوئیچ هایی را که پس از تکمیل تعادل بار منتقل‌شده‌اند نشان‌دهد. بنابراین، می‌توانیم از C*Num و T*Num برای توصیف سربار ارتباط و زمان مهاجرت استفاده‌کنیم. زمان مهاجرت سوئیچ تحت تاثیر فشار بار فعلی کنترل کننده اصلی آن است. با فرض اطمینان از ضرورت، هر چه زودتر سوئیچ جابجا شود، زمان انتقال کمتر مصرف‌می‌شود و زمان از کار افتادن سیستم را می‌توان کوتاه‌کرد. این طرح بر اساس حداقل زمان مهاجرت و مهاجرت‌های سوئیچ نسبتاً کمی است که باعث کاهش سربار ارتباط و زمان مهاجرت در فرآیند کلی تعادل می‌شود. روش پیشنهادی تعداد ۲ پکت گم شه منجر به مهاجرت دارد.

```
Simulation complete.
Packet loss rate at node 5 = 0.030303
Packet loss rate at node 6 = 0.010101
```

شکل ۱۳: پکت‌های گم‌شده روش پیشنهادی

با توجه به نتایج حاصل در شکل ۱۳روش پیشنهادی کمترین تعداد مهاجرت را براساس کارایی داشته‌است.

۵- نتیجه گیری

شبکه‌های تعریف‌شده با نرم‌افزار به دلیل مزایایی مانند قابلیت برنامه‌ریزی و مدیریت متمرکز، یکی از امیدوارکننده‌ترین حالت های توسعه اینترنت آینده درنظرگرفته‌می‌شود. یک کنترلر متمرکز منفرد ممکن است باعث مشکلات قابلیت اطمینان و مقیاس پذیری شود. اگرچه چندین کنترلر می توانند مشکلات مقیاس‌پذیری و قابلیت اطمینان یک کنترلر متمرکز را حل‌کنند، یک مکانیسم انعطاف‌پذیر برای متعادل-کردن بار مورد نیاز است. بارهای ترافیکی بین کنترلرها به راحتی می تواند منجر به توزیع نامتعادل بار بین آن‌ها شود. برای کنترلرهای توزیع‌شده چندگانه، یک طرح مهاجرت سوئیچ دو وزنی متعادل‌کننده بار SDN مبتنی بر پیش‌بینی پیشنهادشده‌است. این طرح بار ترافیک گذشته را به عنوان داده های تاریخی برای پیش‌بینی بار ترافیک آینده درنظرمی‌گیرد.

در حال حاضر، کار تحقیقاتی استراتژی متعادل‌سازی بار چند کنترل‌کننده عمدتاً به دو نوع مدل تقسیم‌می‌شود: تصمیم‌گیری متمرکز و تصمیم‌گیری توزیع‌شده. در یک معماری متمرکز، کنترلر ریشه برخی از کنترل کننده‌های محلی مسئول سوئیچ‌ها را مدیریت می کند. سیستم شامل یک کنترلر هماهنگ‌کننده برای مدیریت جدول اطلاعات بارگذاری کنترل‌کننده‌های جهانی است. از طریق جدول اطلاعات بار، کنترلر تصمیم‌می‌گیرد که آیا عملیات متعادل‌سازی بار را انجام‌دهد یا خیر. تصمیم‌گیری متمرکز می‌تواند مشکل توزیع بار نابرابر را در بین چندین کنترلر حل‌کند. با این حال، آن‌ها دو مرحله اساسی دارند: جمع-آوری اطلاعات بار تمام کنترلرها در زمان واقعی و ارسال دستورهای متعادل‌کننده بار به کنترلرهای بارگذاری‌شده. هر کنترل‌کننده مسئول برخی از سوئیچ‌ها در یک معماری توزیع‌شده‌است و هر کنترل‌کننده اطلاعات را با کنترلرهای دیگر مبادله‌می‌کند، کنترل‌کننده‌ها را به‌طور مسطح سازمان‌دهی‌می‌کند و تصمیم را به یک کنترل‌کننده محلی-می‌دهد و زمان تنظیم جدول جریان را به حداقل می‌رساند. دو مشکل، یعنی پیکربندی چند کنترلر و تعادل ترافیک کنترل‌کننده با روش پیشنهادی حل‌می‌شود. به صورت پویا بار کنترل‌کننده ریشه را با توجه به بار فعلی هر کنترل‌کننده محلی تنظیم‌می‌کند.

از طریق فناوری پیش‌بینی، ما زمان بارگذاری بیش از حد کنترلر را می‌دانیم، به‌طوری که عملیات مهاجرت سوئیچ می‌تواند از قبل انجام‌شود. ما همچنین یک الگوریتم اطلاعات بار راه‌اندازی را برای حل پردازش اضافی و سربار ارتباطی صفحه کنترل مورد نیاز برای اطلاعات بار فعال دوره‌ای بین کنترل‌کننده‌های توزیع‌شده پیشنهادمی‌کنیم. با توجه به اطلاعات گذشته، طرح پیشنهادی پیشنهادمی‌کند که مدیریت سوئیچ‌های خاص بین کنترل‌کننده‌ها منتقل‌شود. ما بار تاریخی و بار آینده سوئیچ ها را درنظرمی‌گیریم و یک الگوریتم مهاجرت سوئیچ با وزن دوگانه پیشنهادمی‌کنیم که فرکانس مهاجرت سوئیچ را کاهش‌می‌دهد. آزمایش‌ها ثابت‌کرده‌اند که این طرح می‌تواند به سرعت بار بین کنترل‌کننده‌ها را متعادل‌کند و تعداد مهاجرت سوئیچ‌ها را کاهش‌دهد. نتایج شبیه‌سازی نشان‌می‌دهد که این طرح می‌تواند کنترل‌کننده اضافه بار را از قبل مهاجرت‌کند و به سرعت بار کاری کنترل‌کننده اضافه بار را کاهش‌دهد. بنابراین، طرح پیشنهادی می‌تواند به‌سرعت و به‌طور مؤثر به تعادل بار چندین کنترل‌کننده SDN دست‌یابد.

مراجع

1. Rana, D.S., S.A. Dhondiyal, and S.K. Chamoli, *Software defined networking (SDN) challenges, issues and solution*. International journal of computer sciences and engineering, 2019. **7**(1): p. 884-889.
2. Mohammadi, H. and S. Mostafavi, *A Prediction-Based Load Distribution Approach for Software-Defined Networks*. Nashriyyah -i Muhandisi -i Barq va

Muhandisi -i Kampyutar -i Iran, 2022. 4(19): p. 289-301.
3. Srivastava, V. and R.S. Pandey, *Load balancing for software-defined network: a review.* International Journal of Computers and Applications, 2021: p. 1-14.
4. Deng, X., et al. *Efficient SDN Controller Load Balancing Using Deep Reinforcement Learning*. in *2021 4th International Conference on Hot Information-Centric Networking (HotICN)*. 2021. IEEE.
5. Yeo, S ,.et al., *Achieving Balanced Load Distribution with Reinforcement Learning-Based Switch Migration in Distributed SDN Controllers.* Electronics, 2021. **10**(2): p. 162.
6. Malbašić, T., et al., *Hybrid SDN Networks: A multi-parameter server load balancing scheme* .Journal of Network and Systems Management, 2022. 30(2): p. 1-28.
7. Salehi, M. and M.A. Irandoost, *A Learning Automata Approach for Load Balancing in Software-Defined Networks.* International Journal of Smart Electrical Engineering, 2022.
8. Liu, Y., et al. *CASM: A Cost-Aware Switch Migration Strategy for Elastic Optical Inter-Datacenter Networks*. in *Photonics*. 2022. Multidisciplinary Digital Publishing Institute.
9. Li, Z., et al. *SDN controller load balancing based on reinforcement learning*. in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*. 2018. IEEE.
10. Zhong, H., et al., *Prediction-based dual-weight switch migration scheme for SDN load balancing.* Computer Networks, 2022: p. 108749.
11. Cabarkapa, D. and D. Rancic, *Performance Analysis of Ryu-POX Controller in Different Tree-Based SDN Topologies.* Advances in Electrical and Computer Engineering, 2021. **21**(3): p. 31-38.
12. J. Cui, Q.L., H. Zhong, M. Tian, L. Liu, *A load-balancing mechanism for distributed SDN control plane using response time.* IEEE Trans. Netw. Serv.Manag. 15 (4) (2018) 1197–1206., 2018.