

A novel approach based on gray wolf evolutionary algorithm for controller load balancing in software defined networks using dynamic switch migration

Saeid Bakhtiari^{1*}, Vosough Guran Ourimi²

1- Department of Fata, Faculty of Cyber, Amin University, Tehran, Iran.

2* - Department of Computer, Faculty of Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran.

^{1*}s.b.fata@police.ir, ²vosugh.guran@gmail.com

*Corresponding author's address: Saeid Bakhtiari, Amin University, Tehran, Iran.

Abstract- Software Defined Network (SDN) has become a popular model for centralized control and management in many modern network scenarios. However, for large data centers with hundreds of thousands of servers and several thousand switches, a single controller mode causes the system to suffer from lack of scalability and reliability. The use of distributed architecture can improve system performance, but the main limitation of this work is the static mapping between the switch and the controller, which may lead to load imbalances in the controllers. The use of multiple controllers distributed in SDN has been used to improve scalability and reliability, where each controller manages a fixed partition of the network. In the proposed method, a balanced controller (BalCon) is used as a scheme to dynamic migrate the SDN switch to achieve load balance between SDN controllers with low migration cost. The decision to migrate is made by the monitor based on network traffic. To balance load in the software-based network, a staged switch migration strategy is used and the target controller is selected by the gray wolf algorithm. The results show that the response time of the proposed method is 13% better than the BalCon method. Also, by increasing the sending rate to 8.33%, the controller's throughput improves compared to the BalCon method. Therefore, the load balance of the proposed method has a better performance than previous works.

Keywords: Software Defined Network, Switch Migration, Load Balancing, Gray Wolf Algorithm

ارائه روشی نوین مبتنی بر الگوریتم تکاملی گرگ خاکستری برای توازن بار کنترل کننده در شبکه‌های مبتنی بر نرم‌افزار با استفاده از مهاجرت پویای سوئیچ

سعید بختیاری^{۱*}، وثوق گوران اوریمی^۲

۱ - گروه فتا، دانشکده سایبری، دانشگاه امین، تهران، ایران.

۲* - گروه کامپیوتر، دانشکده فنی و مهندسی، دانشگاه آزاد اسلامی واحد تهران مرکزی، تهران، ایران.

¹*s.b.fata@police.ir, ²vosugh.guran@gmail.com

* نشانی نویسنده مسئول: سعید بختیاری، تهران، دانشگاه پلیس.

چکیده - شبکه مبتنی بر نرم‌افزار (SDN) در بسیاری از سناریوهای مدرن شبکه به مدلی محبوب برای کنترل و مدیریت متمرکز تبدیل شده است. با این حال، برای مراکز داده بزرگ با صدها هزار سرور و چندین هزار سوئیچ، حالت کنترل کننده تک با عت می شود سیستم از عدم مقیاس پذیری و قابلیت اطمینان رنج ببرد. استفاده از معماری توزیع شده می تواند عملکرد سیستم را بهبود بخشد، اما محدودیت اصلی این کار نداشت ایستا بین سوئیچ و کنترل کننده است که ممکن است منجر به عدم تعادل بار در کنترل کننده ها شود. استفاده از چندین کنترل کننده توزیع شده در SDN برای بهبود مقیاس پذیری و قابلیت اطمینان، جایی که هر کنترل کننده یک بخش ثابت از شبکه را مدیریت می کند، استفاده شده است. در روش پیشنهادی، یک کنترل کننده متعادل (بالکن)، به عنوان طرحی برای انتقال پویای سوئیچ SDN برای دستیابی به تعادل بار بین کنترل کننده های SDN با هزینه مهاجرت کم به کار گرفته شده است. تصمیم گیری برای مهاجرت توسط نظارت بر اساس ترافیک شبکه انجام می شود. برای تعادل بار در شبکه مبتنی بر نرم‌افزار، از یک استراتژی انتقال سوئیچ مرحله‌ای استفاده می‌شود و کنترل کننده هدف توسط الگوریتم گرگ خاکستری انتخاب می‌گردد. نتایج نشان می‌دهد که زمان پاسخ روش پیشنهادی ۱۳٪ بهتر از روش بالکن است. همچنین با افزایش نرخ ارسال به ۸.۳۳٪، توان عملیاتی کنترل کننده در مقایسه با روش بالکن بهبود می‌یابد. لذا، تعادل بار روش پیشنهادی عملکرد بهتری نسبت به کارهای پیشین دارد.

واژه‌های کلیدی: شبکه مبتنی بر نرم‌افزار، مهاجرت سوئیچ، تعادل بار، الگوریتم گرگ خاکستری

۱- مقدمه

جداول جریان، تأخیر اضافی، افزایش احتمال خرابی شبکه، پیکربندی ایستای نداشت بین سوئیچ و یک کنترل کننده، همچنین مشکل تنها نقطه شکست که سبب افزایش آسیب پذیری شبکه در برابر حملات و اختلالات و بی ثباتی شبکه می‌شود. بنابراین با توجه به محدودیت عملکرد و ظرفیت پردازشی کنترل کننده واحد و به منظور رفع کاهش نرخ انتقال بخش داده، استفاده بهینه تر از منابع، همچنین رفع چالش‌هایی مانند: دسترس پذیری، مقیاس پذیری، مشکل تک نقطه شکست، بروز سربار، کاهش کیفیت سرویس، کاهش گذردهی، عدم استفاده بهینه از منابع و تأخیر انتقالات داده به خصوص در مراکز داده بزرگ، نیاز به یک معماری کنترل کننده توزیع شده با رویکرد تعادل بار در کنترل کننده جهت دستیابی به مقیاس پذیری و قابلیت اطمینان، به شدت احساس می‌شود. از آنجایی که هدایت بسته در

با مطرح شدن مفهوم شبکه مبتنی بر نرم‌افزار^۱، رویکرد جدیدی در صنعت شبکه ارائه شد که بتواند نیازهای نسل جدید سامانه‌های مبتنی بر شبکه را نیز پوشش دهد. اگرچه SDN نسبت به شبکه‌های سنتی، به دلیل بخش کنترلی قابل برنامه ریزی، مؤثرتر عمل می‌کند و سبب توزیع مناسب بار ترافیک، بهبود کیفیت سرویس^۲ و کاهش تأخیر می‌گردد، اما هم‌زمان با رشد شبکه‌ها، افزایش تعداد سوئیچ‌ها و تنوع در بار ترافیک، شبکه‌های SDN با یک کنترل کننده متمرکز با مشکلاتی روبرو شده‌اند از جمله: نیاز به توان محاسباتی، ذخیره سازی داده، گذردهی بیشتر در یک کنترل کننده برای تحویل ترافیک، چالش مقیاس پذیری برای شبکه‌های بزرگ، ترافیک انفجاری در تعداد حالات ارسالی در

نوسان بار ناشی از مهاجرت سوئیچ نامناسب ممکن است به طور مکرر منجر به انتقال سوئیچ شود، که نه تنها سربار شبکه را افزایش می‌دهد، بلکه تعادل شبکه را کاهش داده و بر عملکرد شبکه تأثیر می‌گذارد. از طرفی مهاجرت کمینه و مناسب سوئیچ‌ها می‌تواند هزینه اتصالات جدید آن‌ها را پس از انتقال کاهش دهد. در این راستا، این پژوهش به دنبال یافتن روشی نوین جهت بهبود عملکرد شبکه در ساختار مبتنی بر نرم‌افزار است. کنترل‌کننده‌های توزیع چندگانه پویا در شبکه‌های مبتنی بر نرم‌افزار در جهت افزایش مقیاس‌پذیری و قابلیت اطمینان مورد استفاده قرار می‌گیرند، جایی که هر کنترل‌کننده یک بخش^۶ ایستای شبکه را مدیریت می‌کند. با توجه به اینکه نگاهت پویا بین سوئیچ‌ها و کنترل‌کننده‌ها می‌تواند بهره‌وری را در مدیریت تغییرات بار ترافیکی بهبود بخشد، از مهاجرت سوئیچ برای رسیدن به جوابی بهینه‌تر استفاده می‌گردد. در این راستا استفاده از خوشه‌بندی مناسب نیز می‌تواند موجب افزایش کارایی شبکه گردد. بر همین اساس رویکرد پیشنهادی به دنبال استفاده از یک استراتژی مهاجرت سوئیچ مرحله‌ای^{۱۱} با به‌کارگیری الگوریتم بهینه‌سازی گرگ خاکستری^{۱۱} برای متعادل‌سازی بار ترافیک در شبکه و انتخاب بهترین خوشه از سوئیچ‌های منتخب با حداقل هزینه انتقال (انتقال حداقل سوئیچ (ها)) و نوسان بار در شبکه می‌باشد.

این رویکرد بهینه‌سازی را در سه بخش انجام می‌دهد. در بخش اول سربارترین کنترل‌کننده به‌عنوان کنترل‌کننده مبدأ از طریق مقایسه میانگین درجه درخواست سوئیچ هر کنترل‌کننده انتخاب می‌گردد. بخش دوم به انتخاب سریع و مناسب کنترل‌کننده کم‌بار به‌عنوان کنترل‌کننده هدف به‌واسطه بگاری الگوریتم گرگ خاکستری مرتبط است و بخش سوم به واسطه انتخاب بهترین سوئیچ(های) کنترل‌کننده مبدأ با حداقل نوسان و هزینه انتقال از طریق بررسی اعتبار سوئیچ و حداکثرسازی چگالی نسبی در ساخت خوشه منتخب، فرآیند بهینه‌سازی را تکمیل می‌کند.

۳- پیشینه تحقیق

تاکنون تلاش‌های بسیاری در زمینه شبکه‌های مبتنی بر نرم‌افزار انجام گرفته است که هرکدام نقش بسزایی در بهبود کارایی این شبکه‌ها داشته است. در این بخش به‌طور اجمالی نگاهی به کارهای پیشین در زمینه بهبود شبکه‌های مبتنی بر نرم‌افزار خواهد شد و مزایا و معایب هرکدام به‌اختصار بیان می‌گردد. جدول ۱ به‌صورت مروری به مقالات بررسی شده می‌پردازد.

صفحه داده^۲ انجام می‌شود، بیشتر تحقیقات قبلی، روش‌های مؤثری را برای فراهم کردن کیفیت سرویس بهتر (مثل راندمان بالا و زمان پاسخ کم) برای کاربران با غلبه بر محدودیت‌های مختلف که به صفحه داده ارتباط داده می‌شود، مثل محدودیت ظرفیت پیوند داده [۱] و محدودیت اندازه جدول جریان [۲] طراحی می‌کنند؛ حال آنکه اثر محدودیت ظرفیت پیوند کنترل بر کیفیت سرویس نادیده گرفته می‌شود. در حقیقت، ظرفیت پیوند کنترل هم در پیوند بالا^۴ و هم در پیوند پایین^۵ محدود است [۳]. در واقع از پیوند بالا برای تمایز کانال کنترل از سوئیچ‌ها به کنترل‌کننده و از پیوند پایین برای تمایز آن از کنترل‌کننده به سوئیچ استفاده می‌گردد. هنگامی که کنترل‌کننده‌ها از بار ناهموار رنج می‌برند، کنترل‌کننده سربار الگوریتم تعادل بار را اجرا می‌کند تا توزیع نقشه را بین کنترل‌کننده‌ها و سوئیچ‌ها برای تحقق توازن بار کنترل‌کننده انجام دهد. کارهای قبلی بسیاری [۴-۶] در زمینه تعادل بار بر اساس انتقال سوئیچ انجام شده است. این روش‌ها تعادل بار را از سه روش حل می‌کنند: ۱. انتخاب سوئیچ مهاجر. ۲. انتخاب کنترل‌کننده هدف. ۳. مهاجرت^۶ سوئیچ. باید به این نکته توجه داشت که انتخاب نامناسب سوئیچ مهاجر و کنترل‌کننده هدف باعث بروز مشکل نوسان بار می‌گردد [۷].

رویکرد بهینه تعادل بار را می‌توان در شبکه‌های مرکز داده بزرگ، مخابراتی، سیار، همچنین شبکه‌های جریان^۷ و تحویل محتوا^۸ یا شبکه‌های سازمانی با تعداد زیادی دستگاه شبکه به کار برد.

ادامه مقاله به شرح زیر سازمان‌دهی شده است. در بخش دوم انگیزه کار بیان می‌گردد. در بخش سوم مروری بر کارهای پیشین در زمینه توازن بار بین کنترل‌کننده‌های شبکه‌های مبتنی بر نرم‌افزار انجام خواهد گرفت. مدل سیستم، طریقه عملکرد و الگوریتم روش پیشنهادی در بخش چهارم بیان می‌گردد. در بخش پنجم، ارزیابی نتایج حاصل از آزمایش‌ها و مقایسه با روش‌های پیشین ارائه می‌گردد. در بخش ششم، به بحث و بررسی در مورد کارایی روش پیشنهادی پرداخته می‌شود و در نهایت در بخش هفتم، یک نتیجه‌گیری کلی ارائه خواهد شد.

۲- انگیزه

اگرچه روش‌های موجود مبتنی بر استراتژی مهاجرت سوئیچ می‌توانند توزیع نابرابر بار را در چندین کنترل‌کننده حل کنند، اما برخی از این روش‌ها توازن شبکه را در نظر نمی‌گیرند. لذا در جهت نیل به یک ساختار پویا در همبندی این شبکه‌ها، به کار بردن روشی که بتواند بارهای موجود در این شبکه‌ها را در حالت متعادل نگه دارد امری ضروری است. علاوه بر این، وجود مشکل

جدول ۱: مروری بر مقالات پیشین

سال	نویسنده	عنوان مقاله	روش	مزایا	معایب
۲۰۱۹	لین و سای [۸]	سیستم کنترل‌کننده سلسله مراتبی SDN لیه - ابر با تخصیص بهینه منابع تطبیقی برای تعادل بار	ارائه یک سیستم سلسله مراتبی کنترل‌کننده SDN لیه - ابر (HECSDN) با سه ویژگی: اشتراک گذاشتن منابع محاسباتی در لیه و ابر، طراحی مدل صف از معماری پیشنهادی، ارائه یک الگوریتم تعادل بار کارآمد	<ul style="list-style-type: none"> افزایش مقیاس‌پذیری شبکه کاهش تأخیر محاسبات تخصیص عادلانه منابع پایداری سیستم در دوره‌های کوتاه 	<ul style="list-style-type: none"> مختل شدن کل شبکه در صورت از کار افتادن کنترل‌کننده مرکزی کاهش کارایی به دلیل افزایش تأخیر به واسطه افزایش فاصله بخش ابر از لیه
۲۰۱۹	النم و کوربا [۹]	تعادل بار از طریق انتقال سوئیچ در شبکه مبتنی بر نرم‌افزار	در این رویکرد حرکات انتقال بار مبادله می‌شود. مشکلات انتساب کلی گنجایش داده شده است.	<ul style="list-style-type: none"> مقاومت تحت شکست جزئی صفحه کنترل ایجاد موازنه عملکرد - سرعت انجام مهاجرت‌ها فقط در صورت بهبود 	<ul style="list-style-type: none"> کاهش عملکرد با کاهش استفاده از کنترل‌کننده‌های فرعی در نظر نگرفتن پردازنده به‌عنوان یک عامل مهم در مهاجرت
۲۰۲۰	لو [۱۰]	AMcon ¹ : یک کنترل‌کننده SDN سازگار با توزیع در شبکه‌های مرکز داده	از تئوری جامعه شبکه پیچیده برای انتخاب سوئیچ کلید برای قرار دادن کنترل‌کننده‌ای که به سوئیچ‌های کنترل‌شده در یک زیر شبکه نزدیک‌تر است، استفاده می‌شود. همچنین از کنترل‌کننده‌های قابل تنظیم تطبیقی و نگاشت‌کننده برای طراحی صفحه داده استفاده شده است.	<ul style="list-style-type: none"> استفاده از هر دو روش تطبیق تعداد کنترل‌کننده در صورت اضافه‌بار شبکه و مهاجرت سوئیچ 	<ul style="list-style-type: none"> پیچیدگی محاسباتی بالای مدل آموزشی شبکه عصبی
۲۰۲۰	روپانی [۱۱]	تعادل بار پویا در شبکه‌های مبتنی بر نرم‌افزار با استفاده از یادگیری ماشینی	بار توسط مدل شبکه عصبی آموزش دیده برای کوتاه‌ترین مسیرهای مختلف ارائه‌شده توسط الگوریتم دایکسترا ^۱ پیش‌بینی شده است.	<ul style="list-style-type: none"> استفاده از شبکه عصبی آموزش دیده برای محاسبه کوتاه‌ترین مسیر و انتقال داده با حداقل تأخیر 	<ul style="list-style-type: none"> پیچیدگی محاسباتی بالای مدل آموزشی شبکه عصبی
۲۰۲۰	منظر القانم و یغمایی [۱۲]	اینترنت اشیا ^۱ مبتنی بر نرم‌افزار با بار متغیال و آگاه از کیفیت سرویس	از شبکه مبتنی بر نرم‌افزار برای تحقق کیفیت سرویس و تعادل ترافیک بین سرویس‌های IoT استفاده می‌شود. همچنین یک مکانیسم اکتشافی ^۱ پیش‌بینی‌کننده و فعال بر اساس تحلیل سری زمانی و منطق فازی ارائه شده است.	<ul style="list-style-type: none"> بهبود پارامترهای کیفیت سرویس IoT تعادل ترافیک بین سرویس‌های IoT 	<ul style="list-style-type: none"> عدم استفاده از الگوریتم‌های بهینه‌سازی برای جواب بهینه‌تر
۲۰۲۱	کولاکو [۱۳]	تعادل بار در شبکه‌های مبتنی بر نرم‌افزار با استفاده از مسیر یابی چندراهی	اطلاعات مسیریابی با استفاده از الگوریتم مسیریابی موج اصلاح‌شده تولید می‌شود. چندین مسیر به‌طور هم‌زمان بین گره‌های شبکه تشکیل می‌شود که باعث ایجاد توازن پویای کانال‌ها می‌شود.	<ul style="list-style-type: none"> مسیریابی پویا بدون محاسبه مجدد کل مسیر کاهش مدت‌زمان لازم برای تشکیل چندین مسیر و ساده‌سازی روند تعادل ترافیک تغییر بار کانال‌ها به‌صورت پویا 	<ul style="list-style-type: none"> بار کنترل‌کننده در نظر گرفته نمی‌شود
۲۰۲۱	زاو [۱۴]	PLOFR ¹ : یک چارچوب مسیر جریان آلاین برای صرفه‌جویی در مصرف برق و تعادل بار در SDN	مسیری با کمترین افزایش مصرف برق انتخاب می‌شود. همچنین از یک الگوریتم برنامه‌ریزی کارآمد انرژی برای انتقال کلیه جریان‌های عبوری از پیوند کم‌بار به پیوند بار متوسط و سپس خواباندن دستگاه‌های بی‌کار برای دستیابی به صرفه‌جویی در مصرف برق استفاده می‌گردد. برای جلوگیری از بروز ازدحام پیوند و بهبود عملکرد تعادل بار، الگوریتم برنامه‌ریزی تعادل بار پیشنهاد می‌شود.	<ul style="list-style-type: none"> بازده انرژی بالا کاهش انحراف استاندارد بهره‌وری از پیوند عملکرد بهتر تعادل بار 	<ul style="list-style-type: none"> بالاترین سربار پیام به‌خصوص در سناریوهایی با بار بالا تأخیر بیشتر به دلیل تبادل پیوسته پیام‌ها بین سوئیچ‌ها و کنترل‌کننده‌ها
۲۰۲۱	وئوق گوران اوربومی	ارائه روشی نوین مبتنی بر الگوریتم تکاملی گرگ خاکستری برای توازن بار کنترل‌کننده در شبکه‌های مبتنی بر نرم‌افزار با استفاده از مهاجرت پویای سوئیچ	استفاده از مهاجرت سوئیچ جهت توازن بار کنترل‌کننده با به‌کارگیری الگوریتم گرگ خاکستری جهت تعیین کنترل‌کننده هدف و یک استراتژی انتقال سوئیچ مرحله‌ای همراه با انتخاب مناسب کنترل‌کننده مبدأ و همچنین خوشه‌بندی و اعتبارسنجی سوئیچ قبل از انتقال	<ul style="list-style-type: none"> انتخاب بهینه کنترل‌کننده مبدأ، کنترل‌کننده مقصد و سوئیچ مهاجر مناسب استفاده از الگوریتم گرگ خاکستری جهت کاهش بار کنترل‌کننده و اجزای سریع‌تر روال و پیمایش کردن سریع‌تر جواب. کاهش نوسان سوئیچ در مهاجرت کاهش هزینه‌های انتقال 	<ul style="list-style-type: none"> نتایج بهتر تنها در شبکه‌های بزرگ سربار اضافی در پردازش و ارتباط بین کنترل‌کننده‌ها امکان مهاجرت بار تنها یک کنترل‌کننده در هر زمان

۴- روش پیشنهادی

مسیر در کنترل کننده SDN S_i ، بار محاسباتی با نرخ f_{S_i, S_i} تولید می کنند که به صورت زیر است:

$$\mathcal{K}(f_{S_i, S_i}) \quad (2)$$

ج- بار نصب قوانین^{۱۶}

بار محاسباتی در کنترل کننده به دلیل نصب قوانین در سوئیچ S_i مساوی است با:

$$\sum_{S_j \in \mathcal{S}} G(f_{S_i, S_j}) + G(f_{S_i, 0}) \quad (3)$$

معادله (۳) نرخ جریانی را که در حال عبور از S_i و رفتن به سوئیچ های دیگر یا خارج شدن از شبکه SDN است را بیان می کند. مجموعه سوئیچ های SDN که توسط کنترل کننده SDN C_m کنترل می شوند با P_m نشان داده می شوند. مجموعه \mathcal{S} سپس در $|C|$ بخش تقسیم می شود، با $P_m \subset \mathcal{S}$ ، $P_m \cap P_n = \emptyset$ ، $n \neq m$.

حال، بار کلی محاسباتی در کنترل کننده C_m (L_{C_m}) به صورت زیر محاسبه می شود:

$$L_{C_m} \triangleq \sum_{S_i \in P_m} \mathcal{K}(f_{0, S_i}) + \sum_{\substack{S_j \notin P_m \\ S_i \in P_m}} \mathcal{K}(f_{S_i, S_j}) + \sum_{\substack{S_i \in P_m \\ S_j \in \mathcal{S}}} G(f_{S_i, S_j}) + \sum_{S_i \in P_m} G(f_{S_i, 0}) \quad (4)$$

افزافه بار کنترل کننده SDN پاسخگویی^{۱۷} آن را کاهش می دهد و باعث تخریب عملکرد آن می شود. زیرا جریان ها باعث می شوند که یک تأخیر غیرمنتظره تجربه گردد که باعث کاهش عملکرد سیستم می گردد.

یک کنترل کننده SDN را زمانی دارای بار بیش از حد یا دارای ازدحام گویند، وقتی بار کلی محاسباتی آن به صورت زیر باشد:

$$L_{C_m} > L \quad (5)$$

که در آن L نشانگر حداکثر بار محاسباتی قابل تحمل در هر کنترل کننده SDN است. مسئله CLB را می توان به عنوان یک مسئله بهینه سازی ریاضی بیان کرد که آن را Optimal CLB (OCLB) می نامند و به صورت زیر تعریف می شود:

$$\begin{aligned} \min_{P_1, \dots, P_{|C|}} \max_{C_m \in \mathcal{C}} L_{C_m}; \\ \text{Subject to } P_m \cap P_n = \emptyset, m \neq n; \\ UP_m = \mathcal{S}. \end{aligned} \quad (6)$$

در ابتدا سناریوهای کنترل کننده چندگانه SDN با مسئله تعادل بار کنترل کننده بهینه (OCLB¹²) توسعه داده می شود و ثابت می شود که این یک مشکل NP-Complete است. سپس مسئله OCLB به عنوان یک مسئله تقسیم بندی گراف مدل سازی شده و بالکن^{۱۳} [۱۵] و بالکن پلاس^{۱۴} [۱۶] توسعه می یابد. علاوه بر این، برای تعادل بخشیدن به بارهای کنترل کننده، یک استراتژی مهاجرت مرحله ای متشکل از الگوریتم گرگ خاکستری [۱۷] و SSMS [۱۸] انتخاب و در شبکه چند دامنه ای پیشنهاد شده است.

۴-۱- مدل سیستم

سناریوی SDN از مجموعه ای از سوئیچ های SDN، $S_i \in \mathcal{S}$ تشکیل شده است، که توسط مجموعه \mathcal{C} از کنترل کننده های SDN، $C_m \in \mathcal{C}$ اداره می شود. در واقع شبکه SDN توزیع شده به چندین زیر دامنه متصل شده با پیوندهای فیزیکی تقسیم می شود. صفحه کنترل از M کنترل کننده تشکیل شده است که با $\mathcal{C} = \{C_1, \dots, C_M\}$ نشان داده می شود. تعداد سوئیچ ها در صفحه داده N است و مجموعه سوئیچ ها $\mathcal{S} = \{S_1, \dots, S_N\}$ است. کل شبکه به M زیر دامنه تقسیم شده است، بنابراین $G = \{G_1, \dots, G_M\}$. رابطه بین سوئیچ S_i و کنترل کننده C_j یک متغیر باینری x_{ij} است، جایی که $x_{ij} = 1$ نشان می دهد که S_i به C_j متصل است. نرخ ورود جریان های جدید به سوئیچ S_i SDN، از خارج از شبکه SDN و $f_{S_i, 0}$ نرخ جریان های ورودی جدیدی که شبکه SDN را از سوئیچ S_i ترک می کنند، نشان می دهد، در حالی که f_{S_i, S_i} نرخ ورود فعلی جریان های جدید را نشان می دهد که پیوند بین دو سوئیچ SDN متصل S_i و S_j را طی می کند. به عبارت دیگر، f_{S_i, S_j} نشان دهنده نرخ ورود جریان های جدید به سوئیچ S_j SDN است که از سوئیچ SDN S_i آمده است.

الف- بار محاسبه مسیر^{۱۵} برای جریان های خارجی

هنگامی که دسته ای از جریان ها با نرخ f_{0, S_i} از خارج شبکه به S_i می رسند، با توجه به محاسبه مسیر در کنترل کننده SDN S_i ، یک بار محاسباتی تولید می کنند که برابر است با:

$$\mathcal{K}(f_{0, S_i}) \quad (1)$$

ب- بار محاسبه مسیر جریان از سایر دامنه های SDN

هنگامی که دسته ای از جریان ها از S_j به S_i می رسند (یک سوئیچ کنترل شده توسط یک کنترل کننده SDN دیگر)، به دلیل محاسبه

۴-۲- OCLB به‌عنوان مسئله بخش‌بندی گراف^{۱۸}

مسئله OCLB را می‌توان به‌صورت بخش‌بندی گراف بیان کرد. مسئله محاسبه L_{C_m} می‌تواند مستقیماً روی گراف اعمال شود. به‌طور خاص، شبکه SDN به‌عنوان گرافیکی جهت‌دار و لبه‌دار وزنی و رأس وزن‌دار $G(\mathcal{S}, \mathcal{E})$ نشان داده می‌شود که در آن سوئیچ‌های SDN رأس‌هایی با وزن $l(S_i)$ ، $S_i \in \mathcal{S}$ و لبه‌های $\mathcal{E} = \{(S_i, S_j) : S_i, S_j \in \mathcal{S}, l(S_i, S_j) > 0\}$ اتصالات بین سوئیچ‌های SDN هستند و $l(S_i, S_j)$ وزن‌های لبه (S_i, S_j) هستند. به این معنا که:

$$l(S_i) = \mathcal{K}(f_{0,S_i}) + \sum_{S_j \in \mathcal{S}} \mathcal{G}(f_{S_i,S_j}) + \mathcal{G}(f_{S_i,0}) \quad (7)$$

$$l(S_j, S_i) = \mathcal{K}(f_{S_j,S_i}) \quad (8)$$

بار کلی در C_m با L_{C_m} مشخص می‌شود. پس‌از آن مجموع وزن رئوس متعلق به بخش آن با مجموع وزن لبه‌های هدایت‌شده به بخش C_m جمع می‌شود. به‌طور مشخص:

$$L_{C_m} = \sum_{S_i \in P_m} l(S_i) + \sum_{\substack{S_j \notin P_m \\ S_i \in P_m}} l(S_j, S_i) \quad (9)$$

۴-۳- به‌کارگیری بالکن در روش پیشنهادی

مهاجرت بهینه سوئیچ SDN به دلیل پیچیدگی محاسباتی آن، غیرعملی است (مسئله OCLB، NP-complete است) و می‌تواند منجر به مهاجرت نامطلوب بیش‌ازحد سوئیچ شود. یک رویکرد عملی‌تر باید شامل تنظیمات افزایشی بخش‌های سوئیچ باشد، یعنی فقط تعداد کمی از سوئیچ‌های SDN مهاجرت کنند.

بالکن یک الگوریتم اکتشافی است که در طول زمان اجرای شبکه عمل می‌کند و قادر به شناسایی و رفع ازدحام در کنترل‌کننده‌های SDN از طریق انتقال و مهاجرت مناسب سوئیچ می‌باشد. اگر به قابلیت متوالی بودن نیازی نباشد (یعنی، پیام‌های سوئیچ بتوانند خارج از ترتیب پردازش شوند)، دوره خاموشی ذکرشده در [۱۹، ۶] را می‌توان حذف کرد و راه‌اندازی اتصالات جدید تأخیر بیشتری را تجربه نخواهد کرد. استفاده از کدهای بالکن در روش پیشنهادی برای چنین مواردی مناسب است. اگر شبکه به قابلیت توالی‌پذیری نیاز داشته باشد، انتقال سوئیچ بر اساس بالکن ممکن است باعث اختلال موقت در سرویس شود. برای حل این مسئله، یک نسخه بهبودیافته از بالکن به نام بالکن پلاس پیشنهاد می‌شود تا از اختلال در سرویس در هنگام انتقال جلوگیری شود.

۴-۴- انتخاب دامنه برون مهاجرت

در روش پیشنهادی با استفاده از مزایای مرحله اول استراتژی انتقال سوئیچ مرحله‌ای، اگر کنترل‌کننده‌های دامنه‌های مختلف در مقایسه با آستانه بار L به‌عنوان کنترل‌کننده‌های سربار تعیین - شوند، بار این کنترل‌کننده‌ها مورد مقایسه قرار گرفته و دامنه‌ای که بیشترین بار کنترل‌کننده را دارد به‌عنوان دامنه برون مهاجرت (Gh) انتخاب می‌گردد. دامنه برون مهاجرت دامنه‌ای است که سوئیچ‌های خود را جهت کاهش بار کنترل‌کننده منتقل خواهد کرد. برای تشخیص کنترل‌کننده‌ای که بیشترین بار را در بین کنترل‌کننده‌های سربار در همه دامنه‌ها دارد، از درجه درخواست سوئیچ استفاده می‌گردد. مفهوم درجه درخواست سوئیچ در ذیل معرفی شده است. درجه درخواست سوئیچ. تعداد درخواست‌های Packet-In از سوئیچ S_i به کنترل‌کننده به‌عنوان درجه درخواست سوئیچ r_{S_i} تعریف شده است. در زیر دامنه G_j ، n_j تعداد گره‌ها است، R_{G_j} مجموع درجه درخواست سوئیچ در معادله (۱۰) است و $R_{avg}^{G_j}$ میانگین درجه درخواست سوئیچ را در معادله (۱۱) نشان می‌دهد [۱۸].

$$R_{G_j} = \sum_{i=1}^{n_j} r_{S_i} \cdot x_{ij} \quad (10)$$

$$R_{avg}^{G_j} = \left(\sum_{i=1}^{n_j} r_{S_i} \cdot x_{ij} \right) / n_j \quad (11)$$

کنترل‌کننده‌های سربار از نظر میانگین درجه درخواست سوئیچ اولویت‌بندی می‌شوند و کنترل‌کننده با بیشترین میزان میانگین درجه درخواست سوئیچ به‌عنوان کنترل‌کننده دامنه Gh انتخاب می‌شود. دلیل این اولویت‌بندی این است که درخواست‌های Packet-In ارسال‌شده از سوئیچ‌ها به کنترل‌کننده بیشترین بار برای کنترل‌کننده است و این درخواست‌ها پردازنده را سریع‌تر اشباع می‌کنند.

۴-۵- انتخاب دامنه درون مهاجرت با استفاده از الگوریتم گرگ خاکستری

رفتار محاصره گرگ‌های خاکستری برای شکار طعمه را می‌توان با معادله (۱۲) و (۱۳) بیان کرد [۲۰، ۱۷].

$$\vec{D} = | \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) | \quad (12)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (13)$$

در اینجا "t" تکرار فعلی را نشان می‌دهد. X_p و X به ترتیب بردار مکان طعمه و گرگ خاکستری هستند. A و C نیز بردار ضرایب هستند که با معادله‌ی (۱۴) و (۱۵) به دست می‌آیند.

سوئیچ‌های منتقل شده از دامنه G_p به دامنه G_q به عنوان درجه انتقال زیر دامنه $M(G_p, G_q)$ تعریف می‌شود که در معادله (۱۹) بیان شده است.

$$M(G_p, G_q) = | (R_{avg}^{G_p} - R_{avg}^{G_q}) / \sum_{j=1}^M R_{avg}^{G_j} | \quad (19)$$

سپس سوئیچ بعدی در \mathcal{A} انتخاب شده و در یک خوشه خالی \mathcal{T} وارد می‌شود. وقتی حداکثر اندازه لیست سوئیچ اولیه ($mssls^{20}$) به دست آمد، همه جایگزین‌های مهاجرت با استفاده از تابع AlternativeEvaluation ارزیابی می‌شوند. بهترین گزینه تشکیل شده توسط \mathcal{T}^0 (خوشه) و کنترل‌کننده SDN هدف (کنترل‌کننده‌ای که \mathcal{T}^0 را دریافت خواهد کرد) انتخاب می‌شوند و مهاجرت می‌تواند رخ دهد. سرانجام، همبندی جدید SDN با توجه به نتایج مهاجرت ایجاد می‌شود. شبه کد مرحله اول الگوریتم پیشنهادی در ذیل ذکر شده است:

شبه کد الگوریتم پیشنهادی - مرحله اول

- Input:** Edge- and node-weighted graphs $G(S, \mathcal{E})$, SDN subdomains $\{G_1, G_2, \dots, G_M\}$, Controller load L
Output: Switch immigration domain G_i
 Switch emigration domain G_h
- 1: $D = \{ \}$
 - 2: **For** C_i in $\{C_1, \dots, C_M\}$
 - 3: **If** C_i is congested SDN controller in domain G_i
 - 4: Compute $R_{avg}^{G_i}$
 - 5: $D = D \cup R_{avg}^{G_i}$
 - 6: **Select** $R_{avg}^{G_h} = \text{MAX}(D)$
 - 7: **Select** the subdomain with $R_{avg}^{G_h}$ as G_h
 - 8: Adjacent subdomain of G_h is set as $B = \{G_{h-1}, G_{h-2}, \dots, G_{h-k}\}$
 - 9: Set the initial values of the population size B as grey wolves X_i ($i = 1, 2, \dots, n$), parameter a , maximum number of iteration, Coefficient Vector $\vec{A} = 2 \vec{a} \cdot \vec{r}_1 - \vec{a}$, $\vec{C} = 2 \cdot \vec{r}_2$
 - 10: Parameters Initialization
 - 11: Calculate initial population $X_i(t)$ randomly
 - 12: Calculate the fitness function of each search agent $f(X_i)$
 - 13: Assign the values for the first, second, third best solution X_α, X_β and X_δ
 - 14: $t = 0$
 - 15: **While** ($t < \text{Max number of iterations}$)
 - 16: **For** each search agent
 - 17: Update the position of wolves:

$$\vec{X}_n(t+1) = \vec{X}_p(t) - \vec{A} \cdot |\vec{C} \cdot \vec{X}_p(t) - \vec{X}_n(t)|$$

$$\vec{X}(t+1) = \frac{(\vec{X}_1(t) + \vec{X}_2(t) + \vec{X}_3(t))}{3}$$
 - 18: ComputeMigrationAlternatives محاسبه می‌شوند. الگوریتم، بعداً یک حلقه while را اجرا می‌کند که در آن خوشه به طور مداوم با تابع IncreaseCluster بزرگ می‌شود و با تابع ComputeMigrationAlternatives ارزیابی می‌شود. الگوریتم در صورت تحقق یکی از دو شرط توقف متوقف می‌شود: خوشه به اندازه از پیش تعیین شده حداکثر اندازه خوشه (mcs^{19}) برسد، یعنی اندازه $(\mathcal{T}) > mcs$ ، یا خوشه افزایش یافته برابر با نمونه قبلی شود ($new\mathcal{T} = \mathcal{T}$). در صورت لزوم می‌توان حداکثر اندازه خوشه را با درجه انتقال زیر دامنه به طور خودکار تعیین کرد. تعداد
 - 19: **End for**
 - 20: Update a , A , and C
 - 21: Calculate the fitness values of all search agents and grade them

$$\vec{A} = 2 \vec{a} \cdot \vec{r}_1 - \vec{a} \quad (14)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (15)$$

در اینجا مقدار "a" از ۲ به ۰ به صورت خطی در هر تکرار کاهش می‌یابد. بردارهای تصادفی r_1 و r_2 در محدوده ۰ تا ۱ قرار دارند. ماهیت شکار گرگ خاکستری توسط معادله‌ی (۱۶) تا (۱۸) نشان داده شده است. در این روش موقعیت‌های گرگ آلفا، بتا و دلتا به روز می‌شود و در نظر گرفته شده است که آلفا مناسب‌ترین راه‌حل است.

$$\vec{D}_\alpha = |\vec{C}_1 \vec{X}_\alpha(t) - \vec{X}(t)|, \quad \vec{D}_\beta = |\vec{C}_2 \vec{X}_\beta(t) - \vec{X}(t)|, \quad \vec{D}_\delta = |\vec{C}_3 \vec{X}_\delta(t) - \vec{X}(t)| \quad (16)$$

$$\vec{X}_1(t) = \vec{X}_\alpha(t) - \vec{A}_1 \vec{D}_\alpha,$$

$$\vec{X}_2(t) = \vec{X}_\beta(t) - \vec{A}_2 \vec{D}_\beta,$$

$$\vec{X}_3(t) = \vec{X}_\delta(t) - \vec{A}_3 \vec{D}_\delta \quad (17)$$

$$\vec{X}(t+1) = \frac{(\vec{X}_1(t) + \vec{X}_2(t) + \vec{X}_3(t))}{3} \quad (18)$$

روش کار به این صورت است که از بین دامنه‌های همسایه G_h دامنه‌ای که کمترین بار کنترل‌کننده را داراست، توسط الگوریتم گرگ خاکستری تعیین می‌گردد. این دامنه، دامنه درون مهاجرت (G_i) است. دامنه درون مهاجرت، دامنه‌ای است که پذیرای سوئیچ‌های انتخاب شده از دامنه G_h می‌باشد.

۴-۶- الگوریتم پیشنهادی

الگوریتمی که در روش پیشنهادی بکار می‌رود، اساساً بر تکرار سه تابع استوار است. ابتدا الگوریتم یک لیست زیرمجموعه را از دامنه G_h استخراج می‌کند که شامل گره‌های اولیه استفاده شده برای ساخت خوشه است. می‌توان این لیست را، به عنوان مثال، با جستجوی سوئیچ‌های SDN که افزایش قابل توجهی در نرخ ورود جریان دارند محاسبه کرد. اولین سوئیچ SDN متعلق به این لیست انتخاب شده و در خوشه خالی \mathcal{T} وارد می‌شود. گزینه‌های مهاجرت سوئیچ‌های SDN متعلق به \mathcal{T} از طریق تابع ComputeMigrationAlternatives محاسبه می‌شوند. الگوریتم، بعداً یک حلقه while را اجرا می‌کند که در آن خوشه به طور مداوم با تابع IncreaseCluster بزرگ می‌شود و با تابع ComputeMigrationAlternatives ارزیابی می‌شود. الگوریتم در صورت تحقق یکی از دو شرط توقف متوقف می‌شود: خوشه به اندازه از پیش تعیین شده حداکثر اندازه خوشه (mcs^{19}) برسد، یعنی اندازه $(\mathcal{T}) > mcs$ ، یا خوشه افزایش یافته برابر با نمونه قبلی شود ($new\mathcal{T} = \mathcal{T}$). در صورت لزوم می‌توان حداکثر اندازه خوشه را با درجه انتقال زیر دامنه به طور خودکار تعیین کرد. تعداد

شبه کد تابع افزایش خوشه

```

1: Function IncreaseCluster( $\mathcal{T}$ );
2: neighbors $\mathcal{T}$  = ComputeNeighborsOfCluster( $\mathcal{T}$ );
3: foreach  $S_i \in$  neighbors $\mathcal{T}$  do
4:   if  $V(S_i) \neq 0$ 
5:     new $\mathcal{T} = \mathcal{T} \cup S_i$ ;
6:     savedDensities = [savedDensities;  $S_i$ ,
Density(new $\mathcal{T}$ )];
7:    $S_i^0 = \text{argmax}_{\text{savedDensities}} \text{Density}(\text{new}\mathcal{T})$ ;
8:   return  $\mathcal{T} \cup S_i^0$ ;
    
```

اعتبار سوئیچ، که نشان دهنده دوره های حالت معتبر سوئیچ در هنگام انتقال است، به عنوان $V(s)$ تعریف می شود. اعتبار سوئیچ شامل دو پارامتر سال $Y(s_i)$ و عمر $L(s_i)$ است. دوره فعلی انتقال سوئیچ S_i است. بعد از اینکه S_i از یک فرآیند مهاجرت عبور کرد، سپس $Y(s_i)++$. $L(s_i)$ تعداد کل نوبت چرخشی^{۱۱} در فرآیند مهاجرت است، $V(s_i) = L(s_i) - Y(s_i)$. اگر $V(s_i) = 0$ باشد، از مهاجرت S_i صرف نظر می شود [۱۸].

ب- تابع محاسبه گزینه های مهاجرت

این تابع خوشه \mathcal{T} را به صورت مجازی به مقصد کنترل کننده کم بار SDN منتقل می کند و در صورتی که بار کنترل کننده از حد آستانه بیشتر نباشد بار هر کنترل کننده و هزینه انتقال migrationSize (تعداد سوئیچ هایی که منتقل می شوند) محاسبه می شود.

شبه کد تابع محاسبه گزینه های مهاجرت

```

1 Function ComputeMigrationAlternatives( $\mathcal{T}$ ,  $C_i$ );
2 “virtual” migrate cluster  $\mathcal{T}$  to SDN controller  $C_i$  from
domain  $G_i$ ;
3 If  $L_{C_i} < L$  then
4   Compute  $L_{C_n}, \forall C_n \in \mathcal{C}$ ;
5   Compute migrationSize for this new configuration;
6   Save them in lastAlternatives;
7   Return lastAlternatives;
    
```

با توجه به بردار alternatives, EvaluateMigrationAlternatives بهترین جایگزین \mathcal{T}^0 را در بین آن ها انتخاب می کند، که با یکی از روش های ارزیابی زیر بهینه می شود:

minMax - به حداقل رساندن حداکثر بار کنترل کننده ها:

$$\underset{\text{alternatives}}{\text{argmin}} (\max [L_{C_1}, \dots, L_{C_i}]) \quad (21)$$

minSum - به حداقل رساندن مجموع بار کنترل کننده ها:

$$\underset{\text{alternatives}}{\text{argmin}} \sum_{C_m \in \mathcal{C}} L_{C_m} \quad (22)$$

```

22: Update the positions of  $X_\alpha, X_\beta,$  and  $X_\delta$ 
23:  $t = t+1$ 
24: End while
25: Select  $X_\alpha$  in the final iteration as subdomain with  $L_{\min}$ 
named  $G_i$ 
    
```

شبه کد مرحله دوم الگوریتم پیشنهادی به شرح ذیل است:

شبه کد الگوریتم پیشنهادی - مرحله دوم

Input: Switch emigration domain G_h , Switch immigration domain G_i , Old Topology

Output: New Topology With Balanced Load

```

1:  $P_h$ : Set of SDN switches controlled by the congested
SDN controller  $C_h$  in domain  $G_h$ ;
2:  $\mathcal{A} = \text{ComputeStartingSwitchesList}(P_h)$ 
3: foreach  $S_i \in \mathcal{A}$  do
4:    $\mathcal{T} = \{S_i\}$ ;
5:   alternatives = alternatives  $\cup$ 
ComputeMigrationAlternatives ( $\mathcal{T}$ );
6:   while 1 do
7:     new $\mathcal{T} = \text{IncreaseCluster}(\mathcal{T})$ ;
8:     if size( $\mathcal{T}$ ) > mcs || new $\mathcal{T} = \mathcal{T}$  then
9:       break;
10:     $\mathcal{T} = \text{new}\mathcal{T}$ ;
11:    alternatives = alternatives  $\cup$ 
ComputeMigrationAlternatives ( $\mathcal{T}$ );
12: [ $\mathcal{T}^0$ ] ← EvaluateMigrationAlternatives(alternatives);
13: Migrate cluster  $\mathcal{T}^0, G_h = G_h \setminus \{\mathcal{T}^0\}, G_i = G_i \cup \{\mathcal{T}^0\}$ 
    
```

در ادامه توضیح مفصلی از تابع افزایش خوشه و تابع محاسبه گزینه های مهاجرت داده خواهد شد.

الف- تابع افزایش خوشه

با شروع از خوشه \mathcal{T} ، این تابع مجموعه neighbors \mathcal{T} را تشکیل می دهد که از تمام سوئیچ های SDN همسایه \mathcal{T} تشکیل شده است. سوئیچ SDN S_i همسایه \mathcal{T} است اگر $l(S_i, S_j) \neq 0, l(S_j, S_i) \neq 0$ $\exists S_j \in \mathcal{T}$ چگالی نسبی (Density) خوشه تازه ایجاد شده را به حداکثر برساند. منظور از چگالی نسبی نسبت درجه داخلی به تعداد لبه های برخورد شده (متصل) است:

$$\text{Density}(\mathcal{T}) = \frac{\sum_{S_i, S_j \in \mathcal{T}, S_i \neq S_j} l(S_i, S_j)}{\sum_{S_i, S_j \in \mathcal{T}, S_i \neq S_j} l(S_i, S_j) + \sum_{S_i \in \mathcal{T}} l(S_i, S_i)} \quad (20)$$

دلیل اصلی این حداکثر سازی چگالی نسبی این است که فقط سوئیچ های SDN با اتصالات قوی باید در همان خوشه گروه بندی شوند. سپس خوشه بین کنترل کننده ها به طور کلی منتقل می شود تا از پیچیدگی کلی محاسبات کنترل کننده ها بکاهد. به این تابع شرطی اضافه شده تا اعتبار سوئیچ را برای انتقال بررسی کند.

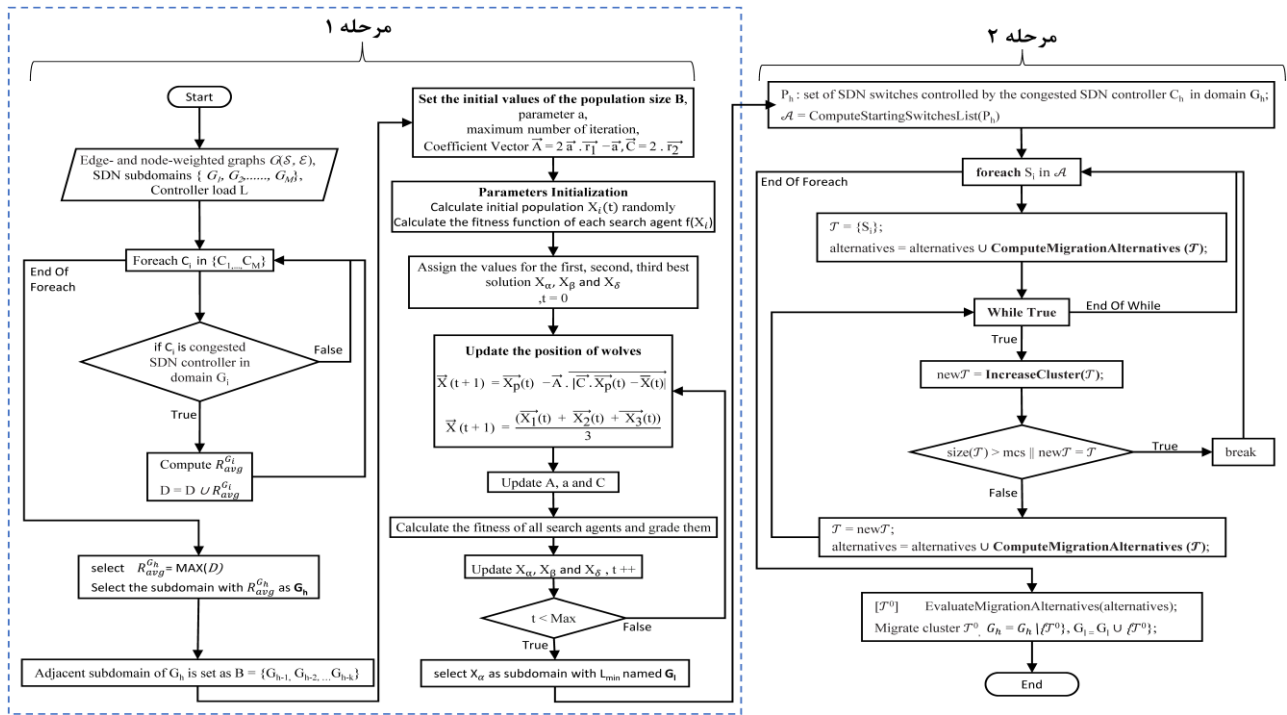
فلوچارت الگوریتم پیشنهادی در شکل ۱ و جزئیات توابع در شکل ۲ به نمایش درآمده است. همان طور که مشاهده می‌گردد، بهبود نسبت به روش بالکن در سه بخش اتفاق می‌افتد. در بخش اول، انتخاب متراکم‌ترین کنترل‌کننده از بین کنترل‌کننده‌های سربار، شبکه را سریع‌تر به حالت عادی خود بر می‌گرداند. در بخش دوم، جستجو توسط الگوریتم گرگ خاکستری سبب حذف حلقه از تابع محاسبه گزینه‌های مهاجرت و ساده‌سازی تانسور^{۳۲} تشکیل شده در این مرحله (که ورودی تابع ارزیاب می‌باشد) می‌گردد. لذا سرعت ارزیابی گزینه‌های مهاجرت بهبود می‌یابد. در بخش سوم، اعتبار-سنجی گزینه‌های مهاجرت، مشکل نوسان سوئیچ را حل می‌کند.

انتگرال - به حداکثر رساندن فاصله از تنظیمات بار کنترل‌کننده‌ها در صورت ازدحام:

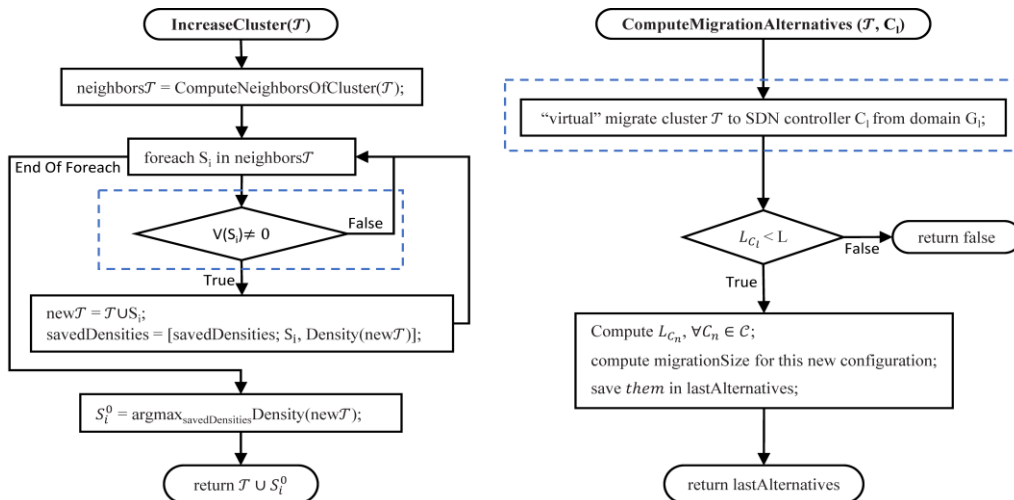
$$\operatorname{argmax}_{\text{alternatives}} \mathcal{D}([L_{C_1}, \dots, L_{C_{|C|}}], [\hat{L}_{C_1}, \dots, \hat{L}_{C_{|C|}}]) \quad (23)$$

با $[\hat{L}_{C_1}, \dots, \hat{L}_{C_{|C|}}]$ بردار کنترل‌کننده‌ها هنگامی که ازدحام دقیقاً قبل از بالکن ظاهر می‌شود بارگیری می‌شود و تابع $\mathcal{D}(u, v)$ به شرح زیر تعریف می‌شود:

$$\mathcal{D}(u, v) = \sum_i \int_{u_i}^{v_i} x^2 dx \quad (24)$$



شکل ۱: فلوچارت الگوریتم پیشنهادی.



شکل ۲: فلوچارت جزئیات توابع الگوریتم پیشنهادی.

۵- نتایج آزمایش‌ها

به منظور بررسی و تحلیل عملکرد روش پیشنهادی، در ابتدا پیاده‌سازی شده و با روش‌های دیگر، مقایسه می‌گردد. علت انتخاب روش بالکن جهت مقایسه، برتری آن نسبت به سایر روش‌های موجود از لحاظ ساختار توزیعی، پیچیدگی زمانی پایین، در نظر گرفتن نرخ توازن بار شبکه جهت تصمیم به انجام یا عدم انجام توازن بار و در نظر گرفتن آستانه بار قابل تنظیم بر اساس بار همه کنترل‌کننده‌ها است. نتایج آزمایش‌های این پژوهش و محیط شبیه‌سازی برای آن به شرح ذیل است:

۵-۱- محیط و شرایط شبیه‌سازی

الگوریتم پیشنهادی در محیط انکوندا^{۲۳} و نرم‌افزار اسپایدر^{۲۴} و با زبان پایتون پیاده‌سازی شده است. تمام آزمایش‌های این پژوهش در کامپیوتری با پردازنده Intel® core i5 و رم ۸ گیگابایت انجام شده است. از کتابخانه مینی‌نت^{۲۵} و ریو^{۲۶} در پایتون برای شبیه‌سازی شبکه SDN استفاده می‌گردد. از طرف دیگر، سوئیچ‌ها توسط ماژول اوپن‌وی سوئیچ^{۲۷} تعبیه شده در مینی‌نت، برای شبیه‌سازی رفتار سوئیچ جریان‌باز، مانند حذف، اضافه کردن و اصلاح قوانین جریان در سوئیچ‌ها، پیاده‌سازی می‌شوند. مینی‌نت و ریو از پروتکل جریان‌باز پشتیبانی می‌کنند که سازگاری تبادل اطلاعات بین آن‌ها را تضمین می‌کند. مؤلفه‌های فوق مبتنی بر سیستم اوبونتو^{۲۸} هستند.

۵-۲- تعیین پارامترهای ارزیابی

در این بخش، از جدول ۲ به منظور ارزیابی روش پیشنهادی و از معیارهای زیر جهت مقایسه استفاده می‌گردد. همچنین، برای دستیابی به اثرات بهتر شبیه‌سازی، پارامترهای آزمایش به شرح زیر تنظیم می‌شوند: ۱. کنترل‌کننده‌ها عملکرد یکسانی دارند. ۲. ظرفیت کنترل‌کننده ۱۰ مگابایت است. ۳. میانگین نرخ جریان ۱۲۰ KB/s است. از طرفی برای انجام شبیه‌سازی‌های سناریوی پویا، روال پیاده‌سازی شده‌ای ایجاد می‌گردد که به دنبال یک فرآیند پواسون، ورود و خروج جریان را در گره‌های لبه ایجاد می‌کند. برای پیاده‌سازی این روال از کتابخانه نامپای^{۲۹} در پایتون استفاده می‌گردد.

جدول ۲: شبیه‌سازی همبندی برای تحلیل عملکرد [۱۵].

نام	تعداد لبه	تعداد هسته	درجه اتصال		تعداد کنترل‌کننده‌ها
			اتصال گره‌های هسته-لبه	درجه اتصال گره‌های هسته-هسته	
همبندی ۱	۵۰	۴۰	۲	مش کامل	۵
همبندی ۲	۵۰	۴۰	۵	مش کامل	۵
همبندی ۳	۵۰	۴۰	۲	تعداد گره‌های هسته ۵	۵
همبندی ۴	۵۰	۴۰	۵	تعداد گره‌های هسته ۵	۵

الف- میانگین تعداد بسته‌های Packet-In دریافتی

این معیار، با محاسبه میانگین تعداد بسته‌هایی Packet-In که از طرف سوئیچ‌ها به کنترل‌کننده ارسال شده‌اند به منظور ارزیابی تأثیر اجرای الگوریتم توازن بار پیشنهادی بر بار ورودی به کنترل‌کننده و اندازه‌گیری توان عملیاتی کنترل‌کننده استفاده می‌گردد.

ب- میانگین تعداد بسته‌های Packet-Out خروجی

این معیار با محاسبه میانگین تعداد بسته‌های Packet-Out که توسط کنترل‌کننده به سمت سوئیچ‌ها ارسال می‌شوند و به منظور بررسی توان پاسخ‌دهی کنترل‌کننده در طی زمان با توجه به نرخ ورودی پیام‌های Packet-In استفاده می‌گردد.

ج- بار محاسباتی کلی

این معیار با اندازه‌گیری بار کنترل‌کننده‌ها در طی یک مدت‌زمان خاص به منظور ارزیابی فواصل و تعدد اجرای الگوریتم پیشنهادی در زمان تعیین شده مورد استفاده قرار می‌گیرد.

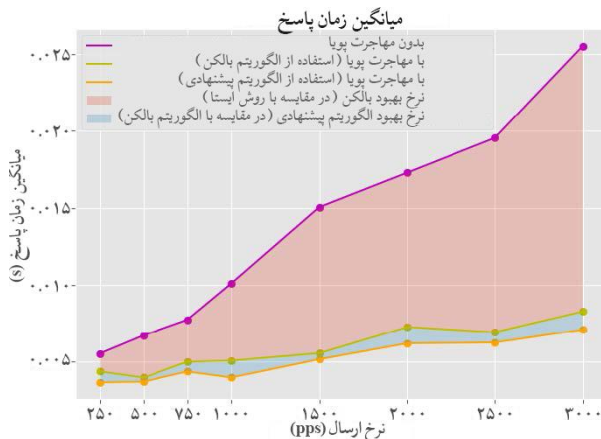
۵-۳- شرح نتایج

در ادامه، نتایج به دست آمده بر اساس معیارهای ارزیابی به طور جداگانه شرح داده شده است.

الف) میانگین تعداد بسته‌های Packet-Out خروجی

در بخش اول، عملکرد زمان پاسخ قبل از مهاجرت و در زمان قبل از اولین اجرای الگوریتم پیشنهادی برای دو کنترل‌کننده C4 و C5 در همبندی ۱ بررسی می‌گردد که در شکل‌های ۳ و ۴ نمایش داده شده است.

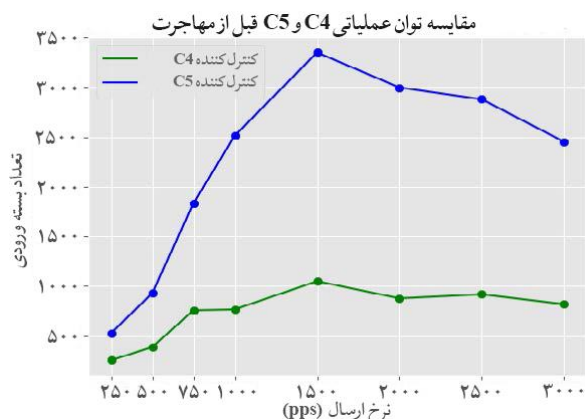
زمان پاسخ می‌گردد. همچنین نسبت به روش ایستا بسیار بهتر است.



شکل ۵: مقایسه میانگین زمان پاسخ در سه روش ایستا، بالکن و روش پیشنهادی.

ب) میانگین تعداد بسته‌های Packet-In دریافتی

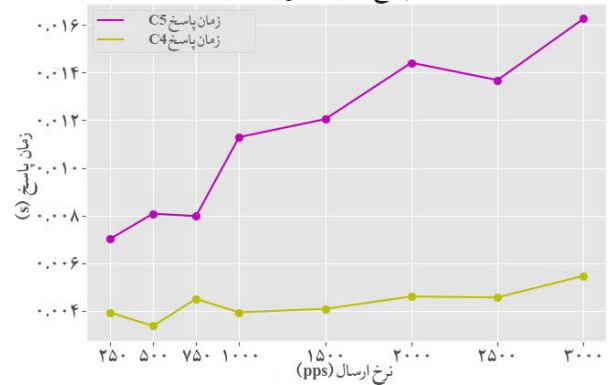
در این بخش کارایی تعداد بسته‌های ارسالی از لحاظ توان عملیاتی قبل و بعد از مهاجرت پس از اجرای اولین بار الگوریتم پیشنهادی در شکل ۶ و ۷ نشان داده شده است.



شکل ۶: مقایسه توان عملیاتی کنترل کننده C4 و C5 قبل از مهاجرت.

کاملاً واضح است که در شکل ۶ کنترل کننده C5 به دلیل افزایش نرخ ورود بسته‌های ورودی خارج از آستانه کنترل کننده به شدت بارگیری شده است، در حالی که کنترل کننده C4 هنوز پهنای باند استفاده نشده زیادی دارد. با این حال، عملکرد کلی کنترل کننده شبکه پس از نرخ ارسال ۱۵۰۰ بسته در ثانیه به دلیل تراکم، به سقف رسیده است. لذا کنترل کننده C5 به دلیل تأخیر در پردازش بسته‌های ورودی به دلیل افزایش نرخ ورودی آن، قادر به پذیرش بسته‌های ورودی جدید نیست و ممکن است به دلایلی از جمله پر بودن بافر کنترل کننده برخی از بسته‌ها دور انداخته شوند.

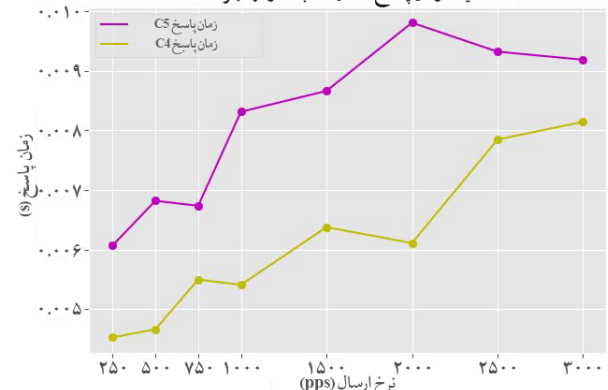
مقایسه زمان پاسخ C4 و C5 قبل از مهاجرت



شکل ۳: مقایسه زمان پاسخ C4 و C5 قبل از مهاجرت.

همان‌طور که در شکل ۳ مشاهده می‌شود، زمان پاسخ کنترل کننده C5 بالاتر از کنترل کننده C4 است زیرا C5 تحت فشار ترافیکی بیشتری است و بار بیشتری دارد. حال زمان پاسخگویی بعد از مهاجرت در شکل ۴ نشان داده شده است.

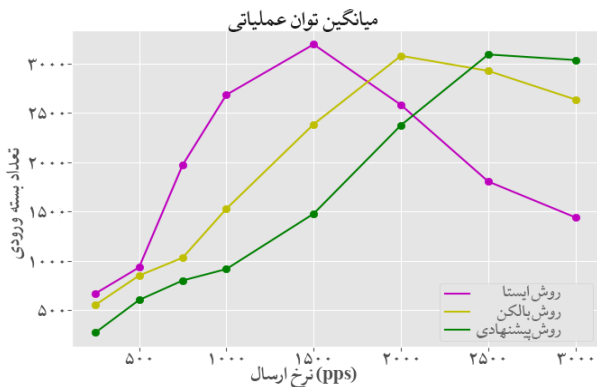
مقایسه زمان پاسخ C4 و C5 بعد از مهاجرت



شکل ۴: مقایسه زمان پاسخ C4 و C5 بعد از مهاجرت.

می‌توان مشاهده کرد که پس از مهاجرت زمان پاسخ کنترل کننده C5 به دلیل توازن بار میان دو کنترل کننده بسیار کاهش می‌یابد در حالی که زمان پاسخ کنترل کننده C4 به دلیل پذیرش بخشی از بار کنترل کننده C5 افزایش می‌یابد. همچنین مشاهده می‌شود که این افزایش به نسبت مزایای کاهش بار و زمان پاسخ در C5 ناچیز است.

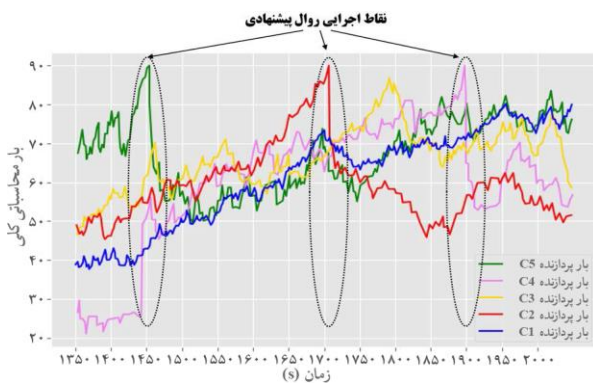
در بخش دوم، عملکرد سه روش پیشنهادی، بالکن و همچنین روش ایستا از لحاظ میانگین تعداد بسته‌های Packet-Out خروجی مقایسه شده است. همان‌طور که در شکل ۵ دیده می‌شود، روش پیشنهادی نسبت به روش بالکن در میانگین زمان پاسخ بهتر عمل می‌کند؛ زیرا الگوریتم پیشنهادی به دلیل استفاده از روال هوش جمعی گرگ خاکستری سریع‌تر به جواب بهینه می‌رسد. از طرفی این روال به علت حذف حلقه و انتقال مجازی تنها به کنترل کننده هدف، بار شبکه را سریع‌تر متعادل می‌سازد که سبب تأثیر بهتر در



شکل ۸: مقایسه میانگین توان عملیاتی در سه روش ایستا، بالکن و روش پیشنهادی.

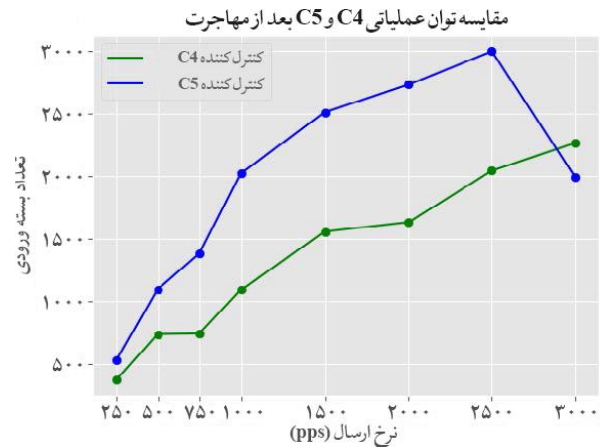
ج) فواصل و تعدد اجرای روال پیشنهادی

شکل ۹ بار محاسباتی ۵ کنترل کننده را در شبیه سازی همبندی ۱ در مدت ۳۴ دقیقه نشان می دهد (۰ به معنای عدم ازدحام است، درحالی که ۱۰۰ نشان دهنده اضافه بار است). خط سبز نشان دهنده سطح ازدحام کنترل کننده C5 است. به محض رسیدن به آستانه ۹۰ $L =$ الگوریتم پیشنهادی با استفاده از اندازه لیست سوئیچ اولیه ۲۰ و حداکثر اندازه خوشه ۲۰ اجرا می شود. روال های اجرایی مختلف الگوریتم پیشنهادی با بیضی نقطه چین سیاه نشان داده شده است.



شکل ۹: بار محاسباتی ۵ کنترل کننده در طول سناریوی پویا و تأثیر الگوریتم پیشنهادی در شبیه سازی با همبندی ۱. خط آبی L_{C1} ، خط قرمز L_{C2} ، خط زرد L_{C3} ، خط بنفش L_{C4} و خط سبز L_{C5} است.

با توجه به شکل ۹ می توان دریافت که تعداد اجرای الگوریتم پیشنهادی در زمان مشخص نسبت به روال بالکن کاهش پیدا کرده است (اجرای ۳ روال در روش پیشنهادی نسبت به ۴ روال بالکن). لذا فواصل اجرای بین دو روال نیز طولانی تر و یکنواخت تر شده است. دلیل احتمالی می تواند انتخاب بهتر سوئیچ ها برای مهاجرت و کاهش بار محاسباتی پردازنده به دلیل استفاده از الگوریتم گرگ خاکستری باشد. حداکثر بار محاسباتی در طول ۳ مورد اجرای روال پیشنهادی به طور متوسط ۱۸ درصد کاهش می یابد و به طور



شکل ۷: مقایسه توان عملیاتی کنترل کننده C4 و C5 بعد از مهاجرت.

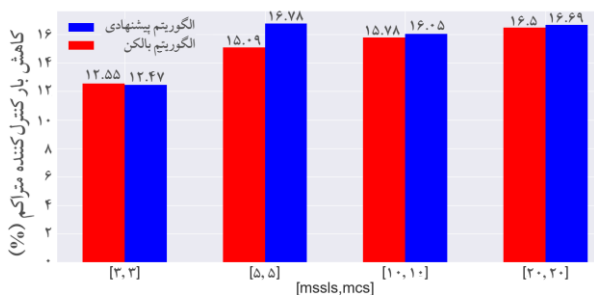
پس از مهاجرت، پیشرفت چشمگیری حاصل می شود. مجموع این دو خط پس از مهاجرت به طور قابل توجهی بیشتر است، به این معنی که توان کلی بهبود می یابد. آستانه نرخ ارسال در مقایسه با ۱۵۰۰ بسته در قبل از مهاجرت، به حدود ۲۵۰۰ بسته در ثانیه می رسد. چراکه کنترل کننده C5 به دلیل مهاجرت سوئیچ (های) آن و کاهش بار پردازنده در نرخ ارسال کلی بالاتری به مقدار آستانه پذیرش بسته های ورودی خود می رسد. نتایج نشان داده است که پس از مهاجرت، بار بین دو کنترل کننده به خصوص برای توان عملیاتی متعادل می شود و توان کلی پس از مهاجرت افزایش یافته است.

همچنین با توجه به شکل ۸ می توان مشاهده کرد که آستانه نرخ ارسال روش پیشنهادی نسبت به روش ایستا ۲ برابر شده است. همچنین نسبت به روش بالکن بهبود چشمگیری داشته است. از طرفی میانگین توان عملیاتی روش پیشنهادی به دلیل استفاده از الگوریتم گرگ خاکستری و بهبود سرعت در اجرای روال با شیب کمتری رو به افول است. چراکه سریع تر بار کنترل کننده را به زیر حد آستانه بر می گرداند. از طرفی با انتخاب بهتر سوئیچ هایی که باید منتقل شوند مانع از اجرای مجدد روال در کوتاه مدت می گردد. لذا کنترل کننده قادر خواهد بود بسته های ورودی جدید را سریع تر پردازش کند.

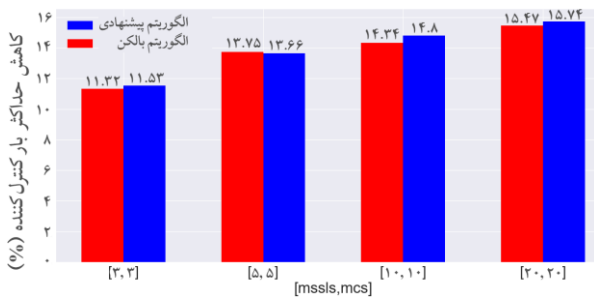
ذکر است در روش پیشنهادی تکرار ۲۰۰ مرتبه در نظر گرفته شده است که می‌تواند با افزایش تکرار، جواب بهینه‌تر باشد. نیاز به تکرار کمتر این الگوریتم نسبت به روش بالکن جهت پیدا کردن کنترل‌کننده هدف به‌خصوص در شبکه‌های بزرگ با تعداد کنترل‌کننده‌های بالا به‌کارگیری آن را برای این نوع شبکه‌ها مناسب‌تر می‌کند.

ر) عملکرد نسخه‌های بالکن شامل mcs و mssls

در این بخش به بررسی نتایج حاصل از عملکرد نسخه‌های بالکن شامل mcs و mssls در مقایسه با روش پیشنهادی با همبندی اول و minMax به‌عنوان متد ارزیابی نشان داده شده که در شکل‌های ۱۲ تا ۱۵ نمایش داده شده است.



شکل ۱۲: مقایسه عملکرد نسخه‌های بالکن شامل mcs و mssls در روش پیشنهادی (کاهش بار کنترل‌کننده).



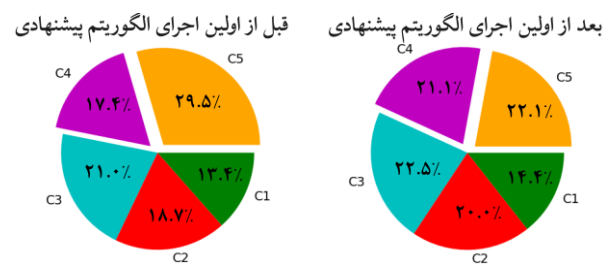
شکل ۱۳: مقایسه عملکرد نسخه‌های بالکن شامل mcs و mssls در روش پیشنهادی (کاهش حداکثر بار کنترل‌کننده).

همان‌طور که در شکل‌های ۱۲ و ۱۳ نشان داده شده است، بیشترین کاهش بار برای روش پیشنهادی با مقادیر [۵،۵] به‌دست آمده است. همچنین کاهش حداکثر بار کنترل‌کننده‌ها نیز در حالت با مقادیر [۲۰،۲۰] به دست آمد. برای محاسبه کاهش بار کنترل‌کننده از معادله ذیل استفاده شده است:

$$\frac{L_C^p(C_m^*) - L_C^{con}(C_m^*)}{L_C^{con}(C_m^*)} \cdot 100 \quad (25)$$

که در آن ثابت $L_C = [L_{C_1}, \dots]$ بردار نشانگر بار کنترل‌کننده‌ها، L_C^{con} بار کنترل‌کننده‌ها هنگام ظاهر شدن ازدحام درست قبل از

متوسط ۲.۳ سوئیچ در هر روال مهاجرت می‌کند. زمان محاسبه ۰.۶۱ ثانیه است و این به دلیل ماهیت اکتشافی الگوریتم گرگ خاکستری است که جواب بهینه را در زمان کمتری پیدا می‌کند. زیرا برخلاف الگوریتم بالکن نیاز به مجازی‌سازی مهاجرت به تمام کنترل‌کننده‌ها ندارد. لذا پردازش تابع ارزیاب ساده می‌گردد. در این حالت روال پیشنهادی می‌تواند بار محاسباتی را به‌طور مؤثر متعادل کرده و با انتقال سوئیچ‌های درست و با تعداد کم، مشکل اضافه‌بار در کنترل‌کننده را حل کند. همان‌طور که مشاهده می‌شود، روال پیشنهادی در زمان اجرا بار کنترل‌کننده‌ها را زیر آستانه نگه می‌دارد. در شکل ۱۰ نیز نسبت بار ۵ کنترل‌کننده در اولین اجرای روال پیشنهادی و همچنین درست قبل از اجرای آن به نمایش درآمده است.

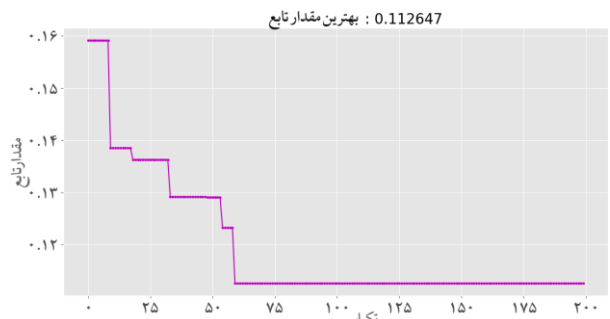


شکل ۱۰: بهبود در تعادل بار بعد از اجرای اولین بار الگوریتم پیشنهادی.

با توجه به شکل بالا، تعادل بار بین دو کنترل‌کننده C4 و C5 به‌خوبی مشهود است که ناشی از مهاجرت مناسب سوئیچ (ها) می‌باشد.

د) تأثیر بهینه‌سازی الگوریتم گرگ خاکستری در کنترل تعادل بار

در این بخش به تأثیر انتخاب راه‌حلی بهینه برای حل مسئله تعادل بار در شبکه‌های مبتنی بر نرم‌افزار پرداخته می‌شود که در شکل ۱۱ نشان داده شده است.



شکل ۱۱: عملکرد تابع برازندگی در انتخاب راه‌حل بهینه

با مشاهده شکل ۱۱ به‌خوبی می‌توان دریافت کرد با تکرار ۶۰ بار، جواب بهینه تابع برازندگی ۲۰ به سمت صفر نزدیک می‌شود. لازم به

کاهش هزینه‌های مجازی‌سازی انتقال می‌باشد. زیرا با حذف حلقه در تابع محاسبه گزینه‌های مهاجرت و ساده‌سازی تانسور به وجود آمده در این تابع، ارزیابی گزینه‌های مهاجرت نیز در شبکه‌های بزرگ‌تر آسان می‌گردد.

۶- بحث و بررسی

نتایج ارزیابی‌های صورت گرفته در محیط آزمون نشان می‌دهد که روش پیشنهادی منجر به توازن بار در کنترل‌کننده‌ها، نرخ خروج بسته‌های Packet-Out متناسب با بسته‌های ورودی در کنترل‌کننده‌ها، افزایش توان عملیاتی در نتیجه افزایش نرخ پذیرش بسته‌های Packet-In در کنترل‌کننده‌ها، فواصل بیشتر بین اجرای دو روال و تعدد کمتر اجرای روال پیشنهادی می‌گردد. به‌عنوان جمع‌بندی ویژگی‌های منحصر به فرد روش پیشنهادی می‌توان به نکات زیر اشاره نمود:

روش پیشنهادی، تنها به مهاجرت بار یک کنترل‌کننده در هر بازه زمانی می‌پردازد و بدین ترتیب و با انتخاب مناسب‌تر سوئیچ‌ها از مهاجرت‌های احتمالی متوالی ناشی از وقوع سربار در کنترل‌کننده مقصد جلوگیری می‌نماید. لذا از انتخاب یک کنترل‌کننده مقصد توسط دو یا چند کنترل‌کننده مبدأ جلوگیری می‌شود. به‌علاوه، به‌منظور اطمینان از کاهش بار در کنترل‌کننده‌ای که دچار سربار شده و همچنین جلوگیری از بروز سربار در کنترل‌کننده مقصد، پس از انتخاب سوئیچ در صورتی عمل مهاجرت انجام می‌گردد که پس از مهاجرت و انتقال بار سوئیچ منتخب، انحراف بار از میانگین بار برای هر دو کنترل‌کننده مبدأ و مقصد نسبت به قبل از مهاجرت کاهش یابد. در غیر این صورت، اگر انحراف از میانگین بعد از مهاجرت بیشتر یا مساوی قبل از مهاجرت گردد، عملیات مهاجرت کارآمد نخواهد بود و ترجیحاً عملیات توازن بار صورت نمی‌گیرد. از طرفی با اندازه‌گیری بار تمام کنترل‌کننده‌ها پس از مهاجرت سوئیچ و انتخاب بهترین خوشه برای مهاجرت برای کاهش بار کلی نتیجه بهتری حاصل می‌گردد. همچنین اولویت‌بندی در اجرای متراکم‌ترین کنترل‌کننده شبکه را سریع‌تر به حالت عادی خود برمی‌گرداند.

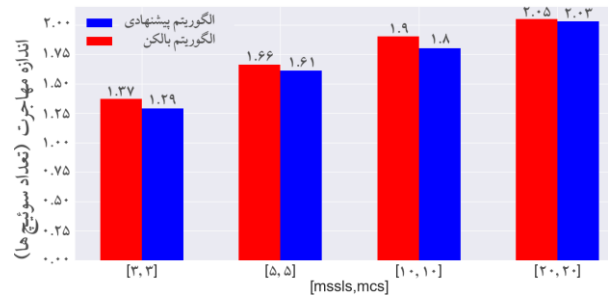
۷- نتیجه‌گیری

در روش پیشنهادی به ارائه روشی کارآمد، توزیعی و پایدار برای توازن بار پویا بین کنترل‌کننده‌ها در شبکه‌های مبتنی بر نرم‌افزار پرداخته شد. شبیه‌سازی‌ها نشان‌دهنده برتری نسبی این روش نسبت به روش‌های پیشین است. هرچند به دلیل عدم دسترسی به شاه‌سیم^{۳۱}‌های شهری (با نرخ ارسال بالا) جهت بررسی و آزمون ترافیک واقعی با اتصال آن‌ها به شبکه مورد آزمون، ممکن است

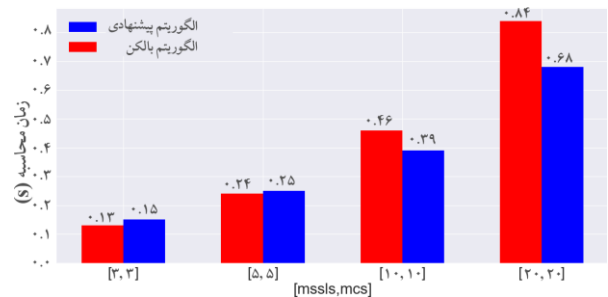
به کار بردن الگوریتم پیشنهادی و L_C^P بارها پس از اجرای روال پیشنهادی است. همچنین $C_m^* = \operatorname{argmax} L_C^{\operatorname{con}}$ ثابت کنترل‌کننده متراکم است و بار کنترل‌کننده متراکم برابر $L_C^{\operatorname{con}}(C_m^*)$ می‌باشد.

کاهش حداکثر بار کنترل‌کننده نیز به‌صورت درصدی ذیل محاسبه شده است:

$$\frac{\max L_C^P - \max L_C^{\operatorname{con}}}{\max L_C^{\operatorname{con}}} \cdot 100 \quad (26)$$



شکل ۱۴: مقایسه عملکرد نسخه‌های بالکن شامل mssls و mcs با روش پیشنهادی (اندازه مهاجرت).



شکل ۱۵: مقایسه عملکرد نسخه‌های بالکن شامل mssls و mcs با روش پیشنهادی (زمان محاسبه).

همان‌طور که در شکل ۱۴ نشان داده شده است، در روش پیشنهادی متوسط اندازه مهاجرت یا به عبارتی متوسط تعداد سوئیچ‌های منتقل شده به نسبت روش بالکن کمتر است. دلیل احتمالی می‌تواند وجود شرط بررسی اعتبار سوئیچ در مرحله خوشه‌بندی الگوریتم باشد؛ زیرا با حذف سوئیچ‌های نامناسب از انتقال پی‌درپی سوئیچ بین کنترل‌کننده‌ها که باعث تکرار بالای اجرای الگوریتم و کاهش کارایی مهاجرت می‌گردد، صرف‌نظر می‌کند و به‌این‌ترتیب به حل مشکل نوسان سوئیچ کمک می‌کند. اندازه مهاجرت پایین‌تر هزینه‌های انتقال را کاهش می‌دهد. زیرا هزینه اتصالات جدید کاهش می‌یابد. همچنین با توجه به شکل ۱۵ زمان محاسبه روش پیشنهادی در مقادیر [10,10] و [20,20] پایین‌تر از روش بالکن است و این امر به دلیل به‌کارگیری الگوریتم اکتشافی گرگ خاکستری در پیدا کردن کنترل‌کننده هدف و

افزایش تعداد گره‌ها در شبکه، کاهش در حداکثر بار کنترل‌کننده نسبت به روش بالکن بهبود می‌یابد. جدا از اعتبارسنجی سوئیچ قبل از انتقال، انتخاب متراکم‌ترین کنترل‌کننده بر اساس میزان بار تحمیلی از سوی درخواست‌های packet-In نیز می‌تواند در کاهش سریع‌تر بار شبکه در هر بار اجرای روال پیشنهادی تأثیرگذار باشد؛ هرچند انتخاب کنترل‌کننده مبدأ با روش فوق سربار اضافی در پردازش و ارتباط بین کنترل‌کننده‌ها ایجاد می‌کند. از طرفی اثر خوشه‌بندی نیز به دلیل انتخاب سوئیچ‌هایی با اتصالات قوی سبب می‌گردد تا هزینه انتقال و بار محاسباتی ایجادشده در جهت کشف همبندی جدید کاهش یابد. لذا بار کلی شبکه کاهش می‌یابد.

به‌طور کلی راه‌حل پیشنهادی با توجه به امکان مهاجرت بار تنها یک کنترل‌کننده در هر زمان و انتخاب سوئیچ مهاجر به لحاظ بهبود در توازن بار، از وقوع سربار در کنترل‌کننده مقصد و ناپایداری و مهاجرت‌های متوالی ناشی از آن جلوگیری می‌نماید. همچنین، به لحاظ هزینه ناشی از مهاجرت، تعادل را بین کارایی شبکه و هزینه انجام مهاجرت حفظ می‌نماید.

در کارهای آتی در نظر داریم مسئله صرفه‌جویی انرژی را به‌طور هم‌زمان با مسئله توازن بار در نظر بگیریم و بین آن‌ها موازنه ایجاد کنیم. برای این کار می‌توان از بهترین جواب الگوریتم گرگ خاکستری پس از جواب آلفا (یعنی جواب بتا) برای انتخاب کنترل‌کننده‌ای که کمترین بار را بعد از کنترل‌کننده هدف منتخب دارد استفاده کرد. لذا می‌توان بار حاصل از سوئیچ‌های این کنترل‌کننده را به یکی از دامنه‌های همسایه‌اش منتقل کرد تا بتوان سوئیچ‌های آن را جهت مصرف کمتر انرژی خاموش کرد. با تعیین یک آستانه برای افزایش بار کنترل‌کننده همسایه انتخاب‌شده و بررسی عدم سربار پس از انتقال کنترل‌کننده‌ای که قرار است بار حاصل از سوئیچ‌ها به آن منتقل شود، می‌توان از بروز یک کنترل‌کننده سربار دیگر در شبکه و کاهش کارایی آن جلوگیری کرد. همچنین در کارهای آتی در نظر گرفته شده که میزان پهنای باند شبکه پس از اجرای الگوریتم بررسی و مقایسه گردد.

مراجع

- [1] C.-Y. Hong et al., "Achieving high utilization with software-driven WAN," presented at the SIGCOMM'13: ACM SIGCOMM 2013 Conference, Aug. 2013.
- [2] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "On the effect of forwarding table size on SDN network utilization," presented at the IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, Apr. 2014.
- [3] A. Wang, Y. Guo, F. Hao, T. V. Lakshman, and S. Chen, "Scotch," presented at the CoNEXT '14: Conference on emerging Networking Experiments and Technologies, Dec. 2014.

نتایج به‌دست‌آمده با آزمون واقعی اندکی تفاوت داشته باشد. اما تلاش شده است تا با استفاده از روش‌های آماری همچون استفاده از تابع پواسون برای تولید نرخ ترافیک تصادفی تا حد امکان جواب نهایی به جواب واقعی و بهینه نزدیک باشد.

با بررسی میانگین تعداد بسته‌های packet-In دریافتی توسط کنترل‌کننده، روش پیشنهادی با افزایش نرخ ارسال به ۸.۳۳ درصد نسبت به روش بالکن از طریق انتقال مناسب سوئیچ‌ها توان عملیاتی کنترل‌کننده را بهبود می‌بخشد. از طرفی به دلیل سرعت اجرای روال در ایجاد توازن بار در شبکه، میانگین توان عملیاتی در ترافیک سنگین با سرعت کمتری افت پیدا می‌کند. همچنین این روش در مقایسه با روال بالکن زمان محاسبه را ۱۱.۸ درصد کاهش می‌دهد. وجود الگوریتم اکتشافی گرگ خاکستری به کاهش این زمان کمک می‌کند. زیرا هزینه پردازش مهاجرت مجازی را به طرز چشمگیری کاهش می‌دهد. بنابراین زمان محاسبه در این روش بهبود می‌یابد. در روال پیشنهادی به‌طور متوسط ۲.۳ سوئیچ منتقل می‌شوند که نشان‌دهنده برتری این روش نسبت به روش‌های پیشین است. دلیل این کاهش حجم مهاجرت بررسی اعتبار سوئیچ‌هاست. انتخاب حداقل سوئیچ‌هایی که باید منتقل شوند، هزینه مهاجرت را به دلیل کاهش هزینه‌های اتصال پس از مهاجرت کم می‌کند. لذا این روش هزینه انتقال را کاهش می‌دهد. همچنین در این روش با بررسی میانگین تعداد بسته‌های packet-In خروجی و بر اساس نرخ ورودی پیام‌های packet-In مشخص می‌گردد که زمان پاسخ به میزان ۱۳ درصد نسبت به روش بالکن کاهش می‌یابد. دلیل این کاهش، مهاجرت خوشه‌ای سوئیچ‌ها و اجرای سریع‌تر روال تعادل بار به دلیل زمان محاسبه پایین جهت کاهش بار کنترل‌کننده‌هاست. کاهش کلی زمان پاسخ می‌تواند تجربه استفاده از شبکه را بهبود بخشد و نیاز به بافر کردن بسته‌ها را در زمان تراکم پردازنده کاهش دهد. از طرفی، با اندازه‌گیری بار کنترل‌کننده‌ها در طی یک مدت‌زمان خاص می‌توان نتیجه گرفت که این روش تعداد اجرای روال پیشنهادی را کاهش می‌دهد که می‌تواند به دلیل انتخاب مناسب‌تر سوئیچ‌هایی باشد که قرار است منتقل شوند. اعتبارسنجی سوئیچ‌ها قبل از انتخاب آن‌ها برای انتقال مشکل نوسان سوئیچ را برطرف می‌سازد. لذا فواصل اجرای روال پیشنهادی طولانی‌تر و یکنواخت‌تر می‌گردد که خود در کاهش بار شبکه ناشی از پردازش‌های این روال تأثیرگذار است. همچنین به‌کارگیری الگوریتم گرگ خاکستری به کاهش بار محاسباتی پردازنده کمک می‌کند. در روال پیشنهادی بار کنترل‌کننده به نسبت روش بالکن به میزان ۳.۴۵ درصد کاهش می‌یابد. همچنین حداکثر بار کنترل‌کننده نیز به میزان ۱.۵۵ درصد کاهش می‌یابد. از شبیه‌سازی‌ها می‌توان نتیجه گرفت که با

- [12] A. Montazerolghaem and M. H. Yaghmaee, "Load-Balanced and QoS-Aware Software-Defined Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3323–3337, Apr. 2020.
- [13] Y. Kulakov, A. Kohan, S. Kopychko, and R. Cherevatenko, "Load Balancing in Software-Defined Networks Using Multipath Routing," in *Advances in Computer Science for Engineering and Education III*, Springer International Publishing, pp. 384–395, January 2021.
- [14] Y. Zhao, X. Wang, Q. He, C. Zhang, and M. Huang, "PLOFR: An Online Flow Route Framework for Power Saving and Load Balance in SDN," *IEEE Systems Journal*, vol. 15, no. 1, pp. 526–537, Mar. 2021.
- [15] M. Cello, Y. Xu, A. Walid, G. Wilfong, H. J. Chao, and M. Marchese, "BalCon: A Distributed Elastic SDN Control via Efficient Switch Migration," presented at the 2017 IEEE International Conference on Cloud Engineering (IC2E), Apr. 2017.
- [16] Y. Xu et al., "Dynamic Switch Migration in Distributed Software-Defined Networks to Achieve Controller Load Balance," *IEEE J. Select. Areas Commun.*, vol. 37, no. 3, pp. 515–529, Mar. 2019.
- [17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014.
- [18] T. Hu, J. Zhang, L. Wang, and D. Qiao, "An Improved Switch Migration Approach to Controller Load Balancing in SDN," presented at the 2017 2nd International Symposium on Advances in Electrical, Electronics and Computer Engineering (ISAEECE 2017), 2017.
- [19] A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, and R. Kompella, "Towards an elastic distributed SDN controller," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 7–12, Sep. 2013.
- [20] O. Bozorg-Haddad, Ed., "Advanced Optimization by Nature-Inspired Algorithms," Springer Singapore, 2018.
- [4] J. Yu, Y. Wang, K. Pei, S. Zhang, and J. Li, "A load balancing mechanism for multiple SDN controllers based on load informing strategy," presented at the 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), Oct. 2016.
- [5] C. Liang, R. Kawashima, and H. Matsuo, "Scalable and Crash-Tolerant Load Balancing Based on Switch Migration for Multiple Open Flow Controllers," presented at the 2014 Second International Symposium on Computing and Networking (CANDAR), Dec. 2014.
- [6] A. A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, and R. Kompella, "ElastiCon," presented at the the tenth ACM/IEEE symposium, 2014.
- [7] Y. Zhou, Y. Wang, J. Yu, J. Ba, and S. Zhang, "Load balancing for multiple controllers in SDN based on switches group," presented at the 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), Sep. 2017.
- [8] F. P.-C. Lin and Z. Tsai, "Hierarchical Edge-Cloud SDN Controller System With Optimal Adaptive Resource Allocation for Load-Balancing," *IEEE Systems Journal*, vol. 14, no. 1, pp. 265–276, Mar. 2020.
- [9] F. Al-Tam and N. Correia, "On Load Balancing via Switch Migration in Software-Defined Networking," *IEEE Access*, vol. 7, pp. 95998–96010, 2019.
- [10] W. Liu, Y. Wang, J. Zhang, H. Liao, Z. Liang, and X. Liu, "AAMcon: an adaptively distributed SDN controller in data center networks," *Front. Comput. Sci.*, vol. 14, no. 1, pp. 146–161, Mar. 2019.
- [11] K. Rupani, N. Punjabi, M. Shamdassani, and S. Chaudhari, "Dynamic Load Balancing in Software-Defined Networks Using Machine Learning," in *Proceeding of International Conference on Computational Science and Applications*, Springer Singapore, 2020, pp. 283–292.

پاورقی‌ها:

- ¹ Software-Defined Network
- ² Quality of Service
- ³ Data Plane
- ⁴ Uplink
- ⁵ Downlink
- ⁶ Migration
- ⁷ Streaming Networks
- ⁸ Content Delivery Networks (CDN)
- ⁹ Partition
- ¹⁰ Staged Switch Migration Strategy (SSMS)
- ¹¹ Grey Wolf Optimization Algorithm
- ¹² Optimal Controller Load Balancing
- ¹³ BalCon
- ¹⁴ BalConPlus
- ¹⁵ Path Computation Load
- ¹⁶ Rules Installation Load
- ¹⁷ Responsiveness
- ¹⁸ Graph Partitioning Problem
- ¹⁹ Max Cluster Size
- ²⁰ Max Starting Switch List Size
- ²¹ Round-Robins
- ²² Tensor
- ²³ AnaConda
- ²⁴ Spyder
- ²⁵ Mininet
- ²⁶ Ryu
- ²⁷ OpenvSwitch
- ²⁸ Ubuntu
- ²⁹ Numpy
- ³⁰ Fitness Function
- ³¹ Trunk