مقاله پژوهشی

مروری بر مکان‌یابی کنترلرها در شبکه‌های تعریف‌شده نرم‌افزاری

مهدی سربازی و محمد فتحی

چکیده: شبکه تعریف‌شده نرم‌افزاری یکی از جدیدترین پیشرفت‌ها در زمینه شبکه‌های کامپیوتری است که با جداسازی سطح داده از سطح کنترل توانسته است منابع شبکه را به‌خوبی مدیریت کند. کنترلرها در این شبکه‌ها، سیستم عامل شبکه هستند و مدیریت کل شبکه را با ارائه خدمات به برنامه‌های کاربردی بر عهده دارند. لایه کنترل در شبکه‌های پیچیده متشکل از چندین کنترلر است، ولی تعیین تعداد مناسب و مکان بهینه برای نصب کنترلرها در شبکه مسئله مهمی است. تعداد کنترلر بهینه مورد نیاز و بحث تعیین مکان بهینه برای نصب کنترلرها در شبکه در تحقیق‌های مختلفی بررسی شده‌اند؛ اما تعدادی از پارامترهای اساسی شبکه در این تحقیق‌ها نادیده گرفته شده‌اند و با وجود راه‌حل‌های ارائه‌شده، هنوز مسائل باز و چالش‌های حل‌نشده‌ای در مکان‌یابی کنترلرهای شبکه تعریف‌شده نرم‌افزاری وجود دارد که نیاز به توجه محققان دارد. این تحقیق مسئله مکان‌یابی کنترلرها را معرفی می‌کند و تعدادی از تحقیق‌های اخیر در این زمینه را دسته‌بندی و بررسی می‌نماید. در ادامه چالش‌ها و مسائل باز این مسئله مطرح می‌گردد و پیشنهادهایی برای تحقیق‌های آینده به محققان ارائه می‌شود.

کلیدواژه: **شبکه تعریف‌شده نرم‌افزاری، راه‌حل‌های فراابتکاری، مکان‌یابی کنترلر، یادگیری ماشین.**

۱- مقدمه

شبکه تعریف‌شده نرم‌افزاری، فناوری جدیدی است که طراحی شبکه مبتنی بر نرم‌افزار را ارائه می‌دهد. این تکنولوژی با جداکردن سطح کنترل از دستگاه‌های سطح داده، مدیریت شبکه رایانه‌ای را ساده‌تر می‌کند. جداسازی سطح کنترل از سطح داده با پشتیبانی از قابلیت برنامه‌ریزی شبکه، موجب سادگی در پیکربندی و کنترل شبکه شده و مدیریت شبکه را کارآمدتر می‌کند [۱].

معماری مرجع شبکه تعریف‌شده نرم‌افزاری در شکل ۱ نشان داده شده است. این معماری سه لایه دارد که شامل یک لایه زیرساخت در پایین، یک لایه کنترل در وسط و یک لایه کاربرد در بالا می‌باشد [۲]. لایه زیرساخت از عناصر هدایتی مانند مسیریاب‌ها، سوئیچ‌ها، نقاط دسترسی بی‌سیم و ... تشکیل شده است. این دستگاه‌ها وظیفه جمع‌آوری اطلاعات وضعیت شبکه و ارسال آنها به کنترلر را دارند. آنها همچنین مسئول ارسال بسته‌ها با استفاده از قوانین دریافت‌شده از کنترلر هستند. دستگاه‌های ارسال داده با استفاده از رابط باند شمالی با کنترلر ارتباط برقرار می‌کنند.

این مقاله در تاریخ ۲۹ مهر ماه ۱۴۰۲ دریافت و در تاریخ ۱۲ فروردین ماه ۱۴۰۳ بازنگری شد.

مهدی سربازی، گروه امور فناوری اطلاعات و امنیت فضای مجازی، معاونت اداری و مالی، دانشگاه کردستان، سنندج، ایران، (email: sarbazi.mahdi@gmail.com).

محمد فتحی (نویسنده مسئول)، گروه مهندسی برق، دانشکده مهندسی، دانشگاه کردستان، سنندج، ایران، (email: mfathi@uok.ac.ir).

هر سوئیچ یک یا چند جدول جریان را نگه می‌دارد که ورودی‌های آنها توسط کنترلر تأمین می‌شود. هر ورودی جدول جریان از سه قسمت تشکیل می‌شود: شناسه ورودی، اقدام و آمار. پس از ورود بسته، سرایند بسته با ورودی‌های جدول جریان مطابقت داده می‌شود و عملکرد مربوط به آن ورودی انجام می‌شود؛ مانند ارسال رو به جلوی بسته به یک پورت یا رهاکردن بسته. در صورت عدم تطابق، بسته برای پردازش به کنترلر ارسال می‌شود [۲].

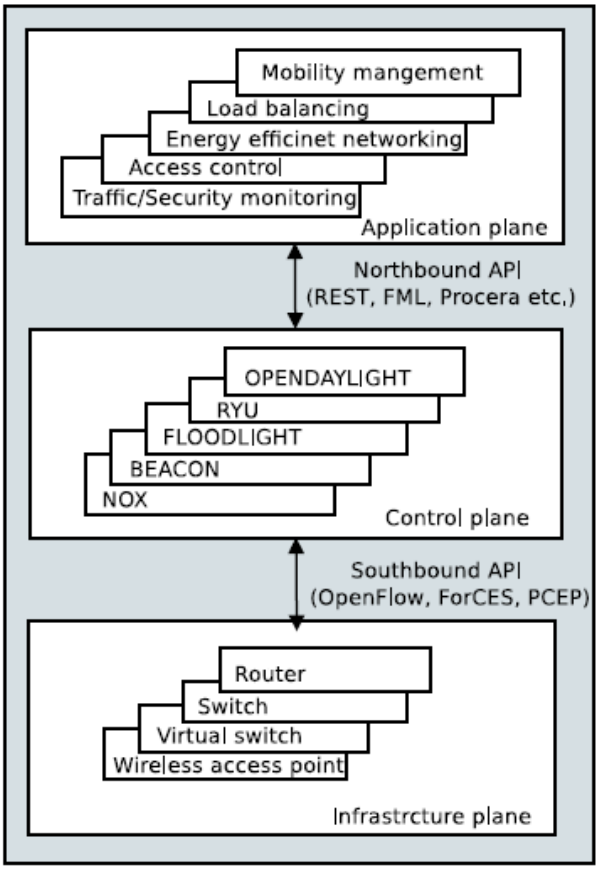
لایه کنترل از یک یا چند کنترلر برای کنترل عناصر زیرساخت تشکیل شده است. این کنترلرها با استفاده از اطلاعات جمع‌آوری‌شده توسط دستگاه‌های ارسال، وظیفه حفظ نمای کلی شبکه را دارند. آنها همچنین مسئول ارائه قوانین ارسال به سوئیچ‌ها هستند. کنترلرها با دستگاه‌های ارسال داده با استفاده از رابط باند جنوبی به مانند OpenFlow ارتباط برقرار می‌کنند [۳]. وقتی چندین کنترلر در لایه کنترل وجود داشته باشد، آنها با استفاده از رابط‌های متصل به باند شرق و غرب با یکدیگر ارتباط برقرار می‌کنند. POX [۴]، Ryu [۵]، Maestro [۶] و Floodlight [۷]، برخی از پیاده‌سازی‌های کنترلرهای باز و جامعه‌محور هستند که از استاندارد OpenFlow استفاده می‌کنند [۲].

شبکه تعریف‌شده نرم‌افزاری با اجرای قابلیت‌هایش به عنوان یک برنامه در نرم‌افزار بر روی سرور، استفاده از جعبه‌های میانی مانند متعادل‌کننده‌های بار، دیوارهای آتش و NAT را از بین می‌برد. این برنامه‌ها به وسیله کنترلرها با استفاده از رابط شمالی به هم متصل می‌شوند؛ مانند انتقال حالت بازنمایی[1]. همان طور که در شکل ۱ آمده است، لایه کاربرد بالای لایه کنترل است. مدیریت تحرک [۸]، کنترل دسترسی [۹]، بهره‌وری انرژی [۱۰]، مهاجرت ماشین مجازی [۱۱]، مسیریابی تطبیقی، تعادل بارگذاری [۱۲]، چندپخشی [۱۳]، مجازی‌سازی شبکه [۱۴] و نظارت امنیتی [۱۵] برخی از ویژگی‌هایی هستند که می‌توانند در لایه کاربرد پیاده‌سازی شوند [۲].

رابط‌های باند جنوبی مانند ForCES، Open-Flow، PCEP و غیره پروتکل ارتباطی بین گره‌های ورودی و عناصر سطح کنترل را تعریف می‌کنند. رابط‌های باند شمالی مانند REST، FML، Procera و غیره رابطی برای توسعه برنامه‌ها به توسعه‌دهندگان برنامه ارائه می‌دهند. رابط‌های متصل به شرق/غرب مانند ALTO، Hyperflow و غیره پروتکل ارتباطی بین چندین کنترلر در شبکه را تعریف می‌کنند [۲].

شبکه تعریف‌شده نرم‌افزاری [۱۶] اولیه با کنترلر منفرد و متمرکز توسعه یافته بود که هدف اصلی آن نمایش کل شبکه بود؛ اما در صورت خراب‌شدن کنترلر مرکزی قابلیت اطمینان و مقیاس‌پذیری را از دست می‌داد. به همین دلیل شبکه تعریف‌شده نرم‌افزاری با چندین کنترلر ایجاد شد و از این رو سطح کنترل باید با تمام کنترلرهای توزیع‌شده منطقی

1. REST



شکل ۱: معماری شبکه تعریف‌شده نرم‌افزاری [۲].

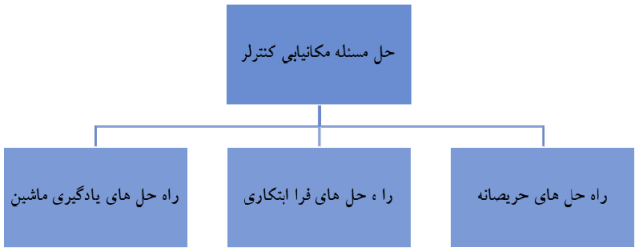
همگام شود. مکان‌یابی یک کنترلر در شبکه فعال تعریف‌شده نرم‌افزاری کار ساده‌ای نیست؛ زیرا شناسایی و عادی‌سازی پارامتر از مجموعه پارامترهایی که فرد می‌تواند بر اساس نیاز خود به دنبال راه‌حل باشد، دشوار است. بنابراین با پارامترهای بیشتر، مسئله پیچیده‌تر می‌شود و ثابت می‌شود که یک مسئله NP-Hard است. برای مکان‌یابی کنترلرها در شبکه‌های فعال شبکه تعریف‌شده نرم‌افزاری باید این پارامترها را در ذهن داشت و باید بر این اساس تصمیم گرفت. برخی از مهم‌ترین پارامترهای مؤثر بر کارایی این فرایند، قابلیت اطمینان، در دسترس بودن شبکه، تأخیر انتشار، امنیت، تعادل بار و مصرف انرژی است.

کنترلرهای شبکه تعریف‌شده نرم‌افزاری، مغز شبکه در نظر گرفته می‌شود. کنترلرها سیستم عامل‌های شبکه هستند و مدیریت کل شبکه را با ارائه خدمات به برنامه‌های کاربردی کاربر بر عهده دارند تا کاربر شبکه بتواند با سخت‌افزار شبکه و با سایر دستگاه‌های کل شبکه ارتباط برقرار کند. کنترلرها انواع مختلفی دارند. کنترلرهای شبکه تعریف‌شده نرم‌افزاری را می‌توان در دو دسته تقسیم کرد:

۱) کنترلرهای فیزیکی متمرکز و کنترلرهای فیزیکی توزیع‌شده

۲) کنترلرهایی که از نظر فیزیکی متمرکز هستند. در ابتدا شبکه تعریف‌شده نرم‌افزاری با کنترلر متمرکز از نظر فیزیکی توسعه یافت؛ اما به دلیل شکست در مقیاس‌پذیری، معماری توزیع‌شده فیزیکی بر آن ترجیح داده شد. برای افزایش عملکرد کنترلر متمرکز از نظر فیزیکی از تکنیک‌های چندرشته‌ای برای تقسیم منطقی یک کنترلر واحد استفاده شد [۱۷].

توزیع فیزیکی را می‌توان به توزیع منطقی و متمرکز منطقی تقسیم کرد. توزیع منطقی از معماری چند کنترلر پشتیبانی می‌کند که هر کنترلر دارای نمای زیرشبکه محلی است و به آن متصل و پاسخگو است. توزیع فیزیکی با تمرکز منطقی نیز از معماری چند کنترلر با مسئولیت‌های



شکل ۲: دسته‌بندی روش‌های حل مسئله مکان‌یابی کنترلر.

توزیع‌شده در بین آنها پشتیبانی می‌کند و فقط مانند یک کنترلر برای لایه زیرین به نظر می‌رسد. همه کنترلرها برای پذیرش هر گونه تغییر در شبکه و داشتن یک نمای کلی از کل شبکه هماهنگ شده‌اند [۱۷].

خصوصیات اصلی معماری شبکه‌های تعریف‌شده نرم‌افزاری، جداسازی سطح کنترل و سطح داده‌ها و نیز تعریف سطح کنترل متمرکز منطقی است. توپولوژی شبکه جدید، مزایای قابل توجهی را برای مرکز داده‌های ابری و ارائه‌دهندگان خدمات در شبکه‌های گسترده فراهم می‌کند؛ زیرا باز ارسال داده‌ها را ساده و امکان مدیریت یک شبکه را به روشی انعطاف‌پذیر فراهم می‌کند. با این حال، شبکه تعریف‌شده نرم‌افزاری تحت تأثیر چالش‌هایی است که باید به طور کلی رفع شود. برای به دست آوردن یک توپولوژی متعادل و مقاوم در برابر شکست، لازم است که حداقل تعداد کنترلرها و محل قرارگیری آنها تعیین شود [۱۸]؛ ضمن این که اطمینان از تأخیر کم بین گره‌ها و کنترلرهای مرتبط و همچنین بین کنترلرها مورد توجه باشد. به طور همزمان لازم است که میزان پردازش کنترلرها و در حد توان عملیاتی آنها متعادل باشد [۱۹]. بنابراین هر راه‌حل پیشنهادی برای مسئله مکان‌یابی کنترلر شبکه تعریف‌شده نرم‌افزاری (CPP) باید با در نظر گرفتن موارد زیر به تعداد کنترلر مورد نیاز و محل قرارگیری آنها بپردازد [۱۸]: الف) ظرفیت کنترلرها، ب) تأخیر انتشار بین کنترلرها و گره‌ها، و بین کنترلرها و ج) توزیع بار بین کنترلرها. راه‌حل‌های خوب، توانایی شبکه در واکنش سریع به حوادث را افزایش می‌دهد و منجر به توپولوژی قوی‌تری می‌شود؛ حتی در صورت بروز خرابی. در شبکه تعدادی سوئیچ و مکان آنها داده می‌شود. هدف اصلی از مسئله مکان‌یابی کنترلر این است که اولاً تعداد دقیق کنترلرهای مورد نیاز را شناسایی و دوم این که محل نصب کنترلرها را در توپولوژی تعیین کنیم [۲۰]. مکان‌یابی کنترلرها کار ساده‌ای نیست؛ زیرا برخی از مؤلفه‌های حیاتی وجود دارند که تأثیر این روش را تحت تأثیر قرار می‌دهند. برخی از آنها به شرح زیرند:

- قابلیت اطمینان شبکه
- تأخیر انتشار (تأخیر)
- بار در کنترلر
- مصرف انرژی

برای به‌دست آوردن بهترین جایگاه کنترلرها در شبکه داده‌شده باید بین این معیارها توازن برقرار کرد، مسئله را با توجه به نیاز آنها انتخاب و فرموله کرد و بهترین روش ممکن را انتخاب نمود [۱۷].

هدف این تحقیق، معرفی مسئله مکان‌یابی کنترلرهای شبکه نرم‌افزاری است. بدین منظور پس از معرفی مسئله در بخش ۱، تعدادی از تحقیقات انجام‌شده در این زمینه در سه گروه راه‌حل‌های با رویکرد یادگیری ماشین، حریصانه و فراابتکاری دسته‌بندی شده و در بخش ۲ بررسی می‌شود. در ادامه در بخش ۳ چالش‌ها و مسائل باز این مسئله مطرح می‌گردد. بخش ۴ جمع‌بندی و بخش ۵ پیشنهادهایی را برای تحقیق‌های آینده ارائه می‌دهد.

۲- کارهای پیشین

۲-۱ راه‌حل‌های یادگیری ماشین

در سال‌های اخیر، تحقیقات زیادی در زمینه مرور مکان‌یابی کنترلر در شبکه تعریف‌شده نرم‌افزاری انجام گردیده است. مقالات زیر به مطالعه و بررسی مروری این موضوع می‌پردازند:

مقاله مروری [۲۱]: این مقاله مطالعه‌ای جامع را بر استراتژی‌های مکان‌یابی کنترلر ارائه می‌دهد. نویسندگان مقاله مزایا و معایب هر استراتژی را به بحث می‌گذارند و فاکتورهای کلیدی را که باید در هنگام انتخاب استراتژی مکان‌یابی در نظر گرفته شوند شناسایی می‌کنند.

مقاله مروری [۲۲]: این مقاله بر روی چالش‌های مکان‌یابی کنترلر در شبکه‌های بزرگ و پیچیده تمرکز دارد. نویسندگان مقاله، الگوریتم‌های مختلفی را برای حل این چالش‌ها مرور و بررسی می‌کنند.

مقاله مروری [۲۳]: این مقاله نیز مطالعه‌ای بر استراتژی‌های مکان‌یابی کنترلر با تمرکز بر معیارهای عملکرد ارائه می‌دهد. نویسندگان مقاله به بررسی تأثیر مکان‌یابی کنترلر به کمک معیارهایی مانند تأخیر، پهنای باند و قابلیت اطمینان می‌پردازند.

مقاله مروری [۲۴]: این مقاله طبقه‌بندی جامعی از مسائل مربوط به مکان‌یابی کنترلر ارائه می‌دهد. نویسندگان مقاله به بررسی روش‌های مختلف حل این مسائل نیز می‌پردازند.

مقاله مروری [۲۵]و۷: این مقاله به بررسی چالش‌های مکان‌یابی کنترلر در شبکه‌های با منابع محدود می‌پردازد. نویسندگان مقاله الگوریتم‌های مختلفی را برای حل این چالش‌ها ارائه می‌دهند.

مقاله مروری [۲۶]: این مقاله مطالعه‌ای بر استراتژی‌های مکان‌یابی کنترلر در شبکه‌های ارائه‌دهندگان خدمات اینترنت[1] و مخابراتی ارائه می‌دهد. نویسندگان مقاله به بررسی چالش‌های خاص این نوع شبکه‌ها می‌پردازند و راه‌حل‌های مناسب برای آنها ارائه می‌دهند.

این بخش به طور مختصر برخی از کارهای مربوط به مکان‌یابی کنترلر شبکه تعریف‌شده نرم‌افزاری را مرور می‌کند. بدین منظور، تعدادی از پژوهش‌های اخیر در این زمینه بر اساس نوع راه‌حل آنها دسته‌بندی شده‌اند. همان طور که در شکل ۲ آمده، راه‌حل‌ها ممکن است بر پایه روش‌های یادگیری ماشینی، حریصانه یا فرابتکاری باشند. روش‌های زیادی برای تعیین مکان‌های بهینه کنترلر یا با تبدیل مسئله به مدل ریاضی یا با تطبیق یک رویکرد فرابتکاری ارائه شده‌اند. برخی روش‌های حریصانه را برای مکان‌یابی کنترلر پیشنهاد کردند. از بررسی ادبیات بدیهی است که بیشتر کارهای پیشنهادی برای مکان‌یابی بهینه کنترلر، تعداد کنترلرها را به عنوان پارامتر ورودی در نظر گرفته‌اند. هر روش پیشنهادی مزایا و معایب خاص خود را دارد. تعداد کمی از آثار به موضوع شناسایی تعداد کنترلر مورد نیاز برای مکان‌یابی در شبکه پرداخته‌اند که در ادامه به مرور تعدادی از روش‌های پیشنهادی جدید می‌پردازیم.

۲-۲ راه‌حل‌های یادگیری ماشین

مسئله مکان‌یابی کنترلرها اولین بار در سال ۲۰۱۲ در [۱۸] معرفی شد. نویسندگان با هدف به حداقل رساندن تأخیر انتشار از گره‌ها به کنترلر اختصاص داده شده، محل قرارگیری کنترلرها در شبکه تعریف‌شده نرم‌افزاری را مطالعه کردند. رویکرد آنها شامل استفاده از الگوریتم CPP در طیف وسیعی از مسائل محلی‌سازی بود؛ مانند مسائل K-Median و K-Center. هدف آنها به ترتیب به حداقل رساندن تأخیر انتشار متوسط و تأخیر انتشار در بدترین حالت بود [۲۷].

گوا و همکاران تأثیر قرارگیری کنترلر بر انعطاف‌پذیری شبکه را با نمودارهای وابستگی متقابل و تجزیه و تحلیل شکست آبشار بررسی کردند و یک روش تقسیم‌بندی و انتخاب را برای مکان‌یابی کنترلر برای بهبود انعطاف‌پذیری پیشنهاد دادند [۲۸]. هاک و همکاران یک چارچوب انعطاف‌پذیر برای مقابله با تأثیرات قطع‌شدن پیوندها در شبکه‌های تعریف‌شده نرم‌افزاری معرفی کردند و یک روش مکان‌یابی کنترلر بهینه مبتنی بر پارتو[2] را برای به حداکثر رساندن تأخیرهای گره به کنترلر و انعطاف‌پذیری از نظر تحمل خرابی و تعادل بار پیشنهاد کردند [۲۹].

نویسندگان در [۳۰]، یک شبکه مبتنی بر شبکه تعریف‌شده نرم‌افزاری را در نظر گرفتند که دارای یک کنترلر واحد است و تمام سوئیچ‌ها با استفاده از یک درخت به کنترلر متصل هستند. هر سوئیچ به یک گره اصلی در درخت متصل است. در صورتی که یک سوئیچ از طریق تغییر مسیر سریع محلی از خرابی گره اصلی خود محافظت شود، وزن آن صفر در نظر گرفته می‌شود و در غیر این صورت، وزن آن برابر با تعداد سوئیچ‌هایی است که در زیرمجموعه آن سوئیچ قرار دارند. در این تحقیق، مسئله یافتن درخت کنترلر با حداقل وزن (mwCT) بررسی می‌شود؛ به طوری که وزن درخت حاصل‌جمع تمام وزن‌های سوئیچ است. چهار کار اصلی در این تحقیق انجام شده است. اول، یک مکانیزم جدید محافظت از همزادها برای به حداکثر رساندن تعداد سوئیچ‌های محافظت‌شده طراحی شده است. دوم ثابت می‌شود که mwCT مسئله NP سخت است. سوم برنامه‌ریزی خطی عدد صحیح برای حل مسئله mwCT فرموله شده است. سرانجام یک الگوریتم ابتکاری کارآمد برای ساخت درخت کنترلر به نام درخت مرتب‌شده با درجه فاصله (DDOT) ارائه شده است. برخلاف برنامه‌ریزی خطی عدد صحیح، DDOT نه‌تنها وزن درخت بلکه میانگین فاصله کنترلر تا سوئیچ را نیز به حداقل می‌رساند.

میشرا و همکاران [۳۱]، مکانیزمی مبتنی بر آنتروپی را برای مکان‌یابی کنترلرها و درک اینکه شبکه مورد حمله قرار گرفته است، ارائه کرده‌اند. این مقاله تکنیک کاهش را نیز برای کاهش شدت حمله ارائه می‌کند.

۲-۳ راه‌حل‌های حریصانه

روش حریصانه یا الگوریتم حریصانه از روش‌های مشهور و پرکاربرد طراحی الگوریتم‌هاست که با ساختاری ساده در حل بسیاری از مسائل استفاده می‌شود. این روش اغلب در حل مسائل بهینه‌سازی استفاده شده و در پاره‌ای مواقع جایگزین مناسبی برای روش‌هایی مانند برنامه‌ریزی پویا است. در حالت کلی این روش سرعت و مرتبه اجرایی بهتری نسبت به روش‌های مشابه خود دارد؛ اما متناسب با مسئله ممکن است به یک جواب بهینه سراسری ختم نشود. این الگوریتم‌ها در مسئله مکان‌یابی کنترلرها مورد استفاده قرار گرفته‌اند.

هو و همکاران [۳۲] در شبکه تعریف‌شده نرم‌افزاری، مکان‌یابی قابل اعتماد کنترلرها را معرفی کردند. آنها مکان‌یابی بر اساس brute force، مکان‌یابی تصادفی و الگوریتم‌های حریصانه را برای مکان‌یابی قابل اعتماد کنترلرها ارائه کردند. در مطالعه‌ای دیگر هو و همکاران معیاری را به نام درصد پیش‌بینی‌شده از دست دادن مسیر کنترل به دلیل عدم موفقیت یک جزء شبکه برای توصیف قابلیت اطمینان شبکه‌های تعریف‌شده نرم‌افزاری معرفی کردند و الگوریتم ابتکاری $w-l-$حریصانه را برای تجزیه و

1. ISP

2. Pareto

تحلیل معامله بین قابلیت اطمینان و تأخیر پیشنهاد دادند. انتظار می‌رود از دست دادن مسیر کنترل به تعداد مسیرهای کنترل عبوری از یک جزء و احتمال خرابی مؤلفه مربوط باشد [۳۳].

نویسندگان [۳۴] مسئله مکان‌یابی کنترلر شبکه تعریف‌شده نرم‌افزاری را برای خرابی‌های تک‌پیوندی و چندپیوندی در نظر گرفتند. الگوریتم مکان‌یابی حریص (GPA) را برای خرابی‌های تک‌پیوندی با هدف کاهش هزینه‌های محاسباتی تعریف کردند. برای خرابی‌های چندپیوندی، با توجه به آن که سناریوهای خرابی پیوندهای بیشتری برای خرابی‌های چندلینک وجود دارد، برای کاهش سربار محاسباتی، شبیه‌سازی مونت‌کارلو و GPA را برای شبیه‌سازی سناریوهای خرابی چندلینک با توجه به نرخ خرابی لینک، ترکیب کرده و کنترلرها را به صورت حریصانه و تکراری مکان‌یابی کردند. نتایج تجربی روی توپولوژی شبکه واقعی نشان داد که GPA و شبیه‌سازی مونت‌کارلو روش‌های کارآمد و قابل اعتمادی برای حل مسئله مکان‌یابی کنترلر برای خرابی‌های تک‌پیوندی و چندپیوندی هستند.

مرجع [۳۵] نیز یک الگوریتم حریصانه برای قرارگیری پویای کنترلر در شبکه‌های نرم‌افزاری تعریف‌شده ارائه می‌دهد. این الگوریتم با در نظر گرفتن ترافیک شبکه، کنترلرها را به گونه‌ای قرار می‌دهد که بار ترافیک بین آنها به طور متعادل تقسیم شود.

مرجع [۳۶] یک مدل ریاضی را برای مسئله قرارگیری کنترلر در شبکه‌های نرم‌افزاری تعریف‌شده ارائه می‌دهد. این مدل هزینه راه‌اندازی شبکه و تأخیر بسته‌های کنترلی را در نظر می‌گیرد و سپس یک الگوریتم حریصانه برای حل این مدل ارائه می‌شود.

۲-٤ راه‌حل‌های فراابتکاری

از انواع الگوریتم‌های تصادفی، الگوریتم‌های فراابتکاری، فراتکاملی یا فرااکتشافی هستند که برای یافتن پاسخ بهینه به کار می‌روند. لانژ و همکاران، چارچوب POCO را با استفاده از روش‌های اکتشافی گسترش دادند تا از شبکه‌های بزرگ یا شبکه‌های پویا با ویژگی‌هایی که با گذشت زمان تغییر می‌کنند، پشتیبانی کند [۳۷]. مولر برای افزایش بقای شبکه‌ها استراتژی مکان‌یابی کنترلر به نام Survivor را ارائه کرد. این استراتژی با کشف تنوع مسیر، تعداد مسیرهای جداگانه بین سوئیچ‌ها و کنترلرها را به حداکثر می‌رساند [۳۸].

در [۳۹]، یک چارچوب فراابتکاری برای تخصیص کنترلر پویا برای نسل ۵ فناوری تلفن همراه (G5) پیشنهاد شد. در این چارچوب از یک نسخه پرآشوب از بهینه‌ساز گرگ خاکستری (GWO) استفاده گردید که یک الگوریتم جدید متغیر مبتنی بر PSO است. نتایج شبیه‌سازی نشان داد که این چارچوب می‌تواند به نتایج قابل قبول و همچنین دقیقی دست یابد و به دلیل مقیاس‌پذیری و نیز سادگی الگوریتم فراابتکاری معرفی‌شده، می‌تواند در هر مسئله مرتبط دیگری مورد استفاده قرار گیرد. بهینه‌ساز گرگ خاکستری پرآشوب با الگوریتم فراابتکاری PSO مقایسه شد و نتایج نشان داد که GWO پرآشوب می‌تواند در طول تکرار کمتر در مقایسه با PSO نتایج دقیق‌تری بدهد.

در [۴۰] چارچوبی برای تعیین و پیش‌بینی تعداد کنترلرهای مورد نیاز در شبکه‌های تعریف‌شده نرم‌افزاری بر اساس مدل SRGM ارائه شده است. این مدل قادر به شبیه‌سازی فرایند تصادفی ظهور باگ در کنترلرهای شبکه‌های تعریف‌شده نرم‌افزاری منبع باز است. در این تحقیق برای مدیریت کارایی شبکه‌های تعریف‌شده نرم‌افزاری، روی اپلیکیشن‌های مختلف چارچوب SRGM تمرکز شده است. در این تحقیق راهنماهایی برای اپراتورهای شبکه فراهم شده تا بر اساس نیازهای قابلیت اطمینان اپلیکیشن‌های شبکه، تصمیم بگیرند چه زمانی لازم است از نرم‌افزار کنترلر در محیط عملیاتی استفاده کنند.

در [۴۱] برای مکان‌یابی کنترلرها با هدف دستیابی به یک تعادل بین تأخیر متوسط و زمان فرایند، یک الگوریتم جدید و گسسته مبتنی بر الگوریتم جستجوی فاخته ارائه شده است. شبیه‌سازی در دو توپولوژی شبکه انجام می‌شود که شبیه دنیای واقعی است. نتایج نشان می‌دهند که الگوریتم جستجوی فاخته جدید عملکرد بهتری نسبت به الگوریتم ژنتیک و بهینه‌ساز تجمیع ذرات دارد.

در [۴۲] یک الگوریتم بهینه‌سازی پویا که مبتنی بر الگوریتم بهینه‌سازی ازدحام (SSOA) است با معرفی نقشه‌های آشفته برای افزایش عملکرد بهینه‌ساز توسعه یافته است. این الگوریتم تعداد بهینه کنترلرها و اتصالات بهینه بین سوئیچ‌ها و کنترلرها در شبکه‌های تعریف‌شده نرم‌افزاری در مقیاس بزرگ را به صورت پویا ارزیابی می‌کند. نتایج شبیه‌سازی نشان می‌دهند که الگوریتم پیشنهادی از نظر زمان اجرا و قابلیت اطمینان بهتر از الگوریتم‌های فراابتکاری و یک الگوریتم مبتنی بر تئوری بازی عمل می‌کند.

نویسندگان در [۴۳] روی مسئله مکان‌یابی کنترلر برای شبکه توزیع‌شده و به طور خاص برای شبکه اینترنت اشیا تمرکز می‌کنند و مفهوم زیرماژول را برای حل کلاس مسئله مکان‌یابی کنترلر که در سیستم‌های واقعی متداول است، اعمال می‌کنند. چارچوب پیشنهادی را از مبانی اساسی تا مشکلات پیچیده مکان‌یابی کنترلر تحت نظریه زیرماژول ارائه می‌دهند. نویسندگان جنبه‌های مختلفی از مسئله مکان‌یابی کنترلر را تحت مفهوم زیرماژول بررسی می‌کنند تا نشان دهند این، چارچوبی امیدوارکننده برای تدوین و پرداختن به اهداف مختلف ارائه‌دهندگان شبکه است. برای هر فرمول، یک الگوریتم ابتکاری برای پرداختن به آن بر اساس الگوریتم معروف Nemhauser ارائه می‌دهند تا عملکرد زیرماژول از پیش تعریف‌شده را به حداکثر برساند. از طریق شبیه‌سازی گسترده، عملکرد و کارایی مدل تحت تنظیمات مختلف شبکه ارزیابی می‌شود. ارزیابی نتیجه برجسته چارچوب را از نظر زمان اجرا، تعداد کنترلرها و تأخیر شبکه نشان می‌دهد. حتی اگر روش زیرماژول با مسئله مکان‌یابی کنترلر مطابقت داشته باشد، از دید کنترلر شبکه تعریف‌شده نرم‌افزاری در شبکه‌های اینترنت اشیا، چارچوب پیشنهادی رابطه بین کنترلرها را در کانال کنترل به طور مشخص در نظر نمی‌گیرد که در واقعیت مستقل نیست.

در [۴۴] یک بهینه‌سازی مبتنی بر وارنا (VBO) برای حل مسئله مکان‌یابی کنترلر قابل اطمینان پیشنهاد می‌شود که متوسط تأخیر کلی شبکه تعریف‌شده نرم‌افزاری را به حداقل می‌رساند. نتایج تجربی نشان می‌دهند که VBO، عملکرد بهتری نسبت به الگوریتم‌های مبتنی بر یادگیری معلم TLBO PSO و Jaya برای توپولوژی‌های مشهوری که در دسترس عموم است ارائه می‌دهد. روش آنها مکان‌یابی کنترلرها را از طریق روش‌های ابتکاری حل می‌کنند که قابلیت اطمینان را نیز برای کنترلرهای داده‌شده تضمین می‌کند.

مسئله اصلی در شبکه‌های نرم‌افزاری تعریف‌شده، شناسایی تعداد و موقعیت بهینه کنترلرهاست. مقاله [۴۵] به بررسی این موضوع می‌پردازد و سه هدف عمده را دنبال می‌کند:

۱) شناسایی تعداد کنترلرهای مورد نیاز: برای این منظور از رویکرد نظریه نمودار استفاده شده است.

۲) شناسایی موقعیت بهینه کنترلرها در شبکه: الگوریتم جستجوی تلفیقی پارتو (PITS) برای این هدف ارائه شده است.

۳) انجام مهاجرت کنترلرها برای کنترل موقعیت‌های پویا مانند خرابی

لینک و بار عدم تعادل کنترلر. یک رویکرد ابتکاری برای انجام مهاجرت پیشنهاد شده است.

در توپولوژی‌های زمان واقعی از مجموعه داده توپولوژی اینترنتی باغ وحش برای آزمایش روش پیشنهادی استفاده شده است. نتایج تجربی نشان می‌دهند این روش در تمام شرایط آزمایش‌شده، عملکرد مطلوبی از نظر تعداد کنترلرها و تأخیر دارد.

در [۲۷] یک مدل ریاضی جامع از مکان‌یابی کنترلرها پیشنهاد می‌شود که تأخیر در انتشار و ظرفیت کنترلر را محدود می‌کند و همزمان حداقل تعداد کنترلرها، مکان آنها و اختصاص گره‌ها به هر یک را تعیین می‌کند؛ در حالی که توزیع بار متعادل بین کنترلرها را حفظ می‌کند. در این تحقیق یک رویکرد ابتکاری نیز ارائه شده است. هدف اصلی این روش به دست آوردن حداقل تعداد کنترلرهاست؛ به طوری که توزیع بار بین کنترلرها را متعادل می‌کند و فاصله بین گره‌ها را به حداقل می‌رساند تا تأخیر قابل قبول از گره‌ها به کنترلر اختصاص داده شده آنها و همچنین بین کنترلرها به دست آید. بنابراین حداقل تعداد خوشه‌های گرهی که همه این شرایط را برآورده می‌کند، ساخته می‌شود و پس از آن برای هر خوشه، بهترین مکان گره در مورد زمان تأخیر و بار برای میزبانی کنترلر انتخاب می‌شود. شبیه‌سازی برای ۶۰ سناریو شبکه نشان می‌دهد که این روش در زمان ناچیز راه‌حل‌های متعادل و انعطاف‌پذیر به دست می‌آورد که ثابت می‌شود برای ٪۹۰ موارد ارزیابی‌شده، بهینه یا تقریباً بهینه است.

در [۴۶] نویسندگان بهینه‌سازی کنترلر و اختصاص سوئیچ برای به حداقل رساندن ترافیک کنترل کل در یک شبکه اینترنت اشیای مبتنی بر شبکه تعریف‌شده نرم‌افزاری را بررسی کردند. آنها با استفاده از IQP این مسئله را که NP سخت است، فرموله کردند و مجموعه‌ای از الگوریتم‌های ابتکاری را پیشنهاد می‌دهند که مکان‌یابی کنترلر و تخصیص سوئیچ را تسریع می‌کند؛ در حالی که با توجه به راه‌حل بهینه، ترافیک افزایش می‌یابد. آزمایش با تعداد زیادی نمودار توپولوژی از مجموعه توپولوژی اینترنتی باغ وحش، ساده‌ترین راه‌حل‌های ابتکاری از جمله تعداد کنترلرها با رگرسیون خطی و مکان‌یابی کنترلرها با معیارهای مرکزی به طور متوسط تقریباً ٪۴٫۵ پهنای باند بیشتری نسبت به راه‌حل بهینه داشته است؛ در حالی که پیشرفته‌ترین آنها به طور متوسط افزایش تقریباً ۱ درصدی داشته و به ویژه، بدترین حالت کمتر از ٪۳ بوده است.

در [۴۷] یک استراتژی برای مکان‌یابی کنترلر پیشنهاد می‌شود تا هنگام اعمال محدودیت‌های قرارگیری و تعادل، عدم تعادل بار بین کنترلرها را به حداقل برساند. همچنین یک الگوریتم مقیاس‌پذیر برای محاسبه راه‌حل‌های نزدیک به بهینه مسئله در شبکه‌های بزرگ ارائه می‌شود. این الگوریتم از چند توازن پایدار برای توزیع کسری سوئیچ‌ها به طور مساوی بین کنترلرها استفاده می‌کند تا عدم تعادل بار را کاهش دهد. با در نظر گرفتن تأخیر سوئیچ‌ها و بار کنترلرها، باقیمانده سوئیچ‌ها به نزدیک‌ترین کنترلرها اختصاص می‌یابند. الگوریتم برخی سوئیچ‌ها را نیز از کنترلری که در حال حاضر به آن اختصاص داده شده جابه‌جا می‌کند تا تأخیر بین سوئیچ‌ها و کنترلرها کاهش یابد. این الگوریتم در دو شبکه پرکاربرد Chinanet و Interoute که از توپولوژی باغ وحش استخراج شده‌اند. نتایج نشان می‌دهند که الگوریتم پیشنهادی نسبت به عدم تعادل بار و تأخیر بین کنترلر بدون تأثیر بر میانگین و انحراف استاندارد در تأخیر تغییر کنترلر، از راه‌حل‌های موجود که برای کنترلر موجود برای نرم‌افزار تعریف شده است، بهتر عمل می‌کند.

اکثر مطالعات با تمرکز بر روی مسئله مکان‌یابی کنترلر، فقط تأخیرهای انتشار کنترلر سوئیچ و تأخیرهای بین کنترلر را در نظر می‌گیرند؛ اما برای کنترل تأخیرهای انتها به انتها باید به ظرفیت کنترلرها توجه شود. در حالی که کنترلرهایی با نرخ پردازش، تعداد پورت‌ها و هزینه‌های مختلف در بازار هست، ارائه‌دهندگان خدمات اینترنت باید با استفاده حداکثری از منابع شبکه‌های خود، مقرون‌به‌صرفه‌بودن و توانایی جبران نیازهای خود را در نظر بگیرند. برای پیشنهاد راه‌حلی برای این مسئله مهم، نویسندگان در [۴۸] مسئله تعیین تعداد کنترلرها را با کوله‌پشتی حل کردند و مسئله GSOCCPP[1]، یک الگوریتم فراابتکاری را با تکرارهای جدید و شرایط جفت‌گیری معتدل برای مکان‌یابی کنترلرها فرموله کردند. این الگوریتم برای به دست آوردن حداقل تأخیرها از مقدار معقولی از زمان محاسبه استفاده می‌کند. تعدادی از مجموعه‌های داده‌های توپولوژی اینترنتی باغ وحش با انواع گره‌های صفحه داده در مقیاس کوچک تا بزرگ برای ارزیابی الگوریتم GSOCCPP بر اساس ظرفیت کنترلرهای مختلف مورد تجزیه و تحلیل قرار می‌گیرند. نتایج شبیه‌سازی نشان می‌دهند الگوریتم GSOCCPP علاوه بر عملکرد بهتر الگوریتم‌های مشابه فراابتکاری و خوشه‌بندی مانند Firefly، بهینه‌سازی ازدحام ذرات و k-means++، در دستیابی به کمترین زمان اجرا در بین الگوریتم‌های تحلیل‌شده نیز موفق است. همچنین راه‌حل پیشنهادی آنها در مقایسه با الگوریتم‌های دیگر برای مکان‌یابی کنترلر در توپولوژی‌های مختلف شبکه، مصرف حافظه کارآمدتری دارد.

در [۴۹]، یک بهینه‌ساز جمعیتی تطبیقی برای مکان‌یابی بهینه کنترلر با نام APB-CO برای شبکه‌های حسگر بی‌سیم پیشنهاد شده است. در این رابطه، هزینه، زمان و قابلیت اطمینان تنها چندین محدودیتی هستند که در نظر گرفته می‌شوند. در پایان APB-CO با ارزیابی همگام‌سازی، هزینه کنترلرها و پوشش شبکه تست می‌شود. مزیت این مدل بهبود کیفیت خدمات و کاهش تأخیر و محدودیتش، نیاز به بهبود انعطاف‌پذیری روش است.

عابدینی بقا و همکاران در [۵۰]، ابتدا شبکه را با الگوریتم خوشه‌بندی فازی C-Means (FCM) تقسیم‌بندی کردند و سپس مکان‌های مناسبی را برای کنترلرها توسط الگوریتم بهینه‌سازی مرغ دریایی تعیین نمودند. روش آنها با توجه به تأخیر انتشار میان کنترلرها و کنترلر تا سوئیچ‌ها توانست تأخیر را بهبود دهد و همچنین تعادل بار در سطح کنترل را فراهم کند؛ ولی راه‌حل آنها تعداد مورد نیاز کنترلر را نمی‌یابد و فقط برای تعداد داده‌شده کنترلر، مکان بهینه را می‌یابد.

مقاله [۵۱] از نظریه بازی و یک الگوریتم فراابتکاری هیبریدی گسسته برای حل مسئله قرارگیری کنترلر در شبکه‌های نرم‌افزاری تعریف‌شده استفاده می‌کند. این الگوریتم به دنبال یافتن بهترین محل برای قرارگیری کنترلرهاست؛ به طوری که هزینه کلی شبکه به حداقل برسد. همچنین [۵۲] یک چارچوب جدید برای قرارگیری کنترلر در شبکه‌های نرم‌افزاری تعریف‌شده با ظرفیت محدود ارائه می‌دهد. این چارچوب از الگوریتم PSO برای یافتن تعادلی بین تأخیر و قابلیت اطمینان در شبکه استفاده می‌کند.

۲-۵ مقایسه راه‌حل‌های مکان‌یابی کنترلرها در شبکه‌های تعریف‌شده نرم‌افزاری

جدول ۱ مقایسه‌ای را از روش‌های مختلف مکان‌یابی کنترلرها در شبکه‌های تعریف‌شده نرم‌افزاری ارائه می‌دهد. این دسته‌بندی از روش‌ها با هدف مطالعه دید کلی از بهبود عملکرد و بهینه‌سازی مکان‌یابی کنترلرها مورد بررسی قرار گرفته‌اند. همچنین این روش‌ها با توجه به مزایا

1. Garter Snap Optimization Capacitated Controller Placement Problem

جدول ۱: مقایسه راه‌حل‌های مکان‌یابی کنترلرها.

دسته‌بندی	مقالات	هدف	مزایا	معایب
راه‌حل‌های یادگیری ماشین	[۱۸] و [۲۷] تا [۳۱]	بهبود کارایی و بهینه‌سازی مصرف انرژی روش‌های مکان‌یابی کنترلرها	افزایش دقت و کارایی مکان‌یابی کنترلرها، بهبود مصرف انرژی	نیاز به داده‌های بزرگ برای آموزش، پیچیدگی بالا در پیاده‌سازی
راه‌حل‌های حریصانه	[۳۲] تا [۳۶]	بهینه‌سازی اطمینان روش‌های مکان‌یابی کنترلرها	سرعت اجرای بالا، سادگی در پیاده‌سازی	احتمال رسیدن به جواب‌های بهینه محلی، محدودیت در بهینه‌سازی چندهدفه
راه‌حل‌های فراابتکاری	[۳۷] تا [۵۲]	بهبود تأخیر شبکه با مکان‌یابی مناسب کنترلرها	ارائه راه‌حل‌های کارآمد و انعطاف‌پذیر	پیچیدگی بالای محاسباتی، احتمال رسیدن به جواب‌های بهینه محلی

و معایب خود در زمینه‌های بهبود تأخیر انتشار، تعادل بار، مصرف انرژی و تعداد کنترلرها مورد ارزیابی قرار گرفته‌اند.

۳- چالش‌ها و مسائل باز

۳-۱ تعداد کنترلرها

افزایش تعداد کنترلرها باعث کاهش تأخیر کلی شبکه می‌شود؛ اما هزینه ارتباطی بین کنترلرها را افزایش می‌دهد و بنابراین باید تعداد بهینه‌ای از کنترلرها در شبکه نصب شود. برای بهبود عملکرد شبکه و حفظ تعادل بین تأخیر و هزینه ارتباط بین کنترلرها باید یک رویکرد بهینه‌سازی شبکه اعمال شود. بهترین روش برای این کار، نصب تعداد بهینه‌ای از کنترلرها در شبکه است. با این روش، تأخیر کلی شبکه و هزینه ارتباط بین کنترلرها کاهش می‌یابد؛ بنابراین بهینه‌سازی تعداد کنترلرها در شبکه، یکی از موارد حیاتی برای بهبود عملکرد شبکه است.

۳-۲ ظرفیت کنترلرها

ظرفیت کنترلرها به حداکثر تعداد درخواست‌هایی اشاره دارد که هر کنترلر در واحد زمان می‌تواند پردازش کند. در تحقیق‌های انجام‌شده، دو رویکرد نسبت به ظرفیت کنترلرها وجود دارد. تحقیقاتی که کنترلرها را ۱) دارای ظرفیت بی‌نهایت (CPP بدون ظرفیت) و ۲) با ظرفیت محدود (CPP با ظرفیت) تعریف کرده‌اند. علاوه بر این، همه کنترلرها ممکن است دارای ظرفیت‌های نامتقارن باشند. در بررسی عملکرد الگوریتم‌های مختلف باید همه حالات در نظر گرفته شوند.

۳-۳ تأخیر

تأخیر از مهم‌ترین عوامل در عملکرد شبکه‌های تعریف‌شده نرم‌افزاری است و به دلیل اهمیت آن، مکان‌یابی کنترلرها نیز باید به گونه‌ای باشد که تأخیر در شبکه کمینه شود. تأخیر در شبکه تعریف‌شده نرم‌افزاری می‌تواند به دلیل مسیریابی نامناسب، تأخیر در پردازش درخواست‌ها توسط کنترلر، تأخیر در انتظار دریافت سرویس توسط کنترلرها، تأخیر انتشار ناشی از کنترلر تا کنترلر و یا تأخیر انتشار سوئیچ تا کنترلر و برعکس باشد. بنابراین مکان‌یابی کنترلرها باید به گونه‌ای باشد که تأخیر در شبکه کمینه و عملکرد بهینه شود. برای بهینه‌سازی مکان‌یابی کنترلرها، الگوریتم‌های مختلفی ارائه شده که به دلیل اهمیت تأخیر، هدف آنها کاهش تأخیر در شبکه است. این الگوریتم‌ها با در نظر گرفتن توپولوژی و میزان ترافیک و تأخیر در شبکه، بهینه‌ترین مکان‌یابی کنترلرها را انتخاب می‌کنند؛ ولی این موضوع هنوز جای کار دارد.

۳-۴ مصرف انرژی

یکی از مسائل مهم در شبکه‌های تعریف‌شده نرم‌افزاری، مصرف انرژی است. با توجه به این که کنترلرها مسئول مدیریت و کنترل شبکه هستند، به دلیل پردازش داده‌های بسیار زیاد، مصرف انرژی آنها نیز بسیار بالاست؛ بنابراین مکان‌یابی کنترلرها با هدف کاهش مصرف انرژی، یکی از مسائل مهم در شبکه‌های شبکه تعریف‌شده نرم‌افزاری است. با توجه به این که کنترلرها ممکن است دارای ظرفیت‌های نامتقارن باشند، مکان‌یابی آنها باید به گونه‌ای باشد که همه کنترلرها به طور متوسط بهینه باشند و در نتیجه مصرف انرژی کمتری داشته باشند.

۴- پیشنهاد کارهای آتی

با وجود تحقیقات گسترده در زمینه مکان‌یابی کنترلر که به آنها اشاره شد، هنوز چالش‌های حل‌نشده‌ای در این زمینه وجود دارد که به تعدادی از آنها اشاره گردید. این چالش‌ها می‌توانند به طور گسترده مورد مطالعه و تحقیقات مرتبط می‌تواند به کمک پیشنهادهای زیر بررسی و تکمیل شود:

- استفاده از الگوریتم‌های چندهدفه: توسعه الگوریتم‌هایی که علاوه بر تأخیر بین سوئیچ‌ها و کنترلرها به عواملی مانند تأخیر بین کنترلرها، ظرفیت کنترلرها، بار ترافیک و مصرف انرژی نیز توجه کنند.
- بهره‌گیری بیشتر از الگوریتم‌های یادگیری ماشین، حریصانه و فراابتکاری: مطالعه و بررسی الگوریتم‌هایی مانند ژنتیک، کلونی مورچه‌ها و جستجوی ممنوعه برای یافتن تعداد بهینه کنترلر و مکان مناسب نصب آنها
- بررسی بیشتر شبکه‌های بزرگ و پیچیده: توسعه الگوریتم‌های مقیاس‌پذیر و استفاده از تکنیک‌های تجزیه و تحلیل داده برای شناسایی زیرشبکه‌های بحرانی
- مطالعه در شبکه‌های با منابع محدود: توسعه الگوریتم‌های سازگار با منابع محدود و استفاده از تکنیک‌های مجازی‌سازی کنترلرها برای به اشتراک‌گذاری منابع
- مطالعه شبکه‌های تعریف‌شده نرم‌افزاری پویا: توسعه الگوریتم‌هایی که خودکار با تغییرات شبکه سازگار شوند و استفاده از تکنیک‌های یادگیری تقویتی برای آموزش الگوریتم‌های استفاده‌شده در کنترلرها.

با تمرکز بر روی این موضوعات، تحقیقات آتی می‌تواند به حل چالش‌های موجود در زمینه مکان‌یابی کنترلر در شبکه‌های تعریف‌شده نرم‌افزار کمک کند و منجر به ارتقای عملکرد، پایداری و کارایی این شبکه‌ها شود.

۵- جمع‌بندی

شبکه تعریف‌شده نرم‌افزاری یکی از جدیدترین پیشرفت‌ها در زمینه شبکه‌های کامپیوتری است که با جداسازی سطح داده از سطح کنترل، توانسته است که منابع شبکه را به‌خوبی مدیریت کند. کنترلرهای شبکه تعریف‌شده نرم‌افزاری، مغز شبکه در نظر گرفته می‌شوند. کنترلرها سیستم عامل شبکه هستند و مدیریت کل شبکه را با ارائه خدمات به برنامه‌های کاربردی کاربر بر عهده دارند تا کاربر شبکه بتواند با سخت‌افزار شبکه و با

سایر دستگاه‌های کل شبکه ارتباط برقرار کند. کنترلرها انواع مختلفی دارند. شبکه تعریف‌شده نرم‌افزاری اولیه با کنترلر منفرد و متمرکز توسعه یافته بود که هدف اصلی آن نمایش کل شبکه بود؛ اما در صورت خرابی کنترلر مرکزی قابلیت اطمینان و مقیاس‌پذیری را از دست می‌داد. به همین دلیل شبکه تعریف‌شده نرم‌افزاری با چندین کنترلر ایجاد شد. در این تحقیق پس از معرفی شبکه تعریف‌شده نرم‌افزاری و معماری آن، چالش‌های مطرح در سطح کنترلر، بیان و مسئله مکان‌یابی کنترلرها معرفی شد. مسئله مکان‌یابی کنترلرها از چند زیرمسئله تشکیل شده که می‌توان به یافتن تعداد کنترلر بهینه مورد نیاز شبکه بر اساس ترافیک و مقیاس شبکه اشاره کرد که تعداد کمی از تحقیقات انجام‌شده به این چالش توجه داشته‌اند. از طرفی بحث تعیین مکان بهینه برای نصب کنترلر در شبکه وجود دارد که برخی تحقیق‌ها با رویکرد حریصانه و برخی با رویکرد فراابتکاری و برخی با رویکرد یادگیری ماشین سعی در حل این مسئله داشته‌اند. اغلب آنها به پارامتر تأخیر توجه داشته‌اند و سایر پارامترها به دست فراموشی سپرده شده است. با وجود راه‌حل‌های ارائه‌شده، هنوز مسائل باز و چالش‌های حل‌نشده‌ای در مکان‌یابی کنترلرهای شبکه شبکه تعریف‌شده نرم‌افزاری وجود دارد که نیاز به توجه محققان دارد. از آن جمله می‌توان تأخیر شبکه، تعادل بار کنترلرها، مصرف انرژی و قابلیت اطمینان را نام برد.

مراجع

[1] A. A. Qaffas, *et al.*, "Adaptive population-based multi-objective optimization in SDN controllers for cost optimization," *Physical Communication*, vol. 58, Article ID: 102006, Jun. 2023.

[2] B. P. R. Killi and S. V. Rao, "Controller placement in software defined networks: a comprehensive survey," *Computer Networks*, vol. 163, Article ID: 106883, Nov. 2019.

[3] N. McKeown, *et al.*, "Openflow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69-74, Apr. 2008.

[4] POX, [Online]. Available at: http://www.noxrepo.org/pox/about-pox/.

[5] Ryu., [Online]. Available at: http://osrg.github.com/ryu/.

[6] Maestro, [Online]. Available at: http://code.google.com/p/maestro-platform.

[7] Floodlight. [Online]. Available at: http://www.projectfloodlight.org/.

[8] K. K. Yap, *et al.*, "OpenRoads: empowering research in mobile networks," *SIG-COMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 125-126, Jan. 2010.

[9] A. K. Nayak, A. Reimers, N. Feamster, and R. Clark, "Resonance: dynamic access control for enterprise networks," in *Proc. 1st ACM Workshop on Research on Enterprise Networking*, pp. 11-18, Barcelona Spain, 21-21 Aug. 2009.

[10] B. Heller, *et al.*, "ElasticTree: saving energy in data center networks," in *Proc. 7th USENIX Conf. on Networked Systems Design and Implementation*, 17 pp., San Jose, CA, USA, 28-30 Apr. 2010.

[11] P. S. Pisa, *et al.*, "OpenFlow and Xen-based virtual network migration," in *Proc. Communications: Wireless in Developing Countries and Networks of the Future*, pp 170-181, Brisbane, Australia, 20-23 Sept., 2010.

[12] O. K. M. Koerner, "Multiple service load-balancing with openflow," in *Proc. IEEE 13th Int. Conf. on High Performance Switching and Routing*, pp. 210-214, Belgrade, Serbia, 24-27 Jun. 2012.

[13] D. Kotani, K. Suzuki, and H. Shimonishi, "A design and implementation of OpenFlow controller handling IP multicast with fast tree switching," in *Proc. IEEE/IPSJ 12th Int. Symp. on Applications and the Internet*, pp. 60-67, Izmir, Turkey, 16-20 Jul. 2012.

[14] R. Sherwood, et al., FlowVisor: A Network Virtualization Layer, Technical Report, 2009.

[15] G. Lu, R. Miao, Y. Xiong, and C. Guo, "Using CPU as a traffic co-processing unit in commodity switches," in *Proc. 1st Workshop on Hot Topics in Software Defined Networks*, pp. 31-36, Helsinki, Finland, 13-13 Aug. 2012.

[16] H. S. Chiang, A. K. Sangaiah, M. Y. Chen, and J. Y. Liu, "A novel artificial bee colony optimization algorithm with SVM for bio-inspired software-defined networking," *Int. J. Parallel. Prog.*, vol. 48, pp. 310-328, Apr. 2020.

[17] S. Chattopadhyaya and A. K. Shahoo, "Software defined networks: current problems and future solutions," in *Materials Today: Proceedings*, vol. 49, no. 8, pp. 2989-2993, 2020.

[18] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. of the 1st Workshop on Hot Topics in Software Defined Networks*, pp. 473-478, Helsinki, Finland, 13-13 Aug. 2012.

[19] B. P. R. Killi and S. V. Rao, "Capacitated next controller placement in software defined networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, no. 3, pp. 514-527, Jun. 2017.

[20] A. Sallahi and M. St-Hilaire, "Optimal model for the controller placement problem in software defined networks," *IEEE Commun. Lett*, vol. 19, no. 1, pp. 30-33, Jan. 2015.

[21] T. Das, V. Sridharan, and M. Gurusamy, "A survey on controller placement in SDN," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 472-503, Aug. 2019.

[22] M.A Gunavathie, S. Jacophine Susmi, S. S. Sivasankari, and M. Venkatesh, " Exploring the challenge of controller placement in software-defined networking: a comprehensive review," in *Proc. 2nd Int. Conf. on Automation, Computing and Renewable Systems*, pp. 1950-1955, Pudukkottai, India, 11-13 Dec. 2023.

[23] B. Isong, R. R. S. Molose, A. M. Abu-Mahfouz, and N. Dladlu, "Comprehensive review of SDN controller placement strategies," *IEEE Access*, vol. 14, pp. 170070-170092, 2020.

[24] A. Shirmarz and A. Ghaffari, "Taxonomy of controller placement problem (CPP) optimization in software defined network (SDN): a survey," *J. of Ambient Intelligence and Humanized Computing*, vol. 12, no. 12, pp. 10473-10498, Dec. 2021.

[25] A. Kumari and A. S. Sairam, "Controller placement problem in software-defined networking: a survey," *Networks*, vol. 78, no. 2, pp. 195-223, Sep. 2021.

[26] B. Sapkota, B. R. Dawadi, and S. R. Joshi, "Controller placement problem during SDN deployment in the ISP/Telco networks: a survey," *Engineering Reports*, vol. 6, no. 2, Article ID: e12801, Feb. 2024.

[27] G. Schütz and J. A. Martins, "A comprehensive approach for optimizing controller placement in software-defined networks," *Computer Communications*, vol. 159, pp. 198-205, Jun. 2020.

[28] M. Guo and P. Bhattacharya, "Controller placement for improving resilience of software-defined networks," in *Proc. 4th Int. Conf. on Networking and Distributed Computing,* pp. 23-27, Los Angeles, CA, USA, 21-24 Dec. 2013.

[29] D. Hock, *et al.*, "Pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. of the 25th In. Teletraffic Congress,* 9 pp., Shanghai, China, 10-12 Sept. 2013.

[30] Z. Yang and K. L. Yeung, "Minimum weight controller tree design in SDN," *Computer Networks*, vol. 165, Article ID: 106949, Dec. 2019.

[31] A. Mishra, N. Gupta, and B. B. Gupta, "Defense mechanisms against DDoS attack based on entropy in SDN-cloud using pox controller," *Telecommun Syst*, vol. 77, pp. 47-62, 2021.

[32] Y. N. Hu, W. D. Wang, X. Y. Gong, X. R. Que, and S. D. Cheng, "On the placement of controllers in software-defined networks," *J. China Univ Posts Telecommun*, vol. 19, sup. 2, pp. 92-171, Oct. 2012.

[33] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, "On reliability-optimized controller placement for software-defined networks," *China Commun.*, vol. 11, no. 2, pp. 38-54, Feb. 2014.

[34] S. Yang, L. Cui, Z. Chen, and W. Xiao, "An efficient approach to robust SDN controller placement for security," *IEEE Trans. on Network and Service Management*, vol. 17, no. 3, pp. 1669-1682, May 2020.

[35] G. Ramya and R. Manoharan, "Traffic-aware dynamic controller placement in SDN using NFV," *The J. of Supercomputing*, vol. 79, no. 2, pp. 2082-2107, Feb. 2023.

[36] A. Naseri, M. Ahmadi, and L. PourKarimi, "Placement of SDN controllers based on network setup cost and latency of control packets," *Computer Communications*, vol. 208, pp. 15-28, Aug. 2023.

[37] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, and M. Hoffmann, "Heuristic approaches to the controller placement problem in large scale SDN network," *IEEE Trans. Netw. Serv. Manag.*, vol. 12, no. 1, pp. 4-17, Feb. 2015.

[38] L. Müller, R. R. Oliveira, M. C. Luizelli, L. P. Gaspary, and M. P. Barcellos, "Survivor: an enhanced controller placement strategy for improving SDN survivability," in *Proc. IEEE Global*

Communications Conf., pp. 1909-1915, Austin, TX, USA, 8-12 Dec. 2014.
[39] A Farshin and S. Sharifian, "A chaotic grey wolf controller allocator for software deÞned mobile network (SDMN) for 5th generation of cloud-based cellular systems (5G)," *Computer Communications*, vol. 108, pp. 94-108, Aug. 2017.
[40] P. Vizarreta, *et al.*, "Assessing the maturity of SDN controllers with software reliability growth models," *IEEE Trans. on Network and Service Management*, vol. 15, no. 3, pp. 1090-1104, Sept. 2018.
[41] F. Li and X. Xu, "A discrete cuckoo search algorithm for the controller placement problem in software defined networks," in *Proc. IEEE 9th Annual Information Technology, Electronics and Mobilelectronics and Mobile Communication Conf.,* pp. 292-296, Vancouver, Canada, 1-3 Nov. 2018.
[42] A. A. Ateya, *et al.,* "Chaotic salp swarm algorithm for SDN multi-controller networks," *Engineering Science and Technology, an International J.*, vol. 22, no. 4, pp. 1001-1012, Aug. 2019.
[43] A. K. Tran, M. J. Piran, and C. Pham, "SDN controller placement in IoT networks: an optimized submodularity-based approach," *Sensors*, vol. 19, no. 24, pp. pp. 1-12, Dec. 2019.
[44] A. K. Singh, S. Maurya, N. Kumar, and S. Srivastava, "Heuristic approaches for the reliable SDN controller placement problem," *Trans. on Emerging Telecommunications Technologies*, vol. 31, no. 2, Article ID: e3761, Feb. 2019.
[45] G. Ramya and R. Manoharan, "Enhanced optimal placements of multicontrollers in SDN," *J. of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 8187-8204, 2020.
[46] K. Choumas, D. Giatsios, P. Flegkas, and T. Korakis, "SDN controller placement and switch assignment for low power IoT," *Electronics*, vol. 9, no. 2, Article ID: 325, 2020.
[47] B. R. Killi and S. V. Rao, "Poly-stable matching based scalable controller placement with balancing constraints in SDN," *Computer Communications*, vol. 154, pp. 82-91, Mar. 2020.
[48] S. Torkamani-Azar and M. Jahanshahi, "A new GSO based method for SDN controller placement," *Computer Communications*, vol. 163, pp. 91-108, Nov. 2020.
[49] S. Taghavi Motlagh, A. Ibrahim, S. Shah Heydari, K. El-Khatib., "Multi-objective optimization for managing disruption risk in SDN," in *Proc. 20th International Conference on the Design of Reliable Communication Networks*, 8 pp., Montreal, Canada. 6-9 May 2024.
[50] M. Abedini Bagha, K. Majidzadeh, M. Masdari, and Y. Farhang, "Improving delay in SDNs by metaheuristic controller placement," *International J. of Industrial Electronics Control & Optimization*, vol. 5, no. 3, pp. 286-296, Dec. 2022.
[51] M. Khojand, K. Majidzadeh, M. Masdari, and Y. Farhang, "Controller placement in SDN using game theory and a discrete hybrid metaheuristic algorithm," *the J. of Supercomputing*, vol. 80, no. 5, pp. 6552-6600, Mar. 2024.
[52] G. D. Singh, *et al.*, "A novel framework for capacitated SDN controller placement: balancing latency and reliability with PSO algorithm," *Alexandria Engineering*, vol. 78, pp. 77-79, Jan. 2024.

مهدی سربازی در سال ۱۳۷۱ در شهر بانه در استان کردستان ایران متولد شد. وی در سال ۱۳۸۹ مدرک دیپلم خود را در زادگاه خود اخذ کرد. در سال ۱۳۹۴ مدرک کارشناسی خود را در رشته علوم کامپیوتر از دانشگاه تبریز دریافت نمود. سپس در سال ۱۳۹۷ مدرک کارشناسی ارشد خود را در رشته مهندسی فناوری اطلاعات گرایش شبکه‌های کامپیوتری از دانشگاه آزاد اسلامی واحد تهران جنوب اخذ کرد. وی در حال حاضر دانشجوی دکتری مهندسی کامپیوتر در دانشگاه آزاد اسلامی واحد سنندج است و به عنوان کارمند بخش فناوری اطلاعات در دانشگاه کردستان مشغول به کار است.

محمد فتحی مدرک کارشناسی ارشد و دکتری خود را در رشته مهندسی برق به‌ترتیب در سال‌های ۱۳۸۲ و ۱۳۸۹ از دانشگاه صنعتی امیرکبیر، تهران، ایران دریافت کرده است. وی در حال حاضر استادیار دانشکده مهندسی برق دانشگاه کردستان، سنندج، ایران است. حوزه‌های پژوهشی مورد علاقه وی شامل یادگیری ماشین، بهینه‌سازی تخصیص منابع در شبکه‌های بی‌سیم، رایانش ابری و لبه، و امنیت شبکه می‌باشد.