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

Winter 2020, 1 (2), 17-32
DOR: 20.1001.1.27832570.1399.1.2.5.1

Received: 15 Nov. 2019
Accepted: 1 Jan. 2020

A Framework for Using Network Virtualization Functions for Video Streams

Ali Asghar Khavasi

Principal Lecturer, Centria University of Applied Sciences, Finland. aliasghar.khavasi@centria.fi

Abstract

Introduction: This paper investigates a way to stream online videos using SDN / NFV that can help to reduce latency and packet loss and cost. This method reduces Capex and Opex costs due to its software and low hardware dependency. Software-centric networks and virtualization of network functions have been introduced in the computer networks arena as a new way to design, deploy and manage network services. Those can provide more flexible and more flexible and adaptable to the consumer and provider according to the needs of end users and service providers to consolidate and deliver network components using standard IT virtualization technologies. This paper presents the architecture and prototype of a video streaming service compatible with SDN / NVF-enabled networks, allowing simplified networking and applications and reduced management costs.

Method: The simulation framework of this paper conforms to the MANO architecture to implement the standard NFV evaluation method. In CloudSim, each simulation occurs by sending and receiving events between modules.

Results: As expected, NoScale-Min policies in both LFF and MFF methods lead to exceptionally long response times due to resource shortages in VNFs. On the other hand, the delay of the NoScale-Max policy is short due to resource constraints enough is assigned to the VNF.

Discussion: In this paper, we proposed CloudSimSDN-NFV as a framework for new simulation tool of video streams to evaluate NFV features in edge and cloud environments along with other features of SDN and cloud computing environments. This framework is designed and developed on CloudSimSDN. We describe the modeling and simulation of NFV and edge computing and the detailed design and implementation of our framework. Two used scenarios were presented to help understand how to use the new tool, and several algorithms were implemented and evaluated based on the framework.

Keywords: Software-Defined Networking (SDN), virtual network functions, CloudSim, virtual simulation, video streams.

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

یک چارچوب برای استفاده از توابع مجازی‌سازی شبکه برای جریان‌های ویدئویی

تاریخ دریافت: ۱۳۹۹/۰۸/۲۵
تاریخ پذیرش: ۱۳۹۹/۱۰/۱۲

دورهٔ اول، زمستان ۱۳۹۹
شمارهٔ دوم، صص: ۱۷ – ۳۲

علی‌اصغر خواصی

دانشیار، دانشگاه سنتریا، فنلاند aliasghar.khavasi@centria.fi

چکیده: بررسی روشی برای پخش ویدئوهای آنلاین با استفاده از sdn/nfv که بتواند به در کاهش تأخیر و ازدست‌رفتن بسته‌ها و کاهش هزینه‌ها کمک‌کند. در این روش به‌دلیل نرم‌افزاری بودن و وابستگی کم به سخت‌افزار باعث کاهش هزینه‌های capex و opex میشود. شبکه‌های نرم‌افزارمحور و مجازی‌سازی توابع شبکه در مهندسی فناوری اطلاعات به‌عنوان روشی جدید برای طراحی، استقرار و مدیریت خدمات شبکه معرفی‌شده‌اند. با استفاده از، آن‌ها می‌توانند برای تثبیت و تحویل مؤلفه‌های شبکه با استفاده از فناوری‌های استاندارد مجازی‌سازی IT، از طریق زیرساخت‌های حامل، با توجه به نیازهای مصرف‌کنندگان نهایی و ارائه‌دهندگان خدمات انعطاف‌پذیرتر و سازگارتر بیشتری برای مصرف‌کننده و ارائه‌دهنده ارائه‌نمود. این مقاله معماری و نمونه اولیه مربوط به یک سرویس پخش ویدیویی متناسب با شبکه‌های فعال‌شده با SDN/NVF را ارائه‌می‌دهد، و این امکان را فراهم‌می‌کند که توابع شبکه و برنامه‌های کاربردی ساده‌سازی‌شده و هزینه‌های مدیریت کاهش‌یابد. در چنین روشی، راه‌حل پیشنهادی، ارائه‌دهندگان محتوای کوچک/ متوسط و غیرمعمول را قادرمی‌سازد بدون نیاز به اتخاذ یک زیرساخت تحویل‌داده ویژه و گران‌قیمت می‌باشد، رویدادهایی را با تعداد محدودی از کاربران علاقه‌مند به ا شتراک بگذارند. روش اجرا بدین شرح می‌با شد ابتدا ساختار مورد نظر را از طریق برنامه‌نویسی java به شبیه ساز معرفی کرده و سپس شرایط شبیه سازی را تعیین می‌کنیم . این نرم افزار شبیه سازی را انجام داده و خروجی‌های لازم برای ما ایجادمی‌کند و سپس با نتایج به‌د ست‌آمده با دیگر روش‌ها مقایسه شده و نتیجه‌گیری به‌د ست‌آمده نشان‌دهنده افزایش کارایی نسبت به دیگر روش‌ها می‌باشد.

واژه‌های کلیدی: شبکه‌های نرم‌افزارمحور، توابع مجازی شبکه، کلودسیم، شبیه‌سازی مجازی، جریان‌های ویدئویی،

۱. مقدمه

پخش ویدئو و بازی‌های رایانه‌ای از محبوب‌ترین و بالاترین رسانه‌های مصرف‌کننده پهنای باند در اینترنت هستند. محتوای ویدئویی حدود ۷۰٪ کل پهنای باند امروزی را در اینترنت مصرف می‌کند. پیشرفت در ابزارهای تولید رسانه، قدرت پردازش بالا و اتصال با سرعت بالا باعث ایجاد نسل رسانه‌های زنده، تعاملی و چندمنظوره شده‌است. شناختی و چندنفره بازی های آنلاین افق های جدیدی را برای تجربه بازی باز کرده‌اند. با این حال، این‌گونه بازی‌های تعاملی، و فیلم‌های چند نمایه و نمایش ۳۶۰ درجه، در حال حاضر توسط عدم‌تحمل تأخیر و استفاده بیش از حد از پهنای باند محدود شده‌است[۷].

رویکرد پیشنهادی، به لطف ویژگی "درون‌شبکه‌ای" آن که ذاتی در الگوی SDN / NFV است، در مقایسه با رویکرد OTT[1] که به طور گسترده مورد استفاده قرارمی‌گیرد، مزایای بسیاری را ارائه‌می‌دهد. در حقیقت، با استفاده از الگوی SDN / NFV، بستر توزیع محتوای ویدیو فقط در شبکه اصلی اجرامی‌شود، که به طور معمول دارای ابعاد بعدی است. علاوه بر این، گره‌های شبکه اصلی که متوجه تکثیر جریان و رمزگذاری‌می‌شوند تصمیم‌می‌گیرند و می‌توانند توسط ارکستر در زمان اجرا با توجه به وضعیت فعلی شبکه متنوع باشند و بدین ترتیب، سکوی سرویس را مستحکم تر از احتراق احتمالی شبکه زیربنایی می‌کند[4].

در این پایان نامه ما چارچوبی را برای شبیه‌سازی قابلیت‌های NFV در هر دو محیط لبه و ابر ارائه‌می‌دهیم. علاوه بر ویژگی‌های اساسی شبکه که توسط SDN در CloudSimSDN پشتیبانی می‌شود، ما ویژگی‌های جدید NFV مانند تخصیص توابع شبکه مجازی، انتقال و مقیاس خودکار را با پشتیبانی از عملکردهای شبکه مربوطه مانند تعادل بار جریان، تغییر مسیر و زنجیره عملکرد سرویس نگهداری (SFC) اضافه‌کردیم[14].

ما چارچوب شبیه‌سازی خود را با مقیاس‌گذاری خودکار و سیاست‌های قرارگیری SFC در محیط‌های لبه یکپارچه و محاسبات ابری ارزیابی کردیم. نتایج اثربخشی آن را در اندازه‌گیری و ارزیابی تأخیر، زمان پاسخ، استفاده از منابع، ترافیک شبکه و مصرف برق با الگوریتم‌های مختلف در هر سناریو نشان‌می‌دهد[2].

در دهه گذشته، محاسبات ابری به منظور تأمین انعطاف‌پذیر منابع محاسباتی توسعه یافته‌است. سازمان‌ها و شرکت‌های نوپا می‌توانند خدمات برنامه خود را بر روی مراکز داده ابری بدون سرمایه‌گذاری هزینه‌های هنگفتی برای خرید سرورهای محاسباتی و زیرساخت‌های شبکه ایجادکنند[7].

درعوض، آن‌ها می‌توانند با چند کلیک ماوس از طرف ارائه‌دهندگان خدمات ابری، به اندازه مورد نیاز در عرض چند دقیقه از منابع استفاده کنند و فقط برای مدت زمان و میزان منابع تأمین‌شده هزینه پرداخت کنند. در رایانش ابری، ماشین‌های فیزیکی از طریق هایپروایزرها (hypervisors[2]) مجازی می‌شوند که منابع مجازی‌سازی‌شده (ماشین‌های مجازی ([VMs] می‌توانند مستقیماً به مستأجران یا از طریق سیستم عامل‌ها و خدمات اختصاص یابد[1].

به همین ترتیب، فناوری مجازی‌سازی مفهوم جدیدی از مجازی‌سازی شبکه را در زمینه ارتباطات از راه دور همراه با شبکه تعریف شده توسط نرم افزار (SDN) به ارمغان آورده است. به همین ترتیب، فناوری مجازی‌سازی مفهوم جدیدی از مجازی‌سازی شبکه را در زمینه ارتباطات از راه دور همراه با شبکه تعریف‌شده توسط نرم‌افزار (SDN) به ارمغان آورده‌است. توابع شبکه مانند فایروال، پروکسی و سیستم تشخیص نفوذ (IDS[3])، قبلاً توسط یک هدف سخت‌افزاری گران‌قیمت ساخته‌می‌شدند که فقط برای برخی از توابع شبکه ساخته‌شده‌است[11].

از آنجا که توابع شبکه همان وظایف فشرده‌پردازنده (CPU intensive tasks) را برعهده‌دارند، ارائه‌دهندگان شبکه مجبورند دستگاه اختصاصی را خریداری‌کنند تا عملکردهای مورد نیاز را به مشتریان خود ارائه دهند. مجازی‌سازی توابع شبکه (NFV) به لطف پیشرفت فناوری مجازی‌سازی، که در محاسبات ابری مشاهده کرده‌ایم، به شدت تکامل یافته است[3].

از آنجا که ماشین‌های مجازی می‌توانند به صورت پویا در ابرها مقیاس بندی، تهیه و مهاجرت کنند، توابع شبکه مجازی (VNF) نیز می‌توانند در سراسر ماشین‌های فیزیکی عمومی تهیه‌شوند تا عملکرد شبکه خاصی را فراهم‌کنند. ارائه‌دهندگان ارتباطات از راه دور مراکز داده ابری خود را برای NFV در شبکه‌های خود ایجادمی‌کنند و VNF را در مرکز داده خود قرارمی‌دهند تا منابع محاسباتی و شبکه‌ای آن را به شدت مدیریت‌کند[16].

به منظور کاهش ترافیک شبکه بین دستگاه‌های حسگر اینترنت اشیا و سرورهای محاسباتی مرکزی که معمولاً در یک ابر قراردارند، می‌توان از منابع لبه (به عنوان مثال، روترهای شبکه، نقاط دسترسی بی‌سیم و برج‌های رادیویی) برای فیلترکردن و پیش‌پردازش داده‌های جمع‌آوری شده استفاده کرد[5].

علاوه بر این، قراردادن توابع سرویس در نزدیکی کاربران نهایی می‌تواند با کاهش تأخیر انتقال شبکه بین کاربر نهایی و عملکرد سرویس، تأخیر انتها به انتها را کاهش‌دهد. در NFV، در سالهای اخیر بسیاری از محققان استفاده از منابع لبه برای VNF ها را مورد بررسی قرارداده‌اند. با وجود توجه بیشتر به NFV و محاسبه لبه، ابزار محدودی وجوددارد که می‌تواند تکنیک‌های جدید را در ادبیات تأیید و ارزیابی کند.

چندین سیستم عامل اثبات مفهوم توسط محققان ارائه‌شده‌است تا توانایی بالقوه عملکرد و تنظیم NFV را از طریق رایانه‌های لبه نشان‌دهد. با این حال، ارزیابی یک روش جدید در مقیاس بزرگ برای سیستم عامل‌های پیشنهادی دردسر ساز است زیرا برای ارزیابی در چنین مقیاسی، سیستم تجربی باید در زیرساخت‌های مقیاس بزرگ مستقر شود. ابزارهای سنتی شبیه‌سازی شبکه مانند ns3 الگوی جدید فناوری‌های NFV و SDN را پشتیبانی نمی‌کنند. بنابراین، یک جعبه ابزار شبیه‌سازی در دسترس، قابل‌انعطاف و مقیاس‌پذیر باید معرفی و توسعه داده‌شود تا بتواند تحقیقات و تحقق NFV و محاسبات لبه‌ای را در محیط‌های محاسبات ابری مجهز به SDN تقویت‌کند[8].

به منظور پرکردن شکاف، در این مقاله، ما یک چارچوب شبیه‌سازی جدید، CloudSimSDN-NFV، برای NFV و شبیه‌سازی محاسبات لبه‌ای در ابرهای مجهز به SDN گسترش‌یافته از CloudSimSDN پیشنهادمی‌کنیم[9].

CloudSimSDN را در سال ۲۰۱۵ برای شبیه‌سازی ویژگی‌های SDN در محاسبات ابری توسعه و ارائه کردیم. CloudSimSDN به شبیه‌سازی سیاست‌های مختلف تخصیص و تهیه برای هر دو محاسبات و منابع شبکه و همچنین میزان کار و روش‌های برنامه‌ریزی آگاه از برنامه در ابر کمک کرده است[18].

علاوه بر قابلیت‌های اساسی شبکه که توسط SDN در CloudSimSDN پشتیبانی‌می‌شود، در این مقاله طراحی و اجرای ویژگی‌های جدید برای پشتیبانی از قابلیت‌های NFV، مانند تخصیص VNF، انتقال و مقیاس خودکار ارائه شده‌است[18].

برای تسهیل این ویژگی‌ها، چارچوب شبیه‌سازی بر اساس نگاشت معماری و مولفه‌های components مدیریت و ارکستراسیون NFV ETSI (MANO[4]) است که شامل ارکستر NFV (NFVO[5])، مدیر VNF (VNFM[6]) و مدیر زیرساخت مجازی (VIM[7]) است[4].

علاوه بر این، مفهوم محاسبه لبه با چارچوب شبیه‌سازی جدید، شامل چندین مرکز داده، شبکه‌های بین ابر و ظرفیت‌های متفاوت مرکز داده، ادغام‌شده و پشتیبانی‌می‌شود.

سهم اصلی این مقاله به شرح زیر است:

- مدل‌سازی تأمین منابع برای NFV در محیط محاسبات لبه.
- معماری و طراحی چارچوب شبیه‌سازی NFV در محاسبات لبه‌ای و ابری ؛
- توسعه دقیق و اجرای چارچوب شبیه‌سازی و چالش‌ها ؛
- ارزیابی عملکرد چارچوب با استفاده از سناریوهای مورد استفاده ؛
- پسوندها و جهت‌های بالقوه چارچوب پیشنهادی.

بقیه این مقاله به شرح زیر است.در بخش بعد در مورد سیستم عامل‌ها و ابزارهای شبیه‌سازی موجود بحث‌می‌کند. مدل‌سازی و شبیه‌سازی NFV در محیط‌های لبه و رایانش ابری و به دنبال آن طراحی دقیق و اجرای چارچوب شبیه‌سازی جدید را ارائه‌می‌دهد و از سناریوها و نتایج ارزیابی با استفاده از چارچوب شبیه‌سازی ارائه شده‌است. ما در مورد الحاقات بالقوه چارچوب پیشنهادی بحث‌می‌کنیم، که می‌تواند برای پشتیبانی در سناریوهای مختلف اجراشود. سرانجام خلاصه و نتیجه‌گیری می‌کند.

به منظور ارزیابی و آزمایش الگوریتم‌های جدید در مورد مدیریت NFV و SFC و همچنین ویژگی‌های شبکه مربوطه، باید یک روش مقیاس‌پذیر برای دیدن اثربخشی الگوریتم در مقیاس بزرگ استفاده‌شود. آزمایش چنین الگوریتم‌های آزمایشی در محیط تولید به دلیل نتایج غیرقابل پیش‌بینی روش جدید غیرعملی است. یک آزمایش در یک بستر آزمایش در مقیاس کوچک یک راه‌حل عملی جایگزین برای تأیید اثربخشی است، اما تأثیر آن را در مقیاس بزرگ به‌سختی می‌توان در سیستم آزمایشگاه مقیاس کوچک گرفت. علاوه بر این، اجرای تجربی الگوریتم جدید برای استقرار در بستر آزمایش می‌تواند زمانبر و چالش برانگیز باشد، به‌خصوص اگر در مراحل اولیه باشد. بنابراین، شبیه‌سازی به طور گسترده‌ای در علم به کار گرفته‌شده‌است تا ارزیابی را با دقت منطقی ساده‌کند، که با همان شرایط قابل‌مقایسه با خطوط پایه است. با شبیه‌سازی، می‌توان چندین آزمایش با تنظیمات و تنظیمات مختلف را در مدت زمان کوتاهی با اسکریپت‌های خودکار انجام داد[18].

به همین ترتیب، به منظور کاهش زمان ارزیابی و ساده‌سازی فرآیند، شبیه‌سازی NFV در محیط لبه ابر بسیار مهم است. در نهایت، جعبه ابزار شبیه‌سازی قابل دسترسی، سازگار و مقیاس‌پذیر با کاهش تلاش برای ارزیابی روش‌ها و الگوریتم‌های جدید در این زمینه، باعث ایجاد نوآوری در NFV می‌شود. ایده‌های نوآورانه را می‌توان بدون نیاز به صرف زمان زیادی برای آماده‌سازی زیرساخت آزمایشگاهی و استقرار الگوریتم جدید بر روی سیستم، با جعبه ابزار شبیه‌سازی پیاده‌سازی و ارزیابی کرد[13].

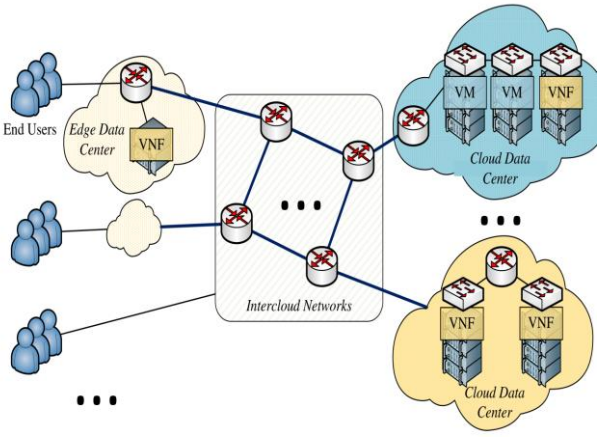
ساختار کلی پیاده‌سازیNFV در محیط ابر-ابر در شکل۱ ارائه‌شده‌است. چارچوب شبیه‌سازی ما بر اساس مؤلفه‌های components معماری ذکرشده در شکل است. مراکز داده Edge مراکز داده‌ای در مقیاس کوچک هستند (به عنوان مثال، مجموعه‌ای از روترهای شبکه، نقاط دسترسی یا یک ایستگاه پایه با چند سرور محاسباتی)، نزدیک به کاربران نهایی قراردارند. کاربران نهایی می‌توانند از طریق شبکه‌های بین ابر (به عنوان مثال شبکه ستون فقرات) از طریق مراکز داده لبه یا مراکز داده ابری به خدمات برنامه دسترسی پیدا کنند. VNF ها را می‌توان در لبه یا مراکز داده ابری قرارداد. مراکز داده ابری می‌توانند ماشین‌های مجازی را برای سرویس‌های برنامه و VNF ها را برای سرویس‌های شبکه فراهم‌کنند. جزئیات بیشتر مدل‌ها و انتزاعات در زیر بخش‌های زیر شرح داده شده‌است[9].

۲. طراحی و اجرا

معماری NFV MANO توسط مؤسسه استاندارد ارتباطات از راه دور اروپا (ETSI) پیشنهاد و در زمینه ارتباطات بسیار پذیرفته‌شده است. MANOسه جز اصلی را برای تنظیم و مدیریت تعریف‌می‌کند NFVO . نیازهای تجاری و عملیاتی را دریافت‌می‌کند و VNF ها را در سیستم برای ارائه خدمات مورد نیاز هماهنگ‌می‌کندNFVO می‌تواند VNF های مختلف را از طریق VNFM مدیریت‌کند تا در واقع VNF هایی را که با عناصر SDN همکاری‌می‌کنند ایجاد و تهیه کند. در زیر VNFM VIMقراردارد که منابع زیرساخت میزبان VNF را کنترل‌می‌کند [17].

چارچوب شبیه‌سازی ما با معماری MANO مطابقت دارد تا روش شبیه‌سازی و ارزیابی استاندارد NFV را همان‌طور که در شکل ۲ نشان داده شده، ارائه‌دهد. زیرساخت مدل‌شده توسط مدیر زیرساخت) به عنوان مثال (VIM برای پیکربندی شبکه، تخصیص آدرس و مسیر فیزیکی مدیریت‌می‌شود. از آنجا کهVNFM می‌تواند VNF ها را در

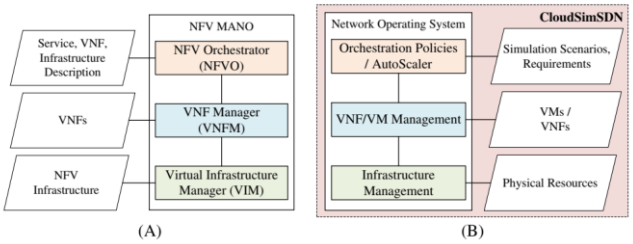
سیستم مدیریت کند، چارچوب شبیه‌سازی می‌تواند VNF ها را به عنوان نمونه‌های VM ایجاد، حذف یا مهاجرت کند. مدیریت VNF توسط ارکستراتور) به عنوان مثال (NFVO تعیین‌می‌شود، که به عنوان سیاست‌های مختلف در چارچوب ما مدل‌سازی می‌شود[11].



شکل ۱: معماری مجازی‌سازی توابع شبکه(NFV)

[Colour figure can be viewed at wileyonlinelibrary.com]

با الزامات ورودی، سناریوها و پیکربندی‌های شبکه مجازی، خط مشی مشخص‌شده توسط کاربر در چارچوب شبیه‌سازی، تصمیم‌گیری در مورد سیاست‌های سازماندهی در بین VNF ها را انجام‌می‌دهد، به عنوان مثال، افزایش منابع برایVNF ، کاهش منابع در صورت عدم استفاده، یا مهاجرت به زیرساخت‌های دیگر می‌باشد.



شکل ۲: ساختار CloudSimSDN-NFV مطابق باNFV MANO

[Colour figure can be viewed at wileyonlinelibrary.com]

علاوه بر ویژگی‌هایNFV ، ما شبیه‌سازی محاسبات لبه را در چارچوب مدل‌سازی و یکپارچه کردیم. مشابه مرکز داده‌های ابری مرکزی، یک مرکز داده لبه با مجموعه‌ای از زیرساخت‌های فیزیکی (میزبان‌ها، سوئیچ‌ها و پیوندها) به یکدیگر متصل‌می‌شوند. مراکز داده لبه با ظرفیت محدود منابع و نوع مرکز داده متفاوت هستند. برای شبکه بین ابر، لبه، ابرهای مرکزی، سوئیچ‌ها و پیوندهای بین ابر برای اتصال آن مراکز داده مدل‌سازی شده‌اند. هر مرکز داده می‌تواند خط مشی‌های مدیریت شبکه و شبکه خود را با سیاست‌های جداگانه شبکه بین ابر برای اتصال در مراکز داده اجراکند. به منظور ساده‌سازی جستجو و مدیریت برای اهداف شبیه‌سازی، اطلاعات مکان در VM ها، VNFها و میزبان‌های فیزیکی در سطح جهان قابل‌مشاهده است.

چارچوب شبیه‌سازی جدید بر رویCloudSim ، یک چارچوب شبیه‌سازی ناخواسته مبتنی بر رویداد که به زبان جاوا اجرا می‌شود، طراحی و توسعه یافته است. از مدل برنامه نویسی شی گرا همان CloudSimاساسی آن پیروی‌می‌کند. مؤلفه‌ها در NFV و رایانه‌های لبه‌ای به عنوان کلاس‌های جاوا طراحی‌شده‌اند که می‌توانند براساس الزامات و سناریوهای شبیه‌سازی با توجه به مدل شی‌گرا تمدید یا جایگزین شوند. در این بخش، ما جزئیات طراحی و اجرای هر جز componentرا در CloudSimSDN-NFV برای شبیه‌سازی NFV و محاسبات لبه ارائه‌می‌دهیم.

۱،۲. شبیه‌سازی رویدادی

در CloudSim، هر شبیه‌سازی با ارسال و دریافت رویدادها بین موجودات رخ‌می‌دهد. به عنوان مثال، هنگامی که یک VM در یک مرکز داده ایجادمی‌شود، یک رویداد درخواست VM برای نهاد مرکز داده ارسال‌می‌شود. سپس، رویداد توسط مرکز داده دریافت‌می‌شود که با استفاده از سیاست تخصیص VM اختصاص داده‌شده به مرکز داده، منابعی را برای ماشین مجازی درخواستی اختصاص‌می‌دهد. به همین ترتیب، یک بار کاریCPU می‌تواند از طریق نهاد مرکز داده به VM ارسال‌شود. برنامه‌ریز پردازش VM رویداد ارسال بار کار را دریافت-می‌کند، زمان پایان بار کار را محاسبه‌می‌کند، سپس رویداد اتمام حجم کار را با زمان پایان محاسبه‌شده، پس‌می‌دهد.

CloudSimSDN-NFV برای شبیه‌سازی انتقال شبکه، ایجاد و حذف VNF و رویدادهای بین ابر از همان اصل پیروی‌می‌کند. رویدادها را بین نهادها می‌فرستد و دریافت می‌کند و تأخیر رویداد توسط خط مشی‌ها و برنامه‌ریزها محاسبه‌می‌شود که می‌تواند با روش‌های مختلف زمان‌بندی و الگوریتم‌ها سفارشی‌شود.

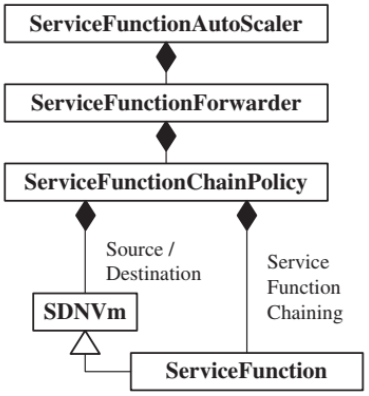
۲،۲. عملکرد شبکه مجازی(VNF)

VNFبا گسترش کلاس VM از CloudSimSDN اصلی طراحی و اجرا می‌شود. MIPS (میلیون دستور در هر ثانیه)، اندازه حافظه و اندازه ذخیره‌سازی. علاوه بر این مشخصات عمومیVM ، VNFها دارای یک زمینه خاص به نامMIPO (میلیون دستورالعمل در هر عملیات) هستند که توان تولید VNF را مدل‌می‌کند MIPO .طول بار CPU را برای یک عملیات شبکه ارائه‌شده توسط VNF مشخص‌می‌کند، که می‌تواند توان VNFرا به همراه MIPS فراهم‌کند. به عنوان مثال، یک VNF با ظرفیت ۱۰۰۰ MIPS و ۱۰MIPO می‌تواند ۱۰۰ درخواست (عملیات) در هر ثانیه را کنترل‌کند. MIPSهمچنین توسط سیاست‌های تخصیص VNF / VMمورد استفاده قرارمی‌گیرد، که تصمیم‌می‌گیرد یک میزبان فیزیکی VM را قراردهد. اگر میزبان فاقد MIPS موجود برای VM / VNFدرخواستی باشد، سیاست تخصیص سعی‌می‌کند آن را در میزبان دیگر اختصاص دهد.

جدول ۱: پارامترهای الگوی توصیف کننده‌های زنجیره‌ای عملکرد

Type	Parameters
Node	name datacenter type subdatacenters size host pes mips mipoper ram
Links	Name

	Source destination bandwidth
Policies	Name Source destination flowname sfc expected_time



شکل ۳: نمودار کلاس برای شبیه‌سازی زنجیره عملکرد سرویس‌ها

۳،۲. زنجیره عملکرد سرویس(SFC)

ما قابلیت شبیه‌سازی SFC را در چارچوب ارائه‌می‌دهیم (شکل 1 را ببینید) SFC .به عنوان یک زنجیره از توابع سرویس چندگانه (به عنوان مثال، VNFها) به عنوان یک لیست مرتب و هدایت‌شده تعریف‌می‌شود. اجرای SFC در ServiceFunctionForwarder تعیین‌می‌شود، که وظیفه ارسال بسته همسان به زنجیره SF ها را دارد.

فوروارددر هر ServiceFunctionChainPolicy تعریف‌شده در پیکربندی شبیه‌سازی فعلی را بررسی‌می‌کند، که از ماشین‌های مجازی مبدا و مقصد تشکیل‌شده‌است. به عنوان مثال، اگر یک جریان شبکه از VM1به VM2 مجبور باشد از دو زنجیره دوVNF ، VNF1و VNF2 عبورکند، در این صورت می‌توانیم با ایجاد دو ماشین مجازی و دو VNF در مرکز داده، تعریف SFC با VNF1 وVNF2 ، و اجرای سیاستی برای هدایت جریان شبکه از VM1 به VM2 برای عبور از SFC تعریف شده‌است.

این SFC ها و سیاست‌های اجرایی در پیکربندی شبیه‌سازی تعریف می‌شوند، هنگام شروع و اجرای شبیه‌سازی برای همه ترافیک‌های شبکه مطابق با خط مشی، بارگیری و مستقر می‌شوند.

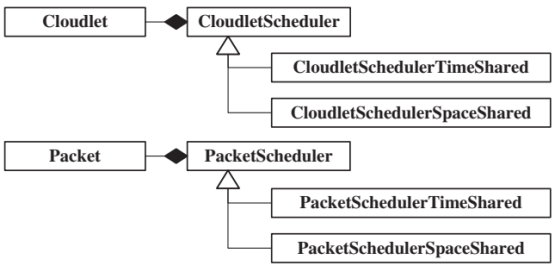
پیکربندی شبیه‌سازی توپولوژی مجازی (جدول ۱) به زبان JSON شرح داده‌شده شامل سه دسته است:

۱) الگوی گره نمونه (VNF، VM، کانتینر)، منابع مورد نیاز (اندازه دیسک، CPU [pe]، سرعت پردازنده [mips] ، اندازه حافظه [ram])و سرعت پردازش بسته‌ها MIPO برای نوع گره (mipoper) VNF و ۲) نیاز تخصیص اختیاری برای نمونه (مرکز داده، مرکزهای فرعی و میزبان) و ۳) نیاز به تخصیص اختیاری برای نمونه (مرکز داده، مرکز داده فرعی و میزبان) تعریف‌می‌شود.

(2)الگوی پیوندهای مجازی نام پیوندهای مجازی بین منبع اصلی منبع / ظرف به مقصد / VM ظرف با پهنای باند خاص را به عنوان یک گزینه تعریف‌می‌کند. اگر نام پیوند مجازی "پیش‌فرض" باشد، برنامه ریز بسته پس از جلب رضایت پیوندهای غیرپیش‌فرض، پهنای باند باقیمانده را به پیوند اختصاص‌می‌دهد. سیاست مربوط به پردازش بسته را می‌توان بر اساس سناریوهای شبیه‌سازی کاربر تغییرداد.

(3) قسمت سوم الگو همان توصیف نمودارهای انتقال VNF (VNFFGD) است که درTacker of OpenStack برای ایجاد فوروارد و SFC استفاده شده‌است. با SFC police، منبع و مقصد SFC را مشخص‌می‌کند و اینکه کدام پیوند باید در SFC (flowname)، دنباله زنجیره‌ای VNF جهت‌دار پیوند اجراشود (به عنوان مثال، ["vnf4", "vnf2", "vnf1"]:"sfc") و همچنین کیفیت خدمات پیش‌بینی‌شده (QoS) از نظر تأخیر انتها به انتهای لینک (expect_time) می‌باشد با این حال، کاربران همچنین می‌توانند در هر زمان از شبیه‌سازی، VMs/VNFs جدید را تغییر داده و ایجادکنند و اتصال و همچنین منابع اختصاص‌یافته را با توجه به الگوریتم‌های تعریف‌شده توسط کاربر به صورت پویا تغییردهند.

در سیاست‌های SFC ، سیاست مقیاس‌گذاری خودکار را برای SFC ها اجرامی‌کنیم. در هر بازه زمانی، ServiceFunctionAutoScaler تأخیر متوسط پایان به پایان بسته‌ها را در SFC و استفاده از SF ها را در SFC بازیابی‌می‌کند. براساس سیاست مقیاس‌گذاری خودکار از پیش تعریف شده، فروشنده اتوماتیک می‌تواند ظرفیت SF (vertical scale) و یا تعداد دستگاه‌های SF را برای عملکرد خاص (horizontal scale) افزایش دهد. اگر گلوگاه در شبکه باشد، همچنین می‌تواند پهنای باند اختصاص‌یافته را برای SFC خاص افزایش‌دهد. ما یک سناریو مورد استفاده از مقیاس‌گذاری خودکار SFC را در بخش ارزیابی ارائه‌می‌دهیم.



شکل ۴:PacketScheduler برای شبیه‌سازی بسته‌های شبکه همراه CloudletSchedulerنسخه اصلیCloudSim

۴،۲. برنامه‌ریز بسته

ما برنامه‌ریز بسته را مشابه زمانبندی Cloudlet درCloudSim، همانطور که در شکل ۴ نشان داده‌شده ، مدل می‌کنیم. در CloudSim اصلی، یک بار محاسباتی برای پردازش CPU به عنوان Cloudlet مدل‌سازی می‌شود که دارای CloudletScheduler برای کار پردازش است .وظیفه برنامه‌ریزی حجم کار پردازش در هر ماشین مجازی را بر اساس سناریوی شبیه‌سازی می‌باشد. در

CloudletSchedulerTimeShared، ظرفیت پردازنده به طور مساوی در تمام Cloudlets ارسال‌شده و در حال حاضر در VM پردازش می‌شود. به عنوان مثال، اگر پنج Cloudlet به یک VM ارسال‌شود، ظرفیت CPU VM در میان آن‌ها به اشتراک گذاشته می‌شود تا به هر Cloudlet ۲۰٪ از کل ظرفیت CPU اختصاص‌یابد. از طرف دیگر، CloudletSchedulerSpaceShared هربار فقط یک Cloudlet را پردازش‌می‌کند تا ۱۰۰٪ ظرفیت پردازنده به اولین Cloudlet ارسال شده به VM اختصاص‌یاب Cloudlet های دیگر در لیست انتظار قرار دارند و هنگامی که Cloudlet زودتر پردازش را انجام‌داد، در صف پردازش قرارمی‌گیرند.

درCloudSimSDN-NFV ، ما زمان‌بندی بسته‌های شبکه را با Packet‌و PacketScheduler class در Cloudlet‌و CloudletSchedulerشبیه‌سازی میکنیم Packet Class نشان دهنده بار کاری انتقال شبکه است که اندازه بسته شبکه را دارد. . PacketScheduler پهنای باند شبکه موجود را بین بسته‌هایی که در حال انتقال بسته‌های مجازی با همان مبدا و مقصد هستند، توزیع می‌کند. اگر چندین جریان دارای پیوند فیزیکی یکسان باشند، پهنای باند پیوند فیزیکی بین این جریان‌ها توزیع‌می‌شود، اگر چندین جریان پیوند فیزیکی یکسانی داشته‌باشند، پهنای باند پیوند فیزیکی بین این جریان‌ها توزیع‌می‌شود.

مشابهCloudletScheduler ، ما PacketScheduler را با دو مدل تقسیم زمان و اشتراک فضا پیاده‌سازی‌می‌کنیم. در به اشتراک‌گذاری زمان، پهنای باند موجود به طور مساوی در میان بسته‌ها از همان منبع VMبه همان مقصد VM تقسیم‌می‌شود. دSpaceShared ، کل پهنای باند شبکه مجازی به اولین بسته ارسالی به شبکه اختصاص‌یافته‌است و بقیه در صف منتظر هستند تا انتقال بسته اول به اتمام رسد.

۵،۲. محاسبه لبه

برای شبیه‌سازی محاسبه لبه، ما منبع لبه را به عنوان بخشی از یک مرکز داده لبه در نظر می‌گیریم که دارای ویژگی مشابه یک مرکز داده ابری است. به عنوان مثال، مرکز داده edge دارای میزبان‌های فیزیکی با ظرفیت و سوئیچ محدود برای اتصال میزبان‌ها و کاربران نهایی است. تفاوت بین مراکز داده edge و cloud اندازه و مقیاس منابع ارائه‌شده در مرکز داده خواهدبود. از نظر شبیه‌سازی، یک مرکز داده لبه و یک مرکز داده ابری را می‌توان همان موجود درنظرگرفت، در حالی که مرکز داده لبه در مقایسه با مرکز داده ابری، مقدار کمتری از منابع محاسباتی را تأمین‌می‌کند. به غیر از ظرفیت منابع، هر دو مرکز داده‌های edge و cloud می‌توانند سیاست‌های مدیریت منابع محاسباتی و قوانین شبکه خود را داشته‌باشند. بنابراین، ما از کلاس مرکز داده که در CloudSimSDNاصلی پیاده‌سازی شده‌است، استفاده‌می‌کنیم تا مراکز داده لبه و ابر را نشان‌دهد.

با این حال، شبیه‌سازی مراکز داده‌های edge و cloud در یک سناریو باید چندین مرکز داده را همزمان اجراکند . CloudSimSDN اصلی به طور بالقوه قادر به پشتیبانی از چندین مرکز داده در طراحی آن بود، اما به‌درستی اجرا نشد. در این کار، ما سناریوی مرکز داده multicloud را در نظر می‌گیریم و نهادهای مرتبط را برای کار بدون تداخل اشیا با سایر مراکز داده کارمی‌کنید. چارچوب جدید می‌تواند چندین مرکز داده با تنظیمات مختلف ایجادکند، به عنوان مثال، یک مرکز داده لبه با چند میزبان با ظرفیت کم و یک مرکز داده در مقیاس بزرگ با هزاران میزبان با عملکرد بالا. پیکربندی‌های شبکه را می‌توان به طور مستقل تنظیم کرد تا سناریوی دنیای واقعی را منعکس‌کند.

با اجرای صحیح مراکز داده متعدد، شبکه‌های بین ابر در این کار طراحی و اجرا شده‌اند تا از قاچاق شبکه بین مراکز داده پشتیبانی کنند. ما کلاس سوئیچ اصلی را به کلاس InterCloudSwitch گسترش دادیم تا سوئیچ‌های ستون فقرات را بهمراه متصل کردن مراکز داده توزیع شده نشان‌دهیم. علاوه بر این، کلاس GatewaySwitch برای نشان‌دادن سوئیچ‌های دروازه در یک مرکز داده معرفی شده‌است. ترافیک شبکه از یک مرکز داده به مرکز دیگر باید از طریق دروازه مرکز داده مبدا، سوئیچ‌های توزیع شده بین ابر، دروازه مرکز داده مقصد و سپس به میزبان مقصد عبورکند. توپولوژی‌های مختلف شبکه را می‌توان به صورت جداگانه برای مراکز داده مختلف و شبکه‌های بین ابر تعریف‌کرد. به عنوان مثال، یک مرکز داده لبه را می‌توان با یک توپولوژی درخت متعارف، یک مرکز داده ابر با درخت چربی و شبکه بین ابر با توپولوژی مش تعریف کرد. به‌علاوه، از آنجا که عناصر مسیریابی SDN در اصل SDN از عناصر فیزیکی جدا شده‌اند، هر میزبان و سوئیچ به عنوان یک Node Class برای مسیریابی مجازی در نظر گرفته‌می‌شود. بنابراین، مراکز داده را می‌توان با توپولوژی ترکیبی یا سرور محور، مانندBCube ، Dcellو hypercube، بدون سوئیچ نیز تعریف‌کرد.

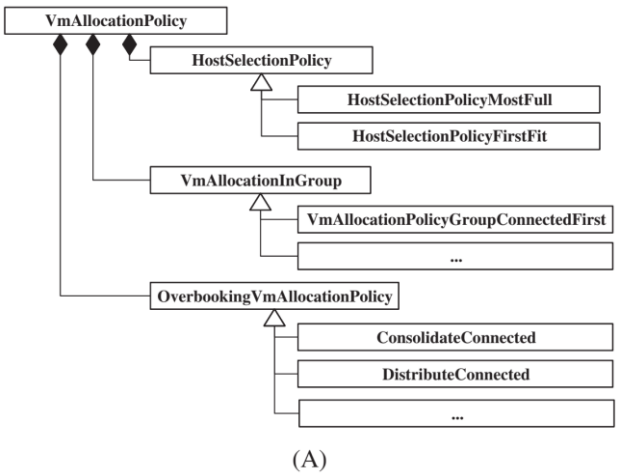
۶،۲. سیاست‌های قابل برنامه‌ریزی

درCloudSimSDN-NFV، ما ویژگی‌هایی را برای ارزیابی و شبیه‌سازی محاسبه لبه و ابر دارای قابلیت SDN-NFV از نظر محاسبات، شبکه، نظارت و مجازی‌سازی مربوطه اجراکردیم (جدول ۲). سیاست‌ها، فناوری مجازی‌سازی و الگوریتم‌ها را می‌توان براساس بهینه-سازی این پارامترها مانند تغییر مسیر جریان پویا بر اساس سرعت پردازش VNF و تأخیر پایان به پایان در امتداد SFC و بهینه‌سازی تخصیص برای جدید، پیاده‌سازی و ارزیابی‌کرد. VNFها را در طول مقیاس‌گذاری خودکار بر اساس اتصال، توپولوژی شبکه و پهنای باند موجود تکرارمی‌کنید. بر اساس چارچوب شبیه‌سازی فعلی، ما الگوریتم تأمین پویای VNF آگاهانه تأخیر را برای استفاده از منابع لبه و ابر و چارچوب یکپارچه الگوریتم‌های مختلف اندازه‌گیری خودکار ارزیابی‌کرده ایم تا هزینه‌ها را با ضمانت نامه‌های سطح سرویس (SLA) و به حداقل برسانیم. تأخیر پایان به پایان در امتداد SFC ها می‌باشد.

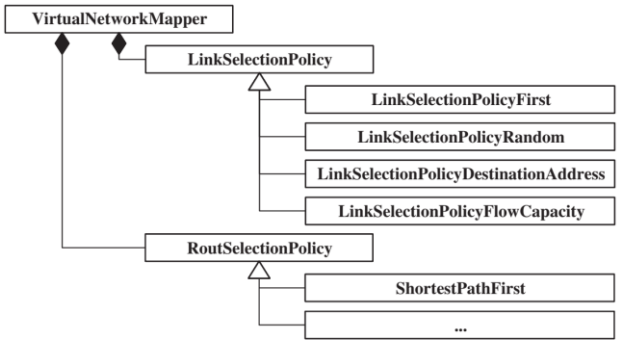
علاوه بر سیاست‌های خودکارسنجی که در بخش ۴،۳ مورد بحث قرارگرفته است، ما سیاست‌های تخصیص VM / VNF / Container و نقشه برداری شبکه را برای سفارشی‌سازی ساده همانطور که در شکل ۵

نشان داده شده ، مدوله‌می‌کنیم. توجه‌داشته‌باشید که ما تمام الگوریتم‌هایی را که ما پیاده‌سازی کرده‌ایم نشان‌نمی‌دهیم، اما سیاست‌های اصلی الگو را نشان‌می‌دهیم VmAllocationPolicy . تعریف‌شده در CloudSim اصلی با میزبان جداشده HostSelectionPolicy گسترش یافته‌است، که می‌تواند بهترین میزبان را برای تخصیص ماشین مجازی در میان میزبان‌ها در لیست نامزدها انتخاب‌کند که براساس سیاست بهینه‌سازی مرتب‌سازی می‌کند. ما دو سیاست را به عنوان الگوی اصلی برای انتخاب میزبان پیاده‌سازی می‌کنیم: اولین روش مناسب: اولین تناسب این است که اولین میزبان ظاهرشده در لیست بازگردد، که منابع کافی برای VM درخواست‌شده در دسترس است. کاملترین روش یافتن بیشترین میزبان در لیست است که همچنان می‌تواند منابع درخواستی را فراهم‌کند. طبق جدول ۲ با توجه به اهداف بهینه‌سازی، توالی لیست میزبان می‌تواند بر اساس سایر پارامترها مانند محاسبات، شبکه و نظارت باشد. بر اساس اتصال بین VNFها، ماشین‌های مجازی و کانتینرها، ما همچنین سیاست VmAllocationPolicyGroupConnectedFirstرا برای نمایش پسوندهای بالقوه پیاده‌سازی‌می‌کنیم. علاوه بر این، ما سیاست‌های تخصیص پیچیده‌تری را برای استفاده بیش از حد و بیش از رزرو کردن پردازنده، حافظه، دیسک و پهنای باند با استفاده از مهاجرت، برای شروع و متعاقباً مدیریت منابع پویای OverbookingVmAllocationPolicy پیاده‌سازی می‌کنیم. بنابراین، سیاست‌های دیگر را می‌توان با رعایت کد منبع موجود با حداقل تلاش برای توسعه به‌راحتی اجراکرد.

برای یک شبکه مجازی، VirtualNetworkMapper Classوظیفه نقشه برداری توپولوژی شبکه مجازی را بر روی عناصر شبکه فیزیکی بر عهده دارد. مشابه HostSelectionPolicy، یک کلاس LinkSelectionPolicyجداگانه پیاده‌سازی‌می‌کند که پیوند را برای انتخاب چندین لینک در توپولوژی فیزیکی پیوند می‌دهد. ما به عنوان مثال چندین روش را در چارچوب اجرامی‌کنیم.



(A)



(B)

شکل ۵: سیاست‌های مهندسی تخصیص و ترافیک شبکه در چارچوب اجرا شده است. A، سیاست انتخاب میزبان ؛B ، خط مشی انتخاب لینک و مسیر

جدول ۲ پارامترهای اساسی پشتیبانی شده برای شبیه‌سازی

Type	Parameters
Computing	CPU Memory Disk Workloads Task scheduling Task priority Overbooking ratio
Networking	Bandwidth Topology Switch buffer Ports Channel priority Control plane Data plane
Monitoring	Statistic Energy consumption Utilization Response time Network delay Fault handling
Virtualization	Connectivity Allocation Lifecycle Flow scheduling QoS SLA

LinkSelectionPolicyFirst بدون درنظرگرفتن ظرفیت شبکه، اولین لینک در لیست را انتخاب‌می‌کند. به همین ترتیب، می‌توان یک پیوند تصادفی را با خط‌مشی تصادفی انتخاب‌کرد. به صورت عملی‌تر، LinkSelectionPolicyDestinationAddress آدرس مقصد شبکه را جستجومی‌کند و پیوند خاصی را محاسبه‌شده توسط یک عملکرد مدول اختصاص‌می‌دهد.

به عنوان مثال، اگر دو پیوند موجود باشد، آدرس مقصد با شماره فرد پیوند اول و عدد زوج پیوند دوم را انتخاب‌می‌کند. LinkSelectionPolicyFlowCapacity پیوند با بیشترین ظرفیت باقی‌مانده مانند پهنای باند موجود یا بافر باقیمانده را در سوئیچ‌ها انتخاب می‌کند.

این ظرفیت شبکه اشغال‌شده لینک‌های موجود در لیست نامزدها را بررسی‌می‌کند و خالی‌ترین آن‌ها را در بین آن‌ها برمی‌گرداند. بنابراین، انتقال شبکه می‌تواند به‌طور مساوی بین چندین مسیر توزیع‌شود. LinkSelectionPolicy نیز برای اجرای یک روش جدید به‌راحتی قابل‌تنظیم است.

علاوه بر این، مطابق شکل B ۵ ، ما همچنین سیاست‌های مسیریابی جریان فعال در شبکه مجازی SDN را اجرا می‌کنیم. بر اساس توپولوژی شبکه فیزیکی و مجازی جهانی که در CloudSimSDN-NFV ارائه شده است، می‌توانیم هر مسیریابی مجازی یا فیزیکی خاصی را به جداول هدایت هر گره هدایت‌کنیم، مانند الگوریتم مبتنی بر کوتاهترین مسیر اول با و از پهنای باند موجود جهانی استفاده کنیم.

علاوه بر این، محققان همچنین می‌توانند الگوریتم فعال مهندسی ترافیک، به عنوان مثال، شبیه‌سازی سیاست‌های امنیت و صلاحیت شبکه را گسترش‌دهند:

برخی از جریان‌ها باید به گروه‌های خاصی از سرورها در حوزه قضایی و جریان داده‌های منطقه‌ای منتقل‌شوند. به طور کلی، برای محققانی که می‌خواهند رویکردهای جدید در تخصیص ماشین مجازی و یا نقشه-برداری شبکه را ارزیابی‌کنند، می‌توانند این سیاست‌ها را با روش پیشنهادی اصلاح کرده و برای ارزیابی اثربخشی رویکرد جدید با خطوط مقدماتی مقایسه‌کنند.

۷,۲. مانیتور مصرف و استفاده از انرژی

چارچوب جدید همچنین شامل monitoring مؤلفه‌های نظارت بر مصرف انرژی و استفاده از منابع است. یک رابط جاوا برای ثبت سابقه استفاده از هر عنصر (به عنوان مثال، یک سوئیچ، یک میزبان و یک ماشین مجازی) تعریف شده‌است. پس از زمان تنظیم شده از قبل نظارت، رویداد نظارت برای اندازه گیری میزان بار کار پردازش شده و محاسبه میزان استفاده از عنصر برای دوره زمانی قبلی آغاز می‌شود.در پایان هر رویداد نظارتی، همان رویداد با فاصله زمانی نظارت دوباره ارسال می‌شود تا سابقه استفاده به صورت دوره‌ای ثبت شود.

علاوه بر ضبط استفاده، برآورد مصرف انرژی برای بازه زمانی نیز با یک مدل خطی محاسبه می‌شود.

در نسخه فعلی، ما از مدل خطی برای تخمین میزان مصرف برق برای میزبان‌ها و سوئیچ‌ها استفاده‌کرده‌ایم، که با توجه به سناریو شبیه‌سازی و نیاز، می‌توان آن را با مدل‌های دیگر جایگزین‌کرد.

۳. ارزیابی با استفاده از دو مورد استفاده

ما چارچوب شبیه‌سازی پیشنهادی را با دو سناریوی مورد استفاده برای NFVو رایانه لبه‌ای ارزیابی‌کردیم. این بخش به بررسی و مقایسه اثربخشی سیاست‌های مختلف در NFV با دو سناریوی مورد استفاده می‌پردازد.

در سناریوی اول، ما تأخیر پایان به پایان و هزینه برآوردشده را با سیاست‌های مختلف اندازه‌گیری خودکار SFC ارزیابی‌می‌کنیم. ما سیاست مقیاس‌گذاری خودکار را برای ظرفیت پردازش VNF ها و ظرفیت پهنای باند برای زنجیره زنی بر چارچوب شبیه‌سازی برای مقایسه تأخیر اندازه‌گیری شده و استفاده از منابع انجام‌می‌دهیم.

در سناریوی دوم، مرکز داده لبه با منابع محدود ایجاد شده‌است تا برخی از VNF ها را میزبانی‌کند.

ما دو سیاست قراردادن VNF را برای انتخاب بین مراکز داده edge و cloud پیاده‌سازی‌می‌کنیم:

استفاده از گره‌های لبه‌ای برای قراردادن VNF یا همه VNF هایی که در ابر قرارمی‌گیرند. این دو سناریو موردی استفاده بالقوه از چارچوب شبیه‌سازی را در تحقیقات مختلف NFV و محاسبات لبه ارائه می‌دهند.برای هر دو سناریو، ما بر اساس ردیابی ویکی پدیا و پیروی از مدل کاربردی سه لایه، بارهای مصنوعی ایجاد کردیم.

۱,۳. آزمایش ۱: سیاست‌های مقیاس‌گذاری خودکار SFC در NFV

اولین مورد استفاده برای ارزیابی سیاست‌های مختلف اندازه‌گیری خودکار SFC در یک مرکز داده ابری است.

در این سناریو، ما یک مرکز داده ابری در مقیاس بزرگ با ۱۲۸ ماشین فیزیکی متصل از طریق ۳۲ سوئیچ لبه، ۳۲ سوئیچ جمع و ۱۶ سوئیچ هسته‌ای ایجادکردیم، که توپولوژی شبکه درخت چربی ۸ غلاف را تشکیل‌می‌دهد .بنابراین، هر غلاف از ۴ سوئیچ لبه، ۴ کلید جمع و ۱۶ ماشین فیزیکی تشکیل‌شده‌است که ۴ ماشین به هر کلید لبه متصل می‌شوند.هر ماشین فیزیکی با ۱۶ هسته پیکربندی شده‌است و هر هسته ۱۰۰۰۰ ظرفیت MIPS دارد. پهنای باند شبکه تمام پیوندها بین سوئیچ‌ها و ماشین‌های فیزیکی به همان اندازه روی ۲۰۰ مگابایت در ثانیه تنظیم شده‌است.

جدول ۳: پیکربندی موارد مختلف در آزمایشات

(VM, virtual machine; VNF, virtualized network function)

Policy	NoScale-Max	NoScale-Min	Flavor
Type (VNF/VM)	lb/ids/fw	lb/ids/fw	web/app/db
Mem (GB)	۸	۸	۲۵۶
CPU (cores)	10/12/16	۸/۶/۲	۱۲/۴/۸
Disk (GB)	۸	۸	۱۰۰۰
Bandwidth (Bytes/s)	2 000 000	500 000	1 500 000
Number of VNFs/VMs	۳/۳/۱	۱/۱/۱	۲/۲۴/۸

جدول ۴: خط مشی‌های زنجیره عملکرد سرویس (SFC) اجرای انتقال شبکه برای هدایت به توابع شبکه مجازی (VNF)

Source	Destination	Subset of VNFs
Web server	Application server	{VNF1, VNF2}
Application server	Database	{VNF3, VNF4}
Database	Application server	{VNF4, VNF3}
Application server	Web server	{VNF2}

۲,۳. توپولوژی مجازی و پیکربندیSFC

برای توپولوژی مجازی، ما برنامه‌های وب سه‌لایه متشکل از سرورهای وب، برنامه و پایگاه داده را شبیه‌سازی‌می‌کنیم تا یک برنامه کامل را تشکیل‌دهیم. ۸ وب سرور، ۲۴ سرور برنامه و ۲ سرور پایگاه داده وجود دارد. پیکربندی انواع مختلف سرورها در جدول ۳ نشان داده شده‌است. همه سرورها به عنوان ماشین مجازی در شبیه‌سازی ایجادمی‌شوند که بارهای متشکل از پردازش CPU و انتقال شبکه را دریافت‌می‌کند.

به منظور ارزیابی عملکردهای NFV، ما ۴ VNF را از سه نوع مختلف نشان‌داده‌شده در جدول ۳ آماده‌می‌کنیم که توسط ۴ نوع مختلف سیاست‌های SFC اعمال‌می‌شوند.

VNF1:VNF Firewall (fw)

Load Balancer (lb): VNF2, VNF3

IDS: VNF4

به‌علاوه، ما MIPO را به ترتیب به‌Load Balancer ، IDS، Firewall به ترتیب ۲۰، ۲۰۰ و ۸۰۰ اختصاص‌می‌دهیم. ترافیک شبکه بین سرورهای VM به SFC های مختلف منتقل‌می‌شود که از زیر مجموعه‌ای از ۴ VNF تشکیل شده‌است همانطور که در جدول ۴ نشان داده شده‌است.

از آنجا که تمام ترافیک شبکه بین این ماشین‌های مجازی برای انتقال از طریق SFC منحرف‌می‌شود، اگر ظرفیت اولیه VNF برای تأمین کل برنامه کافی‌نباشد، این VNF۴ می‌توانند گلوگاه عملکرد برنامه باشند.

۳،۳. سیاست‌های خودکارسازی

در این سناریو، ما سه الگوریتم(NoScale-Min ، NoScale Max و AutoScale) را برای سیاست‌های مختلف اندازه‌گیری خودکار SFC در ترکیب با دو سیاست تخصیص(LFF, MFF) VM / VNF ارزیابی می‌کنیم. پیکربندی NoScale-Min و NoScale-Max انواع مختلف VNF در جدول ۳ نشان‌داده‌شده‌است.

درNoScale-Min ، ویژگی مقیاس‌گذاری خودکار خاموش شده‌است و حداقل منابع برای کل مدت آزمایش به VNF ها اختصاص‌داده‌شده است. از آنجا که VNF ها کمترین میزان منابع را بدون هیچ مقیاس‌بندی اختصاص‌داده‌اند، همیشه کمبود منابع دارند و همین امر باعث‌می‌شود بسته‌های شبکه در انتظار انتظار در VNF ها به تأخیر بیفتند.

از طرف دیگر، با استفاده ازCPU ، پهنای باند و همان نوع VNF بیشتر در یک پلیسSFC ، سیاست NoScale-Max بیش از منابع کافی را به VNFها از ابتدای آزمایش و بدون مقیاس خودکار اختصاص می‌دهد. از آنجا که مقدار زیادی از منابع برای VNF از ابتدا تخصیص یافته است، اگر حجم کار ارسالی کمتر از برآورد باشد، می‌توان از آن‌ها کم استفاده کرد. در سیاستAutoScale ، تخصیص منابع اولیه همان-NoScale Minاست که حداقل منابع را در ابتدای آزمایش اختصاص‌می‌دهد.

با این حال میزان استفاده از VNF ها را دائماً رصدمی‌کند و وقتی سطح استفاده از آستانه تعریف‌شده‌باشد، اندازه VNF را افزایش‌می‌دهد. در این آزمایش، آستانه ۷۰ set را برای اتوکسیلینگ تعیین‌کردیم.

اگر امکان افزایش ظرفیت برای افزایش مقیاس در میزبان فعلی به دو برابر اندازه فعلی وجود داشته باشد (در این حالت دو برابر ظرفیت CPU و پهنای باند تخصیص‌یافته)، پس سیاست تکثیر با تکثیر VNF از همان نوع است یک گزینه نیست.

در غیر این صورت، ما VNF دیگری را ایجاد می‌کنیم و آن را مطابق با سیاست LFF یا MFF تخصیص‌می‌دهیم. علاوه بر سیاست‌های مختلف اندازه‌گیری خودکار، ما اثربخشی سیاست‌های مختلف تخصیص VMو VNF را ارزیابی‌می‌کنیم. الگوریتم LFF(least full first) کمترین ماشین‌های فیزیکی را با مقدار کمتری از منابع اختصاص یافته برای یک ماشین مجازی پیدامی‌کند .در این الگوریتم، VMها و VNF ها در مرکز داده پراکنده‌می‌شوند، زیرا یک ماشین فیزیکی میزبان یک VMدارای اولویت کمتری نسبت به یک ماشین بدون هیچ VM میزبانی است.

در مقابل، الگوریتم MFF(most full first) الگوی اختصاص‌یافته را انتخاب‌می‌کند که ظرفیت کافی برای میزبانی ماشین مجازی را دارد . ماشین فیزیکی انتخاب‌شده هنوز منابع کافی برای ماشین مجازی دارد، اما در حال حاضر منابع بیشتری در مقایسه با سایر ماشین‌های نامزد به سایر ماشین‌های مجازی اختصاص داده‌شده‌است. ماشین‌های مجازی با استفاده از این الگوریتم در تعداد کمتری از ماشین‌های فیزیکی ادغام می‌شوند، که می‌تواند تعداد ماشین‌آلات و مصرف انرژی مرکز داده را کاهش‌دهد.

علاوه بر این، در سناریوهایی که IDS اختیاری است، ما تأثیر طول SFC و تأخیر پردازش VNF را در زمان پاسخ پایان به پایان ارزیابی می‌کنیم. با حذف VNF4 از سیاست‌های SFC که در جدول ۴ نشان داده شده‌است، طول کل سیاست‌های SFC از ۷ به ۵ تغییرمی‌یابد. ما سه خط‌مشی خودکار را با دو سیاست تخصیص VM / VNF ترکیب می‌کنیم تا شش مورد ایجادکنیم و در موارد زیر موجود است.

۴،۳. نتایج ارزیابی

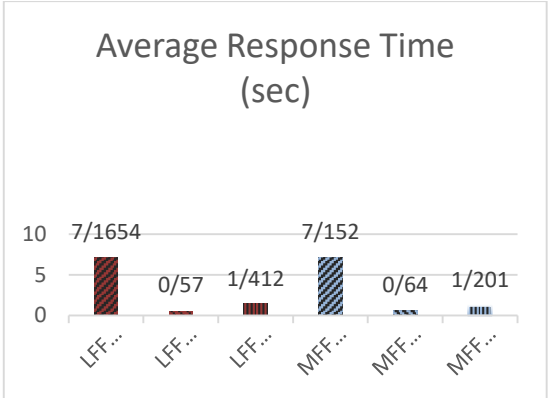
در ابتدا، مطابق شکل ۶، زمان پاسخ تمام بارهای ارسال‌شده به برنامه را اندازه‌گیری‌می‌کنیم. متوسط زمان پاسخ پایان به انتها در شکل A۶ نشان‌داده‌شده‌است. همانطور که انتظار می‌رفت، سیاست‌های NoScale-Min در هر دو LFF و MFF منجر به زمان پاسخ استثنایی طولانی‌تری به دلیل کمبود منابع در VNF ها می‌شود.

از طرف دیگر، تأخیر در سیاست NoScale-Max کوتاه است زیرا منابع کافی به VNF اختصاص‌می‌یابد. در سیاست AutoScale، زمان پاسخ کمی بیشتر از NoScale-Max است اما تا NoScale-Min بیشتر نیست، زیرا VNF های بسیار پرکاربرد پس از مدت زمان کوتاه می‌توانند منابع بیشتری را توسط سیاست مقیاس خودکار به‌دست آورند.

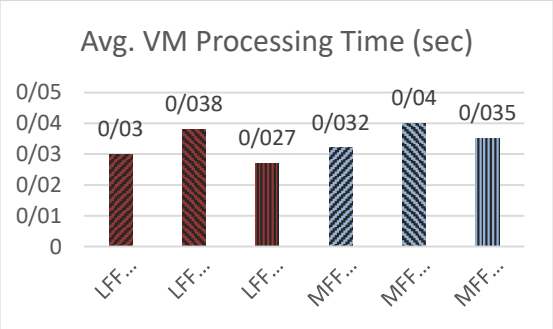
تفاوت مشاهده شده بین LFF و MFF در هر سیاست autoscaling قابل‌توجه نیست. برای توضیح بیشتر در مورد متوسط زمان پاسخ (تأخیر پایان به پایان)، آن را به زمان پردازش سرورهای برنامه و تأخیرهای شبکه در طول SFC ها تفکیک‌می‌کنیم.

شکل B۶ و شکل C۶ به ترتیب میانگین زمان پردازش VM در سرورهای برنامه سرویس (وب، برنامه و VM پایگاه داده) و میانگین زمان انتقال شبکه شامل تأخیر در VNF های اعمال‌شده در SFC را نشان‌می‌دهد.

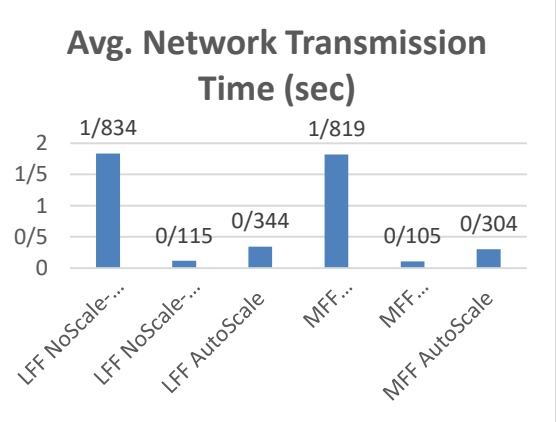
ما می‌توانیم مشاهده‌کنیم که زمان پردازش برنامه در سرورهای VM در هر شش ترکیب یکسان است، که توضیح می‌دهد قدرت پردازش برای سرورهای وب، برنامه و پایگاه داده برای پردازش تمام بارهای کاری در مدت زمان کوتاه کافی است. با این حال، زمان انتقال شبکه به دلیل تأخیر در VNF ها که توسط اجرای SFC رخ‌داده‌است، بین ترکیب سیاست‌های مختلف تفاوت معنی‌داری دارد.



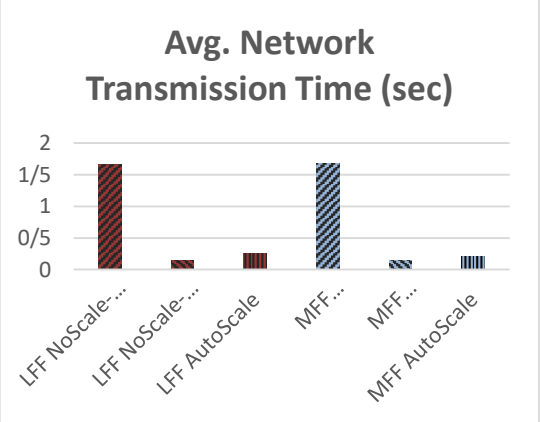
(A)



(B)

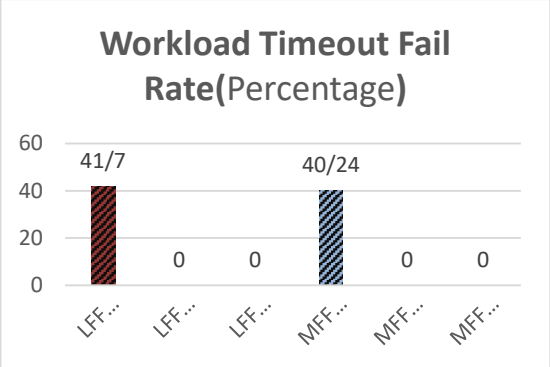


(C)



(D)

نمودار۱: زمان پاسخگویی و پردازش متوسط با سیاست‌های مختلف اندازه‌گیری خودکار



نمودار۲ : میزان توقف خرابی بار کاری

از آنجا که پهنای باند برای خدمات بسیار کافی است، تأخیرهای VNF در طول SFC باعث تأخیر کلی پایان به پایان می‌شود که شامل زمان پردازش بسته و زمان انتظار بافر است. در NoScale-Min، منابع اولیه اختصاص‌داده برای VNF برای پردازش ترافیک شبکه کافی نیستند، که منجر به تأخیر بسته‌ها در SFC می‌شود.

NoScale-Max و سیاست‌های AutoScale زمان انتقال شبکه را به میزان قابل‌توجهی کاهش می‌دهند زیرا منابع لازم برای VNF از ابتدا کافی است (NoScale-Max) یا بعد از مدتی (AutoScale) به اندازه کافی افزایش یافته‌است.

در مورد AutoScale، میانگین تأخیر شبکه در سیاست تخصیص LFF بیشتر از MFF است. سیاست LFF VNF تکرارشده جدید را به میزبان‌های جدید اختصاص‌می‌دهد. در نتیجه، تخصیص توزیع‌شده بیشتر به دلیل داشتن شبکه‌های متغیر شبکه منجر به زمان انتقال شبکه بیشتر می‌شود.

علاوه بر این، در سناریوی عدم وجود IDS در SFC، شکل D۶ نشان می‌دهد که تأخیر شبکه در طول SFC به همین دلیل در نتیجه طول SFC کوتاهتر کاهش‌می‌یابد. ما همچنین در هر خط‌مشی میزان وقفه شبکه را اندازه‌گیری‌کردیم که منجر به خرابی کار می‌شود. آستانه مهلت زمانی بر روی ۵ ثانیه تنظیم‌شده‌است تا بار کاری که بیش از ۵ ثانیه طول بکشد کاهش‌یابد و به عنوان بسته زمان پایان محاسبه‌شود. شکل ۷ میزان وقفه در تمام بارهای کاری را با خط مشی‌های مختلف نشان می‌دهد.

سیاست NoScale-Min منجر به کاهش بیش از ۴۰٪ حجم کار به وقت منقضی می‌شود، در حالی که هیچ‌یک از بارهای کاری در دو سیاست دیگر کاهش‌نمی‌یابد. سپس، ما حداکثر تعداد میزبان‌های همزمان را اندازه‌گیری‌کردیم. مطابق شکل ۸، الگوریتم کلی LFF از میزبان‌های بیشتری استفاده‌می‌کند. برای الگوریتم AutoScale SFC، MFF در مقایسه با LFF، که به طور کامل از منابع محاسباتی استفاده می‌کند، خط‌مشی بهتری است.

اگرچه برای کاهش خرابی کار، تعداد میزبان‌های همزمان استفاده-شده از MFF خودکار در مقیاس بزرگتر از MFF NoScale-Min بیشتر است، اما در مقایسه با MFF NoScale-Max کمتر است. با

استفاده از میزبان کمتر، MFF AutoScale درآمدزایی را برای ارائه-دهنده خدمات به حداکثر می‌رساند و همچنین صرفه‌جویی در هزینه را برای کاربران cloud ارائه می‌دهد.

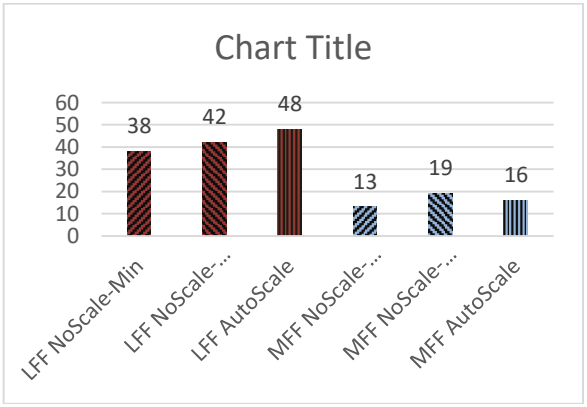
در آخر، ما مطابق شکل ۹ ، میزان انرژی برآورد شده را در مرکز داده برای میزبان‌ها و سوییچ‌ها اندازه‌گیری می‌کنیم. تفاوت واضحی بین LFF و MFF مشاهده‌می‌شود، زیرا LFF در هر سیاست مقیاس خودکار در مقایسه با MFF انرژی بیشتری مصرف‌می‌کند.

به عنوان مثال، LFF با AutoScale 984.49 Wh در میزبان‌ها و ۴۷۴٫۵۶ Wh در کل سوئیچ‌ها ۱۴۵۹٫۰۵ Wh مصرف‌کرد، در حالی که MFF با AutoScale در کل ۴۷۶٫۴۹ (در ۳۷۹٫۶۶ در میزبان‌ها به اضافه ۹۶٫۸۳ در سوئیچ‌ها) مصرف‌کرد.

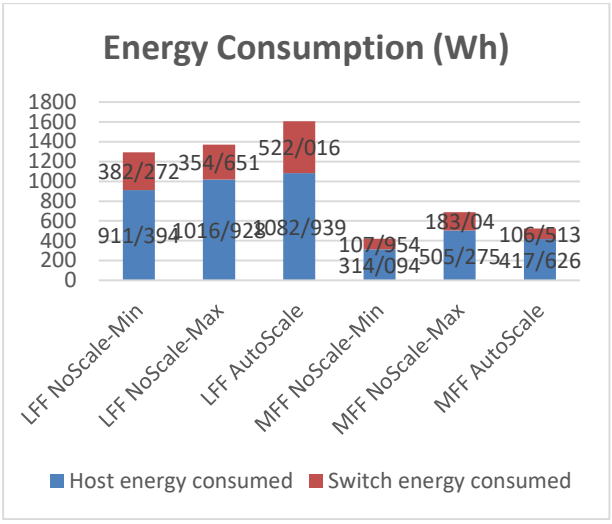
در MFF، تعداد بیشتری از ماشین‌های مجازی در یک میزبان اختصاص‌می‌یابد که منجر به تلفیق ماشین‌های مجازی و حجم کار در تعداد کمتر میزبان می‌شود. میزبان‌های استفاده‌نشده را می‌توان به حالت آماده به کار تبدیل‌کرد، که می‌تواند به میزان قابل‌توجهی در مصرف برق صرفه‌جویی‌کند.

جالب اینجا است که در سیاست تخصیص LFF در AutoScale در مقایسه با NoScale-Max انرژی بیشتری مصرف‌شد زیرا تعداد بیشتری VNF کوچکتر به دلیل سیاست LFF در AutoScale ایجادشد. در LFF، VNF ها در سراسر مرکز داده توزیع می‌شوند که منجر به استفاده از تعداد بیشتری از میزبان‌های فیزیکی می‌شود.

VNF های تکراری ایجادشده از سیاست مقیاس‌گذاری اتوماتیک پس از قرارگیری اولیه نیز پراکنده‌می‌شوند و منجر به استفاده از میزبان‌های فیزیکی بیشتری می‌شوند. علاوه بر این، این VNF های پراکنده باعث افزایش حجم ترافیک شبکه در سیستم می‌شوند که نتیجه آن مصرف بیشتر انرژی در سوئیچ‌ها است.



نمودار۳: به طور همزمان از شماره میزبان استفاده‌می‌شود.



نمودار۴: مصرف انرژی در میزبان‌ها و سوئیچ‌ها در مرکز داده.

۵،۳. آزمایش ۲: سیاست قرار دادن NFV در محیط لبه-ابر

در آزمایش دوم، ما سیاست‌های مختلف قراردادن VNF را در محیط‌های لبه-ابر ارزیابی‌می‌کنیم. ترکیب اجزای مختلف خدمات یک روش مهم برای بهینه‌سازی تخصیص منابع و استفاده از منابع لبه و ابر و همچنین کاهش تأخیر پایان به پایان برای کاربران نهایی است.

یک مرکز داده ابری با منابع بیش از حد برای تأمین زیرساخت برای ارائه خدمات گسترده به مشتریان و ساخته‌شده‌است، در حالی که مراکز داده حاشیه‌ای به منابع محدودی مانند ایستگاه پایه یا نقطه دسترسی بی‌سیم مجهز هستند. بنابراین، تعداد محدود VNF ها را می‌توان در مراکز داده لبه‌ای قرارداد، اگرچه منابع لبه می‌توانند تأخیرهای پایان به پایان را برای کاربران نهایی کاهش‌دهند زیرا ترافیک شبکه می‌تواند نزدیکتر به کاربران پردازش شود.

در این آزمایش، ما یک لبه و یک مرکز داده ابری (به عنوان مثال مرکز داده شبکه) برای قراردادن VNF ها، علاوه بر مرکز داده اضافی (یعنی مرکز داده برنامه) برای میزبانی سرورهای برنامه ایجادکردیم.

مرکز داده edge دارای ظرفیت محدودی با ۴ میزبان (هر کدام با ۱۶ هسته و ۱۰۰۰۰ / MIPهسته) است، در حالی که دو مرکز داده ابری (یکی برای سرورهای برنامه و دیگری برای عملکردهای شبکه) منابع بسیار زیادی برای میزبانی تقریباً تعداد بی‌نهایت ماشین مجازی و VNF نیاز دارد. کاربران نهایی در مرکز داده‌های لبه قرارمی‌گیرند. تأخیر شبکه یک پیوند در همان مرکز داده روی ۱ میلی‌ثانیه تنظیم‌شده‌است.

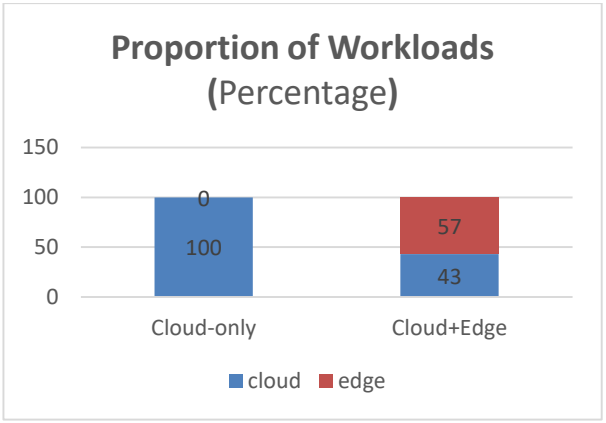
مراکز داده Edge و cloud از طریق چهار سوئیچ بین ابر به صورت خطی به یکدیگر متصل‌می‌شوند و تأخیر شبکه روی ۱۰۰ میلی‌ثانیه تنظیم شده‌است. بنابراین، تأخیر شبکه برای ترافیک بین ابر می‌تواند تا ۴۰۰ میلی‌ثانیه باشد، در حالی که در همان مرکز داده کمتر از ۱۰ میلی-ثانیه است. در این آزمایش، از یک توپولوژی درخت متعارف برای ساده-سازی ایجاد مراکز داده مختلف استفاده‌می‌کنیم.

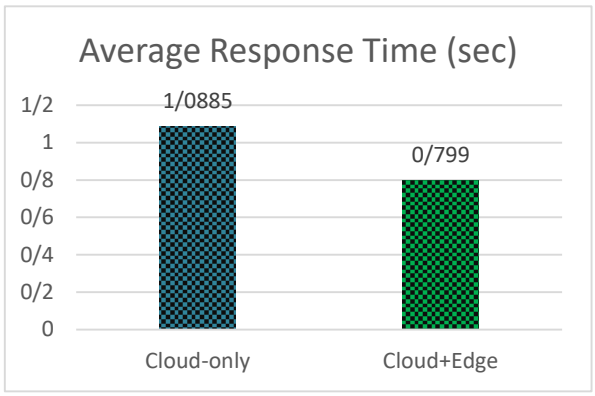
این برنامه از ۱۲ سرور تشکیل‌شده‌است که به‌صورت مجازی کار می‌کنند و در مرکز داده برنامه قرارمی‌گیرند، که توسط ۴ کاربر نهایی ایجادشده در مرکز داده لبه به آن دسترسی دارند. همه ترافیک‌های شبکه از کاربران نهایی تا سرورهای برنامه برای عبور از طریق VNF1 اعمال‌می‌شوند و ترافیک در جهت مخالف از طریق VNF2 انجام‌می‌شوند.

در این آزمایش، سیاست مقیاس‌گذاری خودکار همیشه روشن است، به طوری که اگر استفاده از VNF بیش از آستانه باشد، همانند سیاست مقیاس‌گذاری خودکار که در سناریوی قبلی آزمایش‌شده‌است، می‌توان تعداد VNF را برای خدمت به همان نوع عملکرد افزایش‌داد. هنگامی که یک VNF برای سرویس‌دهی به شبکه‌های بیشتر کپی‌می‌شود، VNF جدید را می‌توان در لبه یا ابر قرارداد. ما از سیاست‌های مختلفی برای انتخاب محل قراردادن VNF های تکراری استفاده‌می‌کنیم.

سیاست‌های قراردادن VNF: برای انتخاب مکانی برای VNF ها، دو سیاست مختلف جای‌گذاری را در شبیه‌سازی پیاده‌کردیم. خط‌مشی فقط ابر این است که فقط از مرکز داده ابری شبکه برای قراردادن VNF و بدون منابع لبه استفاده‌کنید. از آنجا که مراکز داده ابری می‌توانند ظرفیت زیادی فراهم‌کنند، این سیاست نیاز منابع VNF را در نظر نمی گیرد و تمام VNF ها را در ابر قرارمی‌دهد.

از طرف دیگر، سیاست Cloud + Edge مرکز داده edge را به عنوان گزینه در نظر می‌گیرد، بنابراین اگر منابع موجود در لبه وجود داشته باشد، ابتدا VNF جدید در لبه قرارمی‌گیرد. اگر مرکز داده لبه منابع کافی نداشته‌باشد، مرکز داده ابری میزبان VNF خواهدبود. توجه داشته باشید که مرکز داده لبه خود دارای منابع محدودی برای قراردادن VNF های مورد نیاز برای کل کار نیست، بنابراین سیاست فقط لبه در آزمایش در نظر گرفته نشده‌است.





نمودار ۵: نتایج ارزیابی عملکرد سیاست‌های قرارگیری عملکرد شبکه مجازی

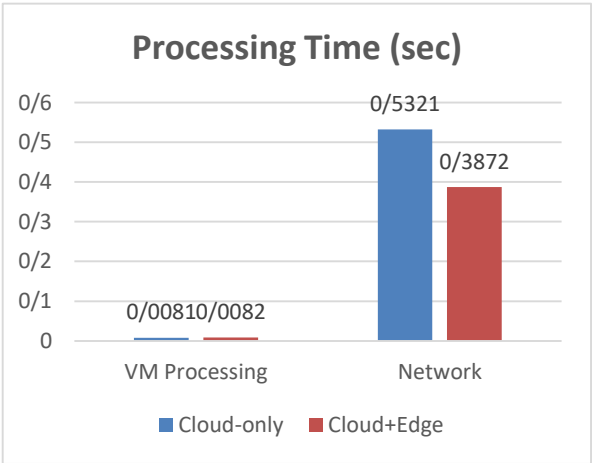
۶،۳. نتایج ارزیابی

نمودار ۱ میانگین زمان پاسخ، CPU و زمان پردازش شبکه و نسبت بارهای پردازش‌شده در هر مرکز داده را که در شبیه‌سازی اندازه‌گیری شده است، نشان‌می‌دهد. زمان متوسط پاسخ برنامه از زمانی که کاربران نهایی از طریق VNF به سرور برنامه درخواست ارسال‌می‌کنند تا رسیدن پاسخ به کاربران نهایی اندازه‌گیری‌می‌شود.

درخواست و پاسخ هر دو برای ارسال از طریق VNF اعمال‌می‌شوند. طبق شکل ۲ A ، با کاهش تأخیر شبکه با قراردادن VNF ها در منابع لبه نزدیک به کاربران نهایی، میانگین زمان پاسخگویی در سیاست Cloud + Edge 26.6٪ کاهش‌می‌یابد و از ۱،۰۸۸۵ ثانیه با سیاست فقط Cloud به ۰،۷۹۹۰ ثانیه می‌رسد.

زمان پردازش VM در سرورهای برنامه در هر دو سیاست به طور مشابه در تقریبی ۰،۰۰۸ ثانیه اندازه‌گیری‌می‌شود، اما زمان انتقال شبکه بین دو خط‌مشی تفاوت معنی‌داری دارد (شکل B۱۰ را ببینید).

این نتیجه نشان‌می‌دهد که کاهش زمان انتقال شبکه در سیاست Cloud + Edge دلیل بهبود زمان پاسخ است. آخرین شکل (۱۰ درجه سانتیگراد) نسبت بارهای VNF پردازش‌شده در مراکز داده لبه و ابر را نشان‌می‌دهد.



در خط مشی فقط Cloud، بدیهی است که همه بسته‌ها فقط در مرکز داده‌های ابری شبکه پردازش می‌شوند، در حالی که حدود ۴۵٪ بسته‌ها در مرکز داده‌های لبه با سیاست Cloud + Edge پردازش می‌شوند.

بقیه (۵۵٪ از بارهای کار) هنوز در ابر شبکه پردازش می‌شوند، به دلیل کمبود منابع در مراکز داده لبه برای VNF های تازه تکرارشده توسط سیاست مقیاس‌گذاری خودکار.

۴. مسیر آینده و تمدید

CloudSimSDN-NFV به عنوان یک چارچوب شبیه‌سازی رویداد محور، از ارزیابی هر دو شبکه و مدیریت منابع محاسباتی در زمینه محاسبات لبه‌ای و ابری پشتیبانی‌می‌کند. برای پردازش شبکه، ما در حال حاضر صحت را در مقایسه با Mininet ارزیابی‌کرده‌ایم.

با این وجود، وقتی نوبت به سناریوهایی می‌رسد که به ارزیابی و پیش بینی دقیق بالا مانند پارامترهای مربوط به زیرساخت سطح پایین نیاز دارند، به پروفایل و مدل‌سازی بستگی دارد.

با پروفایل داده‌های واقعی و شناسایی پارامترهای به‌دست‌آمده، می‌توان مدل‌های ریاضی را با دقت بالاتر به چارچوب شبیه‌سازی ترجمه-کرد.

در این بخش، ما در مورد افکار خود در مورد چگونگی گسترش گسترده CloudSimSDN-NFV برای پشتیبانی از (۱) قابلیت‌های OpenFlow مانند، (۲) ارزیابی مهاجرت زنده و (۳) مدل‌سازی انرژی بحث‌می‌کنیم.

قابلیت‌های OpenFlow مانندCloudSimSDN-NFV فعلی چندین قابلیت SDN مانند مسیریابی بسته فعال پویا، هدایت بسته به مجموعه پورت‌ها و تغییر مقصد بسته یا انداختن بسته را امکان‌پذیر می‌کند.

از آنجا که OpenFlow یک پروتکل SDN واقعی است، می‌تواند از طریق جریان‌های فعال و واکنشی مبتنی بر بسته‌بندی و بسته‌بندی بین کنترل‌کننده و سوئیچ، عملکردها را تسهیل‌کند. برای شبیه‌سازی OpenFlow packet-in و packet-out، می‌توانیم کلاس فعلی بسته را گسترش‌دهیم و برچسب بسته را برای پروتکل OpenFlow مشخص-کنیم.

علاوه بر این، بر اساس اصول شبیه‌سازی رویداد محور، چندین رویداد با توجه به الگوریتم‌ها و سیاست‌های تعریف‌شده توسط کاربر می‌توانند در کلاس SDNHost، SDNDataCenter و NetworkingOperationSystem ایجادشوند.

از این ویژگی‌ها می‌توان در ارزیابی الگوریتم‌های جدید برای قراردادن کنترل‌کننده‌های توزیع‌شده SDN استفاده‌کرد. هدف این الگوریتم‌ها به حداقل رساندن تأخیرهای ارتباطی پیام‌های بسته درون بسته / بسته بین سوئیچ‌های فعال‌شده SDN و کنترل‌کننده‌های SDN است.

۱،۴. مدل‌سازی جامع مهاجرت زنده

به طور کلی، کارهای موجود از مهاجرت زنده به عنوان رویکردی برای تخصیص پویا ماشین‌های مجازی برای تحقق اهدافی مانند تأخیر بهینه پایان به پایان و مصرف انرژی استفاده می‌کنند.

با این حال، مهاجرت زنده می‌تواند با استفاده از منابع محاسباتی و شبکه به طور قابل توجهی بر QoS و SLA تأثیر بگذارد. چارچوب فعلی از مکانیزم مهاجرت زنده پشتیبانی‌می‌کند که اجازه‌می‌دهد نمونه (VNF / VM /کانتینر) در میزبان منبع متوقف‌شود و با بررسی منابع موجود در میزبان مقصد از سر گرفته‌شود.

با این وجود، مدل تقلیدی از کپی حافظه کثیف تکراری از طریق انتقال شبکه در هنگام انتقال را در نظر نمی‌گیرد. این فقط پردازش حجم کار را در CloudletScheduler متوقف‌می‌کند و پس از پایان انتقال، بسته را به مقصد جدید هدایت‌می‌کند. برای یک مدل مهاجرت زنده جامع، می‌توانیم آن را براساس مراحل مختلف و مدل پیش‌بینی گسترش‌دهیم.

یعنی می‌توانیم رفتارهای مهاجرت را در توابع پردازش رویداد SDNDataCenter و NetworkingOperationSystem با توجه به مراحل مهاجرت زنده تعریف‌کنیم: مرحله اول، مرحله هماهنگ‌سازی صفحه، توقف و کپی، و مراحل پس از مهاجرت.

به عنوان مثال، با پشتیبانی مدیر کانال فعلی و برنامه‌ریز Cloudlet، انتقال بسته و پردازش حجم کار را در زمان انتقال از زمان توقف انتقال زنده متوقف‌می‌کند.

بنابراین، هزینه‌های اضافی و عملکرد مهاجرت زنده مانند زمان مهاجرت زنده، زمان خرابی و میزان داده‌های منتقل‌شده قابل‌ارزیابی است. علاوه بر مدل‌سازی مهاجرت زنده، می‌توان جریان مهاجرت و برنامه‌ریزی مهاجرت چندگانه را با توپولوژی شبکه جهانی برنامه‌ریزی کرد.

۲،۴. مدل‌سازی انرژی

محققانی که بر مقایسه عملکرد بین مدل‌های انرژی فعلی و مدل پیشرفته NFV متمرکز هستند، می‌توانند مستقیماً از CloudSimSDN-NFV استفاده‌کنند تا نتایج نسبتاً دقیقی ایجادکنند زیرا NFV به عنوان نرم‌افزار هنوز از همان منابع شبکه (سوئیچ‌ها) استفاده‌می‌کند و منابع محاسباتی (میزبان‌های فیزیکی در مراکز داده ابری).

مدل‌سازی توان مرکز داده مجازی را می‌توان در مدل‌سازی برآورد قدرت تحلیلی و خصوصیات مبتنی بر اندازه‌گیری تجربی دسته‌بندی کرد.

علاوه بر این، ما باید پارامترهای دیگری را برای مدل‌سازی جامع و یکپارچه انرژی در نظر بگیریم، مانند (۱) مدل مصرف برق برای سایر دستگاه‌ها، به عنوان مثال، ذخیره‌سازی و حافظه، و (۲) مدل جریان دما بین اجزای مختلف و سرورها.

بر اساس توانایی نظارت و انعطاف‌پذیری گسترش در حال حاضر موجود، بسیاری از محققان راه‌حل‌های جدیدی را برای مدل‌سازی انرژی ارائه‌داده‌اند.

لوئیس و همکاران CloudSim را برای شبیه‌سازی مصرف برق ذخیره‌شده در مراکز داده گسترش‌دادند.

ایلاگر و همکاران با در نظر گرفتن دمای زمان اجرا، اجزای مدل انرژی سرور را در CloudSim گسترش دادند.

آن‌ها نقاط گرم تولیدشده توسط میزبان‌های فیزیکی در حال اجرا را مدل‌سازی‌می‌کنند، که تأثیر قابل‌توجهی در هزینه انرژی کل خنک کننده و محلول برق ابر و محاسبات لبه دارند.

علاوه بر این، محققان همچنین می‌توانند مدل‌های دیگر مصرف برق برای VM / VNF / ظرف را به عنوان مدل ریاضی در محیط‌های مجازی و سوئیچ‌های OpenFlow برای شبکه SDN پیاده‌سازی‌کنند.

به طور خلاصه، جهت‌های آینده برای مدل‌سازی انرژی NFV و سایر ویژگی‌های مجازی در رایانش ابری و لبه‌ای باید با توصیف رفتار نرم افزار سطح پایین، مدل‌های انرژی حافظه، دیسک و خنک کننده را ادغامکند.

۵. خلاصه و نتیجه گیری

پیشرفت فناوری مجازی‌سازی و انفجار ظرفیت محاسبات منجر به چندین تغییر الگوی نوآورانه در صنعت محاسبات و شبکه از جمله ظهور رایانش ابری، NFV، SDN و محاسبات لبه‌ای می‌شود.

رایانش ابری و SDN به مدت یک دهه در جامعه تحقیقاتی مورد مطالعه قرارگرفته و به طور گسترده‌ای در صنعت پذیرفته‌شده‌است، در حالی که NFV و رایانه لبه‌ای هنوز در مرحله اولیه تحقیق و اجرا در دنیای واقعی هستند.

اگرچه بسیاری از کارهای پیشرفته در ادبیات ارائه شده‌است، اما چارچوب ارزیابی ساده و سریع می‌تواند پیشرفت را به صورت تصاعدی تقویت‌کند.

در این جا، ما CloudSimSDN-NFV، یک چارچوب برای جریان‌های ویدئویی شبیه‌سازی جدید برای ارزیابی ویژگی‌های NFV در محاسبات لبه‌ای و ابری همراه با سایر ویژگی‌های SDN و محیط‌های رایانش ابری، پیشنهاد می‌دهیم. این چارچوب بر روی CloudSimSDN طراحی و توسعه یافته است که بر روی جعبه ابزار CloudSim به خوبی مطالعه شده است. ما مدل‌سازی و شبیه‌سازی NFV و محاسبات لبه‌ای و طراحی دقیق و اجرای چارچوب خود را توصیف کردیم. دو سناریو مورد استفاده برای کمک به درک نحوه استفاده از ابزار جدید ارائه‌شده‌است و چندین الگوریتم بر اساس چارچوب اجرا و ارزیابی‌می‌شوند.

نتایج نشان‌می‌دهد که چارچوب ما می‌تواند برای ارزیابی سریع روش-های مختلف در شبیه‌سازی به طور کارآمد مورد بهره‌برداری قرار گیرد.برای بهبود چارچوب پیشنهادی، می‌توان سیاست‌های داخلی بیشتری را برای قراردادن VNF و ترکیب SFC اضافه و اجراکرد. به عنوان مثال، یک سیاست VNF آگاه از توپولوژی شبکه می‌تواند در شبیه‌سازی پیاده‌سازی و آزمایش‌شود، همچنین یک سیاست ترکیب SFC که با مراکز داده شبکه چند کار می‌کند.

تحت چارچوب شبیه‌سازی پیشنهادی، ما در مورد قابلیت‌های OpenFlow مانند، مدل‌سازی جامع مهاجرت زنده و مدل‌سازی انرژی، برنامه‌های افزودنی و مسیرهای بالقوه آینده را شرح خواهیم داد. بنابراین، ما انتظارداریم که چارچوب شبیه‌سازی ما بتواند محققان را در انجام تحقیقات پیشرفته در NFV، edge و رایانش ابری برای پرورش نوآوری‌های جدید توانمند سازد. برای بهبود چارچوب پیشنهادی، می‌توان سیاست‌های داخلی بیشتری را برای قرار دادن VNF و ترکیب SFC اضافه و اجرا کرد.

به عنوان مثال، یک سیاست VNF آگاه از توپولوژی شبکه می‌تواند در شبیه‌سازی پیاده‌سازی و آزمایش شود، همچنین یک سیاست ترکیب SFC که با مراکز داده شبکه چند کار می‌کند. تحت چارچوب شبیه‌سازی پیشنهادی، ما در مورد قابلیت‌های OpenFlow مانند، مدل‌سازی جامع مهاجرت زنده و مدل‌سازی انرژی، برنامه‌های افزودنی و مسیرهای بالقوه آینده را شرح‌خواهیم‌داد. بنابراین، ما انتظار داریم که چارچوب شبیه‌سازی ما بتواند محققان را در انجام تحقیقات پیشرفته در NFV، edge و رایانش ابری برای پرورش نوآوری‌های جدید توانمند سازد.

مراجع

[1]Baldoni, G., Melita, M., Micalizzi, S., Ramett, C., Schembra, G., & Vassallo, A. (2016). Video broadcasting services over SDN-NFV enabled networks: a prototype. The 3rd International Symposium on Emerging Information, Communication and Networks, 560-565.

[2]Benzekki, K., El Fergougui, A., & Elbelrhiti Elalaoui, A. (2016). Software-defined networking (SDN): A survey". . Security and Communication Networks, 5803-5833.

[3]Costa, P., Migliavacca, M., A, P. P., & L. Wolf. (2012). NaaS: Network-as-a-Service in the Cloud. in Proc. of Hot-ICE'12, 120-145.

[4]D'Amico, V., Lombardo, A., Melita, M., Rametta, C., & G. Schembra. (2016). An SDN/NFV Telco Operator Platform for Video Broadcasting,. in IEEE Communications Magazine, 35-38.

[5]D'Amico, V., Lombardo, A., Melita, M., Rametta, C., & Schembra, G. (2016). An SDN/NFV Telco Operator Platform for Video Broadcasting. IEEE Communications Magazine , 122-128.

[6]Faraci, G., & Schembra, G. (2015). An analytical model to design and manage a green SDN/NFV CPE node. IEEE Transactions on Network and Service Management, 435-450.

[7]Faraci, G., Lombardo, A., & Schembra, G. (2016). A Processor Sharing Scheduling Strategy for NFV Nodes. Journal of Electrical and Computer Engineering, 1-10.

[8]Koumaras, H., Kourtis, M. A., Sakkas, C., Xilouris, G., & S. Kolometsos. (2016). In-service Video Quality assessment based on SDN/NFV techniques. in Proc. of IEEE ICT2016, 16-18.

[9]Koumaras, H., Kourtis, M.-A., Sakkas, C., Xilouris, G., Kolometsos, S., & Demokritos, N. (2016). In-service Video Quality Assessment Based on SDN/NFV Techniques. 23rd International Conference on Telecommunications (ICT) (pp. 126-131). Greece: Institute of Informatics and Telecommunications.

[10]Koumaras, H., Sakkas, C., Kourtis, M. A., Xilouris, C., & Koumaras, V. (2016). Enabling Agile Video Transcoding over SDN/NFV-enabled Networks. International Conference on Telecommunications and Multimedia (TEMU) (pp. 136-141). Greece: Institute of Informatics and Telecommunications NCSR "Demokritos.

[11]Kreutz, D., Ramos, F., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., & Uhlig, S. (2015). Software-Defined Networking: A Comprehensive Survey. in Proceedings of the IEEE, 14-76.

[12]Lombardo, A., Manzalini, A., Schembra, G., Faraci, G., Rametta, C., & V. Riccobene. (2015). An open framework to enable NetFATE (NetworkFunctions at the edge). 1st IEEE Conference on Network Softwarization, (pp. 13-17).

[13]Lombardo, A., Riccobene, A. M., & Schembra, G. (2014). An Open Architecture for Software Defined Services at the Edge. EuCNC2014, 23-26.

[14]Mijumbi, R., Serrat, J., Gorricho, J.-L., Bouten, N., Turck, F. D., & Boutaba, R. (2015). Network Function Virtualization: State-of-the-art and Research Challenges. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 175-205.

[15]P, Q., & T, N. (2014). Service Function Chaining Problem Statement 10. Active InternetDraft, IETF Secretaria, 120-140.

[16]Rückert, J., Blendin, J., & David Hausheer. (2015). Software-Defined Multicast for Over-the-Top and Overlay-based Live Streaming in ISP Networks. Journal of Network and Systems Management, 280-308.

[16] Jungmin Son,TianZhang He, Rajkumar Buyya. (2019). CloudSimSDN-NFV: Modeling and simulation of network function virtualization and service function chaining in edge computing environments . 2019 John Wiley & Sons, Ltd, 1748-1764S

[17] Toosi AN, Son J, Chi Q, Buyya R. ElasticSFC: auto-scaling techniques for elastic service function chaining in networkfunctions virtualization-based clouds. *J Syst Softw*. 2019;152:108-119.

پی‌نوشت

1. Over the Top
2. virtual machine monitor, VMM, virtualizer
3. Intrusion Detection System
4. Management and orchestration
5. network functions virtualization management and network orchestration
6. virtual network functions manager
7. Virtualized Infrastructure Manager