

بارگذاری و اجرای بهینه و پویای برنامه‌های تلفن همراه در لبه شبکه

انتصار حسینی

دانشکده فنی و مهندسی کامپیوتر و فناوری اطلاعات - دانشگاه قم - قم - ایران
پست الکترونیکی: e.Hoesini@stu.qom.ac.ir

محسن نیک رأی*

دانشکده فنی و مهندسی کامپیوتر و فناوری اطلاعات - دانشگاه قم - قم - ایران
پست الکترونیکی: m.nickray@qom.ac.ir

شمس الله قنبری

استادیار دانشکده فنی و مهندسی کامپیوتر - دانشگاه آزاد اسلامی آشتیان - آشتیان - ایران
پست الکترونیکی: myrshg@gmail.com

چکیده

زیاد و حجم داده و پردازش بزرگ، بهتر عمل می‌کنند. در این سیستم از ۵۲۱ کار موجود برای اجرا، تنها ۷۵ کار در دستگاه تلفن همراه به صورت محلی اجرا می‌شوند که این کارها اندازه داده و پردازش و انرژی مصرفی کمتری دارند. بقیه کارها با اندازه داده و پردازش بیشتر برای اجرا به لایه MEC ارسال می‌شوند.

واژه‌های کلیدی: پردازش لبه با دسترسی چندگانه، پردازش لبه سیار، زمان بندی، بهینه سازی لیاپانوف

پردازش لبه با دسترسی چندگانه^۱ و پردازش لبه سیار (MEC)^۲ به کاربران تلفن همراه اجازه می‌دهد تا کارهای پردازشی خود را به لبه شبکه منتقل کنند. MEC قابلیت پردازش لبه و برنامه را در لبه شبکه فراهم می‌کند. در این مقاله، یک مدل بارگذاری پویا براساس بهینه سازی لیاپانوف پیشنهاد شده است که پایداری سیستم را بر اساس وضعیت فعلی سیستم حفظ می‌کند. ما از دو روش زمان بندی در دو ماشین مجازی استفاده می‌کنیم که در روش اول کارها با زمان اجرای بیشتر اولویت بالاتر (HTHP)^۳ و در روش دوم کارها با زمان اجرای کمتر اولویت بالاتر (LTHP)^۴ خواهند داشت. نتایج کار با دو روش به ترتیب ورود و با گردش نوبت مقایسه می‌شوند. مقادیر نشان می‌دهد که صرفه جویی در مصرف انرژی در HTHP و LTHP به ترتیب ۳۸ و ۵۵/۲ درصد است. زمان پاسخ به کارهای ارسالی به ترتیب ۱۹۹/۷۶ و ۱۸۲/۹۶ میلی ثانیه در هر دو روش HTHP و LTHP است که برای کارها با تعداد

۱- مقدمه

امروزه دستگاه‌های تلفن همراه نقش بسزایی در زندگی مردم ایفا می‌کنند. با توسعه شبکه تلفن همراه و اینترنت اشیاء (IoT)^۵، محاسبات، ذخیره سازی و مدیریت شبکه با چالش‌های بسیاری مواجه شده‌اند. این چالش‌ها شامل تاخیرهای زیاد، محدودیت فضای ذخیره سازی، دستگاه‌ها با منابع محدود، لزوم سرویس دهی متناوب بدون وقفه و امنیت بیشتر هستند که با معماری مبتنی بر ابر به طور کامل حل نمی‌شوند [۱ و ۲]. دستگاه‌های سیار مانند تلفن‌های هوشمند، تبلت و دستگاه‌های پوشیدنی فعال در اینترنت

* نویسنده مسئول

1- Multi-access Edge Computing
2- Mobile Edge computing
3- Higher execution Times takes have
4- Lower execution Times takes have the Higher Priority

اشیاء هستند که در آن بسیاری از حسگرها با سایر دستگاه‌ها در منازل، اداره‌ها و جاده‌ها با شبکه‌های پیشرفته امروزی متصل می‌شوند. نسل جدید شبکه سیستم‌های تلفن همراه به نام نسل پنجم (نسل ۵) با هدف تسریع در توسعه برنامه‌های کاربردی دستگاه‌های سیار به وجود آمدند. شبکه‌های نسل پنجم بر استفاده از فناوری‌های نوین از جمله محاسبات ابری سیار و محاسبات لبه سیار برای دستیابی به انعطاف‌پذیری بیشتر و قابلیت پردازش بی‌درنگ تاکید دارند [۳و ۴]. برخی از برنامه‌های کاربردی دستگاه‌های سیار، مانند واقعیت افزوده، بازی و تشخیص چهره نیاز به توان محاسباتی بالایی دارند به این دلیل در هنگام پردازش انرژی زیادی را مصرف می‌کنند [۵]. از طرفی تقسیم منابع در پردازش ابری به دلیل دسترسی دور کاربر باعث تاخیر در زمان پاسخ می‌شود. به این دلیل نیاز به وجود یک فناوری جدید برای سرعت بخشیدن و افزایش کارایی در برنامه‌های حساس به تاخیر و بی‌درنگ احساس می‌شود.

از سال ۲۰۱۶، موسسه استانداردهای ارتباطات اروپا (ETSI)^۶ یک چارچوب و معماری مرجع MEC تولید کرد که عناصر کاربردی آن را پشتیبانی از خدماتی مانند اجرای برنامه و اطلاعات شبکه رادیویی معرفی کرد [۶]. محاسبات لبه در افزایش سرعت زمان پاسخ، رفع محدودیت عمر باتری، صرفه‌جویی در هزینه پهنای باند، تأمین ایمنی داده‌ها و حفظ حریم خصوصی کاربرد نقش دارد که در نزدیکی کاربر این خدمات را ارائه می‌دهد [۷]. در سال ۲۰۱۷ این موسسه نام این معماری را از پردازش لبه سیار به پردازش لبه با دسترسی چندگانه تغییر داد [۶]. با بارگذاری و اجرای کارها^۸ در لبه شبکه، MEC می‌تواند بین توانایی محدود دستگاه‌های کاربر و تقاضای کاربر پیوندی را ایجاد کند. در این مقاله، ما از لایه MEC میان ابر و برنامه‌های کاربردی دستگاه‌های سیار استفاده می‌کنیم. در ضمن، ما کارهای تلفن همراه را به لایه MEC ارسال

می‌کنیم و آن‌ها را در آن لایه پردازش می‌کنیم. کارها بر اساس صف اولویت اجرا می‌شوند. در این مقاله، یک مدل بارگذاری از طریق MEC ارائه می‌شود که دو شاخص کلیدی توان محاسباتی و انرژی مصرفی مورد بحث را بهینه می‌کند. تجزیه و تحلیل پایداری سیستم و پویایی عملکرد با استفاده از الگوریتم بهینه‌سازی لیاپانوف انجام می‌شود. ما همچنین یک تابع هزینه را بر اساس پارامترهای انرژی مصرفی و سرعت زمان پاسخ تعریف می‌کنیم که بر اساس مقادیر آن‌ها تلفن همراه تصمیم می‌گیرد کارها را به MEC ارسال کند و یا به صورت محلی درون خود دستگاه اجرا کند. این تصمیم‌گیری در کاهش میزان انرژی مصرفی و افزایش سرعت اجرای کارها تاثیر دارد. اجرای برنامه در تعدادی ماشین مجازی (VM)^۹ در لایه MEC انجام می‌شود که در هر شکاف زمانی منابعی را با کارایی بالا برای برنامه‌های بی‌درنگ ارائه می‌دهد. در نهایت دو روش پیشنهادی را با دو روش مرسوم زمان‌بندی به ترتیب ورود (FIFO)^{۱۰} و با گردش نوبت (RR)^{۱۱} مقایسه می‌کنیم. در این کار دو پارامتر انرژی مصرفی و زمان پاسخ در هر چهار روش مقایسه می‌شوند. در کار ارائه شده، دستگاه از طریق شبکه ۴/۵ نسل به لایه MEC دسترسی دارد. در نهایت، ما یک سیاست بارگذاری پویا بر اساس دو روش مبتنی بر اولویت با حفظ پایداری سیستم با استفاده از الگوریتم لیاپانوف ارائه دادیم که پارامترهای انرژی مصرفی و زمان پاسخ را نسبت به اجرای محلی بهبود می‌بخشد.

سایر قسمت‌های مقاله به شرح زیر می‌باشد. بخش ۲ به بررسی کارهای موجود در زمینه‌های مشابه این موضوع می‌پردازد. بخش ۳ به ارائه مدل پیشنهادی و فرمول‌ها می‌پردازد. بخش ۴ مدل بارگذاری پویا مورد نظر را توصیف می‌کند. بخش ۵ نتایج و تجزیه و تحلیل عددی را نشان می‌دهد. بخش ۶ به جمع‌بندی مطالب و نتایج این مقاله می‌پردازد.

9- Virtual Machine
10- First Info First Output
11- Round Robin

6- Fifth Generation
7- European Telecommunications Standards Institute
8- Task

در این بخش کارهای انجام شده پیشین در زمینه فناوری پردازش ابری سیار (MCC)^{۱۲} و پردازش لبه سیار را بررسی می‌کنیم. پردازش ابری اولین بار در سال ۲۰۱۲ توسط شرکت سیسکو ارائه شد که بستری برای محاسبات، ذخیره‌سازی و خدمات شبکه میان دستگاه‌های نهایی و کارسازهای ابر سنتی فراهم کرد [۸]. سانگ چین و همکاران یک معماری تعریف کردند که زمینه‌ساز پشتیبانی از تنوع فناوری‌های لبه، Wi-Fi، Bluetooth، ZigBee، LTE و همچنین مجازی‌سازی شبکه و مهندسی ترافیک از طریق مجازی‌سازی عملکرد شبکه و سازوکارهای شبکه بود [۹]. مطالعات دیگری در زمینه چگونگی سازگاری پردازش لبه با قالب‌های اینترنت اشیا از جمله OpenM2M^{۱۳} انجام شد [۱۰]. در این کار، گره‌ها به عنوان واحدهای کنار جاده‌ای (RSU)^{۱۴} در شبکه و سایل نقلیه، برای خدمات مختلف ماشین به ماشین و سیستم‌های مدیریت دستگاه‌های ماشین به ماشین توسعه داده شدند. در یک تجزیه و تحلیل کمی از مصرف انرژی که در آن ۲۵ درصد از برنامه‌های کاربردی اینترنت اشیا نیاز به خدمات زمان واقعی و تاخیر کم داشتند، نشان داده شد که میانگین هزینه انرژی در محاسبات لبه با ۴۸/۴۰ درصد کمتر از مدل رایانش ابری معمولی است [۱۱]. در یک کار توان پردازشی شبکه‌هایی از حسگرهای بی‌سیم جهت کاربردهای نظامی، با ادغام محاسبات ابری و لبه ارائه دادند و بر محدودیت‌ها و آسیب‌های موجود در برنامه‌ها با اجرای زمان واقعی غلبه کردند [۱۲]. یک نمونه کار دیگر، به ارائه مجموعه‌ای از رابط‌های برنامه کاربردی^{۱۵} پرداخته است که اجازه می‌دهد ماشین‌های مجازی برای ارائه خدمات به لبه شبکه دسترسی داشته باشند. با استفاده از APIها، ماشین‌های مجازی می‌توانند به اطلاعات محلی مانند آمار شبکه و داده حسگرها دسترسی پیدا کنند [۱۳].

12- Mobile Cloud Computing

13- Machine to Machine

14- Road Side Units

15- Application Programming Interface (API)

تعدادی از محققان در پروژه‌های تصویر کاملی از تمام جزئیات برای چندین دستگاه هوشمند در محیط ابر ایجاد کردند و کارهای قابل انتقال از دستگاه به سمت تصویر ایجاد شده از دستگاه‌ها منتقل شدند. بنابراین در صورتی که دستگاه خراب یا گم شود نسخه مجازی آن وجود دارد. مزیت دیگر این طرح مقابله با محدودیت سخت‌افزاری دستگاه‌های هوشمند است [۱۴]. در پروژه‌های دیگر امکان ادغام بستر نسل ۵ را با MEC اثبات کردند و نشان دادند که محدوده‌های داخلی و خارجی این شبکه که بدون استفاده بودند، قابل دسترسی هستند [۱۵]. کار دیگری در زمینه تحلیل داده‌های بزرگ انجام شد که از ادغام MCC و MEC استفاده کرد که با توجه به حجم داده‌های پردازشی این طرح محدودیت ترافیک و تاخیر را برای کار آن‌ها برطرف نکرد [۱۶]. نویسندگان در [۱۷] یک مقاله برای پردازش‌ها با کارایی بالا و انرژی مصرفی زیاد ارائه دادند که از الگوریتم زمان‌بندی استفاده می‌کرد و باعث کاهش مصرف انرژی شد. در کارهای [۱۸] و [۱۹] از الگوریتم لیپانوف برای صرفه‌جویی در انرژی دستگاه‌ها در محیط ابری استفاده شد. نویسندگان در [۲۰] یک منبع رادیویی و محاسباتی برای پایانه‌های تلفن همراه با انرژی مصرفی محدود در یک شبکه femtocell ارائه دادند. در کار [۲۱] یک سیاست بارگذاری پیشنهاد شد که به صورت ایستا مصرف انرژی و زمان پاسخ را بهبود می‌بخشد. به منظور ایجاد یک شبکه محاسباتی سیار، نویسندگان در [۲۲] چارچوب تخصیص منابع را برای سازماندهی دستگاه‌های ناهمگن در مجاورت یکدیگر پیشنهاد دادند. در [۲۳] یک طرح بارگذاری پویا بر اساس زنجیره مارکوف^{۱۶} برای کاربران تلفن همراه در یک سیستم ابری کوچک ارائه شد. در [۲۴] از ترکیب الگوریتم لیپانوف و کوله‌پشتی استفاده شد و یک سیاست بارگذاری پویا با استفاده از الگوریتم‌های صف به صورت شبیه‌سازی شده در متلب ارائه شد. در [۲۵] مؤلفان از سه روش زمان‌بندی خروج به ترتیب ورود، همزمانی و اولویت بر حسب تأخیر زمانی در محیط مه پرداختند. در روش

16- Markov Chain

همزمانی، برنامه‌های ورودی بدون توجه به حجم مصرف منابع، اجرا می‌شوند. در روش زمان‌بندی به ترتیب ورود، برنامه‌ها به ترتیب ورود، اجرا می‌شوند و چنانچه کار پایان نپذیرد در صف زمان‌بندی قرار می‌گیرد. در روش سوم، برنامه‌ها بر اساس تأخیر زمانی زمان‌بندی می‌شوند. در این پژوهش از کتابخانه ifogsim برای شبیه‌سازی استفاده شد. نتایج نشان‌دهنده تأخیر زمانی بیشتر در روش همزمانی نسبت به روش‌های خروج به ترتیب ورود و اولویت‌بندی بر اساس تأخیر زمانی است.

در بسیاری از کارها روش‌های مختلفی برای بهبود مصرف انرژی و زمان پاسخگویی برای پردازش ابری و لبه سیار ارائه شده است. اما تصمیم‌گیری پویا با حفظ ثبات سیستم در بسیاری از کارها در نظر گرفته نشده است. ما از بسیاری از ایده‌ها در کارهای ذکر شده استفاده کردیم و روشی بهینه و پویا بر اساس الگوریتم لیاپانوف و روش زمان‌بندی مبتنی بر اولویت برای بارگذاری کارها از تلفن‌همراه به لایه MEC ارائه کردیم. در این کار هر دو پارامتر انرژی مصرفی و زمان پاسخ مقدار بهینه‌تری در لایه MEC را نسبت به اجرای محلی کارها در دستگاه دارند.

۳- تعریف مسئله و فرمول‌بندی

این بخش نحوه ارسال و اجرای کارها در لایه MEC را نشان می‌دهد. برای فهم آسان فرمول‌ها، برخی از پارامترهای اساسی در جدول ۱ ذکر شده‌اند. ما دو VM در لایه MEC در نظر گرفتیم که به کارها سرویس می‌دهند. کارهای محاسباتی می‌توانند به صورت محلی در دستگاه اجرا شوند یا در لایه MEC بارگذاری شوند. با بارگذاری کارهای پردازشی برای اجرا در لبه شبکه، سرعت فرآیند پردازش می‌تواند به طور قابل توجهی بهبود یابد. کانال زمانی در این سیستم به چند شکاف زمانی ثابت تقسیم شده و به هر کار یک شکاف زمانی تعلق می‌گیرد. اگر یک کار در یک شکاف زمانی پایان نپذیرد، به انتهای صف ارسال و در زمان بعدی اجرا می‌شود. ما از شبکه نسل

۴/۵ در جریان ارسال و دریافت استفاده کردیم زیرا شبکه نسل ۵ واقعی در محدوده آزمایش ما در دسترس نیست. به منظور بهبود کارایی برنامه‌ها و کاهش مصرف انرژی، یک برنامه کنترل بر روی دستگاه تلفن همراه نیاز است که تعیین کند کدام کار را به لایه MEC ارسال کند یا به صورت محلی در خود دستگاه اجرا شود. لایه MEC ما دارای m ماشین مجازی است که موازی با هم کار می‌کنند. برای زمان‌بندی بهینه، سیستم از زمان‌بندی مبتنی بر صف اولویت استفاده می‌کند. در هر شکاف زمانی برنامه کنترل، بهترین کار را برای انتقال انتخاب می‌کند. برنامه با حجم بالا پردازش و اندازه داده بزرگ بهترین انتخاب برای بارگذاری است. دو پارامتر $d(t)$ و $s(t)$ به ترتیب اندازه داده ورودی برنامه و اندازه پردازش را توصیف می‌کنند. هر دو بردار به صورت مستقل و یکسان در هر شکاف زمانی t مقدار می‌گیرند. زمان اجرای هر کار پارامتر $T(t)$ است که از رابطه زیر به دست می‌آید:

$$T_{MEC}^n(t) = \frac{R}{S} * (T_r + T_h + T_{tr}) \quad (1)$$

$$T(t) = \sum_{n \in \theta} r(t) T_{MEC}^n(t) + \left[1 - \sum_{n \in \theta} r(t) T_{local}^n(t) \right] \quad (2)$$

درحالی‌که که مقدار R نرخ درخواست مورد انتظار است، $T_h(t) = \frac{s(t)}{C_{vm}(t)}$ زمان سرویس‌دهی در سمت ماشین مجازی است و T_r زمان پاسخ به اضافه زمان ارسال درخواست از VM است، $T_{tr} = \frac{d(t)}{C_{vm}(t)}$ زمان انتقال تخمین زده شده برای بارگذاری کار به VM_n در شکاف زمانی t است و تعداد درخواست‌های همزمانی است که یک VM می‌تواند به آن‌ها سرویس دهد. $T_{local} = \frac{s(t)}{C_{local}(t)}$ زمان اجرا به صورت محلی دستگاه تلفن همراه می‌باشد [۱۹]. فرض می‌کنیم $r(t) \in \{0,1\}$ پارامتری است که اگر $r(t) = 0$ در دستگاه تلفن همراه انجام می‌شوند و اگر $r(t) = 1$ کار در MEC اجرا شده است. انرژی مصرفی از عبارت $E_{vm}(t) + E_{tr}(t)$ محاسبه می‌شود که به ترتیب انرژی مصرفی در زمان انتقال به MEC و انرژی مصرفی

جدول ۱: جدول پارامترها

Factor	تعریف
$d(t)$	اندازه داده‌های ورودی پردازشی برنامه (به عنوان مثال، برنامه و داده‌ها)
$s(t)$	اندازه پردازش‌ها (به عنوان مثال، تعداد چرخه پردازنده)
C_{local}	سرعت پردازشگرهای تلفن همراه (پردازنده‌های در حال اجرا در ثانیه)
e_{idle}	انرژی مصرف شده دستگاه تلفن همراه
e_{tran}	انرژی مصرف شده برای انتقال
e_{active}	انرژی در دسترس دستگاه تلفن همراه
$C_{vm}(t)$	توانایی محاسبه توسط امین ماشین مجازی
t	شکاف زمانی
T	زمان اجرای کار
E	انرژی مصرفی کار
μ	نرخ سرویس‌دهی
λ	نرخ ورود به صف
ϕ	تابع هزینه
pro	اولویت کار در صف
ϑ	طول صف
β	پارامتر هزینه
B	پارامتر وزن در لیاپانوف
V	مقدار ثابت در لیاپانوف
L	فاکتور لیاپانوف

در زمان اجرا در VM را نشان می‌دهند. انرژی مصرفی از عبارت زیر محاسبه می‌شود:

$$E(t) = \sum_{n \in \emptyset} r(t) E_{MEC}^n(t) + \left[1 - \sum_{n \in \emptyset} r(t) E_{local}^n(t) \right] \quad (3)$$

جایی که $E_{local}^n(t) = T_{local} * e_{active}$ انرژی مصرفی در دستگاه تلفن همراه است. دو عبارت ذکر شده در بالا نیز به صورت $E_{tr}(t) = T_{tran} e_{tran}$ و $E_{vm}(t) = C_{vm}(t) * \mu$ محاسبه می‌شوند. پارامتر β به عنوان پارامتر هزینه در

محدوده تعریف می‌شود. اگر $\beta=1$ باشد، مصرف انرژی در تصمیم‌گیری برای بارگذاری به MEC تاثیر ندارد. تابع هزینه برای مصرف انرژی و زمان اجرا به شرح زیر است:

$$\phi(t) = \left(\beta \frac{T(t)}{T_{local}(t)} \right) + \left((1 - \beta) \frac{E(t)}{E_{local}(t)} \right) \quad (4)$$

تمام کارها با نرخ ورود λ به صف می‌رسند. سیستم صف ما با مدل M/M/C پیاده‌سازی می‌شود. شرط پایداری صف ρ طبق عبارت زیر به دست می‌آید:

$$\rho < 1, \lambda < m\mu \quad (5)$$

جایی که μ به معنی نرخ سرویس‌دهی در هر VM با توزیع نمایی است و $m\mu$ نرخ سرویس‌دهی برای ماشین مجازی m است. به طور کلی برای یک صف، اگر $\rho < 1$ باشد، یعنی یک صف پایدار است. لیست به تعویق افتاده‌های صف $q_n(t)$ است که تعدادی از کارها را در انتهای صف نشان می‌دهد. ما مجموعه‌ای از سیاست‌ها را روی صف‌ها اعمال می‌کنیم، زمان بندی اولویت در شکاف‌های زمانی را تعریف می‌کنیم. در مدل ما اولویت بندی بر اساس زمان اجرای کار است که در آن کار با زمان اجرای بالاتر دارای اولویت بالاتر در صف است و در صف دیگر کار با زمان اجرای پایین، دارای اولویت بالاتر است. در این مقاله ما دو سناریو داریم، در اولین سناریو همه کارها با مدل HTHP به VM_1 وارد می‌شوند و دوم همه کارها با مدل LTHP به VM_2 وارد می‌شوند. الگوریتم ۱ مراحل الگوریتم LTHP را نشان می‌دهد که الگوریتم HTHP کاملاً مشابه آن عمل می‌کند و تنها تفاوت در خطوط ۵ و ۷ می‌باشد. پارامترهای ϑ طول صف و $pro(i)$ اولویت هر کار را نشان می‌دهد.

۳- مدل بارگذاری پویا

برای حفظ پایداری سیستم از الگوریتم بهینه‌سازی لیاپانوف استفاده می‌کنیم. راه حل ارائه شده نیاز به دانش قبلی از وضعیت فعلی سیستم ندارد و تصمیم‌گیری بر

Input: $T(t)\mu, m, n, \lambda$.

```

1: repeat
2:   Initialization:  $vp_n, pro(i), t$ ,
3:   do
4:     for all  $i \in 0, 1, n$  and  $i \neq 0$ 
5:       if  $T_i(t)$  is min and  $E(t)$  is max
6:       then highestpriority  $\rightarrow pro(i)$ 
7:       else  $T_i(t)$  is max and  $E(t)$  is max
8:       then lowestpriority  $\rightarrow pro(i)$ 
9:       insert  $(i, pro(i))$ . Insert task  $i$  to the LTHP queue with
       an associated priority  $pro(i)$ 
10:       $v + 1 \rightarrow v$ 
11:      remove  $(i, pro(i))$ . Remove task  $i$  from the LTHP
       queue that has the highest priority and  $T_i(t) \leq t$ 
12:      Update
13:       $v - 1 \rightarrow v$ 
14:      if  $T_i(t) > t$ 
15:        insert  $(i, pro(i))$ . Insert task  $i$  to the LTHP queue with
        an associated priority  $pro(i)$ 
16:       $v + 1 \rightarrow v$ 
17: until All tasks are executed correctly and the queue is
       empty
    
```

$$\Delta L(t) + V\phi(t) \leq B + V + \sum_{n \in \emptyset} r_n(t) \left[\beta V \left(\frac{d(t)C_{local}(t)}{s(t)\mu_n(t)} + \frac{C_{local}(t)}{C_{vm}(t)} - 1 \right) + (1 - \beta)V \left(\frac{d(t)C_{local}(t)e_{tran}}{s(t)\mu_n(t)e_{active}} + \frac{C_{local}(t)e_{idle}}{C_{vm}(t)e_{active}} - 1 \right) + d(t)q_n(t) \right] - \sum_{n \in \emptyset} q_n(t)\mu_n(t) \quad (9)$$

سمت راست رابطه (۹) باید دارای مقدار $P_n(t)$ حداقل باشد. مقدار را برای وضوح بیشتر به صورت زیر تعریف می‌کنیم:

$$P_n(t) = \sum_{n \in \emptyset} r_n(t) \left[\beta V \left(\frac{d(t)C_{local}(t)}{s(t)\mu_n(t)} + \frac{C_{local}(t)}{C_{vm}(t)} - 1 \right) + (1 - \beta)V \left(\frac{d(t)C_{local}(t)e_{tran}}{s(t)\mu_n(t)e_{active}} + \frac{C_{local}(t)e_{idle}}{C_{vm}(t)e_{active}} - 1 \right) + d(t)q_n(t) \right] \quad (10)$$

تصمیم به بارگذاری برای هر VM برابر با مقدار کمینه رابطه (۱۰) است. بقیه جملات معادله (۹) در مقدار تصمیم نهایی تاثیر ندارند چرا که دارای مقدار ثابت هستند. در نتیجه ما $P_n(t)$ را با شرایط زیر تعریف می‌کنیم:

$$r_n(t) = \begin{cases} 1, & P_n(t) < 0 \\ 0, & P_n(t) \geq 0 \end{cases} \quad (11)$$

به همین دلیل، سیستم پایدار باقی می‌ماند و بهترین VM برای سرویس‌دهی انتخاب می‌شود. برای همه کارهای اجرا شده، بردار $r_n(t)$ شکل می‌گیرد. جایی که $\sum_{n \in \emptyset} r_n(t) \leq 0$ هیچ ماشین مجازی مناسبی در شکاف زمانی برای بارگذاری وجود ندارد و اگر $r_n(t) = 1$ باشد، بهترین ماشین مجازی انتخاب می‌شود. استفاده از تابع لیاپانوف سیستم را به سمت یک نقطه‌ی تعادل و پایدار همگرا می‌کند و سیستم در اکثر اوقات در حالت پایدار است.

۴. ارزیابی نتایج

در این بخش، از ماشین مجازی و یک دستگاه تلفن همراه استفاده می‌کنیم. ما درگاه آندروید x86 را در VMware اجرا می‌کنیم. با استفاده از این کار می‌توانیم کارهای رسیده از تلفن همراه را در ماشین مجازی اجرا کنیم. همچنین داده‌های به دست آمده را در نرم افزار R تجزیه و تحلیل می‌کنیم. هر ماشین مجازی با ۵۱۲ مگابایت حافظه و

اساس وضعیت فعلی سیستم انجام می‌شود. الگوریتم بهینه‌سازی لیاپانوف به منظور زمان بندی و بارگذاری بهینه به MEC از دانش جزئی شبکه استفاده می‌کند. تعداد n صف در سیستم به صورت بردار $Q(t) = (q_1(t), \dots, q_n(t))$ نشان داده می‌شود. برای هر فاصله زمانی t ، تابع لیاپانوف درجه دوم عبارت است از:

$$L = \frac{1}{2} \sum_{n \in \emptyset} q_n^2(t) \quad (6)$$

فاکتور Lyapunov به عنوان $\Delta L(t) = L(t+1) - L(t)$ تعریف می‌شود. ما فرض می‌کنیم که لیست معوقه‌های صف در طول زمان با توجه به معادله زیر تغییر می‌کنند:

$$L(t) = \frac{1}{2} \sum_{n \in \emptyset} [q_n^2(t+1) - q_n^2(t)] \leq \sum_{n \in \emptyset} [r_n(t)d(t)q_n(t) - q_n(t)\mu_n(t)] + B \quad (7)$$

مقدار ثابت و محدود $B > 0$ به صورت زیر محاسبه می‌شود:

$$B = \frac{1}{2} \sum_{n \in \emptyset} [(\mu_n^{max})^2 + (d_{max})] \quad (8)$$

فرض کنیم که V یک وزن غیر منفی است که کنترل تعامل میان مقدار اندازه صف و $\phi(t)$ را بر عهده دارد سپس با توجه به (۸)، شرایط $\Delta L(t) + V\phi(t)$ به صورت زیر حفظ می‌شود:

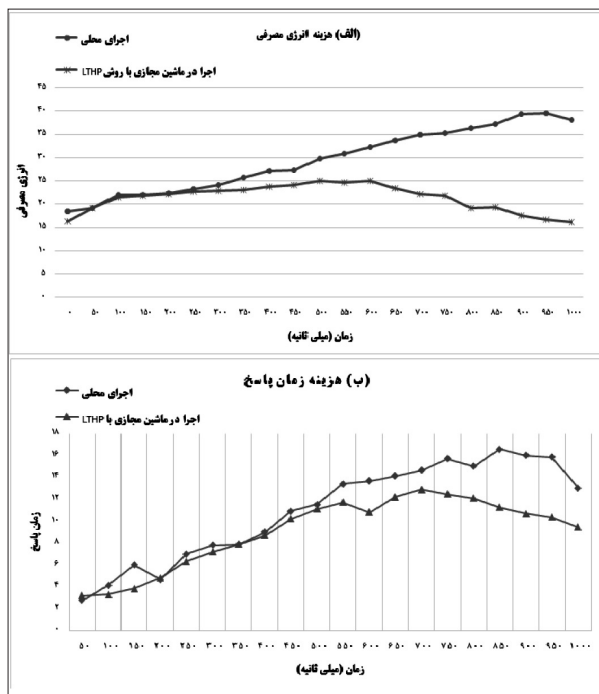
هسته CPU کار می‌کند. دستگاه تلفن همراه با یک پردازنده گیگاهرتزی و اینترنت نسل ۴/۵ کار می‌کند. توان باتری در سه حالت بدون استفاده، استفاده شده از قبل و مصرف شده در حین انتقال داده به ترتیب ۷۹، ۱/۵ و ۲/۲ درصد است. دستگاه تلفن همراه کارها را از طریق شبکه نسل ۴/۵ به ماشین‌های مجازی منتقل می‌کند. برنامه ما یک برنامه از کارهای بی‌درنگ است که اجرای سریع و به موقع کارها در آن اهمیت دارد. میانگین اندازه داده‌های ورودی برای داده‌های سبک ۱۰۳ کیلوبایت و برای داده‌های سنگین ۱۲۵ کیلوبایت است. پارامتر کنترل وزن $\beta = 0.95$ است. متغیر هزینه توسط کاربر با مقدار $\beta = 0.55$ است. برای ارزیابی نتایج، اجرای کارها در MEC را با اجرای محلی در دستگاه تلفن همراه مقایسه می‌کنیم.

شکل (۱) (الف) و شکل (۱) (ب) مقایسه مصرف انرژی و زمان پاسخ را بین سیاست بارگذاری در MEC با صف HTHP و اجرای محلی در دستگاه نشان می‌دهد. این رقم نشان می‌دهد که مصرف انرژی در حدود ۲۰ درصد کاهش می‌یابد و زمان پاسخ حدود ۲۱/۸۸ درصد سریع‌تر شده است. شکل (۲) (الف) و شکل (۲) (ب) مقایسه مصرف انرژی و زمان پاسخ را میان دو روش اجرا در MEC با صف LTHP و اجرای محلی در دستگاه نشان می‌دهد. این رقم نشان می‌دهد که مصرف انرژی در حدود ۱۷/۰۱ درصد کاهش می‌یابد و زمان پاسخ در حدود ۳۹/۱۳ درصد سریع‌تر است. صرفه‌جویی در انرژی و سرعت زمان پاسخ نشان می‌دهد که سیاست استفاده شده ما در زمان بندی مؤثر است. این روش پارامترهای ارزیابی از جمله انرژی مصرفی و زمان پاسخ که باعث محدودیت استفاده در منابع دستگاه می‌شوند را بهبود می‌بخشد. در ادامه لیست انتظار را در هر دو صف اجرا شده در زمان t نشان می‌دهیم. در آن لحظه، کارها به VM_1 صف HTHP و VM_2 با صف LTHP بارگذاری شدند. شکل (۳) لیست انتظار و تغییرات را در چهار روش FIFO، HTHP، LTHP و RR به صورت پویا نشان می‌دهد. کارها در صف‌ها قرار

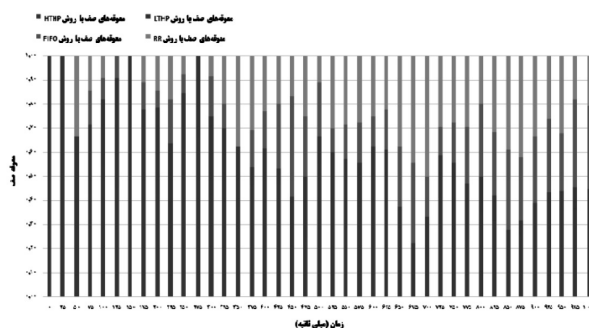
می‌گیرند و در شکاف زمانی تخصیص داده شده اجرا می‌شوند. چنانچه مدت زمان اجرای یکی از کارها بیش از شکاف زمانی در نظر گرفته شده باشد، کار به لیست معوقه‌های صف می‌رود و چنانچه در شکاف زمانی مورد نظر پایان یابد، از صف حذف شده و طول صف کاهش می‌یابد. هر دو صف پایدار هستند اما طول صف LTHP کمتر از طول صف HTHP است. زیرا لیست کارهای منتظر ناتمام در VM_2 با صف LTHP در شکاف‌های زمانی مختلف کمتر است. مقایسه لیست‌های معوقه چهار روش نشان می‌دهد که در هر شکاف زمانی لیست کارهای معوق در صف در روش LTHP نسبت به سه روش دیگر بهتر است و تعداد معوقه‌ها کمتر است. در شروع کار و با تعداد کار کمتر، روش FIFO و RR نسبت به روش HTHP نیز عملکرد بهتری دارند و درصد کارهای معوقه در هر شکاف زمانی در این دو روش کمتر است.

اما در بلند مدت عملکرد روش HTHP بهتر شده زیرا با اختصاص شکاف زمانی به همه کارها تعداد کارهای اجرایی بیشتر می‌شوند و از لیست انتظار حذف شده و لیست کارهای معوقه کوتاه می‌شوند. در کار ارائه شده روش LTHP در متوسط زمان یعنی از ابتدا تا انتها که از تعداد کارها کاسته و افزوده می‌شود، عملکرد بهتری را ارائه می‌دهد. تعداد کارهایی که به لایه MEC فرستاده می‌شود ۵۲۱ کار در هر ثانیه است که از طریق اتصال بی‌سیم نسل ۴/۵ ارسال می‌شود. تعداد کاری که به صورت محلی در تلفن همراه اجرا می‌شود، ۷۵ کار است. این مقدار نشان می‌دهد که بسیاری از کارها به درستی در لایه MEC اجرا شده‌اند. کارهای ارسال شده به MEC در اندازه محاسبات و داده بزرگ هستند در نتیجه احتمال انتقال کار به MEC بالاتر است.

کار سبک‌تر بر روی تلفن همراه اجرا می‌شود و بارگذاری نمیشود. زیرا هزینه ارسال این کارها به MEC بیشتر از اجرای محلی است. این سیاست بارگذاری به درستی کار می‌کند و باعث سودمندی در تلفن همراه و MEC می‌شود.

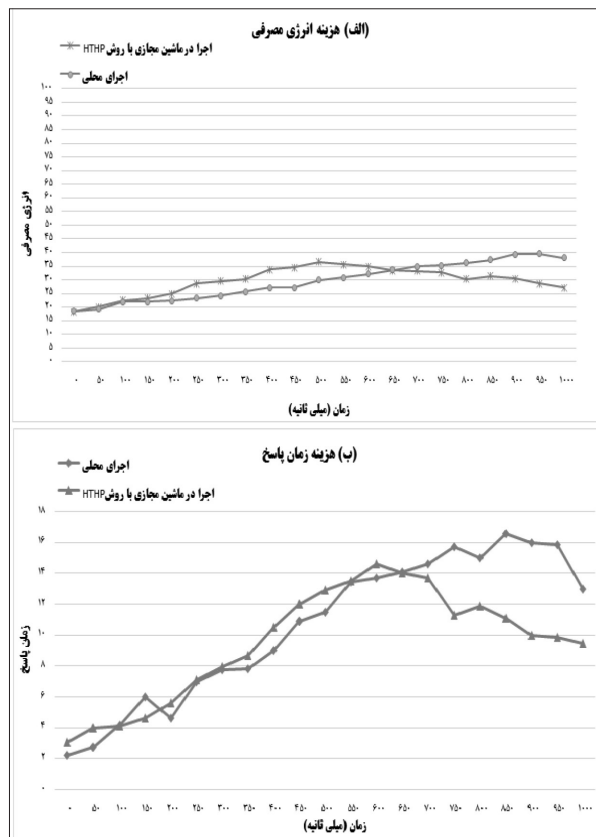


شکل ۲: مقایسه مصرف انرژی و هزینه زمان پاسخ میان دو روش اجرا در MEC با صف LTHP و اجرای محلی در دستگاه



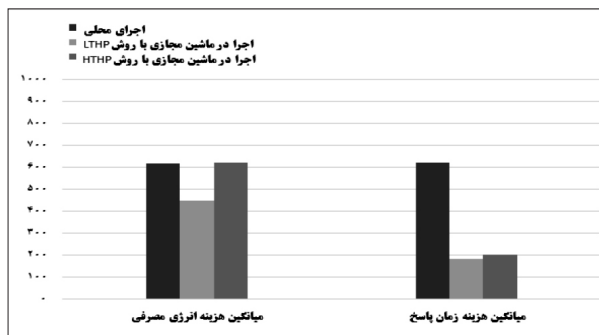
شکل ۳: لیست کارهای معوقه در چهار روش RR و LTHP، HTHP، FIFO

اجرا شده در دستگاه تلفن و ماشین مجازی با دو روش پیشنهادی در شکل (۵) نشان داده شده است. روش LTHP برای هر دو پارامتر نسبت به اجرای محلی و نسبت به روش HTHP بهینه تر است. افزایش تعداد کارها برای اجرا در سمت ماشین مجازی با روش LTHP پاسخ بهتری ارائه می‌کند. پارامتر هزینه انرژی مصرفی در دستگاه تلفن همراه و روش HTHP نزدیک به هم بوده و در طولانی مدت و با تعداد کارهای بیشتر روش HTHP بهینه تر عمل می‌کند. در شکل (۶) میانگین هزینه انرژی مصرفی و زمان پاسخ در چهار روش نشان داده شده است، که در این

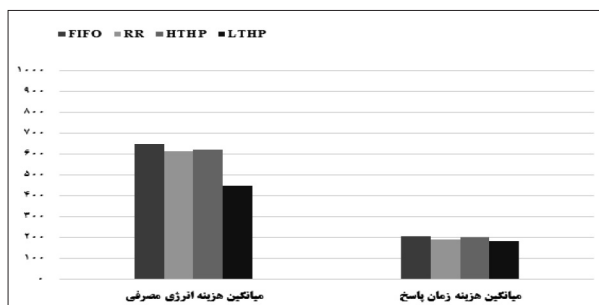


شکل ۱: مقایسه مصرف انرژی و هزینه زمان پاسخ میان دو روش اجرا در MEC با صف HTHP و اجرای محلی در دستگاه

بارگیری کارها با بار محاسباتی سنگین به MEC، انرژی را بر روی تلفن همراه نخیره خواهد کرد. تعداد کاری که به صورت محلی در تلفن همراه اجرا می‌شود، بسیار کمتر است، در نتیجه احتمال انتقال کار به MEC بالاتر است. شکل (۴)، هزینه انرژی مصرفی و زمان پاسخ بین دو روش ارائه شده و روش FIFO و RR را نشان می‌دهد. همان طور که در شکل (۴) الف) مشاهده می‌شود، در استفاده از روش FIFO در زمان‌های بالاتر انرژی مصرفی بیشترین مقدار را نسبت به روش RR و دو روش پیشنهادی دارد. روش LTHP در این میان حتی در بلند مدت نتیجه بهتری را ارائه می‌دهد. در شکل (۴) ب)، در روش FIFO، سرعت پاسخ به دلیل تجمع انبوه کارها در صف با تاخیر همراه است و هزینه سرعت پاسخ بالا است. دو روش پیشنهادی در بلند مدت نسبت به دو روش FIFO و RR بهتر جواب می‌دهد. میانگین هزینه انرژی مصرفی و زمان پاسخ برای کارهای

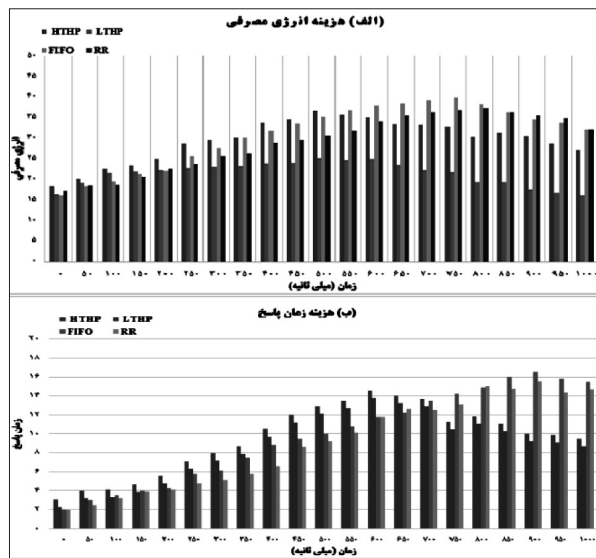


شکل ۵: مقایسه هزینه انرژی مصرفی و زمان پاسخ در سه حالت اجرای محلی، HTHP و LTHP



شکل ۶: مقایسه میانگین هزینه انرژی مصرفی و زمان پاسخ در چهار روش LTHP، HTHP، FIFO و RR

مبتنی بر اولویت با عنوان زمان اجرای بیشتر اولویت بالاتر و زمان اجرای کمتر اولویت بالاتر استفاده کردیم و این دو روش را با اجرای محلی در دستگاه تلفن همراه مقایسه کردیم. استفاده از بارگذاری به MEC در صرفه جویی انرژی و افزایش سرعت پاسخ برای تعداد زیاد کارها و کارها با حجم پردازش و داده بزرگ مناسب است. نتایج عددی ما نشان می‌دهد که دستگاه تلفن همراه حدود ۸۵ درصد از کارها را به MEC ارسال می‌کند و بیش از ۶۰ درصد در مصرف انرژی صرفه جویی کند. در سمت MEC برای ارزیابی وضعیت زمان‌بندی، چهار روش FIFO، HTHP، LTHP و RR با هم مقایسه کردیم و در این میان روش LTHP با ۵۵/۲ درصد صرفه جویی در انرژی و میانگین زمان پاسخ ۱۸۲/۹۶ ثانیه بهترین عملکرد را نسبت به سه روش دیگر دارد. کار آینده در این زمینه شامل موارد زیر است: یک سیستم بارگذاری سریع برای اجرای تعداد بیشتری از VMهای موازی که پیش‌بینی وضعیت آینده سیستم را نیز انجام می‌دهد. از سوی دیگر، از آنجایی که ما روی دو VM متمرکز بودیم،



شکل ۴: مقایسه هزینه انرژی مصرفی و زمان پاسخ در چهار روش LTHP، HTHP، FIFO و RR

مقایسه روش LTHP با ۵۵/۲ درصد صرفه جویی در انرژی بهتر از سه روش دیگر عمل می‌کند. روش RR با ۳۸/۸ درصد صرفه جویی در انرژی برای کارها با زمان اجرای کوتاه‌تر از شکاف زمانی در کوتاه مدت بهتر از روش HTHP عمل می‌کند. مقدار صرفه جویی انرژی در روش HTHP در حدود ۳۸ درصد است که از دو روش LTHP و RR ضعیف‌تر عمل می‌کند. مقدار صرفه جویی در انرژی در روش FIFO حدود ۳۳/۲ درصد است. هر دو روش FIFO و RR برای تعداد کمی از کارها با زمان اجرای کوتاه‌تر از شکاف زمانی عملکرد بهینه‌ای دارند. میانگین سرعت زمان پاسخ در روش LTHP برابر با مقدار ۱۸۲/۹۶ ثانیه است که در مقایسه با سه روش FIFO، HTHP و RR به ترتیب با مقادیر ۲۰۵/۶۶، ۱۹۹/۷۶ و ۱۹۰/۲۳ ثانیه بهتر عمل می‌کند.

۵. نتیجه گیری

اجرای سریع و به موقع برنامه‌های حساس به تاخیر با پردازش‌های سنگین، مهم است. از طرفی، صرفه جویی در انرژی تلفن همراه نیز اهمیت دارد. در این مقاله، ما یک سیاست بارگذاری پویا را ارائه دادیم که برای حفظ پایداری سیستم از الگوریتم لیاپانوف استفاده می‌کند. برای به دست آوردن بهترین راهبرد بهینه‌سازی، از دو روش زمان‌بندی

14. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani and R. Buyya.: Cloud-Based augmentation for mobile devices: Motivation, taxonomies and open challenges. arXiv preprint: 1306.4956, 2013.
15. S. Nunna, A. Kousaridas, M. Ibrahim, M. Dillinger, Ch. Thuemmler, H. Feussner and A. Schneider.: Enabling Real-Time Context-Aware Collaboration through 5G and Mobile Edge Computing. 12th International Conference on Information Technology - New Generations, 2015.
16. H. Kchaou, Z. Kechaou and A. M. Alimi.: Towards an Offloading Framework based on Big Data Analytics in Mobile Cloud Computing Environments.: *Procedia Comput. Sci.* 53, pp. 292–297, 2015.
17. Huang, D. Wang, P. Niyato, D.: A dynamic offloading algorithm for mobile computing. *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.
18. Wang, J. Peng, J. Wei, Y. Liu, D. Jieli, F.: Adaptive application offloading decision and transmission scheduling for mobile cloud computing. published in: 2016 IEEE International Conference on Communications (ICC), 2016.
19. Munoz, O. Pascual-Iserte, A. and Vidal, J.: Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading. *IEEE Trans. Vehicular Technolog.*, vol. 64, no. 10, pp. 4738–4754, Oct. 2015.
20. Lin, Y. E. Chu, T. Lai, Y. and Huang, T.: Time-and-energy-aware computation offloading in handheld devices to coprocessors and clouds. *IEEE Systems Journal*, vol. 9, no. 2, pp. 393–405, June 2015.
21. Zhang, Y. Niyato, D. and Wang, P.: Offloading in mobile cloudlet systems with intermittent connectivity. *IEEE Trans. Mobile Computing*, vol. 14, no. 12, pp. 2516–2529, Dec. 2015.
22. Yang, L. Cao, J. Cheng, H. and Ji, Y.: Multi-user computation partitioning for latency sensitive mobile cloud applications. *IEEE Trans. Computers*, vol. 64, no. 8, pp. 2253–2266, Aug. 2015.
23. Mao, Y. Zhang, J. Song, S.H. and Letaief, K. B.: Power-delay tradeoff in multi-user mobile-edge computing systems. in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, Dec, pp. 1–6, 2016.
24. L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, “Mobility-aware application scheduling in fog computing,” *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, Apr. 2017.
25. Xinchun Lyu, Wei Ni, Hui Tian, Ren Ping Liu, Xin Wang, Georgios B. Giannakis.: Optimal Schedule of Mobile Edge Computing for Internet of Things using Partial Information. in *IEEE Journal on Selected Areas in Communications* PP (99), September 2017.

گسترش به VMهای متعدد توزیع شده و استفاده از انواع دیگر روش‌های زمان‌بندی در سیستم‌های توزیع شده به منظور بهینه‌سازی سیستم‌های چند کاربره، قطعاً یک تعمیم مهم می‌باشد.

مراجع

1. Armbrust, M., R. G. A. Fox, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia.: Above the clouds: A Berkeley view of cloud computing. 2012
2. Zhang, Q., Cheng, L., and R. Boutaba.: Cloud computing: State-of-the-art and research challenges. *Journal Internet Services appl.*, vol. 1, no. 1, pp. 7–18, 2010.
3. Taleb et al, T.: EASE: EPC as a Service to Ease Mobile Core Network. *IEEE Network*, vol. 29, no. 2, Mar. pp. 78–88, 2015.
4. Taleb, T.: Toward Carrier Cloud: Potential, Challenges, and Solutions. *IEEE Wireless Commun.* vol. 21, no. 3, pp. 80–91, June 2014.
5. Fan, B. Leng, S. Yang, K.: A dynamic bandwidth allocation algorithm in mobile networks with big data of users and networks. vol. 30, no. 1, pp. 6–10. *IEEE Network*, Jan. 2016.
6. ETSI, Mobile-Edge Computing Introductory Technical White Paper, 2014, <http://www.etsi.org/technologies-clusters/technologies/mobile-edge-computing-ptc/2018.06.03>.
7. Sh. Weisong, C. Jie, Quan Zhang, L. Youhuizi, X. Lanyu.: Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, pp. 637–646, 2016.
8. F. Bonomi, R. Milito, J. Zhu and S. Addepalli.: Fog computing and its role in the Internet of things. in: *Proceedings of the first Edition of the MCC Workshop on Mobile Cloud Computing*. pp.13–16, 2012.
9. W.S. Chin, H. soo Kim, Y.J. Heo and J.W. Jang.: A context-based future network infrastructure for IoT services. *Procedia Comput. Sci.* 56, pp. 266–270, 2015.
10. S. Datta, C. Bonnet and J. Haerri.: Fog computing architecture to enable consumer centric Internet of things services. in: *Proceedings of the 2015 IEEE International Symposium on Consumer Electronics*, pp. 1–2, 2015.
11. S. Misra and S. Sarkar.: Theoretical modelling of fog computing: a green computing paradigm to support IoT applications. *IET Networks*. Five, pp. 23–29, 2016.
12. D. Lanka, Ch. Lakshmi Veenadhari and D. Suryanarayana.: Application of Fog Computing in Military Operations. *International Journal of Computer Applications* 164(6): pp.10–15, April 2017.
13. M. Zhanikeev.: A cloud visitation platform to facilitate cloud federation and fog computing. *Computer* 48 (5), pp. 80–83, 2015.