

A resource allocation method for fog computing considering user mobility

Samira Ansari Moghaddam¹, Mehri Rajaei^{2*} and Samira Noferesti³

1- Faculty of Electrical and Computer Engineering, University of Sistan and Baluchestan, Zahedan, Iran.

2* - Faculty of Electrical and Computer Engineering, University of Sistan and Baluchestan, Zahedan, Iran.

3- Faculty of Electrical and Computer Engineering, University of Sistan and Baluchestan, Zahedan, Iran.

¹samira.ansary96@gmail.com, ^{2*}rajayi@ece.usb.ac.ir, and ³snoferesti@ece.usb.ac.ir

Corresponding author's address: Mehri Rajaei, Faculty of Electrical and Computer Engineering, University of Sistan and Baluchestan, Zahedan, Iran.

Abstract-With the growing use of mobile devices and delay-sensitive applications, the need to move computing power to the proximity of users increases compared to cloud computing. However, fog nodes have limited computing and storage capacity and, therefore, do not perform well under congestion of requests. One of the causes of congestion and loss of user requests is the mobility of users across fog node areas. The purpose of this paper is to provide an efficient method for resource allocation in fog computing considering user mobility with the aim of increasing the acceptance rate of users' delay-sensitive requests and reducing the response time of requests. The proposed method predicts congestion in the fog node based on the mobility of users and the condition of its queue and, accordingly, decides whether to process the requests in the fog or to send them to the cloud. The experimental results indicated that the proposed method involves a lower response time and a higher acceptance rate than other existing methods. The results of the experiments show that the response time of the proposed method is reduced by 73, 77 and 10 percent compared to the three FCFS, Cloud-only and Delay priority algorithms, respectively. In the proposed algorithm, the request acceptance rate has increased by 44, 25 and 5 percent compared to the above three algorithms, respectively.

Keywords- Resource allocation –Fog computing – User mobility – Request Congestion.

ارائه یک روش تخصیص منابع در رایانش مه با در نظر گرفتن تحرک کاربران

سمیرا انصاری مقدم^۱، مهری رجائی*^۲، سمیرا نوفرستی^۳

۱- دانشکده مهندسی برق و کامپیوتر، دانشگاه سیستان و بلوچستان، زاهدان، ایران.

*^۲- دانشکده مهندسی برق و کامپیوتر، دانشگاه سیستان و بلوچستان، زاهدان، ایران.

^۳- دانشکده مهندسی برق و کامپیوتر، دانشگاه سیستان و بلوچستان، زاهدان، ایران.

¹samira.ansary96@gmail.com, ²*rajayi@ece.usb.ac.ir, ³snoferesti@ece.usb.ac.ir

* مهری رجائی، زاهدان، خیابان دانشگاه، دانشگاه سیستان و بلوچستان، دانشکده مهندسی برق و کامپیوتر.

چکیده- با رشد و گسترش استفاده از دستگاه‌های تلفن همراه و برنامه‌های کاربردی حساس به تاخیر، نیاز به انجام محاسبات در نزدیکی کاربر و در گره‌های مه بجای پردازش در ابر بیشتر می‌شود. با این وجود، گره‌های مه قدرت پردازشی و ذخیره‌سازی محدودی دارند و در نتیجه در شرایط ازدحام درخواست‌ها، نرخ پذیرش و زمان پاسخ مناسبی ندارند. یکی از دلایل بروز ازدحام و از دست رفتن درخواست‌ها، تحرک کاربران و خروج از محدوده تحت پوشش گره مه فعلی می‌باشد. در این مقاله روشی نوین برای تخصیص منابع در رایانش مه با هدف افزایش نرخ پذیرش درخواست‌های حساس به تاخیر کاربران و کاهش زمان پاسخ کلیه درخواست‌ها در شرایط تحرک کاربران پیشنهاد می‌گردد. روش پیشنهادی بر اساس تحرک کاربران و وضعیت صف در گره‌های مه، به پیش‌بینی ازدحام در این گره‌ها پرداخته و بر این اساس در مورد پردازش درخواست‌ها در گره مه و یا انتقال درخواست‌ها به گره ابر تصمیم‌گیری می‌کند. در این مقاله الگوریتم پیشنهادی با سه الگوریتم FCFS، Cloud-only و Delay priority مقایسه شده است. نتایج آزمایشات انجام گرفته نشان می‌دهد که زمان پاسخ روش پیشنهادی نسبت به سه الگوریتم مذکور به ترتیب ۷۳، ۷۷ و ۱۰ درصد کاهش یافته است. در الگوریتم پیشنهادی نرخ پذیرش درخواست‌ها نسبت به سه الگوریتم فوق به ترتیب ۴۴، ۲۵ و ۵ درصد افزایش یافته است.

واژه‌های کلیدی: تخصیص منابع، رایانش مه، تحرک کاربران، ازدحام درخواست‌ها.

۱- مقدمه

پردازشی بالا و فضای ذخیره‌سازی نامحدود را فراهم می‌کند اما به دلیل تاخیر بالای شبکه در مواجهه با درخواست‌های حساس به تاخیر عملکرد نامناسبی دارد [۲]. برخی کاربردها از قبیل سیستم چراغ ترافیک در حمل و نقل هوشمند، مدیریت هوشمند زباله، سیستم سلامت هوشمند، مدیریت هوشمند انرژی، بازی‌های برخط و... به زمان پاسخ بسیار کوتاه و پشتیبانی از تحرک کاربران نیاز دارند [۳]. از طرف دیگر، بسیاری از داده‌های خام نیاز به ذخیره‌سازی در ابر را نداشته و می‌توانند پس از تحلیل، فیلترسازی و کاهش حجم به ابر ارسال شوند.

برای مقابله با محدودیت‌های رایانش ابری، رایانش مه^۴ با هدف فراهم‌سازی امکانات پردازشی، فضای ذخیره‌سازی و ارتباطی در

اینترنت اشیا^۱ (IoT) اخیراً به یکی از بزرگترین تحولات در حوزه فناوری اطلاعات تبدیل شده است. شرکت IDC^۲ پیش‌بینی می‌کند تا سال ۲۰۲۵، ۵۵/۷ میلیارد دستگاه در سراسر جهان به هم متصل خواهند شد که ۷۵ درصد از این دستگاه‌ها از پلتفرم‌های IoT استفاده خواهند کرد. همچنین حجم داده‌های تولید شده توسط دستگاه‌های IoT در سال ۲۰۲۵ نسبت به سال ۲۰۱۹، حدود ۴۰۰ درصد افزایش خواهد داشت [۱].

جهت پردازش و ذخیره‌سازی داده‌های تولید شده توسط دستگاه‌های IoT از پلتفرم رایانش ابری^۳ استفاده می‌شود. رایانش ابری قدرت

۲- مرور کارهای پیشین

در این بخش مطالعات انجام شده در حوزه تخصیص منابع در رایانش مه مورد بررسی قرار می‌گیرد. اغلب تحقیقات موجود به طور خاص بر مسئله زمانبندی وظایف در رایانش مه تمرکز داشته و معیارهای متعددی مانند نرخ پذیرش درخواست‌ها، زمان پاسخ، مصرف انرژی، ترافیک شبکه، تعادل بار، هزینه و ... را مورد استفاده قرار داده‌اند.

شوشتریان و همکاران [۶] از یک متد خوشه‌بندی جهت یافتن گره مه یا خوشه‌ای از گره‌های مه مناسب، با هدف افزایش بهره‌وری منابع، کاهش تاخیر و برطرف کردن نیازمندی‌های کیفیت سرویس برنامه کاربردی، استفاده کرده‌اند. گوورا و همکاران [۷] از زمانبندی مبتنی بر برنامه‌ریزی خطی عددی جهت زمانبندی درخواست‌ها و اختصاص گره مه مناسب به درخواست‌های کاربر استفاده کرده‌اند. بابورو و همکاران [۸] یک روش تخصیص منابع مبتنی بر بهینه‌سازی ازدحام ذرات با هدف کاهش زمان انتظار، تعادل بار میان گره‌های مه و کاهش مصرف بار شبکه ارائه داده‌اند. در این مقاله از الگوریتم بهینه‌سازی گروه بیت بر پایه الگوریتم ازدحام ذرات برای انتخاب بهترین گره مه برای اختصاص درخواست کاربر استفاده می‌شود. وادهوا و همکاران [۹] یک تکنیک اختصاص منابع بر اساس شدت نیاز وظایف کاربر پیشنهاد داده‌اند. در این مقاله ابتدا تمامی منابع در دسترس گره مه شناسایی می‌شوند. منابع در دسترس یک گره مه بر اساس پهنای باند، زمان پاسخ و زمان اتمام کار تعیین می‌گردد. با اتمام منابع موجود در گره مه از منابع در دسترس گره ابر استفاده می‌شود.

وان شبه فام و همکاران [۱۰] با استفاده از یک الگوریتم ابتکاری، با توجه به لزوم تعادل بین حداکثر زمان اجرای درخواست و هزینه اقتصادی سعی در حل مسئله زمانبندی درخواست‌ها در میان گره‌های مه دارند. مدیریت منابع و زمانبندی توسط یک گره واسط صورت می‌گیرد. این گره، جهت انجام زمانبندی، اولویت وظیفه را بر اساس متوسط زمان اجرا و انتقال آن محاسبه می‌کند، به‌صورتی که وظایف سنگین‌تر و کارهای آغاز شده اولویت بالاتری دارند. سپس گره مناسب جهت اجرای وظیفه، با در نظر گرفتن توازن هزینه وظیفه (هزینه انتقال و اجرا) و زمان اتمام کار، از میان گره‌های مه و ابر، انتخاب می‌شود.

سلیم بیتام و همکاران [۱۱] از الگوریتم زنبور عسل برای حل مشکل زمانبندی در مه استفاده کردند. در این مقاله، گره مدیر وظیفه مدیریت درخواست‌ها و تقسیم وظایف یک درخواست در میان سایر گره‌ها را برعهده داشته و با توجه به زمان اجرای پردازنده و حافظه مصرفی به عنوان اجزای تابع شایستگی، سعی در اختصاص وظایف به گره‌های مه ناهمگن دارد. در این مقاله جهت بهبود عملکرد، جهش

نزدیکی کاربران نهایی، در سال ۲۰۱۲ توسط سیسکو معرفی شد. مه یک ابر در نزدیکی کاربران است و سرویس‌های محاسباتی، ذخیره‌سازی و ارتباطی را برای آنها فراهم می‌کند [۴]. گره‌های مه^۵ به عنوان لایه‌ی میانی بین ابر و دستگاه‌های اینترنت اشیا، در نزدیکی کاربران نهایی مستقر می‌شوند. در نتیجه پرداختن به تقاضای هزاران سرویس IoT می‌تواند به راحتی و با تاخیر بسیار کمتر نسبت به ابر فراهم گردد.

از دیگر مزایای رایانش مه، پشتیبانی از کاربران متحرک است. تحرک کاربران و خروج آنها از محدوده تحت پوشش یک گره مه می‌تواند موجب افزایش زمان پاسخ و وقفه در سرویس‌دهی گردد [۵].

یکی از مسائل مطرح در رایانش مه، نحوه تخصیص منابع محدود گره‌های مه به درخواست‌های کاربران، با هدف رسیدن به بیشترین نرخ پذیرش و کمترین زمان پاسخ درخواست‌ها است. تحرک بالای کاربران موبایل در میان گره‌های مه و لزوم اختصاص به‌موقع، مناسب و کارایی منابع در کاربردهای حساس به تاخیر، تخصیص منابع در رایانش مه را به مسئله‌ای چالش برانگیز تبدیل کرده است. تخصیص بهینه منابع می‌تواند موجب کاهش زمان پاسخ، استفاده بهینه از منابع مه، کاهش نیاز به انتقال درخواست کاربر به ابر و افزایش رضایت کاربران گردد.

هدف این مقاله، ارائه روشی نوین برای اختصاص منابع در رایانش مه با در نظر گرفتن تحرک کاربران و ازدحام درخواست‌ها در گره‌های مه می‌باشد. نوآوری‌های روش پیشنهادی به اختصار عبارتند از:

- در نظر گرفتن تحرک کاربران و ارائه روشی برای کاهش درصد درخواست‌های لغو شده کاربران به دلیل خروج از محدوده تحت پوشش گره مه فعلی
- در نظر گرفتن ازدحام درخواست‌ها در یک گره مه و ارائه روشی برای مدیریت ازدحام با توجه به نوع درخواست‌ها

نتایج آزمایشات انجام گرفته نشان می‌دهد که روش پیشنهادی برای تخصیص منابع بر اساس معیارهای نرخ پذیرش، زمان پاسخ و مصرف ترافیک بهتر از روش‌های موجود عمل می‌کند.

سازماندهی مقاله حاضر به شرح زیر است: در بخش ۲ به مرور کارهای پیشین در زمینه تخصیص و زمانبندی منابع در رایانش مه پرداخته می‌شود. در بخش ۳ معماری رایانش مه تشریح شده و راهکار پیشنهادی برای تخصیص منابع در رایانش مه مطرح می‌شود. در بخش ۴ نتایج حاصل از آزمایشات انجام گرفته به منظور ارزیابی روش پیشنهادی ارائه می‌گردد. در انتها در بخش ۵ نتیجه‌گیری و پیشنهادهای آتی بیان می‌گردد.

ابر، در نظر گرفته شده است. در این الگوریتم اختصاص وظایف به گره‌های مه به صورت ایستا و بر اساس اولویت گره‌های مه و به دو صورت اختصاص سری و اختصاص موازی وظایف، انجام می‌شود. هر چه گره به کاربران نزدیکتر باشد، اولویت بالاتری دارد.

هواپینگ ژانگ و همکاران [۱۹] به مسئله قیمت‌گذاری سرویس‌های مه، توسط فراهم‌کنندگان مختلف، توجه کرده‌اند. در روش پیشنهادی این مقاله هر واسط DSO^6 سرویس‌های منحصر به فردی به کاربران ارائه می‌دهد. در حقیقت DSO به عنوان لایه‌ی میانی وظایف زمان‌بندی، قیمت‌گذاری و ارسال درخواست‌ها به گره‌های مه را برعهده دارد. واسط DSO با توجه به اطلاعات سرویس‌های پیشین خریداری شده‌ی کاربر، مقدار منابعی را که در آینده تقاضا خواهد شد، پیش‌بینی کرده و از این پیش‌بینی جهت قیمت‌گذاری خدمات و سرویس‌های خود استفاده می‌کند. در [۲۰، ۲۱] درخواست‌ها به صورت جداگانه به گره‌های مه، بدون در نظر گرفتن تاخیر انتقال، فرستاده می‌شوند. درخواست‌های پردازش نشده از مه به ابر منتقل شده و هدف، اختصاص بار کاری به ابر و یا مه با کمترین میزان مصرف انرژی و تاخیر ارتباطی می‌باشد. لیندونگ لیو و همکاران در روش [۲۲] با استفاده از پایگاه‌داده شامل تراکنش‌های زمان‌بندی سعی در حل مسئله‌ی زمان‌بندی داشته‌اند. تراکنش‌ها نشان‌دهنده‌ی این هستند که چه وظایف و چه گره‌های مه‌ی در زمان‌بندی‌های گذشته شرکت داشته و چگونه زمان‌بندی شده‌اند. براساس این پایگاه تراکنشی و با استفاده از الگوریتم $Apriori$ ، مجموعه قوانین انجمنی تولید می‌شود. سپس زمان‌بندی با توجه به این قوانین و با استفاده از الگوریتم $TSFC^7$ صورت می‌گیرد. الگوهای تولید شده نشان‌دهنده‌ی این است که چه وظایفی در کدام گره‌های مه، اولویت اجرایی دارند.

تعداد معدودی از روش‌های پیشین به مسئله تحرک کاربران و تاثیر آن بر ازدحام و عدم پذیرش درخواست‌ها توجه داشته‌اند.

لوپس بیتنکورد و همکاران [۲۳] درخواست‌ها را به دو دسته درخواست‌های با اولویت بالا و درخواست‌های با اولویت پایین گروه‌بندی کرده‌اند. در این مقاله سه الگوریتم $FCFS^8$ ، $Concurrent$ و اولویت، معرفی و جهت زمان‌بندی استفاده شده‌اند. نویسندگان این مقاله با ارائه یک سناریوی ساده از حرکت کاربران در میان گره‌های مه و در نتیجه ازدیاد درخواست‌های ارسالی یک گره، تحرک و ازدحام را شبیه‌سازی نموده و نشان داده‌اند که با آگاهی از نحوه حرکت کاربران و مهاجرت درخواست‌های غیر حساس به تاخیر به ابر در ابتدای شبیه‌سازی، می‌توان تاخیر اجرای درخواست‌های حساس به تاخیر را کاهش داد. در این روش زمان‌بندی به صورت پویا و بهینه شده با توجه به تحرک کاربران در سناریوی خاص می‌باشد و هدف

در میان راه‌حل‌های ضعیف و با توجه به نزدیکی گره‌های مه به یکدیگر انجام می‌گیرد.

در مقاله‌ی جیانگوا فن و همکاران [۱۲] مسئله زمان‌بندی به عنوان یک مسئله کوله‌پشتی ۰ و ۱ چندبعدی معرفی شده است. در این مقاله، یک الگوریتم کلونی مورچه‌ها با هدف کارایی بالای گره‌های مه و در عین حال توجه به محدودیت مهلت ارسال پاسخ پیشنهاد شده است. در تابع شایستگی به زمان اتمام وظایف با توجه به زمان انجام محاسبات، انتقال داده در میان گره‌ها و وظایف مختلف و همچنین هزینه اقتصادی محاسبات توسط سرورهای ابر توجه شده است.

زامیت کومار میشر و همکاران [۱۳] از الگوریتم ایستای خفاش با هدف تعادل در میان مصرف انرژی و بیشترین زمان پاسخ اجرای درخواست‌ها استفاده کرده‌اند. با داشتن مقادیر مشخص زمان اجرای هر وظیفه در ماشین‌های مجازی گره‌های مختلف مه، در هر تکرار الگوریتم و برای هر ذره، بیشترین زمان اجرا در میان ماشین‌های مجازی و میزان مصرف انرژی در حالت بیکار و مشغول محاسبه شده است. به این ترتیب تابع شایستگی جهت مشخص نمودن خفاش‌های برتر محاسبه می‌گردد.

هدف صباحی کبیرزاده و همکاران [۱۴]، گزینش بهترین الگوریتم زمان‌بندی با تکنیک تست و انتخاب می‌باشد. الگوریتم دارای دو فاز آموزش و تست است. در فاز آموزش با توجه به ویژگی‌های جریان داده‌های مختلف و نتایج الگوریتم‌های زمان‌بندی ژنتیک، ازدحام ذرات، کلونی مورچه‌ها و شبیه‌سازی آبکاری فلزات، یادگیری صورت می‌گیرد. تابع شایستگی با توجه به مجموع مصرف پهنای باند و بهره‌وری پردازنده‌ها محاسبه می‌گردد. سپس در فاز تست، الگوریتم مناسب جهت زمان‌بندی انتخاب می‌گردد. در پژوهشی دیگر صباحی کبیرزاده [۱۵]، هدف زمان‌بندی مبنی بر امنیت می‌باشد. این روش مشابه روش قبلی است، تنها تفاوت در تابع شایستگی جدید است. در این جا تابع شایستگی علاوه بر مصرف پهنای باند و بهره‌وری پردازنده‌ها، سربار امنیتی را نیز شامل می‌شود.

تیهسونی چوداری و همکاران در [۱۶] فرض کرده‌اند که در هر گره مه، تعدادی پردازنده، ماشین مجازی و یک مولفه مدیریت سرور مه قرار گرفته است که الگوریتم زمان‌بندی را اجرا می‌کند. وظیفه الگوریتم زمان‌بندی اختصاص مراکز داده و ماشین‌های مجازی گره مه به درخواست‌های کاربران بر اساس اولویت و مهلت درخواست می‌باشد. یهوا سان و همکاران در [۱۷] از نظریه بازی‌ها برای زمان‌بندی درخواست‌های کاربر و همچنین تشکیل خوشه‌هایی از گره‌های مه استفاده کردند. در این مقاله مسئله زمان‌بندی درخواست کاربر به صورت یک بازی غیر همکارانه در میان گره‌های مه فرموله می‌شود. در مقاله [۱۸] دو لایه محاسباتی مه در لبه و مه در نزدیکی

از شبکه‌های عصبی کانولوشن^{۱۱} بر زمانبندی منابع مه را بررسی می‌کند. از سوی دیگر رویکرد Hunter مطرح شده در [۳۲] را با استفاده از شبکه‌ی عصبی بازگشتی دوطرفه^{۱۲} گسترش می‌دهد. در [۳۳] یک رویکرد پویا بر اساس وضعیت منابع گره‌های مه ارائه شده است. در این مقاله فرض بر این است که می‌توان یک درخواست را به صورت مجموعه‌ای از وظایف در نظر گرفت. این وظایف می‌توانند به صورت موازی در چندین گره مه اجرا شوند. این کار موجب کاهش زمان اجرای درخواست می‌گردد. الگوریتم DCTO^{۱۳} یک الگوریتم بهینه جهت تخلیه وظایف و تخصیص منابع به صورت پویا می‌باشد.

در جدول ۱ تحقیقات پیشین در زمینه تخصیص منابع در رایانش مه بر اساس ویژگی‌های مختلف شامل گره‌های مه همگن/ناهمگن، وجود/عدم وجود کنترل‌کننده‌ی مرکزی، پشتیبانی/عدم پشتیبانی از تحرک، مستقل از تاریخچه/مبتنی بر تاریخچه بودن الگوریتم، ایستا/پویا، حمایت از وظایف مستقل/زنجیره‌ای، زمانبندی با وجود/بدون ابر و ارتباط/عدم ارتباط میان کلاسترها مقایسه شده‌اند. همان‌طور که مشاهده می‌شود، تعداد اندکی از روش‌های موجود تخصیص منابع در رایانش مه از تحرک کاربران پشتیبانی می‌کنند. این درحالی است که چارچوب مه باید توان پاسخ به درخواست‌ها در شرایط تحرک را داشته باشد.

۳- روش پیشنهادی

هدف این مقاله ارائه روشی کارا برای تخصیص منابع در رایانش مه با در نظر گرفتن تحرک کاربران است. از آنجا که تمرکز اصلی پژوهش حاضر بر کاهش زمان پاسخ درخواست‌های حساس به تأخیر می‌باشد، کاربران بر اساس نیازمندی‌های کیفیت سرویس به دو دسته تقسیم می‌شوند. دسته‌ی اول کاربران متصل از طریق برنامه‌ی کاربردی با درخواست‌های حساس به تأخیر هستند. این درخواست‌ها باید در مدت زمان مشخص و از پیش تعیین شده پردازش شوند. دسته‌ی دوم کاربران متصل از طریق برنامه کاربردی با نیازهای غیر حساس به تأخیر هستند. برای پاسخ به درخواست‌های این کاربران محدودیت زمانی وجود ندارد.

منابع دستگاه‌های موجود در لبه‌ی شبکه محدود است، در نتیجه برخی از وظایف پردازشی باید به مراکز ابر ارسال شوند. در صورت انتقال درخواست‌های حساس به تأخیر به ابر با احتمال بالا مهلت زمانی آنها نقض خواهد شد. بنابراین باید سازوکاری برای اجرای وظایف حساس به تأخیر در گره‌های مه و حتی‌الامکان عدم انتقال آنها به ابر ارائه شود. در رویکرد پیشنهادی تلاش برای کاهش تعداد درخواست‌های ارسال شده به ابر و در نتیجه کاهش زمان پاسخ

کاهش تأخیر درخواست‌های حساس به تأخیر است. در [۲۴] نیز درخواست‌ها بر اساس اولویت به دو دسته درخواست‌های واقعی (حساس به تأخیر) و نرم (غیر حساس به تأخیر) دسته‌بندی شده‌اند. در این روش، زمان‌بندی توسط سرور اصلی مه با استفاده از الگوریتم RTES^۹ با اهداف توازن بار در محیط مه، اجرای درخواست‌ها در مهلت تعیین شده، افزایش گذرده‌ی، بهینه‌سازی شبکه و نگهداری موقت داده‌های مورد نیاز کاربر، صورت می‌گیرد.

در [۲۵] شبیه‌سازی جهت پشتیبانی از سیاست‌های مختلف تحرک، خوشه‌بندی و مدیریت میکروسرویس و بهینه‌سازی استفاده از حافظه و زمان شبیه‌سازی پیشنهاد شده است. در این مقاله برای مقابله با تحرک کاربران از مهاجرت مولفه‌های برنامه کاربردی استفاده می‌شود. امینا مسدی و همکاران در [۲۶] یک طرح تخصیص منابع آنلاین با توجه به وضعیت سرورها در گره‌های مه پیشنهاد داده‌اند. در این طرح هدف بهبود کیفیت سرویس بر اساس یک بازی Stackelberg است. در [۲۷] مسئله تخصیص منابع با هدف به حداقل رساندن احتمال مهاجرت به عنوان یک مسئله برنامه‌ریزی غیرخطی صحیح MINLP^{۱۰} در نظر گرفته می‌شود که یک مسئله NP-Hard است. برای حل این مسئله از الگوریتم ژنتیک استفاده می‌شود. در [۲۸] با توجه به شرایط ازدحام درخواست‌ها و تحرک کاربران، وضعیت منابع آزاد گره مه بررسی شده سپس در مورد محل قرارگیری و اجرای درخواست (گره مه یا ابر) تصمیم‌گیری می‌شود. در [۲۹] یک مدل دو گامه برای زمانبندی و اختصاص منابع پیشنهاد شده است. در گام اول، زمانبندی در درون یک خوشه از گره‌های مه صورت می‌گیرد. در این راستا از الگوریتم ازدحام ذرات، جهت زمانبندی و توزیع بار درون خوشه استفاده شده است. در گام بعدی با توجه به نتایج حاصل شده از توزیع بار در گام اول و با در نظر گرفتن زمان انتقال درخواست‌ها در میان خوشه‌های مختلف از الگوریتم تخصیص منابع PGABC استفاده می‌شود. این الگوریتم یک الگوریتم هیورستیک ترکیبی است و از ادغام سه الگوریتم کلونی زنبور عسل، ازدحام ذرات و ژنتیک حاصل می‌شود. در این مقاله هدف کاهش میزان مصرف انرژی و تأخیر می‌باشد.

در الگوریتم دو مرحله‌ای ارائه شده در [۳۰]، هدف اختصاص بهینه منابع، کاهش تأخیر، بهبود مصرف انرژی و کاهش مصرف شبکه می‌باشد. در مرحله اول از الگوریتم بیز ساده، جهت فاز یادگیری و اختصاص منابع به گره‌های مه استفاده می‌شود. در مرحله دوم برای بهبود نتایج از یک الگوریتم هیورستیک که حاصل ترکیب دو الگوریتم کلونی مورچه‌ها و ازدحام ذرات است، استفاده شده است.

در [۳۱] یک رویکرد جدید به نام HunterPlus با هدف کاهش مصرف انرژی و نرخ تکمیل کار پیشنهاد شده است. این رویکرد تأثیر استفاده

جدول ۱: تحقیقات پیشین در زمینه تخصیص منابع در رایانش مه

مرجع	نام الگوریتم	الگوریتم‌های مقایسه شده	ابزار شبیه‌سازی	گره‌های مه ناممکن	وجود کنترل‌کننده مرکزی	پشتیبانی از تحرک	مبتنی بر تاریخچه	پویا	حمایت از وظایف زنجیره‌های	زمان بندی با وجود ابر	ارتباط میان خوشه‌ها
[۶]	Proposed	QOS Aware	MATLAB	*				*			*
[۷]	CASSIA-INT CASSIA-RR	RR ¹⁴ Random	ifogsim	*				*	*	*	
[۸]	EDRAM	RR Throled	Linpack software tool IoT Arduino		*				*		*
[۹]	TRAM	SJF ¹⁵ , FCFS and MPSO ¹⁶	Ifogsim	*	*			*		*	
[۱۰]	Pirority	DLS ¹⁷ -Greedy for cost-HEFT ¹⁸	CloudSim	*	*				*		
[۱۱]	BLA ¹⁹	GA ²⁰ -PSO ²¹	BLA-F c++	*	*					*	
[۱۲]	ACO ²²	FIFO-MINMIIN	-	*				* (دوره‌ای)		*	
[۱۳]	IPSO ²³	PSO	Matlab	*	*				*		
[۱۴]	BAT ²⁴	PSO-BPSO ²⁵	Matlab								
[۱۵]	HH ²⁶	GA-PSO-ACO- SA ²⁷	iFogSim			*			*		
[۱۶]	Pirority	RD ²⁸ -ORC ²⁹ - ERA ³⁰	CloudSim					* (دوره‌ای)		*	
[۱۸]	Pirority	-	Pulp-Gurob							*	
[۱۷]	Proposed	OUS ³¹ -Max SINR ³² - Random	-		*						
[۱۹]	Stackelberg-meny to meny	-	Matlab	*	*		*				
[۲۰]	GBD-Hungarian- convex technique	-	Matlab	*	*					*	
[۲۲]	Iapiori-TSFC	Mimmin- MET ³³ - MCT ³⁴	simGrid	*	*		*	* (دوره‌ای)			
[۲۳]	Pirority	FCFS-concurent	iFogSim			*	*	*	*	*	
[۲۴]	RTES	Pirority-FCFS- multi Obj	CloudSim	*	*			*	*	*	
[۲۶]	Proposed	Naive	eclpis	*	*		*		-	-	-
[۲۷]	PA	AOA ³⁵ , ROA ³⁶ , AARM ³⁷ , NORAM ³⁸	-			*			*	*	-
[۲۹]	PGABC-PSO	SJF-PSO ABC&PSO, MFO ³⁹	-	*	*			*	-	-	*
[۳۰]	2st	Hybrid	Ifogsim	*	*			*	-	*	-
[۳۱]	CNN, BIGGCN	GGCN ⁴⁰	Cosco framework				*	*	*	*	
[۳۳]	DCTO	POST, RCO ⁴¹	-	*	*			*	*		
	روش پیشنهادی در این مقاله	Cloud only, delay priority, FCFS	ifogsim			*		*	*	*	

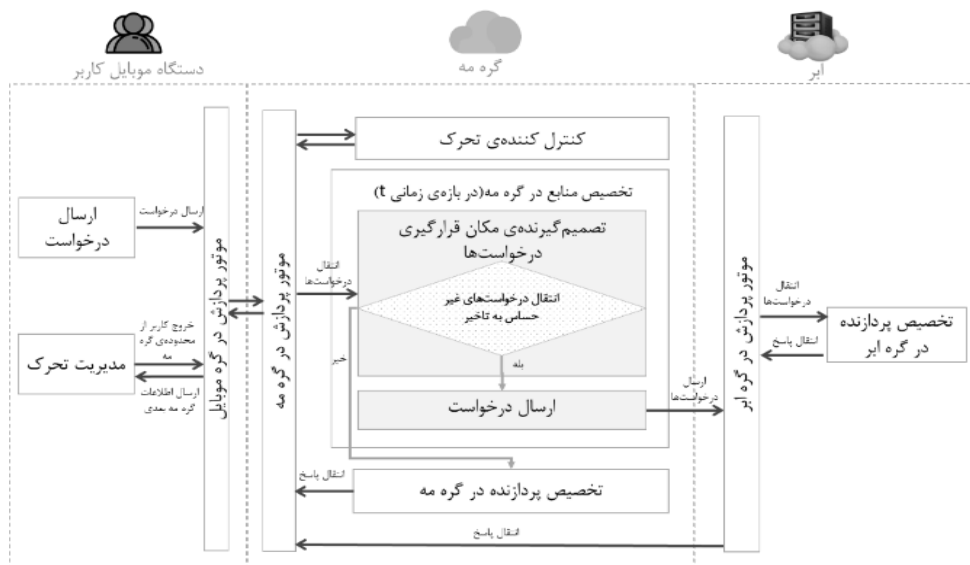
شکل ۲ شبه‌کد مولفه مدیریت تحرک در گره موبایل را نمایش می‌دهد. گره موبایل در هر بازه زمانی، موقعیت آتی خود را براساس سرعت و جهت حرکت ابتدای بازه پیش‌بینی می‌کند (خطوط ۴-۱). پیش‌بینی موقعیت آتی با فرض ثابت بودن جهت و سرعت فعلی کاربر در طول بازه زمانی آتی صورت می‌گیرد (خط ۴). در صورت کوچک در نظر گرفتن بازه زمانی بروزرسانی موقعیت کاربر، این فرض به واقعیت نزدیک است. به این ترتیب احتمال خروج کاربر از محدوده گره مه فعلی در بازه زمانی آتی مشخص می‌شود. در صورت پیش‌بینی خروج از محدوده گره مه فعلی (گره مبدا)، یک درخواست خروج توسط گره موبایل برای گره مه مبدا ارسال می‌شود (خطوط ۶-۵). گره مبدا با پردازش درخواست، اطلاعات گره مه مقصد را برای گره موبایل ارسال می‌نماید (خط ۱۹). از سوی دیگر، ورود کاربر جدید را به گره مه مقصد اطلاع می‌دهد. همچنین درخواست‌های موجود در صف (یعنی درخواست‌های اجرا نشده این گره موبایل) را برای گره مه مقصد ارسال می‌نماید (خط ۱۸). گره مه مقصد می‌تواند درخواست‌ها را بپذیرد یا به ابر ارسال کند. در گره مقصد، از آستانه‌ی λ که در ادامه شرح داده می‌شود برای تصمیم‌گیری جهت ارسال درخواست‌های غیرحساس به تاخیر به ابر و یا پذیرش کلیه‌ی درخواست‌ها استفاده می‌شود.

درخواست‌ها می‌باشد. در ادامه، الگوریتم پیشنهادی تخصیص منابع در شرایط تحرک کاربران تشریح می‌شود.

۳-۱- رویکرد پیشنهادی جهت مدیریت تحرک کاربران

در این پژوهش برای تخصیص منابع به درخواست‌های کاربران متحرک، مطابق شکل ۱، شش مولفه اصلی تعریف شده است. مولفه ارسال درخواست، وظیفه انتقال درخواست به مولفه موتور پردازش 42 را برعهده دارد. مولفه مدیریت تحرک در گره موبایل وظیفه کنترل تحرک و پیش‌بینی خروج کاربر از محدوده گره مه را بر عهده دارد. مولفه تخصیص منابع، در مورد قرارگیری درخواست‌ها در صف گره‌های مه و ابر تصمیم‌گیری می‌کند. مولفه موتور پردازش، نقش واسط را داشته و وظیفه ارسال درخواست‌ها به گره‌های مختلف را برعهده دارد. مولفه تخصیص پردازنده 43 ، وظیفه‌ی آنالیز و پردازش درخواست‌های موجود در صف گره مه را بر عهده دارد. مولفه کنترل‌کننده تحرک 44 ، وظیفه‌ی کنترل و مدیریت تحرک کاربران در گره مه را برعهده دارد.

در روش پیشنهادی زمان به بازه‌های زمانی کوچکی به طول T_1 شکسته می‌شود که در ابتدای هر بازه موقعیت‌های دستگاه‌های موبایل ارزیابی می‌شود بر اساس جهت و سرعت فعلی آنها موقعیت آنها در طول بازه بعدی پیش‌بینی می‌شود.



شکل ۱: مولفه‌های روش پیشنهادی جهت تخصیص منابع

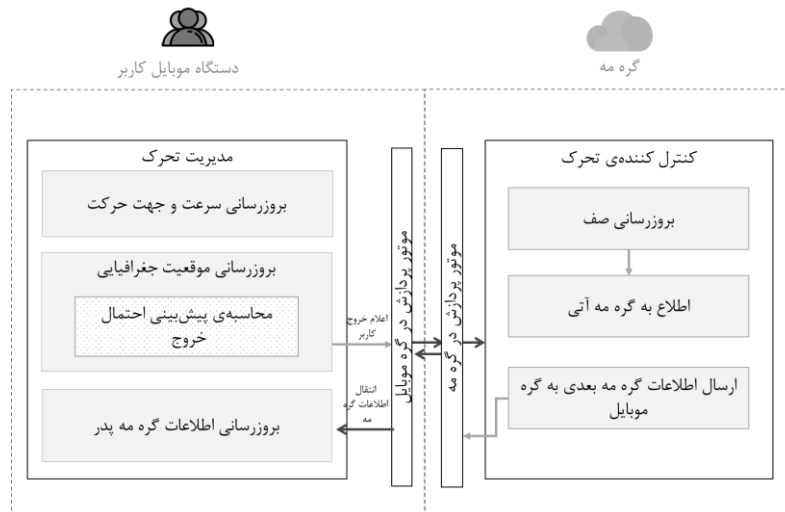
```

Function: Location_Update // Mobile node update location
Input: velocity, geo_loc, direction, parent_id
Output: velocity, geo_loc, direction
1 Begin
2 (velocity, direction) ← Velocity_and_Direction_Update ()
3 new_loc ← movement (velocity, direction, geo_loc); //Movement is a function to calculate new
                                     location
4 next_loc ← movement (velocity, direction, new_loc);
5 If (! current_fog_coverage. covers (next_loc)) // Find new parent (next fog node)
6     Message_Sender (parent_id, next_loc, "Request New Fog Info"); //Ask from current fog node
7 return geo_loc ← new_loc;
8 End

Function: Mobile_Parent_Information_Update //Mobile get new parent info( next fog node) from current fog
Input: message
9 Begin
10 parent_id ← message.get_new_parent_id();
11 current_fog_coverage ← message.get_new_parent_coverage();
12 End

Function: Queue_Update //Current fog node finds next fog info
Input: message
13 Begin
14 mobile_id ← message.get_source_id();
15 mobile_parent_info ← search_fogs_info_table (message.get_location()); //Search_fogs_info_table is
                                     a function to search next fog information
16 leave (mobile_id); //Delete mobile from mobiles list
17 requests ← get_mobile_requests_from_queue(mobile_id);
18 Request_Forwarder (mobile_parent_info. get_new_parent_id(), requests); //Send requests to next fog
node
19 Message_Sender (mobile_id, mobile_parent_info, "New Fog Info"); //Send next fog information to
mobile node
20 End
    
```

شکل ۲: شبه کد بروزرسانی موقعیت کاربر



شکل ۳: مولفه مدیریت تحرک کاربر به منظور مقابله با چالش تحرک در دستگاه متحرک و گره مه.

پردازنده قرار دارند (خط ۶)، می‌باشد. زمان اجرای کلیه درخواست‌ها مطابق معادله ۱ محاسبه می‌گردد (خط ۷).

$$TM_{(i,t)} = TP_i + TQ_{(i,t)} \quad (1)$$

در معادله ۱، $TM_{(i,t)}$ زمان لازم برای اجرای کلیه درخواست‌ها در گره i در بازه زمانی t ام با طول T_2 را نشان می‌دهد. TP_i برابر با زمان اتمام کار درخواست‌های در حال اجرا در گره i در ابتدای بازه t می‌باشد (خط ۵). توجه شود که با توجه به توان پردازشی گره i ، ممکن است چندین درخواست به صورت همزمان اجرا شوند. $TQ_{(i,t)}$ که مطابق معادله ۲ محاسبه می‌گردد، مدت زمان موردنیاز برای پردازش درخواست‌های صف در بازه زمانی t در گره i را نشان می‌دهد (خط ۶).

$$TQ_{(i,t)} = \frac{\sum_{j=1}^n L_{(j,i,t)}}{P_i} \quad (2)$$

که $L_{(j,i,t)}$ طول درخواست j ام در صف گره i در بازه زمانی t و P_i برابر با توان پردازشی گره i است. همان‌طور که در معادله ۲ مشاهده می‌شود، $TQ_{(i,t)}$ به صورت مجموع طول درخواست‌های موجود در صف گره i تقسیم بر توان پردازشی گره i محاسبه می‌شود.

سپس متوسط مهلت باقی‌مانده درخواست‌های حساس به تاخیر محاسبه می‌شود (خط ۸). معادله ۳ نحوه محاسبه متوسط مهلت باقی‌مانده درخواست‌های حساس به تاخیر موجود در صف را نشان می‌دهد.

$$TD_{(i,t)} = \frac{\sum_{j=1}^m (AD_{(j,i,t)} - DQ_{(j,i,t)})}{m} \quad (3)$$

در معادله ۳، متوسط مهلت تقریبی باقی‌مانده درخواست‌های حساس به تاخیر در صف گره i در بازه زمانی t با $TD_{(i,t)}$ نمایش داده می‌شود. $DQ_{(j,i,t)}$ زمان سپری شده درخواست حساس به تاخیر j و $AD_{(j,i,t)}$ مهلت درخواست j موجود در صف گره i در بازه زمانی t می‌باشد. در این معادله m تعداد درخواست‌های حساس به تاخیر موجود در صف گره i در ابتدای بازه زمانی t می‌باشد. سپس ازدحام در گره i بررسی می‌گردد (خطوط ۹ تا ۱۷). این کار با مقایسه مدت زمان لازم برای اجرای کلیه درخواست‌ها (TM) و متوسط مهلت تقریبی باقی‌مانده درخواست‌های حساس به تاخیر در صف گره i صورت می‌پذیرد. بدین منظور دو پارامتر W_1 و W_2 تعریف شده است که وزن مهلت باقی‌مانده درخواست‌های حساس به تاخیر بوده و بر اساس میزان نسبت ورود به خروج کاربران محاسبه می‌گردد. همان‌طور که در شکل ۵ مشاهده می‌شود، این دو پارامترهای ورودی الگوریتم TICC می‌باشند که در ابتدای هر بازه

مطابق شکل ۲ مولفه مدیریت تحرک شامل توابع $Location_Update()$ و $Mobile_Parent_Information_Update()$ می‌باشد، که در گره موبایل اجرا می‌شوند. همچنین تابع $Queue_Update$ در مولفه کنترل‌کننده تحرک در گره مه اجرا می‌شود. دستگاه‌های موبایل و مه از طریق تابع $Message_Sender()$ به تبادل اطلاعات می‌پردازند. شکل ۳ جزئیات مولفه مدیریت تحرک کاربر را نمایش می‌دهد.

۳-۲- روش پیشنهادی جهت تخصیص منابع

در هر گره مه، الگوریتم تخصیص منابع با توجه به وضعیت منابع، تحرک کاربر و اولویت درخواست‌ها، در مورد جای‌گیری درخواست‌ها تصمیم‌گیری می‌کند. درخواست‌ها می‌توانند پس از دریافت در گره مه در سه مکان گره مه، گره ابر و گره‌های مه همجوار پردازش شوند. شکل ۴ مولفه تخصیص منابع در الگوریتم پیشنهادی که $TICC^{45}$ نامیده می‌شود را نمایش می‌دهد. در الگوریتم پیشنهادی، هر گره مه با توجه به معیار ازدحام، در ابتدای هر بازه زمانی به طول T_2 که $T_2 \ll T_1$ و یا هر زمان که درخواستی از گره مه مجاور ارسال شود، در مورد مکان رسیدگی به درخواست‌ها تصمیم‌گیری می‌کند. پیش‌بینی رخداد ازدحام در گره مه با مقایسه متوسط مهلت باقی‌مانده درخواست‌های حساس به تاخیر و زمان مورد نیاز برای اجرای درخواست‌های موجود در صف گره مه که در ادامه شرح داده می‌شود، صورت می‌پذیرد. شبه کد الگوریتم پیشنهادی در شکل ۵ نشان داده شده است.

این الگوریتم در دو حالت فعال می‌شود. حالت اول، در صورت خروج کاربر از محدوده گره مه فعلی، گره مه همسایه پیامی دریافت می‌کند که نشان می‌دهد درخواست‌های کاربر متحرک به این گره مه منتقل می‌شوند و باید در این گره مه زمانبندی شوند. اگر مجموع زمان لازم برای انجام درخواست‌های موجود در صف گره مه دریافت‌کننده از γ بیشتر باشد، فقط درخواست‌های حساس به تاخیر به صف گره مه دریافت‌کننده منتقل می‌شود و سایر درخواست‌ها به ابر ارسال می‌شود، در غیر این صورت، تمامی درخواست‌های منتقل شده از گره مه مجاور به صف گره مه دریافت‌کننده منتقل می‌شوند (خطوط ۲-۳). در مولفه "تخصیص منابع در گره مه همجوار" شکل ۴ جزئیات آن نمایش داده شده است.

حالت دوم، فرا رسیدن بازه زمانی با طول T_2 برای اجرای الگوریتم زمانبندی (خط ۴)، در این حالت ابتدا زمان لازم جهت پردازش کلیه درخواست‌های گره مه محاسبه می‌شود که برابر مجموع زمان لازم برای پردازش درخواست‌هایی که در پردازنده در حال پردازش هستند (خط ۵) و زمان لازم برای اجرای درخواست‌هایی که در صف انتظار

نسبت ورود به خروج، احتمال رخداد ازدحام افزایش می‌یابد و بهتر است بخشی از صف گره مه برای دریافت درخواست‌های بیشتر در آینده خالی شود. به این منظور، پارامترهای W_1 و W_2 مقادیر کمتری می‌گیرد.

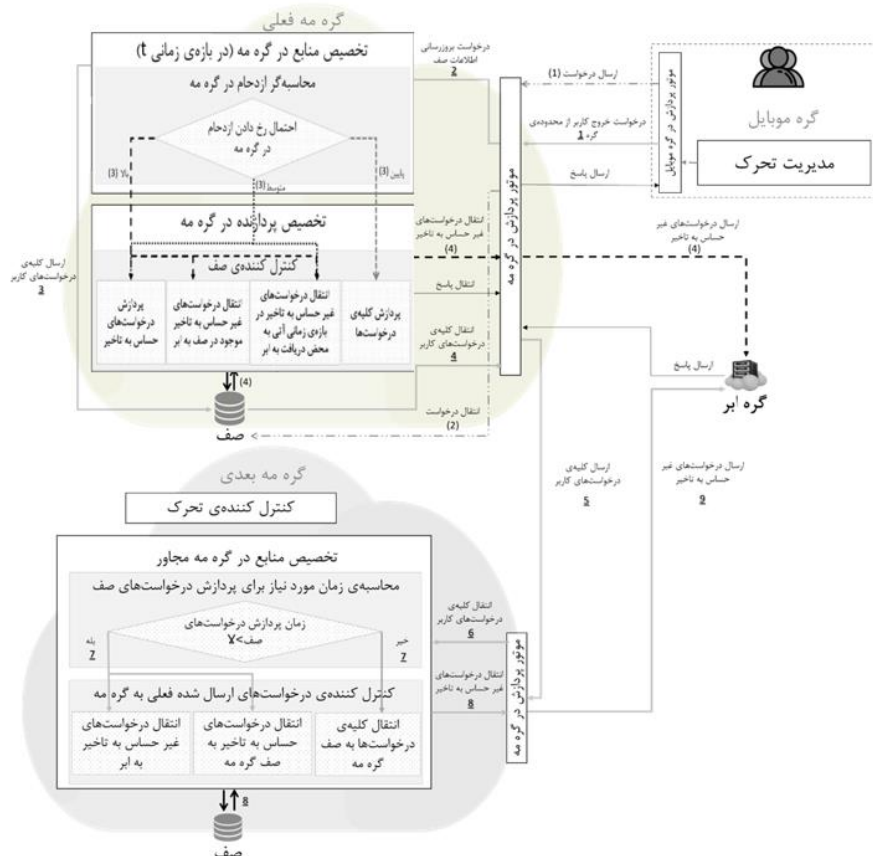
طبق شکل ۴ و شکل ۵ (خطوط ۹-۱۷)، با مقایسه‌ی مقادیر TM و TD سه وضعیت زیر رخ می‌دهد:

۱- $TM < W_1 \times TD$ (خط ۹): در این وضعیت زمان اجرای درخواست‌های موجود در گره مه کمتر از حد پایین متوسط مهلت باقی‌مانده درخواست‌های حساس به تاخیر است. لذا، درخواست‌های حساس به تاخیر فرصت کافی برای اجرا دارند. در حقیقت گره مه توانایی پاسخ به درخواست‌ها را دارد و ازدحامی رخ نداده است. بنابراین نیازی به ارسال درخواست‌ها به گره ابر نیست (خط ۱۰).

قبل از فراخوانی تنظیم می‌شوند بطوریکه $0 < W_1 < W_2 < 1$. نسبت ورود به خروج کاربران (R) بر اساس معادله ۴ محاسبه می‌گردد.

$$R_{(i,t)} = \begin{cases} 0 & \text{if } O_{(i,t)}=0 \ \& \ I_{(i,t)}=0 \\ I_{(i,t)} & \text{if } O_{(i,t)}=0 \ \& \ I_{(i,t)}>0 \\ \left(\frac{I_{(i,t)}}{O_{(i,t)}} \right) & \text{otherwise} \end{cases} \quad (4)$$

هر چه تعداد کاربران وارد شده به گره مه نسبت به تعداد کاربران خارج شده از گره کمتر باشد، مقدار W_1 و W_2 بیشتر است به عبارت دیگر احتمال انتقال درخواست‌ها به ابر کمتر است و بالعکس. با افزایش نرخ ورود کاربران به محدوده گره مه نسبت به خروج آن‌ها، امکان رخداد ازدحام در بازه زمانی آتی بیشتر می‌شود. بنابراین زمانی که این نسبت کم باشد می‌توان فرض کرد که متوسط زمان اتمام کلیه درخواست‌ها در صف گره مه به اندازه‌ی است که نرخ لغو درخواست‌ها به دلیل نقض مهلت زمانی بالا نمی‌باشد. اما با افزایش



شکل ۴: مولفه تخصیص منابع در الگوریتم TICC.

```

Algorithm: TICC // Run on each fog node on interval time t or when receiving message from fog neighbor
Input:
W1, W2, Y: decision parameters
leave_area: the boolean parameter that is True when a message was received from the neighbor fog
message: received message from fog neighbor if the user leaves the fog neighbor's area
Pi: MIPS capacity of fog node i
1 Begin
2 If (leave_area) //If the user leaves the fog node area
3   receive-and-process_message_from_neighbor_fog (message, requests, Y);
4 Else
5   TP ← get_total_length_of_in_process_requests() / pi; //Get time required to process running requests
6   TQ ← get_total_length_of_queue_requests() / pi; // Calculate time required to process queue requests
7   TM ← TP + TQ; //Calculate required time to process all requests in the fog node
8   TD ← get_remaining_time_of_deadline(); //Calculate the average remaining time of delay-sensitive
           requests' deadlines
9   If (TM < W1 × TD)
10    process_requests_in_fog(); //All requests are processed in fog
11   Else if (W1 × TD ≤ TM and TM ≤ W2 × TD)
12    process_delay_requests_in_fog(); //All requests are processed in fog
13    send_new_insensitive_requests_to_cloud_in_next_interval(); //Delay-insensitive requests at the
           beginning of the next interval are sent to the cloud
14   Else if (TM > W2 × TD)
15    process_delay_sensitive_requests_in_fog(); // Delay-sensitive requests are processed in fog
16    send_delay_insensitive_queue_requests_to_cloud(); // Delay-insensitive requests of the queue are
           processed in cloud
17    send_new_insensitive_requests_to_cloud_in_next_interval(); // New delay-insensitive requests at the
           beginning of the next interval are sent to the cloud
16 End
    
```

شکل ۵: شبه‌کد الگوریتم تخصیص منابع TICC

غیر حساس به تاخیر در بازه‌ی زمانی آتی (خط ۱۷)، درخواست‌های غیر حساس به تاخیر موجود در صف گره مه به گره ابر ارسال می‌شوند (خط ۱۶).

۴- شبیه‌سازی

تنظیمات اعمال شده در این مقاله با توجه به مقاله پایه [۲۳] و تنظیمات پیش‌فرض iFogSim انتخاب شده است.

۴-۱- شبیه‌سازی تحرک کاربران

در اینجا سناریویی برای شبیه‌سازی تحرک کاربران در نظر گرفته شده است که گسترش یافته سناریوی مطرح شده در [۲۳] می‌باشد. در این سناریو، مطابق شکل ۶ محدوده تحت پوشش با اندازه ۱۴۴۰۰×۱۴۴۰۰ مترمربع، شامل ۹ ناحیه با اندازه‌های یکسان تعریف شده است. در هر ناحیه یک گره مه در نظر گرفته شده است که تنها به کاربران متحرک آن ناحیه سرویس می‌دهد.

در ابتدای شبیه‌سازی، هر گره موبایل به صورت تصادفی در یک ناحیه قرار می‌گیرد. در طول روند شبیه‌سازی، گره‌های موبایل با تغییر موقعیت جغرافیایی در میان ناحیه تحت پوشش گره‌های مه حرکت

۲- $W_1 \times TD \leq TM \leq W_2 \times TD$ (خط ۱۱): در این وضعیت درخواست‌های حساس به تاخیر در شرف لغو شدن هستند و احتمال ازدحام درخواست‌ها تقریباً زیاد می‌باشد. به این دلیل که زمان موردنیاز جهت پردازش کلیه درخواست‌ها در گره مه، از حد پایین متوسط مهلت باقی‌مانده درخواست‌ها، بیشتر و از حد بالای آن کمتر است. در این حالت گره مه هنوز به وضعیت ازدحام کامل نرسیده است. بنابراین جهت اجتناب از ازدحام، از پذیرش درخواست‌های غیر حساس به تاخیر در بازه‌های زمانی آتی، جلوگیری شده و این درخواست‌ها به ابر منتقل می‌شوند. (خط ۱۳-۱۲)

۳- $W_2 \times TD < TM$ (خط ۱۴): در این وضعیت زمان اجرای درخواست‌های موجود در گره مه از حد بالای متوسط مهلت زمانی باقی‌مانده درخواست‌های حساس به تاخیر بیشتر است. بنابراین امکان لغو درخواست‌های جدید و موجود در صف بسیار زیاد است و گره تقریباً در وضعیت ازدحام قرار گرفته است. در این حالت به منظور جلوگیری از لغو درخواست‌های حساس به تاخیر، علاوه بر اجتناب از پذیرش درخواست‌های

درخواست‌ها در هر سنسور (در حقیقت بازه زمانی ایجاد یک درخواست جدید) به صورت تصادفی بین ۱۰ تا ۱۵ میلی ثانیه انتخاب می‌شود. بازه زمانی بروزسانی موقعیت کاربر (T_1) نیز ۱ ثانیه در نظر گرفته شده است. سرعت حرکت کاربران نیز به صورت تصادفی از بازه ۰ تا ۱۵ کیلومتر بر ساعت انتخاب شده است [۳۷]. محدوده سرعت با توجه به میانگین سرعت افراد پیاده و دوچرخه‌سوار در نظر گرفته شده است.

به منظور ایجاد ازدحام در یک گره خاص (در این مقاله گره ۵) و بررسی عملکرد روش پیشنهادی در شرایط ازدحام، در طول شبیه‌سازی، جهت حرکت کاربران با احتمال ۷۰ درصد به سمت آن گره تنظیم شده است. مهلت زمانی اجرای درخواست‌های حساس به تاخیر به صورت تصادفی از بازه [۲۵۰، ۷۵۰] میلی ثانیه انتخاب می‌شود [۳۸].

همان‌طور که در بخش قبل گفته شد، جهت عملکرد بهتر در مواجهه با شرایط مختلف صف گره مه و تحرک کاربران از دو پارامتر W_1 و W_2 استفاده شده است. در این مقاله مقادیر W_1 و W_2 بر اساس نسبت ورود به خروج (R) و مطابق جدول ۳ تعیین شده‌اند.

جدول ۳: مقادیر پارامترهای W_1 و W_2 بر اساس R.

W_2	W_1	R
۰.۸۵	۰.۵۵	۲-۰
۰.۸	۰.۵	۵-۲
۰.۷	۰.۴۵	>۵

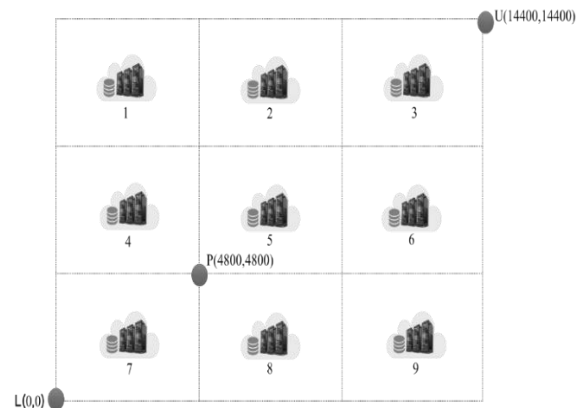
در این پژوهش، درخواست‌های کاربر موبایل هنگام خروج او از محدوده گره مه فعلی به گره مجاور ارسال می‌شوند. در گره مه مجاور از آستانه‌ی λ برای تصمیم‌گیری جهت ارسال درخواست‌های غیر حساس به تاخیر به ابر و یا پذیرش کلیه‌ی درخواست‌ها در گره مه مقصد مورد استفاده قرار می‌گیرد. جهت تعیین مقدار آستانه λ ، در طی ۱۰ بار تکرار شبیه‌سازی تعداد درخواست‌های لغو شده و مقادیر TQ ثبت شده‌اند. میانگین مقادیر TQ زمانی که تعداد درخواست‌های لغو شده در گره مه بین ۱۰۰ و ۱۵۰ باشد، به عنوان آستانه λ در نظر گرفته شده است که مقدار آن برابر ۹.۴۹ می‌باشد.

۳-۴- ارزیابی روش پیشنهادی

جهت ارزیابی روش پیشنهادی سه دسته آزمایش صورت گرفته است. در آزمایش اول به بررسی تاثیر انتقال درخواست‌ها به گره مه همجوار در صورت تحرک کاربر و پیش‌بینی خروج او از محدوده گره مه فعلی در روش پیشنهادی می‌پردازیم.

شکل ۷ نمایش‌دهنده درصد درخواست‌های رد شده در دو حالت ارسال و عدم ارسال درخواست‌ها به گره مه همجوار در شرایط تحرک

می‌کنند. برای بروزسانی موقعیت یک گره موبایل، از دو مشخصه‌ی جهت و سرعت حرکت استفاده می‌شود. به منظور بروزسانی موقعیت، سرعت و جهت حرکت گره موبایل، در بازه‌های زمانی مشخص به طول T_1 به صورت تصادفی برای بازه بعد انتخاب می‌شود.



شکل ۶: محدوده‌ی جغرافیایی در نظر گرفته شده برای شبیه‌سازی.

۴-۲- تنظیمات شبیه‌سازی

جدول ۲ شامل اطلاعات قدرت پردازشی و ذخیره‌سازی گره‌های مه و ابر می‌باشد. این اطلاعات شامل MIPS، میزان حافظه 4^6 ، پهنای باند کانال ارتباطی با لایه‌ی بالا 4^7 و پهنای باند کانال ارتباطی با لایه‌ی پایین 4^8 و پهنای باند کانال ارتباطی با گره‌های مه مجاور 4^9 می‌باشد [۳۴-۳۵].

جدول ۲: مشخصات گره‌های موجود در شبیه‌سازی [۳۴، ۳۵].

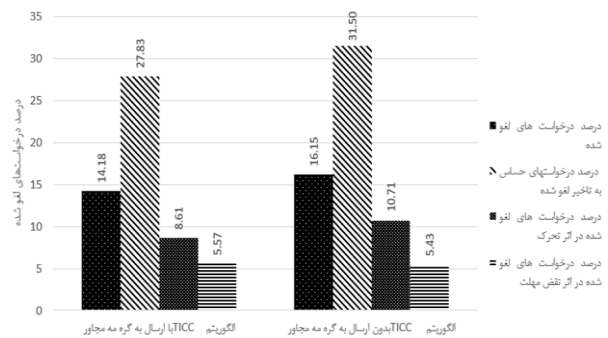
نوع گره	MIPS	حافظه (MB)	پهنای باند کانال ارتباطی با لایه‌ی بالا (Kbps)	پهنای باند کانال ارتباطی با لایه‌ی پایین (Kbps)	پهنای باند کانال ارتباطی با گره‌های مه مجاور (Kbps)
مه	۲۸۰۰	۴۰۰۰	۱۰۰۰۰	۱۰۰۰۰	۱۰۰۰۰۰
ابر	۴۴۸۰۰	۴۰۰۰	۱۰۰	۱۰۰۰۰	-
واسط	۲۸۰۰	۴۰۰۰	۱۰۰۰۰	۱۰۰۰۰	-
موبایل	۱۰۰۰	۱۰۰۰	۲۷۰	۱۰۰۰۰	-

در این پژوهش، تاخیر ارتباطی میان گره‌های ابر و واسط، تاخیر ارتباطی میان گره‌های واسط و مه، تاخیر ارتباطی در بین گره‌های مه و تاخیر ارتباطی میان گره‌های مه و موبایل، به ترتیب ۱۰۰ و ۴ و ۲۰ و ۲ میلی ثانیه در نظر گرفته شده است [۳۵، ۳۶]. هر کاربر موبایل ۵۰۰ درخواست ارسال می‌کند و در مجموع ۲۷۰۰۰ درخواست توسط کاربران ارسال می‌شود. فواصل زمانی بین ارسال

الگوریتم پایه به نام‌های FCFS و Cloud-only که در بسیاری از روش‌های ارائه شده جدید از جمله [۹] و [۲۹] نیز مبنای مقایسه بوده‌اند و الگوریتم مطرح شده در [۲۳] با عنوان Delay-priority که تحرک و ازدحام درخواست‌ها را در نظر گرفته است، مقایسه شده است. در الگوریتم FCFS درخواست‌ها به ترتیب ورود به صف گره می‌شوند و بدون هیچ اولویتی اجرا می‌شوند. در هر بازه‌ی زمانی با توجه به توان پردازشی یک گره می‌شوند. در این پژوهش تنها ظرفیت CPU در نظر گرفته شده است)، چندین درخواست می‌توانند به صورت همزمان اجرا شوند. در الگوریتم Cloud-only تمام درخواست‌ها به گره ابر منتقل می‌شوند و هیچ درخواستی در گره‌های میز پردازش نخواهد شد. در الگوریتم Delay-priority درخواست‌ها مشابه صف FCFS اجرا شده و در صورت نبود ظرفیت خالی، درخواست‌های غیر حساس به تاخیر به ابر منتقل می‌شوند. در ادامه کارایی الگوریتم پیشنهادی بر اساس معیارهای درصد درخواست‌های لغو شده، زمان پاسخ، مصرف ترافیک شبکه و مصرف انرژی با الگوریتم‌های مذکور مقایسه می‌شود.

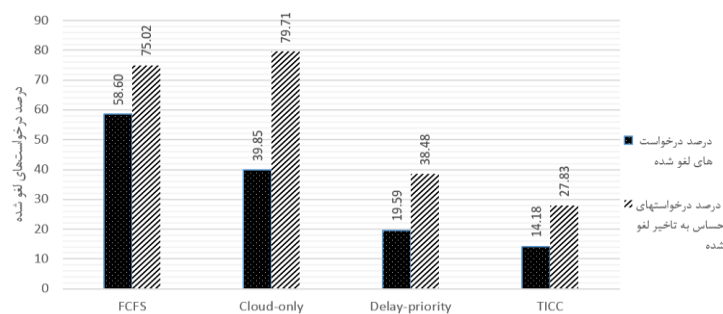
در شکل ۸، درصد درخواست‌های لغو شده و درصد درخواست‌های اجرا نشده‌ی حساس به تاخیر، نمایش داده شده است. نتایج حاصل میانگین ۱۰ بار اجرای الگوریتم‌ها می‌باشد. درصد درخواست‌های لغو شده و درصد درخواست‌های حساس به تاخیر لغو شده در الگوریتم TICC کمتر از سه الگوریتم دیگر می‌باشد. دلیل اصلی این امر، انتقال درخواست‌های غیر حساس به تاخیر با توجه به وضعیت ازدحام در صف درخواست‌ها می‌باشد. با این انتقال تعداد درخواست‌های موجود در صف کاهش می‌یابد، بنابراین درخواست‌های حساس به تاخیر در زمان کمتر و با سرعت بالاتری اجرا خواهند شد. همچنین نقش پیش‌بینی خروج کاربر از محدوده‌ی گره و ارسال درخواست‌های اجرا نشده به گره میز همجوار را نمی‌توان نادیده گرفت. این امر موجب عدم حذف درخواست‌ها در شرایط تحرک و اجرای هر چه بیشتر درخواست‌های کاربران می‌گردد.

کاربر و پیش‌بینی خروج می‌باشد. مطابق شکل ۷ با انتقال درخواست‌های کاربر به گره میز مجاور (در صورت پیش‌بینی خروج کاربر از گره میز فعلی در بازه زمانی آتی و پیش از خروج وی از محدوده‌ی گره میز) در الگوریتم TICC نرخ پذیرش درخواست‌ها به میزان ۱.۹۷ درصد افزایش و درصد درخواست‌های لغو شده حساس به تاخیر به میزان ۳.۶۷ کاهش یافته است. دلیل افزایش نرخ پذیرش درخواست‌ها این است که بسیاری از درخواست‌هایی که به دلیل خروج کاربر از محدوده گره میز فعلی لغو می‌شدند، در این روش به گره میز مجاور منتقل شده و امکان اجرای آنها در این گره فراهم می‌شود. ارسال درخواست‌ها پیش از خروج کاربر در مرز دو گره میز سبب می‌شود تا درخواست‌ها سریع‌تر دریافت شده و از زمان اتصال کاربر به گره میز بعدی تا زمان رسیدگی به درخواست‌ها مدت زمان کمتری سپری شود. همچنین همان‌طور که در شکل ۷ مشاهده می‌شود، درصد درخواست‌های لغو شده در اثر تحرک در شرایط ارسال درخواست‌ها به گره میز مجاور کمتر از شرایط عدم ارسال درخواست‌ها به گره میز مجاور می‌باشد. بنابراین تاثیر تحرک کاربران در لغو درخواست‌ها کاهش می‌یابد.

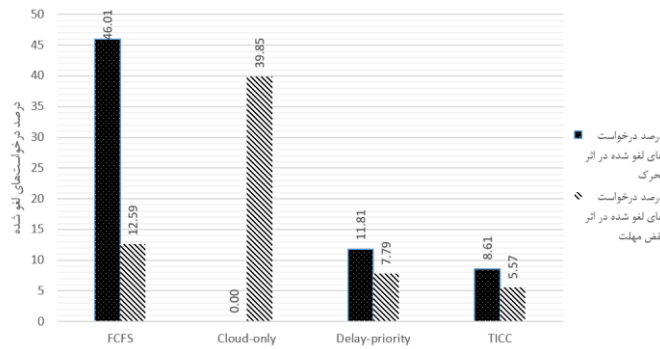


شکل ۷: مقایسه کارایی الگوریتم TICC در دو حالت ارسال و عدم ارسال درخواست‌ها به گره میز همجوار.

در آزمایش دوم، به ارزیابی و مقایسه کارایی روش پیشنهادی نسبت به سایر روش‌های موجود می‌پردازیم. الگوریتم پیشنهادی با دو

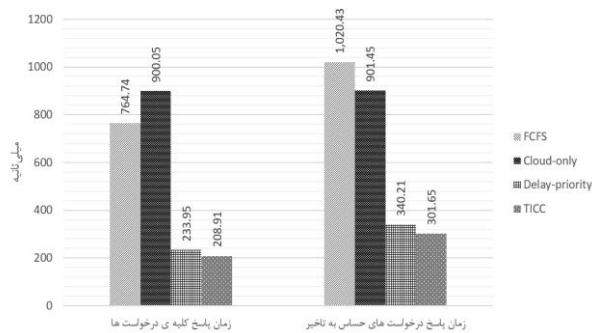


شکل ۸: درصد درخواست‌های لغو شده به تفکیک نوع درخواست در چهار الگوریتم FCFS، Delay-priority، Cloud-only و TICC.



شکل ۹: درصد درخواست‌های لغو شده به تفکیک عوامل لغو در چهار الگوریتم FCFS، Delay-priority، Cloud-only و TICC.

زمان پاسخ درخواست‌های حساس به تاخیر نسبت به متوسط زمان پاسخ کلیه درخواست‌ها بیشتر باشد.



شکل ۱۰: متوسط زمان پاسخ درخواست‌ها با استفاده از چهار الگوریتم FCFS، Delay-priority، Cloud-only و TICC.

در الگوریتم FCFS انتظار درخواست‌ها در صف گره مه، موجب افزایش زمان پاسخ می‌شود. در الگوریتم Cloud-only کلیه درخواست‌ها، به ابر منتقل می‌شوند. بنابراین زمان پاسخ این الگوریتم به خصوص برای درخواست‌های حساس به تاخیر بالاتر از سایر الگوریتم‌ها است. همان‌طور که مشاهده می‌شود متوسط زمان پاسخ در الگوریتم TICC کمتر از سایر الگوریتم‌ها است. زمان پاسخ درخواست‌های حساس به تاخیر در TICC با وجود پذیرش تعداد درخواست‌های بیشتر در مقایسه با الگوریتم Delay-priority حدوداً ۳۸.۵۶ میلی‌ثانیه (۱۰ درصد) و در مقایسه با FCFS در حدود ۷۱۸ میلی‌ثانیه (۷۰ درصد) کاهش یافته است. در الگوریتم TICC با انتقال درخواست‌های غیر حساس به تاخیر به ابر قبل از رخداد ازدحام در گره مه طول صف گره کاهش یافته و در نتیجه زمان پاسخ به درخواست‌های حساس به تاخیر نیز کاهش می‌یابد.

شکل ۱۱ میزان استفاده از ترافیک شبکه نسبت به تعداد درخواست‌های اجرا شده در چهار الگوریتم، FCFS، Delay-priority، Cloud-only و TICC را نشان می‌دهد. مطابق شکل ۱۱، روش Cloud-only بیشترین میزان استفاده از ترافیک شبکه را دارد. این

شکل ۸ نشان می‌دهد که الگوریتم TICC در مقایسه با الگوریتم Delay-priority درخواست‌های لغو شده و درخواست‌های حساس به تاخیر لغو شده را به ترتیب ۶.۲ درصد و ۲.۲۲ درصد کاهش دهد.

در شکل ۹ درصد درخواست‌های لغو شده به تفکیک عامل لغو (نقض مهلت/تحرک) نمایش داده شده است. همان‌طور که در این شکل مشاهده می‌شود، درصد درخواست‌های لغو شده در اثر مهلت در الگوریتم Cloud-only بیشتر از سایر روش‌ها می‌باشد. از آنجایی که در Cloud-only فرض شده تمامی گره‌های موبایل در محدوده‌ی گره‌های مه متصل به گره ابر حرکت می‌کنند و از این محدوده خارج نمی‌شوند، در این الگوریتم عامل تحرک در لغو درخواست‌های کاربران تأثیری ندارد. در سایر الگوریتم‌ها بیشتر درخواست‌های کاربران در اثر تحرک لغو شده است. در الگوریتم FCFS درصد درخواست‌های لغو شده بیشتر از سایر روش‌ها است. در این الگوریتم انتظار درخواست‌های حساس به تاخیر در صف گره مه با ظرفیت پردازشی محدود موجب افزایش زمان پاسخ و در نتیجه عدم اتمام وظیفه در مهلت مقرر می‌گردد. همچنین درصد درخواست‌های لغو شده در اثر تحرک در الگوریتم پیشنهادی کمتر از سایر الگوریتم‌ها می‌باشد. الگوریتم پیشنهادی به دلیل پیش‌بینی گره مه بعدی، نسبت به FCFS و Delay-priority درصد درخواست‌های لغو شده در اثر تحرک را به ترتیب ۳۷.۴ و ۳.۲ درصد کاهش داده است. بعلاوه به دلیل تصمیم‌گیری برای ارسال درخواست‌های غیر حساس به تاخیر به ابر با توجه مهلت انجام درخواست‌های حساس به تاخیر، درصد لغو درخواست‌های حساس به تاخیر را نسبت به روش FCFS و Delay-priority به ترتیب ۳.۹۴ و ۲.۲۲ درصد کاهش داده است.

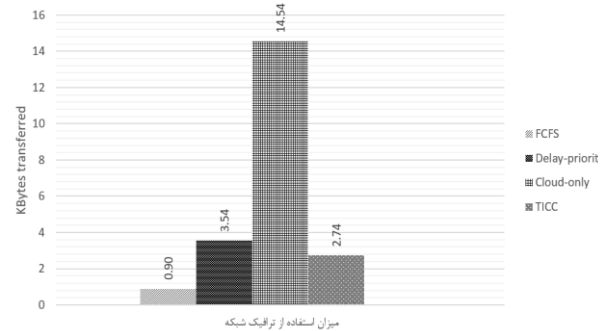
شکل ۱۰ متوسط زمان پاسخ درخواست‌ها در روش پیشنهادی را با سه روش دیگر مقایسه می‌کند. زمان پاسخ، اختلاف زمان از لحظه ارسال درخواست توسط سنسور تا لحظه دریافت پاسخ در عملگر می‌باشد. تفاوت در وظایف درخواست‌های حساس به تاخیر با وظایف درخواست‌های غیر حساس به تاخیر، موجب شده است تا متوسط

توجه به تحرک کاربران و وضعیت ازدحام درخواستها در صف گره‌های مه بررسی شده است. در روش پیشنهادی جهت تخصیص منابع با توجه به تحرک کاربران، وضعیت صف یک گره مه (ازدحام گره) و نوع درخواست، در مورد مکان و زمان رسیدگی به درخواستها (گره مه فعلی، گره مه مجاور یا ابر) تصمیم‌گیری می‌شود. به منظور ارزیابی کارایی روش پیشنهادی سه دسته آزمایش صورت گرفته است. در دسته‌ی اول تاثیر انتقال درخواستها به گره‌های مه همجوار در صورت خروج کاربران از محدوده‌ی گره مه فعلی، بررسی شده و نتایج حاصل نشان می‌دهد که انتقال درخواستها به گره مجاور منجر به کاهش لغو درخواستها به میزان ۱.۹۷ درصد می‌شود. در سری دوم آزمایشها بر اساس معیارهای درصد درخواستهای لغو شده، زمان پاسخ و میزان استفاده از ترافیک شبکه، کارایی الگوریتم پیشنهادی با روشهای پایه FCFS و Cloud-only و نیز روش Delay-priority مقایسه شده است. نتایج آزمایشهای انجام گرفته نشان می‌دهد که الگوریتم پیشنهادی نسبت به سایر الگوریتمهای نامبرده در مواجهه با تحرک کاربران، ازدحام و کلاسهای مختلف برنامه‌های کاربردی با نیازهای متفاوت، نرخ پذیرش بالاتری به ویژه برای درخواستهای حساس به تاخیر دارد. در آزمایش سوم نیز نشان داده شد که روش پیشنهادی در شرایط ازدحام بهتر از سایر الگوریتمها عمل می‌کند و نرخ پذیرش بالاتری دارد.

در این مقاله یک تکنیک ساده جهت حرکت کاربران ارائه شده است در حالی که سناریوهای تحرک کاربران می‌تواند بسیار پیچیده باشد. به عنوان کار آتی می‌توان فرمولهای پیشنهادی برای پیش‌بینی ازدحام را با توجه به الگوهای حرکت کاربران بهبود داد. برای نمونه می‌توان از تکنیکهای داده‌کاوی برای تشخیص الگوهای حرکتی کاربران استفاده کرد. همچنین با استفاده از الگوریتمهای ابتکاری می‌توان معیارهای تصمیم‌گیری دقیق‌تری را در نظر گرفت که سربار محاسباتی کمی داشته باشد و عملکرد الگوریتم را بهبود دهد. همچنین انتخاب خودکار روش زمانبندی در گره مه از میان چندین روش با توجه به الگوهای رفتاری و تحرک کاربران و نوع درخواستهای ارسالی می‌تواند به عنوان کار آتی مورد توجه قرار گیرد.

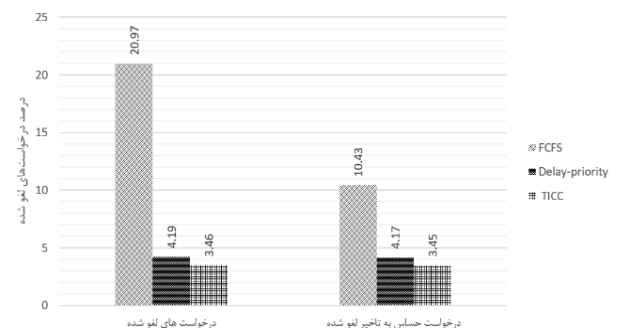
در این مقاله، وظایف درخواستهای برنامه کاربردی مورد استفاده در دستگاه موبایل به هم وابسته هستند و باید به ترتیب انجام شوند در صورتی که می‌توان برنامه‌های کاربردی که وظایف درخواستهای آنها از هم مستقل باشند را نیز لحاظ نمود. از سوی دیگر در این مقاله گره‌های مه همگن فرض شده است.

مقدار در الگوریتم TICC حدود ۲.۷۴ می‌باشد که نسبت به روشهای Delay-priority و Cloud-only کمتر است. الگوریتم FCFS نیز به دلیل عدم انتقال درخواستها به ابر کمترین میزان مصرف شبکه را دارد.



شکل ۱۱: میزان استفاده از ترافیک شبکه نسبت به تعداد درخواستهای اجرا شده در چهار الگوریتم FCFS، Delay-priority، Cloud-only و TICC

در آزمایش سوم، به ارزیابی کارایی روش پیشنهادی در شرایط ازدحام پرداخته شده است. در این شبیه‌سازی به منظور ایجاد ازدحام در یک گره مه، کاربران با احتمال ۷۰ درصد در جهت گره مه انتخابی حرکت می‌کنند. درصد درخواستهای لغو شده و درصد درخواست حساس به تاخیر لغو شده در گره انتخابی برای ازدحام (گره ۵)، در شکل ۱۲ نشان داده شده است.



شکل ۱۲: درصد درخواستهای لغو شده در گره دارای ازدحام با استفاده از سه الگوریتم FCFS، Delay-priority و TICC

نتایج شکل ۱۲ نشان می‌دهد در گره‌ی که با شرایط ازدحام مواجه شده است، با استفاده از الگوریتم TICC درخواستهای کمتری لغو می‌شوند. همچنین مطابق این نمودار در الگوریتم FCFS درصد قابل ملاحظه‌ای از درخواستها (۲۰.۹۷٪) لغو می‌شود. روش پیشنهادی به ترتیب ۰.۷۳ و ۱۷.۵۱ درصد نسبت به FCFS و Delay-priority تعداد درخواستهای لغو شده را کاهش داده است.

۵- نتیجه‌گیری

در این مقاله، مسئله زمانبندی و تخصیص منابع در رایانش مه با

- [19] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F.R. Yu and Z. Han, "Computing Resource Allocation in Three-tier IoT Fog Networks: A Joint Optimization Approach Combining Stackelberg Game and Matching," *IEEE Internet of Things Journal*, Vol.4, No. 5, pp. 1204-1215, 2017.
- [20] R. Deng, R. Lu, C. Lai, T.H. Luan and H. Liang, "Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption," *IEEE Internet of Things Journal*, Vol.6, No.3, pp. 1171-1181, 2016.
- [21] R. Deng, R. Lu, C. Lai and T.H. Luan, "Towards Power Consumption-Delay Tradeoff by Workload Allocation in Cloud-Fog Computing". In 2015 IEEE International Conference on Communications (ICC), London, UK, pp. 3909-3914, 2015.
- [22] L. Liu, D. Qi, N. Zhou, and Y. Wu, "A Task Scheduling Algorithm Based on Classification Mining in Fog Computing Environment," *Wireless Communications and Mobile Computing*, Vol.2018, pp. 1-11, 2018.
- [23] L.F. Bittencourt, J.D. 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, 2017.
- [24] M. Verma, N. Bhardwaj, and A.K. Yadav, "Real Time Efficient Scheduling Algorithm for Load Balancing in Fog Computing Environment," *I.J. Information Technology and Computer Science*, Vol.8, No.4, pp. 1-10, 2016.
- [25] R. Mahmud, S. Pallewatta, M. Goudarzi and R. Buyya, "Ifogsim2: An extended ifogsim simulator for mobility, clustering, and microservice management in edge and fog computing environments," *Journal of Systems and Software*, Vol.190, pp. 1-17, 2022.
- [26] Y. Jie, M. Li, C. Guo and L. Chen, "Game-theoretic online resource allocation scheme on fog computing for mobile multimedia users," *China Communications*, Vol.16, No.3, pp.22-31, 2019.
- [27] D. Wang, Z. Liu, X. Wang and Y. Lan, "Mobility-aware task offloading and migration schemes in fog computing networks," *IEEE Access*, Vol.7, pp.43356-43368, 2019.
- [۲۸] سمیرا انصاری مقدم، سمیرا نوفرستی، مه‌ری رجایی، "ارائه یک روش کارا برای تخصیص منابع در رایانش مه با در نظر گرفتن شرایط ازدحام درخواست‌ها"، نشریه مهندسی برق و مهندسی کامپیوتر ایران، جلد ۱۹، شماره ۱، ۱۴۰۰.
- [29] W. Liu, C. Li, A. Zhen, Z. Zheng, Z. Zhang, and Y. Xiao, "Fog Computing Resource-Scheduling Strategy in IoT Based on Artificial Bee Colony Algorithm," *Electronics*, Vol. 12, N. 7, pp. 1-24, 2023.
- [30] O. Sharma, G. Rathee, C. A. Kerrache, and J Herrera-Tapia, "Two-Stage Optimal Task Scheduling for Smart Home Environment Using Fog Computing Infrastructures," *Applied Sciences*, Vol. 13, N. 5, PP. 1-15, 2023.
- [31] S. Iftikhar, M. M. M. Ahmad, S. Tuli, D. Chowdhury, M. Xu, S. Singh Gill, S. Uhlig, "HunterPlus: AI based energy-efficient task scheduling for cloud-fog computing environments," *Internet of Things*, Vol.21, PP.1-17, 2023.
- [32] R. Dharaniya, V. S. Satanya, S. Babu and S.A. Zaid, "Ai agents at different data centers to minimize the energy spending". In International conference on advanced computing and communication systems (ICACCS), Coimbatore, India, pp.813-818, 2023.
- [33] H. T. Dang and D.S Kim, "Dynamic Collaborative Task Offloading for Delay Minimization in the Heterogeneous Fog Computing Systems," *journal of communications and networks*. Vol. 25, V. 2, 224-259, 2023
- [34] H.R. Refaat and M.A. Mead, "DLBS: Decentralize Load-Balance Scheduling Algorithm for Real-Time IoT Services in Mist Computing," *International Journal of Advanced Computer Science and Applications(IJACSA)*, Vol.10, No.9, pp. 92-100, 2019.
- [35] A. Khalid and M. Shahbaz, "Service Architecture Models for Fog Computing: A Remedy for Latency Issues in Data Access from Clouds," *Transactions on Internet and Information Systems*, Vol.11, No.5, pp. 2310-2345, 2017.
- [36] D. Rathod and C. Girish, "Load Balancing of Fog Computing Centers: Minimizing Response Time of High Priority Requests,"
- [1] A. Kishorl and C. Chakarbarty, "Task Offloading in Fog Computing for Using Smart Ant Colony Optimization," *Wireless Personal Communications*, Vol. 127, No.2, pp. 1683-1704, 2021.
- [2] M. Kaur and R. Aron, "A systematic study of load balancing approaches in the fog computing environment," *The Journal of Supercomputing*, Vol.77, No.8, pp. 9202-9247, 2021.
- [3] S. Jain, S. Gupta, K.K. Sreelakshmi and J.J. Rodrigues, "Fog computing in enabling 5G-driven emerging technologies for development of sustainable smart city infrastructures," *Cluster Computing*, Vol.25, pp. 1-44, 2022.
- [4] S.K. Idrees and A.K. Idrees, "New fog computing enabled lossless EEG data compression scheme in IoT networks," *Journal of Ambient Intelligence and Humanized Computing*, Vol.13, No.6, pp. 3257-3270, 2022.
- [5] G. Javadzadeh and A.M. Rahmani, "Fog computing applications in smart cities: A systematic survey," *Wireless Networks*, Vol.26, No.2, pp.1433-1457, 2020.
- [6] L. Shooshtarian, D. Lan and A. Taherkordi, "A clustering-based approach to efficient resource allocation in fog computing," in *International Symposium on Pervasive Systems, Algorithms and Networks*, Springer, Cham, pp. 207-224, 2019.
- [7] J. C.Guevara, and N.L.S. da Fonseca, "Task scheduling in cloud-fog computing systems," *Peer-to-Peer Networking and Applications*, Vol.14, No.2, pp. 962-977, 2021.
- [8] D. Baburao, T. Pavankumar, and C.S.R. Prabhu, "Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method," *Applied Nanoscience*, Vol.11, No.7, pp. 1-10, 2021.
- [9] H. Wadhwa and R. Aron, "TRAM: Technique for resource allocation and management in fog computing environment," *The Journal of Supercomputing*, Vol.78, No.1, pp. 667-690, 2022.
- [10] X.Q. Pham and E.N. Huh, "Towards Task scheduling in a Cloud Fog Computing System," *Network Operations and Management Symposium (APNOMS)*, Kanazawa, Japan, pp. 1-4, 2016.
- [11] S. Bitam, S. Zeadally and A. Mellouk, "Fog Computing Job Scheduling Optimization Based on Bees Swarm," *Enterprise Information Systems*, Vol.12, No.4, pp. 373-397, 2018.
- [12] J. Fan, X. Wei, T. Wang, T. Lan and S. Subramaniam, "Deadline-Aware Task Scheduling in a Tiered IoT Infrastructure". In 2017-2017 IEEE Global Communications Conference (GLOBECOM), Singapore, pp. 1-7, 2017.
- [13] S.K. Mishra, D. Puthal, J.J.P.C. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable Service Allocation Using a Metaheuristic Technique in a Fog Server for Industrial Applications," *IEEE Transactions on Industrial Informatics*, Vol.14, No.10, pp. 4497-4506, 2018.
- [14] S. Kabirzadeh, D. Rahbari, and M. Nickray, "A Hyper Heuristic Algorithm for Scheduling of Fog Networks". In 2017 21st Conference of Open Innovations Association (FRUCT), Helsinki, Finland, pp. 148-155, 2017.
- [15] S. Kabirzadeh, D. Rahbari, and M. Nickray, "A Security Aware Scheduling in Fog Computing by Hyper Heuristic Algorithm". In 2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS), Shahrood, Iran, pp. 87-92, 2017.
- [16] T. Choudhari, M. Moh and T.S. Moh, "Prioritized Task Scheduling in Fog Computing". In Proceedings of the ACMSE 2018 Conference (ACMSE), New York, NY, USA, pp.1-8, 2018.
- [17] Y. Sun, T. Dang and J. Zhou, "User Scheduling and Cluster Formation in Fog Computing Based Radio Access Networks". In 2016 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB), Nanjing, China, pp.1-4, 2016.
- [18] V.B.C. Souza, W. Ramirez, X.M. Bruin, E.M. Tordera, G. Ren and G. Tashakor, "Handling Service Allocation in Combined Fog-Cloud Scenarios". In 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, pp.1-5, 2016.

- Study Travel Times,” Journal of Transport Geography, Vol.48, pp. 41-51, 2015 .
- [38] S.N. Srirama, K. Ramamohanarao, R. Buyya and M.R. Mahmud, “Quality of Experience (QoE)-aware Placement of Applications in Fog Computing Environments,” Journal of Parallel and Distributed Computing, Vol.132, pp. 190-203, 2018.
- [37] B. Dewulf, T. Neutens, M. Vanlommel and S. Logghe, “Examining Commuting Patterns Using Floating Car Data and Circular Statistics: Exploring the Use of New Methods and Visualizations to

پاورقی‌ها:

- 26 Hyper Heuristic
- 27 Simulated Annealing
- 28 Reconfigure Dynamically
- 29 Optimize Response Time
- 30 Efficient Resource Allocation
- 31 Optimal User Scheduling
- 32 Signal To Interference Plus Noise
- 33 Minimum Execution Time
- 34 Minimum Completion Time
- 35 All Offloading Algorithm
- 36 Randomly Offloading Algorithm
- 37 Allocation Algorithm Regardless Of The Mobility
- 38 Near Optimal Resource Allocation Mechanism
- 39 Moth-Flame Optimization Algorithm
- 40 Gated Graph Convolution Network
- 41 Recursive Computation Offloading
- 42 Process Engine
- 43 Processor Allocator
- 44 Mobility Controller
- 45 Transfer Insensitive Requests To Cloud On Conditions
- 46 RAM
- 47 Up Bandwidth
- 48 Down Bandwidth
- 49 Neighbor Bandwidth
- 1 Internet Of Things
- 2 International Data Corporation
- 3 Cloud Computing
- 4 Fog Computing
- 5 Fog Node
- 6 Data Service Operators
- 7 Task Scheduling In Fog Computing
- 8 First Come First Serve
- 9 Real-Time Efficient Scheduling
- 10 Data Service Operators
- 11 Convolutional Neural Networks
- 12 Bidirectional Gated Recurrent Unit
- 13 Dynamic Collaborative Task Offloading
- 14 Round Robin
- 15 Shortest Job First
- 16 Binary Particle Swarm Optimization
- 17 Dynamic Level Scheduling
- 18 Heterogeneous Earliest Finish Time
- 19 Bees Life Algorithm
- 20 Genetic Algorithm
- 21 Particle Swarm Optimization
- 22 Ant Colony Optimization
- 23 Improved Particle Swarm Optimization
- 24 Bat Algorithm
- 25 Binary PSO