

بهینه‌سازی زمان پاسخ در شبکه‌های اینترنت اشیای نرم‌افزارمحور با استفاده از محاسبات مه-ابر

الهام حاجیان

پردازش ابری یکی از تکنولوژی‌هایی است که باعث رشد اینترنت اشیاء در زمینه پردازش اطلاعات شده است. با این حال پردازش ابری دارای چالش‌هایی از جمله تأخیر و مصرف پهنای باند می‌باشد. پردازش مه^۲، تکمیل‌کننده پردازش ابری^۳ است. با استفاده از شبکه‌های نرم‌افزارمحور و جداسازی سطح کنترل از سطح داده، می‌توان بعضی از پردازش‌ها را در سطح داده در مجاورت حسگرها و سوئیچ‌های مربوطه به عنوان مه و بعضی از آن را با در نظر گرفتن تأخیر در ابر انجام داد [۱]. در این معماری، ابر همان کنترل‌کننده می‌باشد. پردازش در ابر بدون محدودبودن وسایل ذخیره‌سازی و پردازشی انجام می‌شود.

در نتیجه ترکیب مه و محاسبات ابری در اینترنت اشیاء که توسط شبکه‌های نرم‌افزارمحور^۴ (SDN) تسهیل شده است، مزایای قابل توجهی را ارائه می‌دهد. ادغام محاسبات مه نزدیک‌تر به لبه شبکه تأخیر را کاهش می‌دهد، عملکرد سیستم را بهبود می‌بخشد و پردازش بلادرنگ را ممکن می‌سازد. رایانش ابری قابلیت‌های ذخیره‌سازی و پردازش مقیاس‌پذیر را فراهم می‌کند. شبکه‌های نرم‌افزارمحور نقش مهمی را در مدیریت و بهینه‌سازی منابع شبکه، تضمین اتصال یکپارچه و افزایش امنیت ایفا می‌کنند. به طور کلی این ادغام فرصتی برای افزایش اتصال، کارایی و نوآوری در قلمرو اینترنت اشیاست [۲].

با توجه به حجم بالای داده‌های تولیدشده در دستگاه‌های هوشمند، چالش‌هایی از جمله مصرف باتری، توان، فضای ذخیره‌سازی و پهنای باند مطرح هستند که همگی جزء موانع کیفیت سرویس شبکه می‌باشند. در شبکه‌ها با داده‌های بزرگ، حجم، سرعت و تنوع داده‌ها و بعد جغرافیایی مسائلی هستند که باعث می‌شود پردازش ابری همیشه بهترین گزینه نباشد. محاسبات مه به‌عنوان تکمیل‌کننده محاسبات ابری می‌تواند راه‌حلی برای کاهش زمان پاسخ که یکی از اهداف شبکه است، باشد [۳]. در این راه‌حل تا جایی که ممکن است باید درخواست‌ها در مه انجام شوند.

مه از توانایی محاسباتی نزدیک کاربران نهایی استفاده می‌کند. با انجام سرویس‌ها در مه می‌توان به تأخیر پایین دست پیدا کرد [۴]. در شبکه‌های اینترنت اشیای نرم‌افزارمحور، اولین مسأله برای کاهش تأخیر و بهبود زمان پاسخ، توزیع محاسبات به صورت بهینه می‌باشد. اینکه کدام گره‌ها مسئول محاسباتی کدام گره باشد، مهم است. این مسأله به عنوان توزیع محاسباتی غیرمتمرکز شناخته می‌شود [۵]. دومین مسأله، کنترل‌کننده نرم‌افزارمحور می‌باشد که مسیریابی بهینه را برنامه‌ریزی می‌نماید. مکان کنترل‌کننده با توجه به تعداد پیام‌های ردوبدل شده مهم است. این مسأله

چکیده: اینترنت اشیاء از ابر برای پردازش اطلاعات رسیده از وسایل الکترونیکی استفاده می‌کند. سرویس‌های قدرتمند و دور از حسگرها، پردازش را انجام می‌دهند. وسایل اینترنت اشیاء درخواست‌ها را به ابر ارسال کرده و نتایج را از آن دریافت می‌کنند. در بعضی از کاربردهای اینترنت اشیاء، زمان پاسخ و تأخیر مهم می‌باشد؛ بنابراین باید هرچه بیشتر زمان تأخیر را کاهش داد. ارسال اطلاعات به ابر، خود مستلزم ایجاد تأخیر می‌باشد؛ لذا استفاده از مه در کنار ابر در اینترنت اشیاء نقش اساسی را ایفا می‌کند.

استفاده از مه و رایانش ابری در زمینه اینترنت اشیاء، موضوع قابل توجه برای پژوهشگران می‌باشد. برای تسهیل این فرایند، شبکه‌های نرم‌افزارمحور به عنوان یک جزء حیاتی ظاهر شده‌اند. این شبکه‌ها امکان کنترل و مدیریت متمرکز شبکه را فراهم می‌کنند. همچنین با هدایت پویای جریان‌های داده به منابع مه یا ابر بر اساس شرایط زمان واقعی، استفاده بهینه از منابع و اتصال یکپارچه را تضمین می‌کنند. محاسبات مه به استقرار منابع در نزدیکی حسگر شبکه اشاره دارد. با انجام این کار، محاسبات مه با هدف کاهش تأخیر و استفاده از پهنای باند در حالی که عملکرد کلی سیستم را بهبود می‌بخشد، می‌پردازد. این کار را با استفاده از قابلیت‌های محاسباتی محلی برای پردازش داده‌ها انجام می‌دهد.

این پژوهش با استفاده از معماری پیشنهادی برای شبکه‌های اینترنت اشیاء و مدل‌سازی قسمت‌های مختلف این معماری با استفاده از نظریه صف، زمان پاسخ را توسط محاسبات مه-ابر کاهش داده و پارامترهای کیفیت سرویس شبکه را توسط آنالیز ریاضی به دست می‌آورد. در ادامه نیز نمودارهای مقایسه انرژی باقیمانده و تأخیر برای کاربردهای سلامتی و روشنایی و همچنین نمودارهای مقایسه‌ای استفاده و عدم استفاده از محاسبات مه-ابر رسم شده است. نمودارها نشان می‌دهند که استفاده از محاسبات مه-ابر، زمان پاسخ را کاهش می‌دهد و کاربرد روشنایی با استفاده از مه، انرژی باقیمانده بیشتر و کاربرد سلامتی، تأخیر کمتر دارد. شبیه‌سازی با استفاده از نرم‌افزار NS۲ در کاربرد خانه هوشمند انجام شده است.

کلیدواژه: اینترنت اشیاء، خانه هوشمند، شبکه‌های نرم‌افزارمحور، محاسبات ابر، محاسبات مه، نظریه صف.

۱- مقدمه

اینترنت اشیاء^۱ (IoT) نقش مهمی در فناوری‌های ارتباطی بر عهده دارد. با افزایش تعداد وسایل متصل به اینترنت، داده‌های زیادی نیز تولید می‌شوند. انتقال و پردازش این داده‌ها نیاز به مصرف انرژی زیادی دارد.

این مقاله در تاریخ ۶ مرداد ماه ۱۴۰۳ دریافت و در تاریخ ۱۳ بهمن ماه ۱۴۰۳ بازنگری شد.

الهام حاجیان، گروه مهندسی کامپیوتر، دانشگاه بجنورد، بجنورد، ایران، (email: e.hajian@ub.ac.ir)

1. Internet of Things

2. Fog Computing

3. Cloud Computing

4. Software Defined Network

تخلیه بار ترافیکی^۳ بهره گرفته شده که در آن هر گره اینترنت اشیا می‌تواند یکی از شبکه‌های مه و یا شبکه ابر را با توجه به کیفیت خدمات مورد نظرش انتخاب کند. مراحل اجرای طرح پیشنهادی به این صورت انجام گرفته که ابتدا برای رسیدن به مقدار حداکثری کیفیت تجربه^۴، مسأله فرموله شده تا تصمیمات تخلیه بار ترافیکی برای هر کاربر اتخاذ گردد. در این مقاله منظور از کیفیت تجربه، کاهش تأخیر و انرژی محاسباتی می‌باشد. سپس برای تمامی کاربران به منظور تشخیص انتخاب درست، تخلیه بار ترافیکی، مدل بازی و ویژگی‌های فرموله شده و آنالیز شده است و در ادامه تعادل نش با توجه به رشد تعداد کاربران جهت کاهش تأخیر به کار گرفته شده است.

ونگ و همکاران [۱۰] یک شبکه محاسباتی سه‌لایه با استفاده از ارتباط عمودی بین دستگاه‌ها، گره‌های لبه و سرویس‌دهنده‌های ابری با روش جهت متناوب ضرایب و تفاوت برنامه‌نویسی محدب در نظر گرفته‌اند. لی و همکاران [۱۱] بر بهینه‌سازی محاسبه و تخصیص منابع ارتباطی در شبکه‌های اینترنت اشیا بی‌سیم مبتنی بر محاسبات مه با دسترسی چندگانه غیرمترجم متمرکز شده‌اند. طرح پیشنهادی آنها تصمیم بهینه برای انتخاب حالت محاسبات مناسب است. مسأله بهینه‌سازی فرموله شده یک مسأله بهینه‌سازی غیرخطی عدد صحیح مختلط است و سپس الگوریتم ژنتیک بهبودیافته برای حل آن معرفی شده است.

باروس و همکاران [۱۲] از محاسبات مه با تأخیر کم برای انجام کنترل انرژی الکتریکی در میکروشبکه استفاده کردند. محاسبات مه ارائه شده با خدمات مدیریتی از جمله کنترل کردن و الگوریتم‌های برنامه‌ریزی برای لوازم خانگی به منظور کاهش مبلغ قبض با محاسبات مشتق یکپارچه متناسب است.

کاراتاس و همکاران [۱۳] محاسبات ابر و مه را برای قراردادن و سرویس‌دهی کارآمد داده‌های اینترنت اشیا ارائه داده‌اند. راهبردهای قراردادن داده‌ها و الگوریتم‌هایی برای کاهش تأخیر متوسط که توسط برنامه‌های کاربردی وابسته به رده‌بندی داده‌ها استفاده می‌شود، با استفاده از واحدهای محاسباتی مه انجام شده است.

نگوین و همکاران [۱۴] الگوریتم بهینه با پیچیدگی کم را برای بارگذاری محاسبات مشترک و تخصیص منابع برای سیستم محاسبات ابری متحرک مبتنی بر چندورودی، چندخروجی با در نظر گرفتن اطلاعات حالت کانال ایجاد کردند. الگوریتم‌های بهینه و با پیچیدگی کم برای حل برنامه‌نویسی غیرخطی عدد صحیح مختلط اساسی ارائه شده است. ونگ و همکاران [۱۵] بهینه‌سازی تصمیم بارگذاری، قابلیت محاسبات محلی و تخصیص منابع محاسباتی گره مه را در نظر گرفتند. آنها این مسأله را به دو زیرمسأله مستقل تجزیه کرده و یک الگوریتم تصمیم برای بارگذاری حداقل تأخیر مبتنی بر الگوریتم‌های ژنتیک و شبیه‌سازی تبرید ارائه کرده‌اند. برای حل این مسأله برنامه‌نویسی غیرخطی عدد صحیح مختلط با پیچیدگی کم طراحی شده است.

ژوو و همکاران [۱۶] مسأله منبع را بررسی کرده و یک مسأله بهینه‌سازی برای منبع را فرموله کردند تا مجموع وزنی تأخیرهای کاربران را مطابق با کاربردهای خواسته شده به حداقل برسانند. همچنین الگوریتم مبتنی بر بازی ائتلاف در این مرجع پیشنهاد شده است.

فرجی و همکاران [۱۷] یک چهارچوب محاسباتی توزیع شده برای مدیریت منابع خودمختار در زمینه محاسبات مه ارائه دادند. سپس آنها از

به عنوان محل فرارگیری کنترل‌کننده‌ها اهمیت پیدا می‌کند. حل هر دوی این مسائل به بهینه‌سازی زمان تأخیر منجر می‌شود [۶]. در این پژوهش تمرکز بیشتر بر روی مسأله اول می‌باشد؛ اینکه سرویس‌ها روی چه وسیله‌ای اجرا شوند. اجرای سرویس به صورت محلی در مه انجام شود یا اینکه به کنترل‌کننده در ابر فرستاده شود. مسأله دوم به دلیل تعداد یک عدد کنترل‌کننده دارای مکان مشخص در ابر، در این پژوهش نمود پیدا نمی‌کند.

برای مقایسه اهداف شبکه‌هایی با این موارد، کاربرد خانه هوشمند با دو هدف سلامتی و روشنایی مورد مقایسه قرار می‌گیرد. فناوری خانه هوشمند بر اساس اصل اینترنت اشیا عمل می‌کند. این یک مفهوم جدید است که به دستگاه‌هایی اشاره دارد که به شکل شبکه به یکدیگر متصل شده و قادر به برقراری ارتباط با یکدیگر برای انجام وظایف هستند. دستگاه‌های مختلف می‌توانند به روش‌های مختلف با یکدیگر ارتباط برقرار کنند. رایج‌ترین راه از طریق اتصالات شبکه بی‌سیم یا وای‌فای است، اما می‌تواند از طریق امواج رادیویی و سیم‌کشی نیز باشد. شبکه می‌تواند از ترکیبی از این موارد استفاده کند؛ برای مثال استفاده از سیم‌کشی برای زمانی که وای‌فای خاموش است، بسیار کاربردی است. نوآوری‌های این مقاله عبارتند از

۱) ارائه معماری جدید بر اساس محاسبات ابر و مه

۲) مدل‌سازی معماری با استفاده از نظریه صف

۳) کاهش زمان پاسخ و بررسی پارامترهای کیفیت سرویس شبکه با استفاده از آنالیز ریاضی

۲- تاریخچه

در ادامه تعدادی از مقالات مطالعه شده در این زمینه بیان می‌گردد. در [۷] مبارکلی و همکاران از شبکه مه به عنوان شبکه‌ای که نقش اساسی در انتقال و توزیع داده‌ها در اینترنت اشیا دارد، به عنوان لایه میانی بین ابر و اشیا به منظور افزایش سرعت برای برنامه‌های حساس به زمان یاد کرده‌اند. برای نظارت بر این شبکه‌ها از شبکه‌های نرم‌افزارمحور و مکانیزم مجازی‌سازی استفاده شده است. در این مقاله با استفاده از پیش‌بینی و به‌کارگیری الگوریتم‌های انتقال برنامه‌های حساس به تأخیر، به کاهش تأخیر دست یافته‌اند.

در استفاده از شبکه مه- ابر، مصرف انرژی و تأخیر ارتباطی، مواردی هستند که در [۸] توسط باباعلی و همکاران به آن توجه شده است. در این مقاله از روش VSFC^۱ در شبکه‌های مه به منظور کاهش تأخیر استفاده شده است، به طوری که با استفاده از منابع محاسباتی در لبه، مصرف انرژی شبکه و تأخیر را کاهش داده و همچنین با اضافه کردن افزونه‌ای به شبیه‌ساز iFogSim روش پیشنهادی خود را مورد ارزیابی قرار داده‌اند.

در [۹] شاه‌منصوری و همکاران به بررسی محاسبات سلسله‌مراتبی شبکه مه و شبکه ابری برای دستگاه‌ها و سیستم‌های اینترنت اشیا با استفاده از تئوری بازی پرداخته‌اند. شبکه مه به منظور کاهش تأخیر و ترافیک زیرساخت^۲ شبکه استفاده شده است، ولی مدیریت منابع در این شبکه‌ها یکی از چالش‌های مهم است. علاوه بر این در شبکه مه هر کاربر به‌طور خودخواهانه سعی در رسیدن به کیفیت تجربه حداکثری برای خود بدون توجه به کاربران دیگر و اولویت‌های آنها دارد. برای حل این چالش و به منظور افزایش رضایتمندی کاربران از خدمات ارائه شده، از محاسبات

3. Offloading

4. Quality of Experience

1. Volunteer Supported Fog Computing

2. Backbone

جدول ۱: خلاصه‌ای از مطالعات انجام شده.

مرجع	مزایا	معایب
[۷]	کاهش تأخیر	عدم جداسازی برنامه برای مه و ابر
[۸]	کاهش تأخیر	انجام محاسبات در لبه شبکه
[۹]	کاهش ترافیک	عدم مدیریت منبع
[۱۰]	کاهش تأخیر	محاسبات زیاد
[۱۱]	تخصیص منابع به صورت بهینه	محاسبات زیاد
[۱۲]	کاهش داده عبوری به ابر	عدم داشتن اولویت
[۱۳]	اولویت‌دادن داده‌ها برای کاهش تأخیر	اولویت فقط بر اساس نوع داده
[۱۴]	پیچیدگی کم برای تخصیص منابع	تأخیر بررسی نشده است.
[۱۵]	محاسبات محلی و تخصیص منابع	محاسبات زیاد
[۱۶]	کاهش تأخیر بر اساس وزن	عدم مدیریت منبع
[۱۷]	محاسبات توزیع شده و مدیریت منابع	ازدحام شبکه
[۱۸]	استفاده از زمانبند برای انجام محاسبات	عدم اولویت
[۱۹]	کاهش هزینه کلی	عدم مدیریت منابع
[۲۰]	توزیع بار بر اساس حرکت وسایل	بررسی فقط در وسایل سیار
[۲۱]	تبادل بار بر اساس مصرف انرژی	بررسی فقط در وسایل سیار
[۲۲]	زمان‌بندی بر اساس حداقل کردن مصرف انرژی	عدم اولویت

زمان اجرای سرویس در مه بیشتر باشد، در مه انجام می‌شود. - اگر مجموع تأخیر ارسال سرویس به ابر و زمان اجرای آن در ابر از زمان اجرای سرویس در مه کمتر باشد، در ابر انجام می‌شود. علاوه بر تأخیر برای اجرای سرویس در مه باید محدودیت‌های دیگر نیز در نظر گرفته شود؛ زیرا ممکن است مه انرژی کافی برای انجام پردازش را نداشته باشد. هرچند که از لحاظ تأخیر، بهتر است سرویس به صورت محلی انجام شود؛ بنابراین در الگوریتم پیشنهادی علاوه بر تأخیر باید انرژی مصرفی نیز محاسبه گردد. در بخش بعد، معماری سیستم روش پیشنهادی بیان می‌شود.

۳-۱ معماری پیشنهادی

در این پژوهش، معماری جدیدی از اینترنت اشیا بر اساس مه و ابر ارائه شده است. به علت اینکه تمرکز اصلی روی کاربردهایی از سلامتی و روشنایی می‌باشد و ممکن است حسگرهایی از اینترنت اشیا سیار و تعدادی ثابت باشند، بنابراین باید معماری‌ای ارائه شود که برای هر دو نوع حسگر بهینه باشد. در این معماری سه حالت انتقال وجود دارد: حسگر اینترنت اشیا به حسگری دیگر، مه به ابر و بالعکس و حسگر به ابر یا مه و بالعکس. با توجه به حرکت تعدادی از حسگرها، خوشه‌های مه تشکیل می‌شود. هر خوشه سرویس را برای حسگر در محدوده خود تأمین می‌کند؛ بنابراین از انتقال داده سرویس در شبکه جلوگیری می‌شود. پردازش سرویس در خوشه، توسط کنترل‌کننده مه انجام می‌شود [۲۳]. این کنترل‌کننده مرکز تصمیم‌گیری مه است که منابع ناهمگن را بر اساس نوع سرویس مختلف اختصاص می‌دهد و همچنین فشار بار کاری را از روی کنترل‌کننده مرکزی ابر کاهش داده و تصمیمات بدون وقفه را در مجاورت حسگرهای اینترنت اشیا می‌گیرد. کنترل‌کننده مرکزی ابر، مرکز کنترل عمومی است که در ابر قرار گرفته است. این کنترل‌کننده، شبکه را مدیریت می‌کند و قوانین سرویس و سیاست‌ها را برای کنترل‌کننده مه ارسال می‌کند. زمانی که منابع محاسباتی در مه کافی نباشد، کنترل‌کننده ابر سرویس را انجام می‌دهد یا به کنترل‌کننده مه مجاور این سرویس را محول می‌کند. استفاده از مدل کنترل‌کننده دولایه‌ای (مه و ابر)، مدیریت و پیکربندی را ساده می‌کند و کارایی را بهبود داده و باعث گسترش سیستم می‌شود [۲۴]. انجام خوشه‌بندی مه بر اساس برد کنترل‌کننده مه می‌باشد که تا چند متر اطراف خود می‌تواند داده‌ها را دریافت کند. اگر حسگری در برد هیچ کدام از کنترل‌کننده‌های مه قرار نگیرد، مستقیماً تقاضای سرویس را به ایستگاه پایه با پوشش وسیع ارسال می‌کند و آن را به نزدیک‌ترین کنترل‌کننده خوشه مه ارسال می‌کند. شکل ۱ معماری سیستم پیشنهادی را نشان می‌دهد که در آن، حسگرها و کنترل‌کننده مه در سطح داده و کنترل‌کننده ابر در سطح کنترل قرار می‌گیرند. کنترل‌کننده اصلی در شبکه نرم‌افزارمحور که کنترل‌کننده SDN نامیده می‌شود، در ابر وجود دارد. به همین دلیل در این معماری ابر در سطح کنترل پوشش داده می‌شود.

۳-۲ سناریوی پیشنهادی

سناریوی پیشنهادی، دیدی دیگر از معماری پیشنهادی را بررسی می‌کند. هر دید با یک لایه نشان داده می‌شود. ارتباطات بین این لایه‌ها با خطوط مشخص شده است. شکل ۲ مربوط به این سناریو است. در سناریوی پیشنهادی، ۳ لایه در نظر گرفته شده است. لایه فیزیکی، قسمت‌های فیزیکی سیستم را نشان می‌دهد. تمامی وسایل سخت‌افزاری در این لایه قرار می‌گیرند. لایه شبکه، شامل سوئیچ‌های نرم‌افزارمحور و

یک سیستم تأمین خدمات اینترنت اشیا مبتنی بر حلقه کنترل (پایش، تجزیه و تحلیل، برنامه‌ریزی، اجرا و دانش) به همراه یک روش یادگیری تقویت‌کننده و پشتیبانی از روش رگرسیون برداری استفاده کردند. کشور و همکاران [۱۸] به منظور افزایش کیفیت سرویس و کاهش زمان پاسخ در برنامه‌های حساس به زمان، یک زمانبند فراابنکاری با استفاده از الگوریتم بهینه‌سازی کلونی مورچه‌ها که از طبیعت الهام گرفته شده است، در محیط مه ارائه کرده‌اند. این الگوریتم باعث می‌شود زمان پاسخ نسبت به الگوریتم‌های مقایسه‌شده در مقاله کاهش داده شود. در [۱۹] تا [۲۱] که در محیط مه شبیه‌سازی گردیده‌اند، به ترتیب هزینه کلی دستگاه کاربر، نسبت تحویل بسته و مقدار کل انرژی و انحراف بار، بهینه شده است.

عزیزی و همکاران [۲۲] به صورت ریاضی مسأله زمان‌بندی کار را برای به حداقل رساندن مصرف انرژی کل گره‌های مه در حالی که الزامات کیفیت خدمات وظایف اینترنت اشیا را برآورده می‌کند، فرموله کردند. همچنین به حداقل رساندن زمان نقص مهلت در مدل را در نظر گرفته‌اند. در مرحله بعد دو الگوریتم مبتنی بر نیمه‌حریص، یعنی نیمه‌حریص آگاه از اولویت و نیمه‌حریص عدم آگاهی از اولویت با الگوریتم چندشروعی را برای ترسیم کارآمد وظایف اینترنت اشیا به گره‌های مه پیشنهاد دادند. جدول ۱ خلاصه‌ای از تحقیقات مطالعه‌شده را نشان می‌دهد.

۳- روش پیشنهادی و معماری سیستم

این پژوهش برای حل کردن مسأله توزیع محاسباتی غیرمتمرکز در جهت بهبود زمان پاسخ به مکان اجرای سرویس‌ها می‌پردازد. بعضی از سرویس‌ها به صورت محلی در مه اجرا می‌شوند و اجرای بعضی دیگر به ابر محول می‌شود. سرویس‌های سبک‌وزن و سرویس‌هایی که به تعداد زیاد تقاضا می‌شوند، بهتر است به صورت محلی اجرا شود و تأخیر تقریباً صفر دارند. سرویس‌های دیگر به ابر فرستاده می‌شوند.

انتخاب بین اجرای محلی که در این پژوهش اجرا در مه نامیده می‌شوند و اجرا در ابر، بر اساس زیر انتخاب می‌شود:

- اگر مجموع تأخیر ارسال سرویس به ابر و زمان اجرای آن در ابر از

جدول ۲: پارامترهای مورد نیاز برای روش پیشنهادی.

پارامتر	توضیح
μ_s	نرخ سرویس ارسال
λ_s	نرخ ورودی ارسال
D_u	طول داده فراسو
r_u	نرخ فراسو
d	فاصله تا ابر
C_u	سرعت فراسو
M_c	تعداد چرخه پردازش در ابر
f_c	نرخ چرخه پردازش ابر
μ_r	نرخ سرویس دریافت
λ_r	نرخ ورودی دریافت
D_d	طول داده فروسو
r_d	نرخ فروسو
C_d	سرعت فروسو
μ	نرخ سرویس در کل کنترل کننده‌های مه
λ	نرخ ورودی به کل کنترل کننده‌های مه
M_f	تعداد چرخه پردازش در مه
f_f	نرخ چرخه پردازش در مه
e	انرژی مصرفی در هر چرخه
e_s	انرژی مصرفی برای ارسال یک بیت
e_r	انرژی برای دریافت یک بیت
E'	انرژی مصرفی پردازش در مه
T'	تأخیر در مه
E	انرژی مصرفی پردازش در ابر
T	تأخیر در ابر

$$T' = \frac{1}{\mu - \lambda} + \frac{M_f}{f_f} \quad (۱)$$

زمان تأخیر در مه از مجموع تأخیرهای صف و پردازش تشکیل می‌شود. تأخیر صف از معکوس تفریق نرخ ورودی (λ) و نرخ سرویس (μ) به دست می‌آید. تأخیر پردازش برابر با تقسیم تعداد چرخه‌های پردازشی مورد نیاز (هر سرویس ممکن است در چند چرخه انجام شود) برای پردازش محاسبات (M_f) بر نرخ انجام چرخه‌های پردازشی (f_f) در مه است.

انرژی مصرفی در مه وابسته به تعداد پردازش‌ها و انرژی لازم برای انجام هر پردازش می‌باشد. مه باید منابع محاسباتی مورد نیاز برای انجام سرویس را داشته باشد. یکی از این منابع، انرژی است. انرژی مصرفی برای پردازش در مه از (۲) به دست می‌آید

$$E' = M_f \times e \quad (۲)$$

انرژی مصرفی برای پردازش در مه از ضرب تعداد چرخه انجام شده برای محاسبات در انرژی مصرفی برای انجام هر چرخه به دست می‌آید. با توجه به اینکه برای کاهش زمان پاسخ، بهتر است سرویس‌ها در مه پردازش شوند، ولی هر گره به دلیل محدودیت انرژی ممکن است سرویس را اجرا نکند و به ابر ارسال کند.

در صورتی که انرژی مصرفی برای پردازش در مه از انرژی باقیمانده کنترل کننده مه کمتر باشد، در مه انجام می‌شود. تأخیر در ابر برابر با مجموع تأخیرهای موجود در لایه شبکه برای ارتباط وسایل و ارسال به ابر به علاوه تأخیر پردازش در ابر می‌باشد. میزان تأخیر به وجود آمده برای ارسال پیام‌ها به ابر از (۳) به دست می‌آید



شکل ۱: معماری سیستم پیشنهادی.



شکل ۲: سناریو و دید پیشنهادی لایه‌ها.

کنترل کننده است. این لایه مسئول ارتباطات درون شبکه‌ای می‌شود. لایه محاسبات که مهم‌ترین لایه بر اساس روش پیشنهادی است، نحوه انجام محاسبات و وسایل درگیر را نشان می‌دهد. با توجه به انجام محاسبات در مه یا ابر، این لایه شامل کنترل کننده ابر و کنترل کننده مه می‌باشد که محاسبات در اولی به عنوان محاسبات در ابر و پردازش در دومی به عنوان مه شناخته می‌شود. زمان پاسخ (تأخیر کلی) در این سناریو برابر با مجموع زمان اجرا (زمانی که سرویس اجرا می‌شود، چه به صورت محلی و چه در ابر) و تأخیر ارتباطی (زمانی که یک کامپیوتر با کامپیوتر دیگر ارتباط برقرار می‌کند) می‌باشد. الگوریتم پیشنهادی در لایه محاسبات انجام می‌شود. این الگوریتم بیشتر ارتباط بین لایه فیزیکی و لایه محاسبات است و مشخص می‌کند داده‌های وسایل فیزیکی باید از چه طریق پردازش شوند، به صورت محلی و در کنترل کننده مه باید محاسبات انجام شود و یا از طریق ارسال به ابر امکان‌پذیر است. لایه شبکه نیز در صورت ارسال به ابر استفاده می‌شود و ارتباطات بین لایه‌ای را نشان می‌دهد.

۳-۳ ساختار شبکه روش پیشنهادی

ساختار شبکه مدل پیشنهادی با پارامترهای مورد نیاز لیست شده در جدول ۲ بیان شده است. با توجه به پارامترهای موجود در جدول، مکان اجرای سرویس مهم است. علاوه بر در نظر گرفتن تأخیر برای کاهش زمان پاسخ در کاربردهای اینترنت اشیا، باید انرژی مورد نظر را برای انجام سرویس در نظر گرفت. منابع محاسباتی برای انجام سرویس در مه باید به قدر کافی باشند تا قابل انجام باشد؛ در غیر این صورت به ابر واگذار می‌گردد. تأخیر و انرژی محاسبات در مه و ابر توسط معادلات زیر در روش پیشنهادی به دست می‌آید.

اگر سرویسی به صورت محلی (مه) اجرا شود، تأخیر ارتباط در لایه شبکه وجود ندارد و فقط تأخیر در لایه محاسبات محاسبه می‌شود. به دلیل اینکه کنترل کننده مه در نزدیکی حسگرها قرار دارد، تقریباً تأخیر ایجاد شده برابر صفر در نظر گرفته شده است. در این صورت تأخیر در مه با استفاده از (۱) به دست می‌آید

در ارتباط است، ولی فاصله ایستگاه پایه و سرخوشه در نظر گرفته نشده است. (۳) سرخوشه در هر دور تغییر می‌کند و شبکه خوشه‌ها به‌روزرسانی می‌شود و بنابراین مصرف انرژی زیادی دارد [۲۵]. انرژی مصرفی در حالت ارسال و دریافت در الگوریتم Leach برای ارسال l بیت داده به فاصله d به ترتیب با استفاده از (۶) و (۷) به دست می‌آید

$$E_{TX}(l, d) = E_{TX}(l) + E_{TX-amp}(l, d) = \begin{cases} lE_{elect} + lE_{fs}d^\alpha, & d < d_c \\ lE_{elect} + lE_{mp}d^\beta, & d > d_c \end{cases} \quad (6)$$

$$E_{Rx}(l) = E_{RX-elect}(l) = lE_{elect} \quad (7)$$

رابطه (۶) مصرف انرژی در حالت ارسال را نشان می‌دهد که وابسته به فاصله بین فرستنده و گیرنده می‌باشد. رابطه (۷) مصرف انرژی در حالت دریافت را نشان می‌دهد [۲۵]. مقادیر اولیه برای شبیه‌سازی این پارامترها در ادامه آورده شده است.

۳-۵ مدل معماری محاسباتی مه-ابر

۳-۵-۱ توصیف مدل

همان طور که در قسمت قبل گفته شد، حسگرهای اینترنت اشیا درخواست‌های خود را برای پردازش، ابتدا به مه ارسال کرده و سپس با توجه به محدودیت‌های مه، یا در مه انجام شده یا به کنترل‌کننده ابر ارسال می‌شود. با توجه به درخواست‌های زیاد حسگرهای اینترنت اشیا و تأخیر پردازش مه، درخواست‌ها در صف پردازش مه قرار خواهند گرفت. این صف دارای حداکثر طول C می‌باشد. اگر صف پر شده باشد، پیام‌ها گم می‌شوند. تعدادی درخواست با احتمال P به دلیل وجود محدودیت مانند منابع محاسباتی، حافظه ذخیره‌سازی و ... باید به کنترل‌کننده ابر ارسال شوند و با احتمال $1-P$ درخواست انجام شده و نتیجه به ایستگاه پایه ارسال می‌شود. این صف در کنترل‌کننده مه قرار دارد و از نوع $M/M/1/C$ می‌باشد. هر کنترل‌کننده مه یک صف مجزا برای خود دارد و درخواست‌های محدود مه خود را جمع‌آوری می‌کند. تعدادی درخواست که در کنترل‌کننده مه انجام نمی‌شوند به کنترل‌کننده ابر هدایت می‌شوند. در کنترل‌کننده ابر نیز صفی وجود دارد که به دلیل عدم وجود محدودیت در حداکثر تعداد درخواست‌ها در صف، از نوع $M/M/1$ در نظر گرفته شده است. بعد از رسیدن نوبت به درخواست و انجام آن، درخواست از سیستم خارج شده و به ایستگاه پایه ارسال می‌شود. پیام‌های ورودی با توزیع پواسون با نرخ ورودی λ_i توسط هر حسگر وارد صف می‌شوند. i اندیس شماره حسگر است. معادله (۸) نرخ کل ورودی را نشان می‌دهد

$$\lambda = \sum_{i=1}^X \lambda_i \quad (8)$$

X تعداد حسگرها در هر مه و بنابراین λ نیز بر اساس توزیع پواسون است. زمان‌های سرویس در هر کنترل‌کننده مستقل است و به صورت نمایی توزیع شده که نرخ میانگین آن $1/\mu_{FC}$ می‌باشد. شکل ۳ این مدل را نشان می‌دهد.

۳-۵-۲ تحلیل مدل

در این بخش به تحلیل مدل‌سازی معماری پیشنهادی با استفاده از نظریه صف پرداخته می‌شود. برای هر کدام از کنترل‌کننده‌های مه و ابر صفی در نظر گرفته شده که در ادامه تجزیه و تحلیل می‌شود. پارامترهای مورد نیاز برای این صف‌ها در جدول ۳ معرفی می‌شوند.

$$T = \frac{1}{\mu_s - \lambda_s} + \frac{D_u}{r_u} + \frac{d}{C_u} + \frac{M_c}{f_c} + \frac{1}{\mu_r - \lambda_r} + \frac{D_d}{r_d} + \frac{d}{C_d} \quad (3)$$

تأخیر برای رسیدن داده‌ها به ابر از چندین مورد تشکیل شده است. در این تأخیر، تأخیر صف، تأخیر انتقال، تأخیر انتشار در حالت فراسو و همچنین تأخیر صف، تأخیر انتقال و تأخیر انتشار در حالت فرسو به‌علاوه تأخیر پردازشی محاسبه شده است.

$1/(\mu_s - \lambda_s)$ تأخیر صف می‌باشد. D_u/r_u که از تقسیم طول داده فراسو به نرخ ارسال به دست می‌آید، تأخیر انتقال را محاسبه می‌کند. d/C_u که تأخیر انتشار در حالت فراسو را نشان می‌دهد، از تقسیم فاصله تا ابر و سرعت فراسو به دست می‌آید. حالت فرسو نیز به همین ترتیب است. M_c/f_c که تأخیر پردازش محاسبات در ابر را نشان می‌دهد، از تقسیم تعداد چرخه در ابر (هر سرویس ممکن است در چند چرخه انجام شود) به نرخ چرخه در ابر به دست می‌آید. ابر برخلاف مه محدودیت انرژی ندارد و هر تعداد چرخه و سرویس را می‌تواند پردازش کند. انرژی مورد نیاز برای پردازش در ابر از (۴) به دست می‌آید

$$E = D_u \times e_s + D_d \times e_r \quad (4)$$

این انرژی برای پردازش وابسته به لایه‌های شبکه و محاسبات می‌گردد. انرژی مصرفی پردازش در ابر برابر است با مجموع حجم داده ارسالی در حالت فراسو ضربدر انرژی مصرفی برای ارسال هر بیت و همچنین حجم داده دریافتی از فرسو ضربدر انرژی مصرفی برای دریافت هر بیت.

۳-۴ خوشه‌بندی و الگوریتم مسیریابی

به دلیل اینکه مسیریابی در الگوریتم پیشنهادی، الگوریتم Leach است و میزان مصرف انرژی در ابر برابر است با مجموع حجم داده ارسالی در الگوریتم می‌باشد، الگوریتم Leach توضیح داده می‌شود. Leach یکی از الگوریتم‌های مهم در مسیریابی می‌باشد که برای تعادل بار استفاده می‌گردد. این الگوریتم دارای دو فاز شکل‌گیری خوشه و فاز پایداری می‌باشد. سرخوشه به صورت تصادفی انتخاب می‌گردد. در فاز شکل‌گیری خوشه، هر گره، شانس برای انتخاب‌شدن به عنوان سرخوشه دارد. این انتخاب به وسیله یک عدد تصادفی بین صفر و یک انجام می‌گیرد. رابطه (۵) چگونگی تولید عدد تصادفی را نشان می‌دهد

$$T(n) = \begin{cases} \frac{p}{1 - p(r \bmod p^{-1})}, & n \in G \\ 0, & \text{else} \end{cases} \quad (5)$$

که p احتمال انتخاب گره سرخوشه، r شماره دور، $T(n)$ مقدار آستانه برای انتخاب سرخوشه و G مجموعه گره‌هایی است که در P^{-1} دور آخر، سرخوشه نشده‌اند. در این الگوریتم اگر مقدار عدد تولیدشده تصادفی، کمتر از $T(n)$ باشد، گره به عنوان سرخوشه انتخاب می‌گردد. بعد از مشخص‌شدن سرخوشه‌ها، گره‌های معمولی بر اساس شدت سیگنال دریافتی از سرخوشه‌ها به سرخوشه مورد نظر با توجه به بیشترین شدت سیگنال متصل می‌شوند. در فاز پایداری، اعضای خوشه داده‌های خود را به سرخوشه با استفاده از روش تقسیم زمانی^۱ (TDMA) ارسال می‌کنند و سپس داده به ایستگاه پایه ارسال می‌شود. با اینکه الگوریتم Leach مصرف انرژی را در شبکه توزیع می‌کند، ولی این پروتکل دارای ۳ عیب اساسی می‌باشد: (۱) پارامتر انرژی گره‌ها در نظر گرفته نمی‌شود. (۲) سرخوشه به صورت مستقیم با ایستگاه پایه به وسیله ارتباطات تک‌گامی

جدول ۳: تعریف پارامترهای مورد نیاز برای تحلیل صف.

پارامتر	توضیح
C	حداکثر طول صف
P	احتمال درخواست‌های ارسالی به ابر
λ_i	نرخ ورودی درخواست از هر حسگر به کنترل کننده مه
μ_{FC}^j	نرخ سرویس درخواست هر کنترل کننده مه
π_k^j	احتمال حالت پایدار K تقاضای پیام در کنترل کننده z ام
P_{loss}^j	نرخ فقدان پیام با وجود c پیام در صف
X_{FC}^j	میانگین کارایی سرویس
U_{FC}	بهره‌وری کنترل کننده مه
E_{FC}^j	تعداد میانگین تقاضاهای پیام در z امین کنترل کننده مه
L_{FC}^j	میانگین تعداد کنترل کننده‌های مه مشغول انجام سرویس پیام تقاضا
M_{FC}^j	میانگین تعداد پیام‌های منتظر در z امین کنترل کننده مه
W_{FC}^j	میانگین زمان انتظار پیام تقاضای سرویس در z امین کنترل کننده مه
T_{FC}^j	میانگین زمان پاسخ پیام تقاضای سرویس در z امین کنترل کننده مه
$P\lambda$	نرخ تقاضای پیام ورودی به کنترل کننده ابر
π_k	احتمال حالت پایدار K تقاضا در سیستم
E_{cc}	میانگین تعداد پیام موجود در صف ابر
M_{cc}	میانگین درخواست‌های منتظر در صف ابر
X_{cc}	میانگین کارایی سرویس در کنترل کننده ابر
U_{cc}	بهره‌وری صف ابر
W_{cc}	میانگین زمان انتظار در کنترل کننده ابر
T_{cc}	میانگین زمان پاسخ درخواست در کنترل کننده ابر

$$\overline{E_{FC}^j} = \sum_{k=1}^c k \pi_k^j = \begin{cases} \frac{\rho}{1-\rho} \frac{1-(c+1)\rho^c + c\rho^{c+1}}{1-\rho^{c+1}}, & \rho \neq 1 \\ \frac{c}{2}, & \rho = 1 \end{cases} \quad (19)$$

میانگین تعداد کنترل کننده‌های مشغول انجام سرویس پیام تقاضا از (۲۰) به دست می‌آید

$$\overline{L_{FC}^j} = 1 - \frac{1-\rho}{1-\rho^{c+1}} \quad (20)$$

میانگین تعداد پیام‌های منتظر در z امین کنترل کننده مه از (۲۱) به دست می‌آید

$$\overline{M_{FC}^j} = \overline{E_{FC}^j} - \overline{L_{FC}^j} \quad (21)$$

میانگین زمان انتظار پیام تقاضای سرویس در z امین کنترل کننده مه از (۲۲) به دست می‌آید

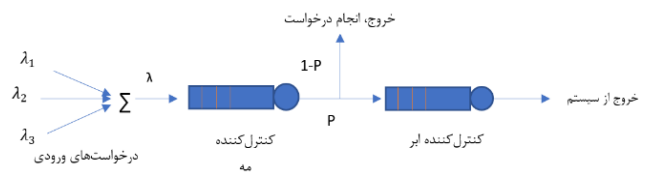
$$\overline{W_{FC}^j} = \frac{\overline{M_{FC}^j}}{\lambda} \quad (22)$$

میانگین زمان پاسخ پیام تقاضای سرویس در z امین کنترل کننده مه از (۲۳) به دست می‌آید

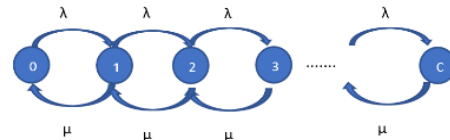
$$\overline{T_{FC}^j} = \frac{\overline{E_{FC}^j}}{X_{FC}^j} = \frac{1}{\mu_{FC} - 1} - \frac{C\lambda^c}{\mu_{FC} - \lambda^c} \quad (23)$$

۳-۲-۵-۳ صف کنترل کننده ابر

اگر تقاضایی در کنترل کننده مه انجام نشود، به کنترل کننده ابر ارسال می‌گردد. یک تقاضای پیام با احتمال p به کنترل کننده ابر ارسال می‌شود



شکل ۳: مدل صف در معماری پیشنهادی.



شکل ۴: مدل صف کنترل کننده مه.

۳-۲-۵-۳ صف کنترل کننده مه

همان طور که در قسمت قبل گفته شد، هر کنترل کننده مه یک صف از نوع $M/M/1/C$ دارد و حداکثر مقدار طول صف C می‌باشد. پیام تقاضا وارد صف می‌شود، مگر اینکه صف پر باشد. احتمال پایداری است که k پیام تقاضا در z امین کنترل کننده مه وجود دارد. شکل ۴ صف مربوط به کنترل کننده مه از نوع $M/M/1/C$ را نشان می‌دهد. با توجه به اینکه تأخیر در کنترل کننده مه که در قسمت قبل گفته شد شامل تأخیر صف می‌باشد، در این قسمت این تأخیر مدل می‌شود.

با استفاده از (۹) تا (۱۴) احتمال حالت پایدار به دست می‌آید

$$\pi_k^j \lambda = \pi_{k-1}^j \mu_{FC} \quad (9)$$

$$\pi_1^j (\lambda + \mu_{FC}) = \pi_0^j \lambda + \pi_1^j \mu_{FC} \quad (10)$$

$$\pi_1^j (\lambda + \mu_{FC}) = \pi_1^j \lambda + \pi_1^j \mu_{FC} \quad (11)$$

$$\pi_k^j (\lambda + \mu_{FC}) = \pi_{k-1}^j \lambda + \pi_{k+1}^j \mu_{FC}, \quad k = 2, \dots, C-1 \quad (12)$$

$$\pi_c^j \mu_{FC} = \pi_{c-1}^j \lambda \quad (13)$$

$$\sum_{k=0}^c \pi_k^j = 1 \quad (14)$$

احتمال حالت پایدار k تقاضای پیام در کنترل کننده z ام از (۱۵) به دست می‌آید. با فرض $\rho = \lambda/\mu$

$$\pi_k^j = \begin{cases} \frac{(1-\rho)\rho^k}{1-\rho^{c+1}}, & \rho \neq 1 \\ \frac{1}{c+1}, & \rho = 1 \end{cases} \quad (15)$$

نرخ فقدان پیام زمانی که c پیام در صف هستند از (۱۶) به دست می‌آید

$$P_{loss}^j = \lambda \pi_c^j = \lambda \frac{1-\rho}{1-\rho^{c+1}} \rho^c, \quad \rho \neq 1 \quad (16)$$

میانگین کارایی سرویس از (۱۷) به دست می‌آید

$$\overline{X_{FC}^j} = \lambda(1 - P_{loss}^j) = \lambda \frac{1-\rho^c}{1-\rho^{c+1}}, \quad \rho \neq 1 \quad (17)$$

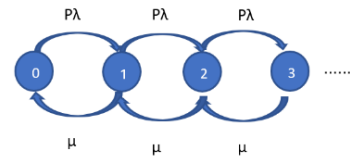
بهره‌وری کنترل کننده مه از (۱۸) به دست می‌آید

$$U_{FC} = \frac{\overline{X_{FC}^j}}{\mu_{FC}} = \rho \frac{1-\rho^c}{1-\rho^{c+1}} \quad (18)$$

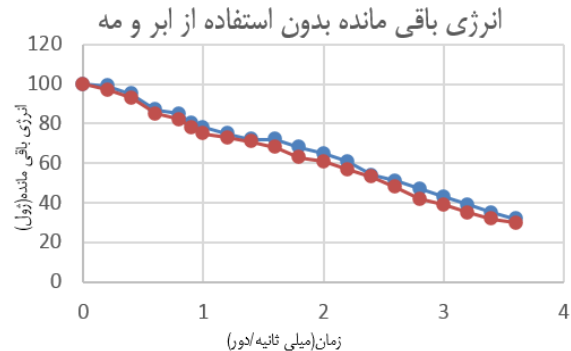
تعداد میانگین تقاضاهای پیام در z امین کنترل کننده مه از (۱۹) به دست می‌آید

جدول ۴: مقادیر مورد نیاز پارامترها برای شبیه‌سازی.

پارامترها	مقادیر
تعداد حسگرها	۱۰۰
کنترل کننده در ابر	۱
اندازه بسته	۴۰۰۰ بیت
انرژی اولیه	۱ ژول
E_c	۵۰ nj/bit
E_{fs}	۵ pj/bit.m ^۲
E_{mp}	۱۳ fj/bit.m ^۲
محیط شبکه	۱۰۰×۱۰۰
مدت شبیه‌سازی	۴۵۰۰ ثانیه
نوع صف	Fifo
اندازه صف	۲۰۰ بسته
نوع آنتن	Omni antenna



شکل ۵: مدل صف کنترل کننده ابر.



شکل ۶: انرژی باقیمانده بدون استفاده از ابر و مه. روشنایی بدون استفاده از ابر و مه (●) سلامتی بدون استفاده از ابر و مه (●)

$$\overline{W}_{cc} = \frac{\overline{M}_{cc}}{p\lambda} = \frac{\eta}{\mu(1-\eta)} \quad (30)$$

میانگین زمان پاسخ درخواست در کنترل کننده ابر توسط (۳۱) به دست می‌آید

$$\overline{T}_{cc} = \frac{\overline{E}_{cc}}{p\lambda} = \frac{1}{\mu(1-\eta)} \quad (31)$$

۴- ارزیابی کارایی

در این قسمت نمودارهای مقایسه با توجه به محیط شبیه‌سازی بررسی می‌شوند.

۴-۱ مشخص کردن پارامترها

در این بخش با استفاده از شبیه‌سازی کامپیوتری، کارایی الگوریتم پیشنهادی نشان داده می‌شود. برای رسیدن به این هدف، شبکه‌ای ساده متشکل از ۱۰۰ گره و ۱ کنترل کننده، تولید شده که در آن گره‌ها به صورت تصادفی توسعه داده شده‌اند. شبیه‌سازی در سیستم عامل Ubuntu ۱۰ توسط شبیه‌ساز ns۲ روی نسخه ماشین مجازی vmware ۱۳ انجام شده است. جدول ۴ تمامی پارامترها و مقادیر مورد نیاز شبیه‌سازی را لیست کرده است.

۴-۲ آنالیز کارایی

برای نشان دادن تأثیر ابر در محاسبات در شبکه‌های اینترنت اشیاى مبتنی بر نرم‌افزار، نمودارهای مربوط به مصرف انرژی و تأخیر بررسی شده‌اند؛ زیرا برای دو کاربرد خانه هوشمند، سلامتی و روشنایی به ترتیب پارامترهای تأخیر و انرژی دارای اهمیت بالایی هستند. در هر دو مورد از الگوریتم مسیریابی Leach برای مسیریابی استفاده می‌گردد.

شکل ۶ انرژی باقیمانده دو کاربرد را در حالت بدون استفاده از ابر نشان می‌دهد. تمامی محاسبات در کنترل کننده انجام می‌شود و به دلیل اینکه تعداد پیام‌های ارسالی به کنترل کننده زیاد می‌باشد، انرژی مصرفی بیشتر است. کاربرد روشنایی و سلامتی نیز فرقی نمی‌کند.

شکل ۷ انرژی باقیمانده در دو کاربرد با استفاده از ابر می‌باشد. به دلیل اینکه انرژی در کاربرد روشنایی اهمیت دارد، با توجه به مقدار پیام‌ها سعی

و با احتمال $1-p$ در کنترل کننده مه انجام می‌شود. صف مربوط به کنترل کننده ابر $M/M/1$ است. یک کنترل کننده ابر وجود دارد که تقاضاها را سرویس می‌دهد. نرخ تقاضای پیام که با فرایند پواسون به کنترل کننده ابر می‌رسند، $p\lambda$ می‌باشد. زمان‌های پردازش مستقل است و به صورت نمایی با نرخ میانگین $1/\mu$ توزیع شده است. صف $M/M/1$ در شکل ۵ دیده می‌شود.

احتمال حالت پایدار سیستم، زمانی که هیچ درخواستی وجود ندارد از (۲۴) به دست می‌آید

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \left(\frac{p\lambda}{\mu}\right)^k} = \frac{1}{1 + \frac{p\lambda}{\mu}} = 1 - \frac{p\lambda}{\mu} \quad (24)$$

احتمال حالت پایداری، زمانی که k تقاضا در سیستم وجود دارد از (۲۵) به دست می‌آید. با فرض $\eta = p\lambda/\mu$

$$\pi_k = \pi_0 \left(\frac{p\lambda}{\mu}\right)^k = (1-\eta) \left(\frac{p\lambda}{\mu}\right)^k = (1-\eta)\eta^k \quad (25)$$

میانگین تعداد پیام که در صف وجود دارد از (۲۶) به دست می‌آید

$$\overline{E}_{cc} = \sum_{k=1}^{\infty} k\pi_k = (1-\eta) \sum_{k=1}^{\infty} k\eta^k = \frac{\eta}{1-\eta} \quad (26)$$

میانگین درخواست‌های منتظر در صف توسط (۲۷) به دست می‌آید

$$\overline{M}_{cc} = \sum_{k=1}^{\infty} k\pi_{k+1} = \frac{\eta^2}{1-\eta} \quad (27)$$

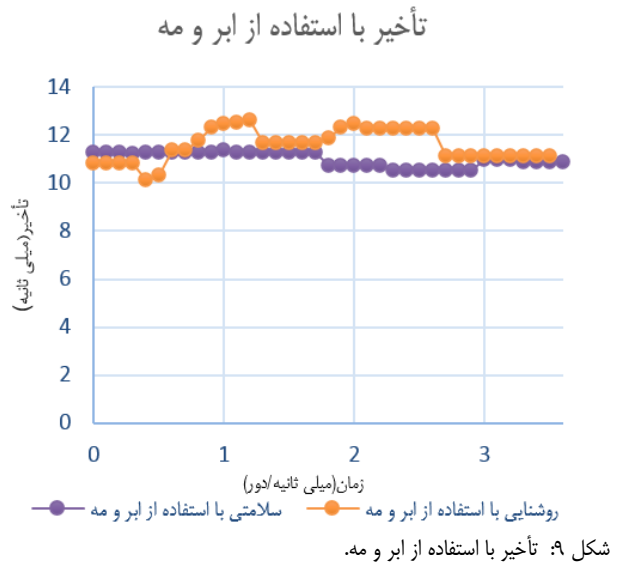
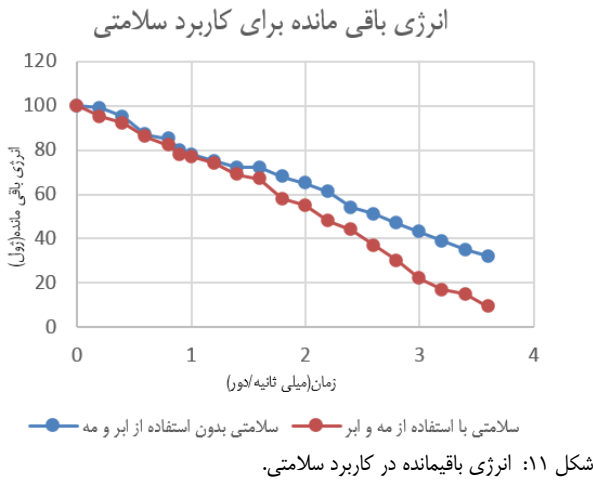
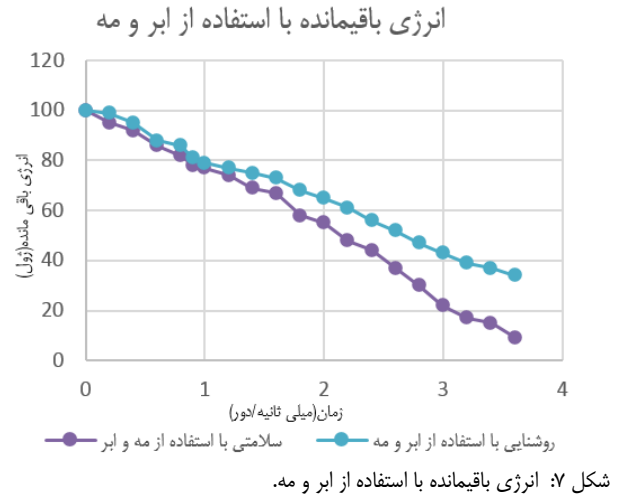
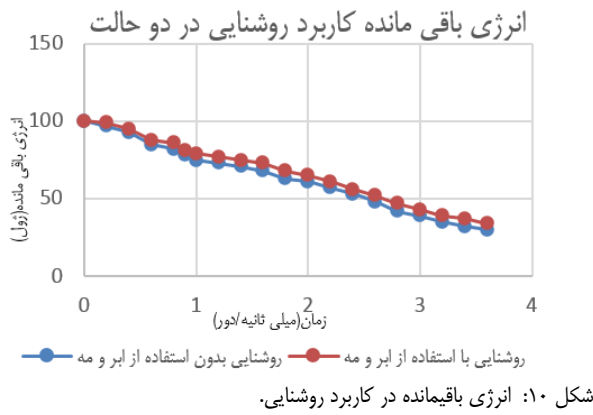
میانگین کارایی سرویس در کنترل کننده ابر توسط (۲۸) به دست می‌آید

$$\overline{X}_{cc} = p\lambda(1-\pi_c) \quad (28)$$

بهرموری صف در زمانی که صف خالی نیست از (۲۹) به دست می‌آید

$$\overline{U}_{cc} = 1 - \pi_c = \eta \quad (29)$$

میانگین زمان انتظار در کنترل کننده ابر برای درخواستی که وارد شده است توسط (۳۰) به دست می‌آید



استفاده از ابر نشان می‌دهد. به دلیل اینکه تأخیر برای کاربرد سلامتی دارای اهمیت بالایی است، محاسبات در مه انجام می‌شود تا سریع‌تر پردازش گردد؛ بنابراین تأخیر پایین‌تر است. ولی برای کاربرد روشنایی به دلیل کم‌اهمیت بودن تأخیر تا جای ممکن محاسبات در ابر انجام می‌شود و بنابراین دارای تأخیر بالاتری می‌باشد.

شکل ۱۰ انرژی باقیمانده را برای کاربرد روشنایی در دو حالت استفاده از ابر و بدون استفاده از آن نشان می‌دهد. همان‌طور که در نمودار دیده می‌شود، انرژی باقیمانده برای حالت استفاده از ابر بیشتر می‌باشد؛ زیرا حسگرها انرژی خود را برای پردازش صرف نمی‌کنند و پیام‌ها در کنترل‌کننده پردازش می‌شود.

شکل ۱۱ انرژی باقیمانده برای کاربرد سلامتی در دو حالت را نشان می‌دهد. در کاربرد سلامتی چون تأخیر اهمیت دارد، بیشتر محاسبات در گره انجام می‌شود. در هر حالت با و بدون استفاده از ابر، ترجیح به پردازش در گره است. انرژی باقیمانده در حالت استفاده از ابر بیشتر می‌باشد؛ زیرا بعضی از بسته‌ها و برنامه‌ها که محدودیت تأخیر آنها به حدی هست که در ابر انجام شود، این کار به ابر سپرده می‌شود ولی بدون استفاده از ابر، تمامی محاسبات در گره است.

شکل ۱۲ تأخیر را برای کاربرد سلامتی در دو حالت با استفاده از ابر و بدون آن نشان می‌دهد. تأخیر در استفاده از ابر بیشتر است؛ زیرا تعدادی برنامه‌ها در ابر انجام می‌شوند، بدون این که در پارامتر تأخیر در آن برنامه مشکلی پیش آید.

شکل ۱۳ تأخیر برای روشنایی را نشان می‌دهد. در کاربرد روشنایی با استفاده از ابر، تمامی پیام‌ها برای ابر ارسال شده و پردازش در ابر اتفاق می‌افتد؛ بنابراین در این حالت نسبت به حالت بدون استفاده از ابر، دارای تأخیر بالایی می‌باشد.

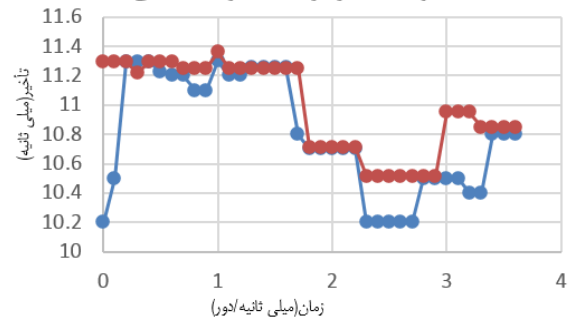
می‌کند محاسبات را در ابر انجام دهد. با فرض اینکه مقدار انرژی مصرفی در ارسال پیام از انجام محاسبات کمتر است. برای کاربرد سلامتی نیز به دلیل اهمیت تأخیر، اگر انرژی گره به حدی است که بتواند جوابگوی محاسبات باشد، در مه انجام می‌شود که تأخیر کمتری را متحمل گردد.

شکل ۸ تأخیر را بدون استفاده از ابر نشان می‌دهد. در این حالت بدون در نظر گرفتن کاربرد بسته‌ها، تمامی بسته‌ها و محاسبات در مه پردازش می‌شوند؛ بنابراین تأخیر پایین است. شکل ۹ تأخیر را در دو کاربرد با

Yang, L., Esposito, A. (eds) *Internet of Everything* Springer, Singapore, pp. 103-130, 2018.

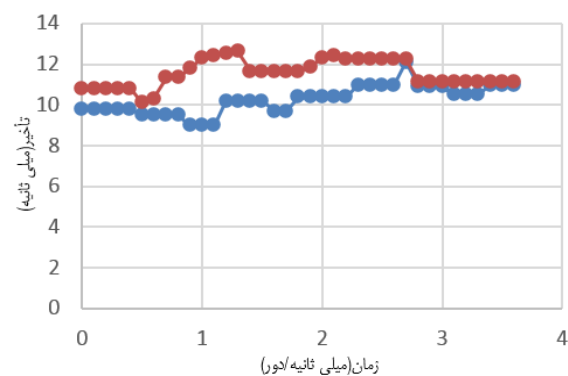
- [4] H. R. Arkian, A. Diyanat, and A. Pourkhalili, "MIST: fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications," *J. Netw. Comput. Appl.*, vol. 82, pp. 152-165, 15 Mar. 2017.
- [5] V. Kumar, A. Laghari, S. Karim, M. Shakir, and A. A. Brohi, "Comparison of fog computing & cloud computing," *Int. J. Math. Sci. Comput.*, vol. 5, no. 1, pp. 31-41, Jan. 2019.
- [6] G. Javadzadeh and A. M. Rahmani, "Fog computing applications in smart cities: a systematic survey," *Wireless Networks*, vol. 26, no. 2, pp. 1433-1457, Feb. 2020.
- [7] A. Mubarakali, et al., "Fog-based delay-sensitive data transmission algorithm for data forwarding and storage in cloud environment for multimedia applications," *Big Data*, vol. 11, no. 2, pp. 128-136, Apr. 2020.
- [8] B. Ali, M. A. Pasha, S. ul Islam, H. Song, and R. Buyya, "A volunteer supported fog computing environment for delay-sensitive IoT applications," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3822-3830, Sept. 2020.
- [9] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: a computation offloading game," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3246-3257, May 2018.
- [10] Y. Wang, et al., "Cooperative task offloading in three-tier mobile computing networks: an ADMM framework," *IEEE Trans. on Vehicular Technology*, vol. 68, no. 3, pp. 2763-2776, Mar. 2019.
- [11] X. Li, et al., "Optimizing resources allocation for fog computing-based Internet of Things networks," *IEEE Access*, vol. 7, pp. 64907-64922, 2019.
- [12] E. Barros, et al., "Fog computing model to orchestrate the consumption and production of energy in microgrids," *Sensors*, vol. 19, no. 11, Article ID: 2642, Jun. 2019.
- [13] F. Karatas and I. Korpeoglu, "Fog-based data distribution service (F-DAD) for internet of things (IoT) applications," *Future Generation Computer Systems*, vol. 93, pp. 156-169, Apr. 2019.
- [14] T. Nguyen, L. Le, and Q. Le-Trung, "Computation offloading in MIMO based mobile edge computing systems under perfect and imperfect CSI estimation," *IEEE Trans. on Services Computing*, vol. 14, no. 6, pp. 2011-2025, Jan. 2019.
- [15] Q. Wang and S. Chen, "Latency-minimum offloading decision and resource allocation for fog-enabled Internet of Things networks," *Trans. on Emerging Telecommunications Technologies*, vol. 31, no. 12, Article ID: 3880, Dec. 2020.
- [16] J. Zhou, X. Zhang, and W. Wang, "Joint resource allocation and user association for heterogeneous services in multi-access edge computing networks," *IEEE Access*, vol. 7, pp. 12272-12282, 2019.
- [17] M. Faraji Mehmandar, S. Jabbehari, and H. Haj Seyyed Javadi, "A dynamic fog service provisioning approach for IoT applications," *International J. of Communication Systems*, vol. 33, no. 14, Article ID: e4541, Jul. 2020.
- [18] A. Kishor and C. Chakarbarti, "Task offloading in fog computing for using smart ant colony optimization," *Wireless Personal Communications*, vol. 127, no. 2, pp. 1683-1704, Nov. 2022.
- [19] W. Bai, et al., "Joint optimization of computation offloading, data compression, energy harvesting, and application scenarios in fog computing," *IEEE Access*, vol. 9, pp. 45462-45473, 2021.
- [20] A. R. Hameed, S. Islam, I. Ahmad, and K. Munir, "Energy-and performance-aware load-balancing in vehicular fog computing," *Sustainable Computing: Informatics and Systems*, vol. 30, Article ID: 100454, Jun. 2020.
- [21] R. Qun and S. M. Arefzadeh, "A new energy-aware method for load balance managing in the fog based vehicular ad hoc networks (VANET) using a hybrid optimization algorithm," *IET Communications*, vol. 15, no. 13, pp. 1665-1676, Aug. 2021.
- [22] S. Azizi, et al., "Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: a semi-greedy approach," *J. of Network and Computer Applications*, vol. 201, Article ID: 103333, May 2022.
- [23] M. Anoushee, M. Fartash, and J. Akbari Torkestani, "An intelligent resource management method in SDN based fog computing using reinforcement learning," *Computing*, vol. 106, no. 4, pp. 1051-1080, Apr. 2024.
- [24] P. Linh-An, N. Duc-Thang, L. Meonghun, D. Park, and T. Kim, "Dynamic fog-to-fog offloading in SDN-based fog computing systems," *Future Generation Computer Systems*, vol. 117, pp. 486-497, Apr. 2021.
- [25] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of the 33rd Annual Hawaii Int. Conf. on System Sciences*, vol. 2, 10 pp., Maui, HI, USA, 7-7 Jan. 2000.

نمودار تأخیر برای کاربرد سلامتی



شکل ۱۲: تأخیر در کاربرد سلامتی.

تأخیر برای کاربرد روشنایی



شکل ۱۳: تأخیر در کاربرد روشنایی.

۵- نتیجه‌گیری

این پژوهش با توجه به معماری پیشنهادی برای شبکه‌های اینترنت اشیا نرم‌افزارمحور و استفاده از ابر و مه به کاهش زمان پاسخ درخواست‌ها می‌پردازد. زمان پاسخ علاوه بر تأخیر پردازش به تأخیر ارتباطات هم وابسته است. سناریوی پیشنهادی و دیدی دیگر از معماری که در این پژوهش به آن اشاره می‌شود، به درک زمان پاسخ کمک می‌کند. در ادامه با استفاده از نظریه صف، مدل را برای محاسبات ابر و مه تحلیل کرده و کیفیت سرویس‌های مورد انتظار شبکه را با استفاده از فرمول‌های ریاضی به دست می‌آورد. برای شبیه‌سازی با استفاده از نرم‌افزار NS۲ تأخیر و انرژی باقیمانده را توسط نمودارها نشان داده و مقایسه می‌کند. در این نمودارها تأثیر استفاده از محاسبات ابری و مه برای اینترنت اشیا آمده است. این مقایسه در کاربردهای مختلف اینترنت اشیا، حساس به تأخیر (سلامتی) و غیرحساس به تأخیر (روشنایی) نشان داده شده است.

مراجع

- [1] M. R. Rezaee, N. A. W. Abdul Hamid, M. Hussin, and Z. A. Zukarnain, "Fog offloading and task management in IoT-fog-cloud environment: review of algorithms, networks, and SDN application," *IEEE Access*, vol. 12, pp. 39058-39080, 2024.
- [2] A. Khakimov, et al., "IoT-fog based system structure with SDN enabled," in *Proc. of the 2nd In. Conf. on Future Networks and Distributed Systems*, ArticleID: 62, 6 pp., Jun. 2018.
- [3] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: a taxonomy, survey and future directions," in *Di Martino, B., Li, K.C.,*

الهام حاجیان تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد مهندسی کامپیوتر به ترتیب در سال‌های ۱۳۸۵ و ۱۳۸۹ از دانشگاه خوارزمی و دانشگاه اصفهان به پایان رسانده است و همچنین مدرک دکتری خود را از دانشگاه اصفهان در سال ۱۴۰۲ اخذ کرده است. او هم‌اکنون عضو هیأت علمی دانشکده فنی و مهندسی دانشگاه بجنورد می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: شبکه‌های کامپیوتری بی‌سیم، شبکه‌های نرم‌افزارمحور، اینترنت اشیا و کیفیت سرویس در شبکه‌ها.