Archive of SID

مقایسه کاهش تاخیر در ارتباطات از نوع ماشین با استفاده از محاسبات مه و محاسبات ابری

زهرا عارفیان[1] محمدرضا خیام باشی[2]

۱- دانشجوی دکتری- دانشکده مهندسی کامپیوتر- دانشگاه اصفهان- اصفهان- ایران
zarefian@eng.ui.ac.ir

۲- دانشیار- دانشکده مهندسی کامپیوتر- دانشگاه اصفهان- اصفهان- ایران
M.R.Khayyambashi@comp.ui.ac.ir

چکیده: تعداد ماشین‌ها و اشیایی که از اینترنت استفاده می‌کنند، در سال‌های اخیر به شدت رو به افزایش است. ارتباطات از نوع ماشین و اینترنت اشیا در همه زمینه‌ها به ویژه در صنعت رشد کرده است. در این نوع ارتباطات، برخی از برنامه‌های کاربردی حساس به تاخیر می‌باشند و باید به صورت زمان حقیقی پاسخ موردنظر خود را دریافت نمایند و به نیازهای محاسباتی آن‌ها پاسخ داده شود، از طرفی با توجه به حجم بالای اطلاعات تولید شده توسط دستگاه‌های هوشمند، نحوه ذخیره‌سازی و پهنای‌باند نیز مطرح می‌باشد. سرویس‌های محاسبات ابری در برخورد با این موارد با مشکلاتی مواجه است. محاسبات مه در کنار محاسبات ابری راه‌حلی قابل قبول است و به عنوان میان‌افزار بین ابر و ماشین‌ها قرار گرفته است. در این مقاله کاهش تاخیر با استفاده از محاسبات مه برای ارتباطات از نوع ماشین مورد بررسی قرار گرفته که عاملی برای کاهش مصرف پهنای باند و نیز کاهش تاخیر می‌باشد. شبیه‌سازی-های صورت گرفته علاوه بر کاهش تاخیر برای این نوع ارتباطات با استفاده از محاسبات مه، به افزایش تعداد درخواست‌های سرویس داده شده نسبت به محاسبات ابری پرداخته است.

واژه‌های کلیدی: ارتباطات از نوع ماشین؛ محاسبات مه؛ کاهش تاخیر؛ مصرف پهنای باند؛ محاسبات ابری.

نوع مقاله: پژوهشی

DOI: 10.29252/jiaeee.18.3.1217

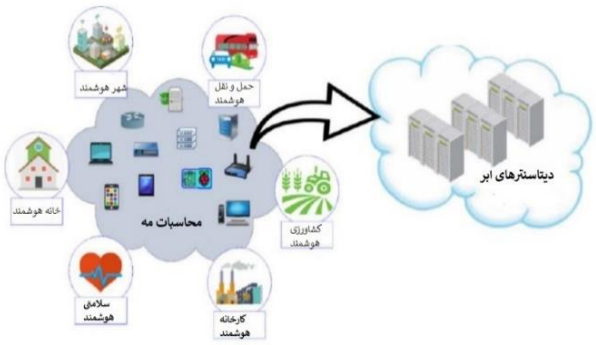
تاریخ ارسال مقاله: ۱۳۹۹/۰۸/۲۴
تاریخ پذیرش مشروط مقاله: ۱۳۹۹/۱۰/۲۷
تاریخ پذیرش مقاله: ۱۳۹۹/۱۲/۰۹
نام نویسنده‌ی مسئول : دکتر محمدرضا خیام باشی
نشانی نویسنده‌ی مسئول: عضو هیئت علمی گروه معماری کامپیوتر، دانشکده کامپیوتر، دانشگاه اصفهان، اصفهان

www.SID.ir

Archive of SID

۱- مقدمه

در سال‌های اخیر ارتباطات از نوع ماشین (MTC[۱]) و یا ارتباطات ماشین با ماشین (M2M[۲]) در همه زمینه‌ها فراگیر شده است و نقش مهمی در شبکه‌های نسل ۵[۳] دارد. قبل از اهمیت ارتباطات MTC در دهه‌های اخیر، ارتباطات دستگاه با دستگاه مطرح بود که توسط آن دستگاه‌ها با یکدیگر از طریق شبکه بی‌سیم در ارتباط بودند، که تعداد آن‌ها هر ساله رو به افزایش بود[۱]. با توجه به جامعیت ماشین‌های MTC از نظر اهمیت به سرعت جای آن را گرفت. با افزایش تعداد دستگاه‌ها و حسگرهای MTC باید بتوان نیازهای محاسباتی، ذخیره-سازی و انتقال آن‌ها را رفع کرد. برخی از برنامه‌های کاربردی حساس به تاخیر می‌باشند و باید به صورت زمان حقیقی[۴] به آن‌ها پاسخ داد[۳] [۲]. با توجه به حجم بالای اطلاعات تولید شده توسط دستگاه‌های هوشمند، چالش‌هایی از جمله مصرف باتری، توان مصرفی، فضای ذخیره‌سازی و پهنای باند مطرح است، که همگی از جمله موانع رسیدن به کیفیت سرویس دهی (QoS[۵]) مطلوب می‌باشند. حجم بالای داده مساله را به سمت داده‌های بزرگ[۶] پیش می‌برد. در این نوع شبکه‌ها سه بعد حجم، سرعت و تنوع وجود دارد، علاوه بر این در شبکه‌های MTC توزیع جغرافیایی نیز به چشم می‌خورد. با وجود این موارد محاسبات ابری برای پردازش این نوع درخواست‌ها به خوبی عمل نمی‌کند. محاسبات مه[۷] می‌تواند راه‌حلی قابل قبول برای غلبه بر این چالش‌ها باشد[۴] [۵]. این مفهوم برای اولین بار در سال ۲۰۱۲ توسط Cisco ارائه گردید که محاسبات، نزدیک دستگاه‌ها و حسگرها انجام می‌شود. از نظر جایگاه سلسله‌مراتبی بین ابر و ماشین‌های MTC به عنوان یک میان افزار[۸] قرار دارد. محاسبات و پردازش‌ها در مه، به صورت توزیع شده صورت می‌پذیرد و هدفش توجه و رفع چالش‌های مربوط به MTC و ابر می‌باشد[۳]. مه در عمل یک ابر کوچکی است که در لبه‌ها به عنوان مکملی در کنار ابر گسترش داده است، و عامل ارتباط میلیاردها دستگاه ناهمگن با یکدیگر است، همانطور که در شکل ۱ مشاهده می‌شود. از محاسبات مه می‌توان در تمامی کاربردهای IoT مثل شهرهای هوشمند[۹]، خانه‌های هوشمند[۱۰] ، شبکه‌های هوشمند، وسایل نقلیه بدون سرنشین[۱۱] و مراقبت‌های سلامتی[۱۲] بهره گرفت [۶], [۷]. برخی از برنامه‌های کاربردی حساس به تاخیر و زمان هستند و اگر انجام محاسبات و ذخیره‌سازی‌های مورد نظرشان در ابر انجام گیرد، نتیجه مطلوبی ندارد و منجر به تاخیر بالا و QoS پایین می‌شوند. بدین منظور انجام محاسبات و ذخیره‌سازی بر اساس ابر در نزدیکی دستگاه‌ها انجام می‌گیرد. شبکه‌های مه از اجزای سنتی و معمول مانند مسیریاب‌ها، سوئیچ‌ها، سرور پروکسی‌ها، ایستگاه‌های پایه[۱۳] و ... تشکیل شده است. این نوع شبکه‌ها قابلیت حرکت، مقیاس پذیری و قابلیت همکاری دارند [۳] و امکان محاسبات، ذخیره‌سازی و منابع شبکه را از طریق کاربران مختلف پیاده‌سازی می‌کنند. یک اتصال پهن و کوتاه به جای اتصال باریک و بلند بین کاربران و مه برقرار است[۸]. تکنولوژی مه جایگزین تکنولوژی ابر نیست و این دو در واقع مکمل یکدیگر می‌باشند. سرویس‌هایی که حساس به زمان هستند و برای پاسخ به زمان حقیقی نیاز دارند از تکنولوژی مه استفاده می‌کنند و سرویس‌هایی که به فضای ذخیره‌سازی زیاد و محاسبات و تجزیه و تحلیل پیچیده نیاز دارند, به دلیل منابع غنی و فضای کافی از ابر استفاده می‌کنند[۹].



شکل (۱): کاربردهای استفاده از محاسبات مه [۱۰]

مه به صورت یک معماری ناهمگن در لبه‌ها توزیع شده است، و امکان مخفی‌سازی ناهمگنی‌ها را دارد، به طوری‌که به صورت یک ساختار یکپارچه، با قابلیت مدیریت و برنامه‌ریزی به نظر رسد. به عبارت دیگر محاسبات مه را می‌توان به صورت، سیاست توزیع شده با دید جهانی و اجرا و پیاده‌سازی محلی تعریف کرد. سیستم محاسبات مه، مانند محاسبات ابری، می‌توان منابع را اجاره کرده و هزینه‌ها از این طریق قابل تامین می‌باشد[۱۱]. سرورهای مه برای اهداف مختلفی در مکان-های مختلف استفاده می‌شوند، این سرورها باید قادر به برقراری ارتباط با کاربران و دستگاه‌ها، سرورهای دیگر و ابر باشند[۸]. در جدول شماره ۱ مقایسه‌ایی از ویژگی‌های ابر و مه به طور خلاصه بیان شده است. روش پیشنهادی بر روی مساله کاهش هزینه تاخیر و تعادل بار تمرکز می‌کند. دو نوع درخواست، حساس به تاخیر و غیرحساس به تاخیر در این روش استفاده شده است. با توجه به اینکه درخواست‌های ارسالی از طرف ماشین‌های MTC از کدام دسته باشند، به لایه مه یا لایه ابر ارسال می‌شوند.

جدول(۱): مقایسه ویژگی‌های ابر و مه [۱۲]

ویژگی‌ها	محاسبات ابری	محاسبات مه
تاخیر	بالا	کم (به صورت محلی)
دسترسی به شبکه	بیسیم و سیمی	عموما بی سیم
محل سرورها	دور و از طریق اینترنت	در لبه و به صورت محلی
دسترس پذیری	٪۹۹,۹	نسبت به ابر خیلی کمتر
مصرف پهنای باند	بالا	کم
امنیت	امنیت پایین و تعریف نشده	امنیت بالاتر و تعریف شده
میزان منابع	بسیار زیاد و قوی	کمتر و محدودتر
قیمت هر سرور	۱۵۰۰-۳۰۰۰ دلار	۵۰-۲۰۰ دلار

ساختار مقاله به صورت بیان شده می‌باشد: در بخش ۲ به بیان کارهای پیشین در این زمینه پرداخته می‌شود در بخش ۳ به بیان روش

www.SID.ir

Archive of SID

پیشنهادی پرداخته می‌شود در بخش ۴ شبیه‌سازی‌ها بیان می‌گردد و نهایتا نتیجه‌گیری در بخش ۵ مطرح می‌شود.

۲- کارهای پیشین

یکی از اهداف اصلی که باعث به وجود آمدن محاسبات مه در کنار محاسبات ابری شد، کاهش تاخیر می‌باشد. در زیر به بررسی برخی از مقالات که به این موضوع توجه کرده است پرداخته می‌شود:

در مقاله [۱۳] دنگ و همکاران روش خود را با بیان ویژگی‌های شبکه‌های مه مانند، تعاملات[۱۴] زمان حقیقی، لختی[۱۵] کم و نرخ سرویس بالا برای کاربران موبایل آغاز کردند. شبکه‌های مه شبیه یک شبکه ابری سبک وزن است که می‌تواند با یک گام با رابط‌هایی از جمله بلوتوث[۱۶] و وایفا[۱۷] به کاربران سرویس ارائه دهد. شبکه مه جایگزین ابر نیست بلکه به عنوان مکملی برای آن می‌باشد و به منظور کاهش بار پهنای باند و کاهش لختی انتقال استفاده می‌شود. در این مقاله توان مصرفی و تاخیر انتقال مورد بررسی قرار گرفته است و حالت بهینه‌ایی برای هر دو پارامتر فرموله شده است، به طوری‌که با تقسیم مساله به چند زیر مساله قابل حل می‌باشد. ابتدا یک چارچوب ریاضی برای ارزیابی مساله تقابل توان و تاخیر با در نظر گرفتن بار کاری بدست آمده است، سپس با استفاده از رویکرد تقریبی به حل آن پرداخته‌اند و نهایتاً کاهش تاخیر نتیجه‌ایی است که از شبیه‌سازی‌ها بدست آمده است. هنگام بررسی مدل ریاضی این روش توان مصرفی برای هر دستگاه در شبکه مه به صورت یک تابع درجه دو بوده، که این تابع با پیچدگی محاسبات و مصرف توان برای دستگاه‌های جانبی ارتباط مستقیم دارد. تابع تاخیر برای دستگاه‌های شبکه مه به صورت تئوری صف مدل شده است.

در مقاله[۱۴] مبارکعلی و همکاران از شبکه مه به عنوان شبکه‌ایی که نقش اساسی در انتقال و توزیع داده‌ها در IoT دارد، و به عنوان لایه میانی بین ابر و اشیا به منظور افزایش سرعت برای برنامه های حساس به زمان، یاد کردند. برای نظارت بر این شبکه‌ها از شبکه‌های SDN و مکانیزیم مجازی سازی استفاده شده است. در این مقاله با استفاده از پیش‌بینی و به کارگیری الگوریتم‌های انتقال برنامه‌های حساس به تاخیر به کاهش تاخیر دست یافته‌اند.

استفاده از شبکه، مصرف انرژی و تاخیر ارتباطی از جمله مواردی است که در مقاله [۱۵] توسط بابارعلی و همکاران به آن توجه شده است. در این مقاله از روش VSFC[۱۸] در شبکه‌های مه به منظور کاهش تاخیر استفاده شده است، به طوری‌که با استفاده از منابع محاسباتی در لبه مصرف شبکه و تاخیر را کاهش می‌دهد. آن‌ها با اضافه کردن افزونه‌ایی به شبیه ساز iFogSim روش پیشنهادی خود را مورد ارزیابی قرار داده‌اند.

در مقاله[۱۶] شاه‌منصوری و همکاران به بررسی محاسبات سلسله-مراتبی شبکه مه و شبکه ابری برای دستگاه‌ها و سیستم‌های IoT با استفاده از تئوری بازی پرداخته‌اند. شبکه مه به منظور کاهش تاخیر و ترافیک زیرساخت[۱۹] شبکه استفاده شده است، ولی مدیریت منابع در این شبکه یکی از چالش‌های مهم است. علاوه بر این در شبکه مه هر کاربر به طور خودخواهانه سعی در رسیدن به QoE حداکثری برای خود بدون توجه به اولویت‌ها و دیگر کاربران دارد. به منظور افزایش رضایت‌مندی کاربران از خدمات ارائه شده از محاسبات تخلیه بار ترافیکی[۲۰] بهره گرفته شده است، که در آن هر گره IoT می‌تواند یکی از شبکه‌های مه و یا شبکه ابر را با توجه به کیفیت خدمات مورد نظرش انتخاب کند. مراحل اجرای طرح پیشنهادیشان به این صورت انجام گرفته است، ابتدا برای رسیدن به مقدار حداکثری QoE[۲۱] مساله را فرموله کرده تا تصمیمات تخلیه بار ترافیکی برای هر کاربر اتخاذ گردد. در این مقاله منظور از QoE کاهش تاخیر و انرژی محاسباتی می‌باشد. سپس برای تمامی کاربران به منظور تشخیص انتخاب درست، تخلیه بار ترافیکی، مدل بازی و ویژگی‌هایش فرموله شده و آنالیز شده است و در ادامه تعادل نش با توجه به رشد تعداد کاربران جهت کاهش تاخیر به کار گرفته شده است.

در مقالات[۱۳] [۱۷] [۱۸] مه را یک ابر نزدیک به زمین معرفی کرده و آن را برای محاسبات سبک وزن مناسب دانسته‌اند. سرورهایی که در مه وجود دارد مناسب برای کاربران موبایل است. به طوری‌که باید با کیفیت بالا و هزینه کم برنامه‌های کاربردی مانند ویدئو را با تاخیر کم در اختیار کاربران قرار دهد. در این مقالات محاسبات مه را یک ساختار سه بعدی موبایل، مه و ابر معرفی شده است؛ که در آن کاربر تلفن همراه از سرورهای مه با استفاده از اتصالات بیسیم محلی سرویس می‌گیرند و سرورهای مه درخواست‌های خود را از ابر با استفاده از شبکه‌های سلولی یا سیمی دریافت می‌کنند. به طور خاص در مقاله [۱۸]، شبکه محاسباتی مه به دو بخش تقسیم می‌شود، پنل کنترل به صورت مرکزی در ابر و پنل داده در مه جایگزاری می‌گردد. جایی که ابر یک کنترل‌کننده است و بدین ترتیب تاخیر را کنترل می-کند. در این مقاله به طور کلی تاخیر در دو بعد مسافت و طول صف‌ها را عنوان می‌کند و حجم داده‌های منتظر برای پردازش کاهش می‌یابد.

در مقاله[۱۹] نیز بحث تاخیر سرویس‌های ابر مطرح شده است. مه با قرارگیری بین ابر و اشیا باعث دسترسی با سرعت بالا و نزدیکی فیزیکی به کاربران شده است، به‌طوری‌که قادر است به برنامه‌های کاربردی به صورت زمان واقعی خدمات ارائه دهد و قابلیت حرکت را نیز حمایت کند. برای کاهش تاخیر و افزایش کیفیت ارائه خدمات به زمان‌بندی داده‌ها پرداخته است که برای این امر از شبکه‌های مبتنی بر نرم افزار کمک گرفته شده است.

روش‌های بسیاری برای کاهش تاخیر مبتنی بر شبکه‌های مه پیشنهاد شده است ولی در بیشتر آن‌ها به مدیریت درخواست‌های ارسالی بر اساس نوع درخواست کمتر توجه شده است. در بخش بعدی به بیان روش پیشنهادی بر اساس تقسیم‌بندی پردازش درخواست‌ها بر اساس نوع آن‌ها پرداخته شده است.

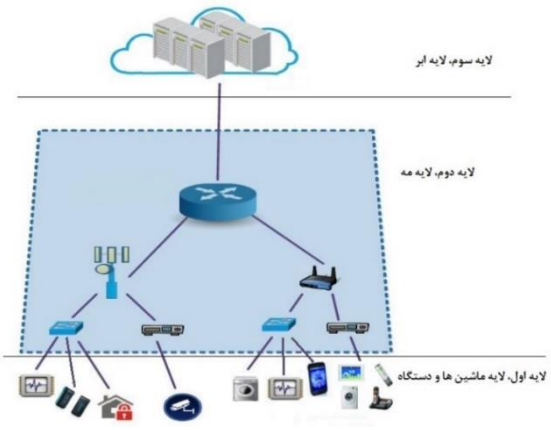
www.SID.ir

Archive of SID

۳- روش پیشنهادی و معماری سیستم

ساختار سلسله‌مراتبی ماشین‌ها، مه و ابر در شکل ۲ مشخص است. همان‌طور که در این شکل مشاهده می‌شود معماری شبکه در سه لایه تعریف گردیده است. در لایه اول تمامی دستگاه‌ها و ماشین‌ها (یا سنسورها و اکچئویترها) قرار گرفته‌اند که می‌توانند داده و سیگنال تولید کرده و این اطلاعات را برای پردازش به لایه بالاتر ارسال نمایند. لایه دوم به عنوان یک میان‌افزار که به لایه مه معروف است، برای پردازش اطلاعات و کارهای شبکه‌ایی و ذخیره‌سازی استفاده می‌شود. این لایه به صورت توزیع شده قرار گرفته و نزدیک به ماشین‌ها می‌باشد به طوریکه علاوه بر فاصله، مصرف پهنای باند و تاخیر پردازش نیز کاهش می‌یابد. در حقیقت یک اتصال کوتاه و باریک به جای اتصال بلند و پهن جایگزین شده است. این لایه ظرفیت محاسبه و ذخیره‌سازی محدودتری نسبت به ابر دارد. لایه سوم که به لایه ابر معروف است، دارای سرورهای بسیار قوی و منابع ذخیره‌سازی نامحدود می‌باشد. درخواست‌های لایه اول می‌توانند با واسطه لایه دوم یا بدون واسطه به لایه ابر ارسال گردد و از منابع غنی برای پردازش بهره بگیرد. مدیریت معماری سه لایه را می‌توان به طریق زیر اعمال کرد:

- اگر درخواستی حساس به زمان باشد، آن درخواست به سمت لایه مه ارسال می‌شود. حال اگر منابع موجود در مه بتوانند قبل از سررسید زمان پایان[۲۲] در خواست را پردازش کنند، به پردازش آن می‌پردازند، در غیر این صورت آن را رد می‌کنند[۲۳].
- اگر درخواست ارسالی از طرف ماشین‌ها حساس به زمان نباشد آن درخواست به لایه ابر فرستاده می‌شود.



شکل (۲): ساختار سلسله‌مراتبی ماشین‌ها، مه و ابر

با انجام این کار از همان ابتدا درخواست‌ها به دو دسته تقسیم می‌شوند. درخواست‌های غیرحساس به زمان که زمان پاسخ[۲۴] برایشان اهمیتی ندارد به ابر ارسال شده و فقط درخواست‌های حساس به زمان در مه پردازش می‌گردد. با انجام این کار می‌توان منابع محدود مه به خوبی مدیریت شده و درخواست‌های حساس به زمان بیشتری در آن پردازش می‌شود. اگر لایه مه قادر به پردازش درخواست‌های حساس به زمان قبل از اتمام زمان آن‌ها نباشد آن درخواست‌ها را رد می‌کند، بنابراین وقت خود را صرف پردازش بدون نتیجه نمی‌کند. با این روش تعداد درخواست‌های منتظر پردازش پشت منابع مه کاهش یافته و بنابراین طول صف کمتر می‌شود، در نتیجه تاخیر سرویس کمتر شده و درخواست‌های بیشتری پردازش می‌شوند. سرورها و منابع موجود در ابر بسیار قدرتمند و نامحدود می‌باشد و برای درخواست‌های غیر حساس به تاخیر که تاخیر انتقال برای آنها حائز اهمیت نمی‌باشد مناسب‌تر است.

ساختار شبکه مدل پیشنهادی به صورت زیر تعریف شده است:

- **T** مجموعه درخواست‌ها و کارهایی که از طرف ماشین‌ها ارسال می‌گردد: $T = \{t_1, t_2, \ldots, t_n\}$
- n تعداد کل درخواست‌ها و **n'** تعداد درخواست‌های حساس به تاخیر؛
- **V** مجموعه حجم درخواست‌های ماشین‌ها: $V = \{v_1, v_2, \ldots, v_n\}$؛
- **S** تعداد منابع لایه مه $S = \{s_1, s_2, \ldots, s_m\}$
- **C** ظرفیت سرورها در لایه مه $C = \{c_1, c_2, \ldots, c_m\}$
- **D** فاصله ماشین‌ها تا لایه مه $D = \{d_1, d_2, \ldots, d_{n'}\}$ و **d** فاصله ماشین‌ها تا لایه ابر $d = \{d_1, d_2, \ldots, d_{n-n'}\}$؛
- **C** سرعت انتشار تا لایه مه و **c'** سرعت انتشار تا لایه ابر؛
- **W** پهنای باند بین دستگاه‌ها تا لایه مه؛
- **W'** پهنای باند بین دستگاه ها تا لایه ابر.

ظرفیت پردازشی سرورها در لایه ابر نامحدود فرض می‌شود به طوریکه به محض ارسال درخواست به ابر، پردازش انجام شده و پاسخ آن به لایه اول ارسال می‌گردد، بنابراین تاخیر پردازشی و تاخیر صف در این لایه نادیده گرفته می‌شود. تاخیر کل در این لایه توسط معادله ۱ محاسبه شده است.

$$T_{total1} = 2 \times (n\text{-}n') \sum_{j=1}^{n-n'} \frac{d_j}{c'} + \sum_{i=1}^{n-n'} \frac{V_i}{W'} + T_{Path}, \quad (1)$$

$$T_{Path} = \frac{1}{\mu_{min} - \lambda}$$

در معادله ۱، $2 \times (n\text{-}n') \sum_{j=1}^{n-n'} \frac{d_j}{c'}$ برابر تاخیر انتشار از ماشین‌های MTC تا لایه ابر که در آن و $\sum_{i=1}^{n-n'} \frac{V_i}{W'}$ برابر تاخیر انتقال است. با توجه به فاصله ماشین‌ها تا ابر بیشترین هزینه تاخیر به تاخیر انتشار تعلق می‌گیرد. در این معادله T_{Path} زمان پردازش در صف سوئیچ‌های مسیر از ماشین MTC تا ابر می‌باشد، که در آن λ برابر نرخ ورود درخواست‌ها به صف سوئیچ‌های مسیر و μ نرخ سرویس کمینه

www.SID.ir

Archive of SID

سوئیچ‌های مسیر می‌باشد. در صورتی که تعداد درخواست‌های بیشتری به ابر ارسال شود این قسمت به نسبت تاخیر انتشار و انتقال سهم بیشتری در تاخیر کل را به خود اختصاص می‌دهد.

زمان پاسخ برای درخواست‌های حساس به تاخیر که در لایه مه پردازش می‌شود توسط معادله ۲ محاسبه می‌شود. تاخیر سرویس برابر با مجموع تاخیر انتشار، تاخیر پردازش، تاخیر انتقال و تاخیر صف می‌باشد. از آنجایی که تاخیر انتشار بسیار ناچیز است از آن صرف‌نظر می‌گردد. معادله ۳ پردازش یا عدم انجام پردازش درخواست‌های حساس به تاخیر را در این لایه مورد ارزیابی قرار می‌دهد.

$$T_{tota2} = \frac{\sum_{i=1}^{k} vi}{C \times m} + \sum_{i=1}^{n'} \frac{V_i}{W} + \frac{1}{\mu - \lambda} + 2n' \sum_{j=1}^{n'} \frac{D_j}{C} \quad (۲)$$

$$T = T_{Deadline} - T_{total2} \quad (۳)$$

در معادله ۲، $\frac{\sum_{i=1}^{k} vi}{C \times m}$ برابر تاخیر پردازش در سرورهای مه می‌باشد که به ظرفیت پردازشی سرورها، تعداد منابع و حجم درخواست‌ها وابسته است. $\sum_{i=1}^{n'} \frac{V_i}{W}$ برابر تاخیر انتقال که به پهنای باند دستگاه‌ها تا لایه مه و حجم درخواست‌ها وابسته است. $2n' \sum_{j=1}^{n'} \frac{D_j}{C}$ برابر تاخیر انتشار تا لایه مه می‌باشد، به علت فاصله بسیار کم ماشین‌ها تا لایه مه تاثیری در تاخیر سرویس ندارد و حتی در مواردی می‌توان از آن صرف‌نظر کرد. $\frac{1}{\mu - \lambda}$ برابر تاخیر صف پشت سر منابع مه می‌باشد، تاخیر صف وابسته به نرخ ورود و نرخ سرویس توسط سرورهای مه است. صف پشت سر منابع به صورت M/M/1 مدل شده و درخواست‌ها به صورت اولین ورودی اولین سرویس (FCFS) پردازش می‌شوند. در معادله ۳ اگر T بزرگتر از صفر باشد درخواست ارسالی در لایه مه پردازش می‌شود، در غیر این صورت آن را باز پس می‌فرستد.

تاخیر کل برابر مجموع تاخیر درخواست‌هایی که در لایه مه پردازش می‌شود به اضافه درخواست‌هایی که در لایه ابر پردازش می‌گردند. می‌توان توسط شبه کد زیر مسیر کار را دنبال کرد:

Input (T,V,S,C) & Output Response time in fog and cloud
{
For i=1 to n
If ti don't delay sensitive then send to cloud and calculate Eq.1
Else ti delay sensitive send to fog and calculate Eq.2
If Ttotal2 < deadline then send to Resource queue
Else drop
}

۴- شبیه‌سازی

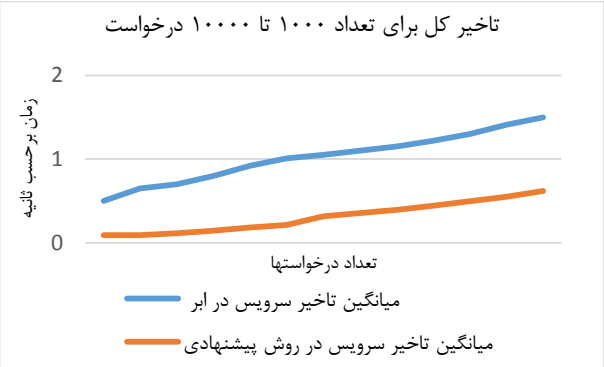
در این بخش جزییاتی از شبیه سازی انجام گرفته، مورد توجه قرار می‌گیرد. برای انجام ارزیابی‌ها از شبیه‌ساز IFogSim استفاده می‌شود [10][20]. زیرا این شبیه‌ساز برای محاسبات مه اخیرا بسیار مورد توجه قرار گرفته است. بنیاد IFogSim بر اساس CloudSim طراحی شده، که یک شبیه‌ساز مناسب و جامع برای شبکه‌ها و محاسبات ابری می‌باشد. شبیه‌سازی‌ها با در نظر گرفتن دو سناریو ادامه داده می‌شود، به طوری‌که سناریوی اول پردازش درخواست‌های ارسالی از لایه اول توسط روش پیشنهادی و سناریوی دوم پردازش درخواست‌ها توسط لایه ابر می‌باشد. ویژگی‌های سناریوهای مورد استفاده در جدول ۲ به طور خلاصه بیان شده است. برای مشخص کردن تعداد درخواست‌های حساس به زمان (درخواست‌هایی که قرار است در مه پردازش شود) با ارزش‌گذاری بیشتر برای این نوع درخواست‌ها از تابع رندوم (∂) استفاده شده است.

جدول (۲): خلاصه‌ایی از پارامترهای شبیه‌سازی

مفاهیم	پارامترها
N تعداد کل درخواست‌ها	۱۰۰۰-۱۰۰۰۰
حجم درخواست‌های حساس به زمان	$60\% < \partial < 70\%$
تعداد تکرار سناریو	۱۰
پهنای باند تا لایه مه	400 Mbps
پهنای باند تا لایه ابر	1Gbps

نتایج نمودار تاخیر کل با توجه به مقادیر بیان شده در جدول ۲ در شکل ۳ ارائه گردیده است و بیانگر کاهش تاخیر در شبکه با استفاده از روش پیشنهادی نسبت به روش سنتی ابر می‌باشد که بیانگر کاهش ۶۵٪ تاخیر سرویس است. تاخیر سرویس بیانگر مجموع تاخیرهای انتقال، انتشار، صف و پردازش می‌باشد. در روش پیشنهادی برابر مجموع معادلات ۱ و ۲ می‌باشد.

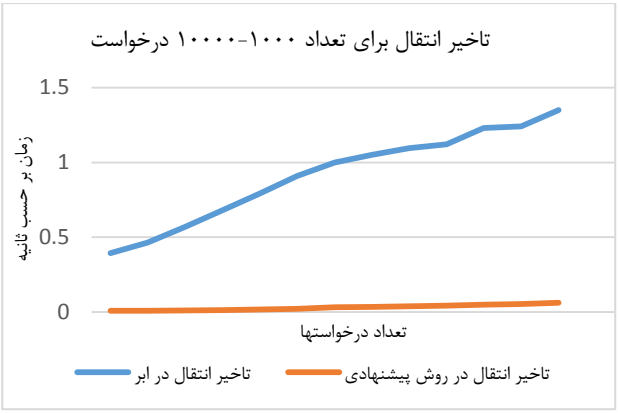
شکل ۴ مربوط به تاخیر انتقال است و برای هر دو سناریو مورد توجه قرار گرفته است همان‌طور که در نمودار مشاهده می‌شود تاخیر انتقال برای ارسال درخواست‌ها به مه در مقایسه با ارسال به ابر بسیار ناچیز است.



شکل (۳): مقایسه تاخیر کل برای تعداد ۱۰۰۰ تا ۱۰۰۰۰ درخواست

www.SID.ir

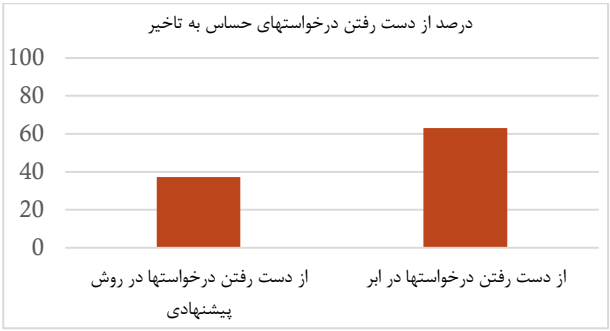
Archive of SID



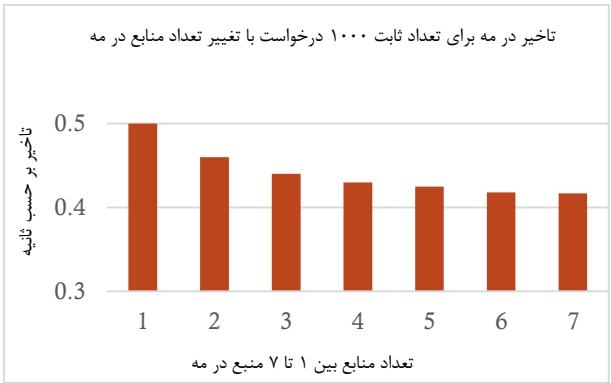
شکل (۴): مقایسه تاخیر انتقال تعداد ۱۰۰۰-۱۰۰۰۰ درخواست

شکل ۵ مربوط به تعداد درخواست‌های حساس به تاخیری است که در هر دو سناریو پردازش نشده‌اند و زمان پردازش را از دست داده‌اند. از آنجایی که درخواست‌های ارسالی از طرف ماشین‌های MTC اکثرا با حجم کم و نیازمند به پردازش کوتاه هستند، بیشترین زمان تاخیر صرف انتقال آن‌ها به ابر می‌شود بنابراین اگر درخواستی حساس به تاخیر باشد با احتمال بیشتری نسبت به وقتی که به لایه مه ارسال شود و در آنجا پردازش شود، از دست می‌رود.

شکل ۶ تاخیر سرویس برای تعداد ثابت درخواست با تغییر تعداد منابع در لایه مه در روش پیشنهادی را نشان می‌دهد. همان‌طور که در نمودار مشخص است افزایش تعداد منابع باعث کاهش زمان سرویس-دهی می‌شود اما پس از رسیدن به تعداد ۵ منبع تاثیر چندانی در کاهش تاخیر سرویس ندارد.



شکل (۵): مقایسه درصد از دست رفتن درخواست‌های حساس به تاخیر



شکل (۶): مقایسه تاخیر سرویس در روش پیشنهادی برای تعداد ثابت درخواست با تغییر تعداد منابع در مه

۵- نتیجه گیری

در این مقاله به بیان مفهوم مه در شبکه‌های MTC پرداخته شد. به طوری‌که وجود آن در کنار ابر، نه به عنوان جایگزین بلکه به عنوان مکملی برای پردازش درخواست‌های ارسالی از سوی ماشین‌های MTC تعریف شده است. لایه مه با جایگیری بین ماشین‌ها و ابر، یک ابر نزدیک به کاربران را ایجاد می‌کند و تاخیرهای پردازشی و انتقال را کاهش می‌دهد. به طوریکه درخواست‌های حساس به زمان را می‌توان در این لایه پردازش کرد و درخواست‌هایی که غیرحساس به زمان هستند به منظور جلوگیری از ازدحام برای پردازش به ابر ارسال کرد. نتایج شبیه‌سازی بیانگر کاهش تاخیر زمانی که از محاسبات مه استفاده می‌شود نسبت به حالتی که از ابر به تنهایی استفاده می‌شود، و همچنین افزایش درصد درخواست‌های پردازش شده می‌باشد. به عنوان کارهای آینده می‌توان قابلیت حرکت را برای دستگاه‌ها در نظر گرفت، علاوه بر این می‌توان منابع موجود در مه را متعدد و متنوع فرض کرد.

مراجع

[1] ع. بیگی, طاهرپور and ,عباس, "کنترل توان در ارتباطات دستگاه به دستگاه زیرلایه در شبکه‌های سلولی نسل پنجم تحت اطلاعات غیر دقیق حالت کانال‌ها," نشریه مهندسی برق و الکترونیک ایران , vol. 17, no. 1, pp. 115–125, 2020.

[2] M. Zhao, A. Kumar, T. Ristaniemi, and P. H. J. Chong, "Machine-to-Machine Communication and Research Challenges: A Survey," Wirel. Pers. Commun., vol. 97, no. 3, pp. 3569–3585, 2017.

[3] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in Internet of everything, Springer, 2018, pp. 103–130.

[4] A. Whitmore, A. Agarwal, and L. Da Xu, "The Internet of Things—A survey of topics and trends," Inf. Syst. Front., vol. 17, no. 2, pp. 261–274, 2015.

[5] 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, 2017.

[6] M. Adhikari, M. Mukherjee, and S. N. Srirama, "DPTO: A deadline and priority-aware task offloading in fog computing framework leveraging multi-level feedback queueing," IEEE Internet Things J., 2019.

[7] S. M. Alizadeh Shabestary, M. Saeedmanesh, A. Rahimi-Kian, and E. Jalalabadi, "Real-Time Frequency and Voltage Control of an Islanded Mode Microgrid," نشریه مهندسی برق و الکترونیک ایران, vol. 12, no. 3, pp. 9–14, 2016.

[8] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, "Fog computing: Focusing on mobile users at the edge," arXiv Prepr. arXiv1502.01815, 2015.

[9] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the Suitability of Fog Computing in the Context of Internet of Things," IEEE Trans. Cloud Comput., vol. 6, no. 1, pp. 46–59, 2015.

[10] R. Mahmud and R. Buyya, "Modelling and simulation of fog and edge computing environments using iFogSim toolkit," Fog edge Comput. Princ. Paradig., pp. 1–35, 2019.

[11] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog

www.SID.ir

Archive of SID

computing: A platform for internet of things and analytics," in Big data and internet of things: A roadmap for smart environments, Springer, 2014, pp. 169–186.
[12] V. Kumar, A. 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, 2019.
[13] 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 things J., vol. 3, no. 6, pp. 1171–1181, 2016.
[14] A. Mubarakali, A. D. Durai, M. Alshehri, O. AlFarraj, J. Ramakrishnan, and D. Mavaluru, "Fog-based delay-sensitive data transmission algorithm for data forwarding and storage in cloud environment for multimedia applications," Big Data, 2020.
[15] 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., 2020.
[16] 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, 2018.
[17] L. Gao, T. H. Luan, S. Yu, W. Zhou, and B. Liu, "FogRoute: DTN-based data dissemination model in fog computing," IEEE Internet Things J., vol. 4, no. 1, pp. 225–235, 2016.
[18] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in 2014 federated conference on computer science and information systems, 2014, pp. 1–8.
[19] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in 2014 Australasian telecommunication networks and applications conference (ATNAC), 2014, pp. 117–122.
[20] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," Softw. Pract. Exp., vol. 47, no. 9, pp. 1275–1296, 2017.

زیرنویس‌ها

[1] Machine Type Communication
[2] Machine-to-Machine Communication
[3] 5G
[4] real time
[5] Quality of Service
[6] Big Data
[7] Fog Computing
[8] Middleware
[9] Smart-Cities
[10] Smart-Home
[11] Self-Driving Vehicles
[12] Healthcare
[13] Base Stations
[14] Interaction
[15] Latency
[16] Bluetooth
[17] Wifi
[18] Volunteer Supported Fog Computing
[19] Backbone
[20] Offloading
[21] Quality of Experience
[22] Deadline
[23] Reject
[24] Response time

www.SID.ir