سال سیزدهم، شماره‌های ۴۷ و ۴۸، بهار و تابستان ۱۴۰۰
صص: ۱۱_۲۲

دو فصلنامه علمی
فناوری اطلاعات و ارتباطات ایران

یک چارچوب بهبودیافته برای بهبود کیفیت و امنیت در شبکه اینترنت اشیاء با استفاده از زنجیره بلوکی و قدرت پردازشی لایه مه

محمدتقی شیخان*، کیانوش آزادی**
* مالک محصول، شرکت موج آینده فرافن
**دانشجوی دکتری، دانشگاه صنعتی امیرکبیر
تاریخ دریافت: ۱۳۹۹/۰۱/۰۲ تاریخ پذیرش: ۱۳۹۹/۰۷/۲۰
نوع مقاله: پژوهشی

چکیده
با گسترش و همه‌گیر شدن اینترنت اشیاء، در آینده‌ای نه‌چندان دور شاهد وابسته شدن زندگی بشر به سرویس‌های آن خواهیم بود. در آن زمان تصور ادامه حیات بدون اینترنت اشیاء دشوار بوده و بروز اختلال در سرویس‌های آن موجب وقوع خسارات جانی و مالی بسیاری خواهد شد. بروز اختلال در سرویس‌های اینترنت اشیاء می‌تواند به دو علت پیش آید: اختلالات شبکه و اختلال ناشی از انجام فعالیت‌های مخرب نفوذگرها. فعالیت‌های مخرب نفوذگرها همچنین می‌تواند منجر به نقض حریم خصوصی افراد شود. در این مقاله راهکاری برای افزایش مقاومت سرویس‌های اینترنت اشیاء در برابر اختلالات شبکه و فعالیت‌های مخرب نفوذگران ارائه شده است. راهکار پیشنهادی با اتکا بر قدرت پردازشی نودهای حاضر در لایه مه، به‌صورت توأمان به کاهش تأخیر (بهبود کیفیت) سرویس و بهبود امنیت و حفظ حریم خصوصی اشیاء می‌پردازد. سایر ویژگی‌های راهکار پیشنهادی عبارت از رعایت عدالت میان اشیاء از منظر کیفیت سرویس دریافتی و حداقل نمودن سربار ناشی از پردازش و انتقال بسته‌های منقضی (بسته‌هایی که قطعاً تأخیر بیش از حد آستانه را تجربه خواهند نمود) است. رعایت عدالت موجب می شود کیفیت سرویس هیچ‌یک از اشیاء قربانی کاهش تأخیر سرویس کل شبکه نشود؛ چراکه ممکن است اشیاء مذکور مورداستفاده در کاربردی حیاتی (مثلاً در حوزه سلامت) باشند.
واژگان کلیدی: اینترنت اشیاء، پردازش مه، کیفیت سرویس، امنیت، حریم خصوصی

۱ مقدمه

پدیده اینترنت از زمان پیدایش تا اکنون شاهد تغییرات بسیاری در سیر تکاملی خود بوده است: شبکه جهانی وب، شبکه‌ای از مستندات HTML مرتبط باهم، سیستم‌های تعاملی دوطرفه با کاربر، شبکه‌های اجتماعی، وب معنایی (قابل‌فهم کردن محتوا برای ماشین‌ها) و غیره. اینترنت اشیاء حوزه‌ای است که جذاب‌ترین تکامل فناورانه آتی در سیر تکاملی اینترنت را رقم خواهد زد [۱] و [۲].

اینترنت اشیاء کاربردهای فراوانی در صنعت و زندگی روزمره انسان‌ها دارد. برخی از محبوب‌ترین کاربردهای آن عبارت‌اند از: خانه هوشمند، تجهیزات پوشیدنی، شهر هوشمند، شبکه توزیع هوشمند، اینترنت صنعتی، اتومبیل‌های متصل، سلامت،

نویسنده مسئول: محمد تقی شیخان، m.sheykhan@fwutech.com

خرده‌فروشی‌های هوشمند، زنجیره تأمین هوشمند و مزارع هوشمند [۳].

تحقق اینترنت اشیاء در حوزه های مربوط به زندگی روزمره و خصوصی مردم، در کنار مزایای بسیاری که فراهم می‌آورد، مخاطراتی نیز به همراه دارد: مخاطرات ناشی از قطعی و افت کیفیت سرویس (به دلیل وابستگی ایجادشده به سرویس) و مخاطرات امنیتی و حفظ حریم خصوصی. مطابق پیش‌بینی‌های انجام‌شده، در سال ۲۰۲۵ میلادی، ۴۱ درصد بازار اینترنت اشیاء متعلق به حوزه سلامت خواهد بود [۴]؛ بنابراین ضروری است که مخاطرات فوق بررسی و مرتفع شوند، چراکه عدم توجه به آن‌ها منجر به فجایع جبران‌ناپذیری خواهد شد.

در سال‌های اخیر پژوهش‌های بسیاری حول موضوعات افزایش کیفیت سرویس، بهبود سطح امنیت و حفظ حریم خصوصی اشیاء با بهره‌گیری از قابلیت های لایه مه[۱] انجام شده است. کیفیت سرویس و امنیت دو نیاز اساسی در شبکه اینترنت‌اشیاء است که عدم ارضاء هریک موجب بروز اشکالات اساسی خواهد شد؛ اما پژوهش‌های پیشین هرکدام تنها به یکی از جنبه‌های بهبود کیفیت سرویس یا حفظ امنیت و صیانت از حریم‌خصوصی پرداخته است. این مسئله موجب باقی ماندن نیاز برای انجام پژوهش‌هایی برای بهبود کیفیت سرویس و حفظ امنیت و حریم خصوصی به‌طور هم‌زمان شده است.

در [۵] تا [۱۰] تلاش شده با آفلود نمودن وظایف پردازشی لایه ابر به لایه مه، موازنه‌ای میان افزایش انرژی مصرفی نودها و بهبود کیفیت سرویس اشیاء برقرار کنند. در این پژوهش‌ها پارامترهای مختلفی برای تصمیم‌گیری در مورد آفلود نمودن یا ننمودن وظایف به لایه مه استفاده‌شده است که برای نمونه می‌توان به سطح باتری و قدرت پردازشی نودهای لایه مه، کمینه تأخیر موردنیاز هر سرویس و میزان توان پرداز شی موردنیاز وظایف پرداز شی[۲] اشاره نمود.

در الگوریتم‌های مبتنی بر آفلود[۳]، زمانبندی[۴] پردازش و ارسال از پارامترهای مهم و تاثیرگذار بر کیفیت سرویس اشیاء است. WFQ [۱۱] و W۲FQ [۱۲] الگوریتم های شناخته شده‌ای برای زمان‌بندی هستند. [۱۳] نیز الگوریتمی برای زمان‌بندی با اولویت[۵] وظایف پردازشی در پردازش ابری ارائه کرده است. الگوریتم‌های فوق کارایی مناسبی از نظر بهبود کیفیت سرویس دارند؛ اما سربار پردازشی زیادی داشته و برای نودهای مورد استفاده در اینترنت‌اشیاء مناسب نیستند. نویسندگان [۱۴] و [۱۵] تلاش کرده‌اند با ارائه الگوریتم‌هایی سبک‌وزن بر این مشکل فایق آیند. الگوریتم ارائه شده در [۱۵] دارای سربار زمان‌بندی $O(K)$ –بدون وابستگی به تعداد وظایف پردازشی- است.

[۱۶] و [۱۷] تلاش کرده‌اند تا با فشرده‌سازی داده‌ها، انتخاب بهترین نود در لایه مه (با توجه به پهنای‌باند نودهای در دسترس) و تغییر پویای الگوی حجم داده های ارسالی، به کم‌ترین تأخیر و درنتیجه بالاترین کیفیت سرویس دست یابند. از مواردی که تأثیر بسیاری بر کیفیت سرویس در این پژوهش ها دارد، الگوریتم فشرده‌سازی مورداستفاده و قابلیت فشرده شدن داده‌های در دسترس است؛ برخی از انواع داده‌ها قابلیت فشرده سازی بالاتری دارند.

در [۱۸] و [۱۹] نسخه‌های سبک‌وزنی از PKI[۶] برای احراز هویت میان لایه اشیاء و لایه مه پیشنهاد شده است. در این پژوهش‌ها تلاش شده با استفاده از زیرساخت PKI چالش احراز هویت اشیاء، نودهای لایه مه و لایه ابر را مرتفع سازند. [۲۰] نیز به موضوع احراز هویت در بستر PKI و بر پایه رمز عبور پرداخته است.

در سال های اخیر پژوهش های فراوانی در زمینه کاربرد زنجیره بلوکی در صنایع مختلف (مالی، خودروهای خودران و غیره) برای افزایش امنیت و صیانت از حریم خصوصی شده است. برای نمونه در [۲۱] تا [۲۳] تلاش شده با استفاده از مفاهیم زنجیره بلوکی، کاربرد[۷]های مختلف را بدون نیاز به یک سرویس‌دهنده متمرکز (مانند سرویس‌دهنده ابری) ایمن سازند.

در [۲۴] تا [۲۹] راهکارهایی برای حفظ حریم خصوصی در شبکه اینترنت اشیاء مبتنی بر لایه مه ارائه نموده‌اند. در این پژوهش‌ها علاوه بر مخفی نگه‌داشتن داده‌ها و اطلاعات خصوصی کاربران، استراتژی‌ها و راهکارهایی برای تضمین عدم توانایی مهاجمین در پیش‌بینی و تخمین اطلاعات حساس اشیاء و نودهای شبکه پیشنهاد شده است. مسائلی که در این مقالات موردبررسی قرار گرفته شامل شناسایی انواع حملات شناخته شده برای استخراج داده‌های حساس (مانند حمله ایجاد تداخل، ایجاد تراکنش جعلی برای موقعیت‌یابی کاربران، تحلیل حجم و تناوب داده‌های ارسالی برای شناسایی نوع اشیاء و غیره) و پیشنهادهایی برای پیشگیری از بروز آن‌ها است.

در این پژوهش راهکاری برای بهبود کیفیت سرویس در شبکه اینترنت اشیاء با حفظ امنیت و حریم خصوصی پیشنهاد می شود. برای این منظور ابتدا یک الگوریتم مکاشفه‌ای برای کاهش تاخیر سرویس اشیاء ارائه شده و میزان بهبود حاصل، نمایش داده می‌شود.

---

[۱] Fog Layer

[۲] Processing Task

[۳] Offload

[۴] Scheduling

[۵] Priority-based

[۶] Public Key Infrastructure

[۷] Application

سپس صیانت از امنیت و حریم خصوصی اشیاء با استفاده از رمزنگاری متقارن و مفاهیم زنجیره بلوکی به الگوریتم پیشنهادی افزوده شده و تاثیر این موضوع بر تاخیر سرویس بررسی می شود. به‌طور اجمالی دستاوردهای راهکار پیشنهادی به شرح زیر است:

- بهبود کیفیت سرویس با ارائه یک الگوریتم مکاشفه‌ای برای کمینه کردن تأخیر سرویس دریافتی اشیاء
- تضمین حفظ امنیت و حریم خصوصی با استفاده از الگوریتم‌ها و مکانیسم‌های امنیتی سبک‌وزن
- رعایت عدالت میان اشیاء از منظر کیفیت سرویس دریافتی
- کاهش سربار ترافیکی و پردازشی شبکه با معدوم ساختن بسته‌های منقضی

در ادامه ابتدا مدل سیستمی و تعاریف بیان می‌شود. سپس در فصل ۳ الگوریتم مکاشفه‌ای پیشنهادی برای کاهش تأخیر سرویس و افزایش امنیت شبکه شرح داده می شود. در فصل ۴ نتایج شبیه‌سازی موردبحث قرار گرفته و در فصل ۵ نتیجه‌گیری ارائه می‌شود.

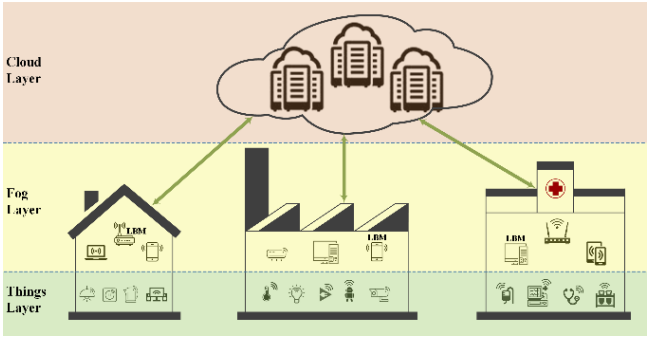
۲ مدل سیستمی و تعاریف

همان‌طور که در **Error! Reference source not found.** مشاهده می شود، شبکه اینترنت اشیاء مفروض در راهکار پیشنهادی شامل اشیاء و تجهیزات هوشمند موجود در محیط های مختلف (خانه، کارخانه، بیمارستان و غیره) است که به صورت سیمی یا بی‌سیم به شبکه جهانی اینترنت متصل هستند. شبکه مفروض متشکل از سه لایه اشیاء، مه و ابر است. در این شبکه اشیاء سرویس‌گیرنده هستند. نودهای حاضر در لایه مه و سرویس‌دهندگان لایه ابر نیز نقش سرویس‌دهنده را ایفاء می‌کنند.

اشیاء به طور ممتد وظایف پردازشی تولید می‌کنند که باید ظرف مدت زمان محدودی پردازش شوند. پردازش هر وظیفه می تواند توسط خود شیء، یک نود در لایه مه یا یک سرویس‌دهنده در لایه ابر انجام شود.

هر شیء به یک نود در لایه مه متصل است و وظایف پردازشی خود را برای آن ارسال می‌کند. نودهای لایه مه نیز دارای اتصالاتی مابین خود و به سرویس‌دهندگان لایه ابر هستند و این امکان را دارند که وظایف پردازشی را برای یکدیگر یا به لایه ابر آفلود نمایند.

ازآنجایی که ممکن است نودهای مخرب در لایه مه نفوذ کنند، از مفاهیم زنجیره بلوکی (BC[۱]) برای افزایش امنیت و حفظ حریم خصوصی استفاده شده است. نودهای لایه مه در هر محیط، یک



شکل ۱. معماری شبکه اینترنت اشیاء در راهکار پیشنهادی

زنجیره بلوکی خصوصی تشکیل می‌دهند و یکی از نودها نقش مدیر بلوک محلی (LBM[۲]) را بر عهده می‌گیرد. انتصاب LBM بر عهده لایه ابر است.

هر نود با یک کلید عمومی (PK[۳]) شناخته می‌شود و تمامی تراکنش‌های انجام شده در هر BC توسط LBM به دفترکل[۴] محلی اضافه می‌شود. تمامی ارتباطات میان نودها و سرویس‌دهندگان به‌صورت متقارن رمزنگاری می‌شوند.

دفترکل هر BC خصوصی تنها توسط نودهای حاضر در همان محیط نگهداری می شوند، اما چکیده[۵] دفتر کل در لایه ابر نگهداری می‌شود تا صحت آن تضمین شود. البته در شرایطی که نودهای لایه مه ظرفیت ذخیره سازی دفترکل در حافظه خود را نداشته باشند، امکان ذخیره دفترکل در حافظه سرویس‌دهندگان لایه ابر (به‌صورت رمزنگاری‌شده) وجود دارد.

نودها و اشیاء حاضر در هر BC خصوصی نیازمند دریافت و توافق کلیدهای رمزنگاری متقارن و نامتقارن خود به صورتی امن هستند. برای این منظور موارد زیر در شبکه اینترنت اشیاء راهکار پیشنهادی فرض شده است:

- هر شیء حاضر در شبکه مجهز به یک eSIM[۶] است. کلیدهای رمزنگاری متقارن و نامتقارن اصلی و مکانیسم احراز هویت در این ماژول به‌صورت پیش‌فرض تعبیه شده است و با استفاده از این ماژول سرویس‌دهنده های لایه ابر و اشیاء یکدیگر را به‌صورت امن احراز هویت می‌کنند.
- هر نود در لایه مه کلیدهای عمومی و خصوصی[۷] خود را تحت زیر ساخت PKI (همانند [۱۸] و [۱۹]) دریافت می‌کند. در این میان سرویس‌دهندگان لایه

[۱] Block Chain
[۲] Local Block Manager
[۳] Public Key
[۴] Immutable Ledger
[۵] Hash
[۶] Embedded SIM Card
[۷] Public and Private Key

ابر نقش مرجع صـــدور گواهینامه (CA[1]) را بازی می‌کنند.

- اشیاء و نودهای لایه مه، هر جریان ترافیکی را با یک کلید فرعی مجزا رمزنگاری می‌کنند. کلید فرعی برای رمزنگاری جریان ترافیکی آتی نیز در پایان جریان ترافیکی فعلی تعیین می‌شود. در صورت قطع ارتباط پیش از انجام تعیین کلید ارتباط بعدی، تبادل کلید توســط یک متد گســترش کلید مانند دفی-هلمن [۳۰] یا F-secure [۳۱] انجام می‌شود.
- اشــیاء و نودهای لایه مه یکدیگر را با کمک سرویس‌دهندگان لایه ابر احراز هویت و اعتبارسنجی می‌کنند. درواقع ســرویس‌دهنده لایه ابر کلید رمزنگاری و مکانیسم احراز هویت را برایشان مشخص نموده و ID نودهای مجاز را نگهداری می‌کند.

موضوعی که اهمیت اساسی در تعیین کیفیت سرویس اشیاء دارد، مقدار تاخیر در دریافت سرویس است. در اینجا فاصله زمانی میان آماده شــدن یک وظیفه پردازشــی تا حصــول جواب آن را تأخیر ســرویس[2] می‌نامیم و تأخیر ســرویس وظیفه jام شــیء i را با τ_j^i نمایش می‌دهیم. تأخیر سرویس شیء i را نیز با d_i نمایش می‌دهیم که برابر با میانگین تأخیر سرویس وظایف پردازشی آن است:

$$d_i = \frac{\sum_j \tau_j^i}{n} \qquad (۱)$$

هر شیء وظایف پردازشی را با احتمال p_i^l خودش پردازش نموده، با احتمال p_i^F برای پردازش به یکی از نودهای واقع در لایه مه سپرده یا با احتمال p_i^C برای سرویس‌دهنده‌های لایه ابر ارسال می‌کند (در اصطلاح آفلود می‌کند). وظایف پردازشی با احتمال b_i از نوع سبک (محاســبات ســاده‌ای مانند میانگین گیری عددی) یا با احتمال $b_i' = ۱ - b_i$ از نوع ســنگین (محاســبات پیچیده‌ای مانند پردازش تصویر) است.

هر نود این امکان را دارد که وظایف دریافتی را برای یک نود دیگر در همان BC یا برای یک ســرویس‌دهنده آفلود نماید. همچنین با هدف افزایش سطح امنیت، نودها تلاش می‌کنند در صورت امکان وظایف را پیش از آفلود، تکه‌تکه[3] نمایند. عمل تکه‌تکه کردن موجب کاهش ارزش اطلاعاتی زیروظایف[4] شده و در صورت بروز یک حمله موفق، اطلاعات ناقص و کم‌ارزشــی به دســت محاجم می‌افتد. با عنایت به موارد فوق، تأخیر سرویس شیء iام را می‌توان به صورت زیر محاسبه کرد:

$$\begin{aligned} d_i = p_i^l \times (A_i) + \left(۱ - p_{drop}^i\right) \\ \times \left[p_i^F \times \left(X_{ij}^{IF} + Y_{ij}^{IF} + E_i + L_{ij}\right)\right. \\ + p_i^C \\ \times (X_{ik}^{IC} + Y_{ik}^{IC} + E_i + E_k + \bar{H}_k + X_{ki}^{CI} \\ \left. + Y_{ki}^{CI}) + T_{BC} + T_{def}\right] + p_{drop}^i \\ \times TTL; \\ j = f(i). k = g(i) \end{aligned} \qquad (۲)$$

در رابطه فوق A_i میانگین زمان پردازش وظایف پردازشــی توســط شــیء iام اســت. p_{drop}^i احتمال مقدور نبودن پردازش وظایف پیش از منقضی شدن و دور ریخته شدن آن است. X_{ij}^{IF} تأخیر انتشار و Y_{ij}^{IF} تأخیر انتقال لینک مابین شیء i و نود j است. همچنین X_{ik}^{IC} و Y_{ik}^{IC} به ترتیب مجموع تأخیر انتشــار و مجموع تأخیر انتقال لینک‌های اتصال‌دهنده شیء i به سرویس‌دهنده k است و X_{ki}^{CI} و Y_{ki}^{CI} به ترتیب مجموع تأخیر انتشــار و مجموع تأخیر انتقال لینک‌های میان ســرویس‌دهنده k و شــیء i اســت. E_i و E_k نیز به ترتیب تأخیر حاصل از سربار رمزنگاری/رمزگشایی بسته‌های حامل وظایف پرداز شی توسط شیء i و سرویس‌دهنده k است. لازم به توضیح است که تأخیر انتظار در صف ار سال هر لینک در پارامتر تأخیر انتقال گنجانده شده است.

پارامتر T_{BC} میانگین تأخیر تحمیل‌شده به اشــیاء در اثر فرایندهای انجام‌شــده در زنجیره بلوکی (مانند تشــکیل بلوک، ماینینگ و محاســبه و تبادل چکیده دفتر کل) اســت. ازآنجایی که فرایندهای انجام‌شــده در زنجیره بلوکی موجب افزایش تأخیر در کل شــبکه می‌شود، این پارامتر برای تمام اشــیاء یکســان فرض شــده اســت. پارامتر T_{def} نیز میانگین تأخیر ناشی از سرهم‌بندی[5] پاسخ وظایف تکه‌تکه شــده در شــیء iام اســت. همچنین $f(i)$ و $g(i)$ توابعی هستند که مشخص می‌کنند شیء iام وظایف خود را برای کدام نود و کدام سرویس‌دهنده ارسال می‌کند.

L_{ij} شــامل تمام تأخیرهای اتفاق افتاده در لایه مه و احتمالاً در لایه ابر (در صورت نیاز به آفلود شدن وظیفه به سرویس‌دهنده لایه ابر) اســت. تأخیرهای لایه مه شــامل تأخیر پردازش و انتظار در صــف پردازشــی نود پردازش‌کننده وظیفه، تأخیر انتشــار و تأخیر انتقال تمام لینک‌های مابین نودهایی که وظیفه میان آن‌ها آفلود شده (در مســیر رفت‌وبرگشــت)، تأخیر حاصــل از تکه‌تکه کردن وظیفه (در صــورت وقوع)، تأخیر حاصــل از رمزنگاری/رمزگشــایی در هر نود و تأخیرهای اتفاق افتاده در لایه ابر شامل تأخیرهای انتشار و انتقال، تأخیر حاصــل از رمزنگاری/رمزگشــایی و تأخیر پردازش وظیفه در سرویس‌دهنده است. همچنین $\bar{H}_k$ شامل تأخیر انتظار در صف و تأخیر پردازش سرویس‌دهنده k است.

---

[1] Certificate Authority
[2] Service Delay
[3] Fragment
[4] Subtask
[5] Defragmentation

ازآنجایی‌که تعداد دفعات آفلود شدن و اینکه آیا وظایف پردازشی به لایه ابر نیز آفلود خواهند شد یا خیر از پیش مشخص نیست، رابطه تأخیر در لایه مه به‌صورت بازگشتی[۱] نوشته شده است. در رابطه زیر $L_{ij}(x)$ نشان‌دهنده تأخیر سرویس پردازش وظایف شیء i (که به نود j متصل است) در لایه مه و احتمالاً لایه ابر، در xامین دفعه آفلود شدن است:

$$\begin{aligned} L_{ij}(x) = P_j \cdot \left(\bar{T}^{j}_{process} + X^{FI}_{ji} + Y^{FI}_{ji} + E_j\right) + \left(1 - P_j\right) \\ \cdot \Big[[1 - \phi(x)] \\ \cdot \Big[X^{FF}_{jj'} + Y^{FF}_{jj'} + E_{j'} + \gamma \times T_{frag} \\ + L_{ij'}(x+1)\Big] + \phi(x) \\ \cdot \left[X^{FC}_{jk} + Y^{FC}_{jk} + \bar{H}_k + X^{CI}_{ki} + Y^{CI}_{ki} + E_j \right. \\ \left. + E_k\right]\Big]; \\ j' = best(j).\, k = h(j) \end{aligned} \quad (۳)$$

که در رابطه فوق $\bar{T}^{j}_{process}$ میانگین زمان پردازش وظایف در نود j، E_j تأخیر حاصل از رمزنگاری/رمزگشایی بسته در نود j و $\phi(x)$ تابع تصمیم‌گیری درباره آفلود کردن وظایف به بهترین نود همسایه یا سرویس‌دهنده لایه ابر است. خروجی تابع $\phi(x)$ یکی از مقادیر صفر یا یک است. مقدار صفر به معنی آفلود برای بهترین همسایه و مقدار یک به معنی آفلود برای سرویس‌دهنده است.

T_{frag} میانگین تأخیر ناشی از تکه‌تکه کردن وظایف پردازشی است که با احتمال γ رخ می‌دهد (برخی از وظایف امکان تکه‌تکه شدن را ندارند). $h(j)$ و $best(j)$ به ترتیب توابع مشخص‌کننده سرویس‌دهنده متصل و بهترین همسایه برای نود jام است.

هریک از نودهای لایه مه یا لایه ابر پس از پردازش وظیفه محوله، پاسخ آن را برای شیء مالک وظیفه ارسال می‌کند. ازآنجایی‌که هیچ مسیر مستقیمی میان اشیاء و سرویس‌دهندگان لایه ابر وجود ندارد، تمامی تبادلات میان اشیاء و سرویس‌دهندگان از طریق نودهای لایه مه صورت می‌پذیرد.

در راهکار پیشنهادی هدف کمینه نمودن میانگین تأخیر سرویس اشیاء با رعایت انصاف است؛ به‌این‌ترتیب کیفیت سرویس هیچ‌یک از اشیاء قربانی کمینه‌سازی میانگین تأخیر سرویس کل نمی‌شود. برای این مهم از تابع انصاف نسبی[۲] استفاده کرده و مسئله را به‌صورت زیر فرموله نموده‌ایم:

$$\text{MIN} \frac{1}{|S^I_P|} \sum_{i \in S^I_P} \frac{|d_i - d^{min}_i|}{d^{min}_i} \quad (۴)$$

$$subject\ to:$$
$$d_i < d^{max}_i : \forall i \in S^I_P$$

در مسئله فوق S^I_P مجموعه اشیاء حاضر در حوزه P است. d^{max}_i بیشینه تأخیر قابل‌پذیرش برای شیء i است و درصورتی‌که یک وظیفه پرداز شی از آن تخطی کند، منقضی شده و دیگر پاسخ آن برای شیء مالک ارزشمند نبوده و دور ریخته می‌شود. همچنین هر شیء دارای کمینه تأخیر سرویس بوده که با d^{min}_i نمایش داده می‌شود. اگر یک شیء تأخیر سرویسی برابر با d^{min}_i داشته باشد به بیشینه کیفیت سرویس قابل حصول خود رسیده است. بنابراین تلاش برای کاهش تأخیر سرویس اشیاء بیش از این مقدار، کمکی به افزایش کیفیت سرویس کل شبکه ننموده و حالت بهینه، رساندن تأخیر سرویس تمام اشیاء به مقدار d^{min}_i است. مقادیر d^{min}_i و d^{max}_i برای هر شیء با توجه به کاربرد تعیین می‌شود.

توپولوژی اتصال میان نودهای لایه مه و اشیاء به‌صورت یک به n است؛ یعنی هر شیء تنها می‌تواند به یک نود در لایه مه متصل باشد اما هر نود در لایه مه می‌تواند به تعداد بسیاری شیء متصل شود. توپولوژی اتصال میان نودهای لایه مه اما از قاعده مشخصی پیروی نمی‌کند. هر دو نود در لایه مه که در خط دید یکدیگر باشند و نسبت سیگنال به نویز و تداخل‌شان[۳] بیشتر از حد آستانه باشد ($SINR_{ij} > SINR_{threshold}$)، به هم متصل شده و در اصطلاح همسایه هستند. توپولوژی اتصال میان نودهای لایه مه و سرویس‌دهندگان لایه ابر نیز به‌صورت n به یک است.

۳ الگوریتم مکاشفه‌ای

در فصل قبل مفروضات راهکار پیشنهادی برای بهبود امنیت و حفظ حریم خصوصی بیان شد. در این فصل الگوریتم مکاشفه‌ای پیشنهادی باهدف کمینه کردن میانگین تأخیر سرویس اشیاء با رعایت عدالت شرح داده می‌شود.

هر نود در لایه مه هنگامی‌که بسته‌ای از اشیاء متصل یا نودهای همسایه دریافت می‌کند، با توجه به میزان ازدحام در صف پردازشی خود تصمیم می‌گیرد که بسته را پردازش کند یا آن را آفلود نماید. همچنین با توجه به میزان ازدحام موجود در صف پردازشی و میزان تاخیر انتقال و تاخیر انتشار لینک ارتباطی با نودهای همسایه، تصمیم می‌گیرد آیا آن را برای یکی از نودهای همسایه در لایه مه ارسال کند یا برای سرویس‌دهندگان لایه ابر؟ نتیجه این تصمیم‌گیری تأثیر بسیاری در کاهش یا افزایش تأخیر سرویس دریافتی اشیاء ایفاء می‌کند؛ به همین سبب الگوریتم پیشنهادی به شیوه تصمیم‌گیری نودهای لایه مه در مورد بسته‌های دریافتی می‌پردازد. در ۰ فلوچارت الگوریتم پیشنهادی نمایش داده شده است.

هر نود هنگامی‌که یک بسته دریافت می‌کند، ابتدا پرچم FSC[۴] را بررسی می‌نماید. این پرچم بیانگر این موضوع است که آیا شیء مالک وظیفه، مُصِر به پردازش شدن وظیفه در لایه ابر است یا خیر؟

[۱] Recursive

[۲] Proportional Fairness

[۳] Signal to Interference plus Noise Ratio (SINR)

[۴] Force Send to Cloud

هنگامی که یک شیء تصمیم به ارسال مستقیم وظیفه‌ای به سرویس دهندگان لایه ابر نمود، پرچم FSC بسته را مقداردهی می‌کند ($Flag_{FSC} = ۱$). درصورتی‌که این پرچم مقداردهی شده باشد، نود دریافت‌کننده بسته بلافاصله و بدون هیچ پردازشی آن را برای سرویس‌دهنده لایه ابر ارسال می‌کند. در غیر این صورت فیلد TTL[۱] بسته بررسی می‌شود. فیلد TTL متناظر با پارامتر d_i^{max} در مسئله (۴) است.

هر شیء در هنگام آفلود نمودن وظایف پردازشی خود به لایه مه یا ابر، فیلد TTL بسته‌ها را مقداردهی می‌کند. مقداردهی فیلد TTL توسط هر شیء در هنگام قرار دادن وظایف پردازشی در بسته‌های IP و متناسب با کاربرد (کنترل دما، پردازش تصویر دوربین‌های ویدئویی و غیره) انجام می‌شود.

هر نود یا سرویس‌دهنده پیش از پردازش وظیفه، مقدار TTL بسته را به‌اندازه زمان تخمینی انتظار خود کاهش می‌دهد. همچنین پیش از ارسال بسته (وظیفه پردازشی یا پاسخ وظیفه)، تأخیر انتشار و تأخیر انتقال از TTL بسته کسر می‌شود. همچنین هنگامی‌که وظیفه‌ای توسط یک نود یا سرویس‌دهنده پردازش شد، فیلد TTL وظیفه در بسته پاسخ کپی می‌شود.

هر نود پس از دریافت یک بسته، *تاخیر صف پردازشی + تأخیر انتقال + تأخیر انتشار* خود را با مقدار فیلد TTL بسته مقایسه می‌نماید. درصورتی‌که از مقدار TTL کوچک‌تر باشد، به این معناست که نود قادر است بسته را پردازش نموده و پاسخ آن را پیش از منقضی شدن، به شیء مالک بازگرداند. بنابراین بسته را در صف پردازشی خود قرار می‌دهد.

در صورتی‌که مقدار TTL از *تاخیر صف پردازشی + تأخیر انتقال + تأخیر انتشار* کوچک‌تر باشد، نود باید بسته را آفلود نماید. برای تصمیم‌گیری در این مورد لازم است بهترین همسایه شناسایی شود. منظور از بهترین همسایه، نودی است که در کوتاه‌ترین زمان بسته را پردازش نموده و پاسخ را بازمی‌گرداند.

هر نود برای مشخص نمودن بهترین همسایه مقدار *تاخیر صف پردازشی + دو برابر تأخیر انتشار + دو برابر تأخیر انتقال* را برای تمام همسایه‌ها محاسبه می‌نماید. علت قرار دادن ضریب دو برای تاخیر انتشار و تاخیر انتقال، در نظر گرفتن هر دو تاخیر پیش‌آمده، ابتدا در مسیر ارسال وظیفه پردازشی و سپس در مسیر دریافت پاسخ آن است. همسایه‌ای که کمترین مقدار را داشته باشد به‌عنوان بهترین همسایه مشخص می‌شود.

پس از مشخص شدن بهترین همسایه، لازم است بررسی شود که آیا بهترین همسایه قادر به پردازش وظیفه و بازگرداندن پاسخ آن درزمانی کمتر از TTL بسته هست یا خیر. برای این مهم TTL بسته با مقدار *تاخیر صف پردازشی + دو برابر تأخیر انتشار + دو برابر تأخیر انتقال* بهترین همسایه مقایسه می‌شود. اگر مقدار محاسبه‌شده بزرگ‌تر از TTL بود یعنی امکان پردازش بسته در لایه مه وجود ندارد و باید برای سرویس‌دهندگان لایه ابر آفلود شود. در غیر این صورت بسته برای بهترین همسایه ارسال خواهد شد.

علت دیگری که باعث می‌شود نود از آفلود بسته برای بهترین همسایه منصرف شود، بیشتر شدن تعداد دفعات آفلود شدن (N_{fwd}) از حد آستانه از پیش تعیین‌شده (e_m) است. علت وجود این پارامتر جلوگیری از گرفتار شدن بسته‌ها در حلقه[۲] مسیریابی است. با این ترتیب بدون نیاز به ذخیره کردن سابقه بسته‌های پردازش‌شده در هر نود، مطمئن می‌شویم هیچ‌گاه بسته‌ای در حلقه مسیریابی گرفتار نمی‌شود.

اگر *تاخیر صف پردازشی + دو برابر تأخیر انتشار + دو برابر تأخیر انتقال* بهترین همسایه کوچک‌تر از TTL بسته باشد، برای بهترین همسایه آفلود می‌شود. قبل از آفلود اما ترجیح بر این است که وظیفه به دو زیروظیفه تقسیم شود. مزیت این عمل مزیت بالا رفتن سطح امنیت داده‌ها است زیرا هریک از زیروظایف ارزش اطلاعاتی پایینی دارند.

پیش از آفلود وظیفه برای بهترین همسایه، پرچم تکه‌تکه شدن (Fragmentation_Flag) و پارامتر شماره تکه (Fragmentation_Number) بسته بررسی می‌شود. در صورتی‌که پرچم تکه‌تکه شدن برابر با False باشد به این معناست که وظیفه مذکور قابلیت تقسیم به دو زیروظیفه را ندارد. همچنین اگر پارامتر شماره تکه مقداردهی شده باشد یعنی بسته مذکور خود یک زیروظیفه است و دیگر نمی‌توان آن را به دو زیروظیفه تقسیم کرد.

اگر مقدار پرچم تکه‌تکه شدن برابر با True و پارامتر شماره تکه مقداردهی نشده باشد، وظیفه مذکور به دو زیروظیفه تقسیم شده و مجدداً هر دو زیروظیفه برای پردازش توسط نود امکان‌سنجی می‌شود. در غیر این صورت یک واحد به پارامتر N_{fwd} بسته اضافه شده و بسته برای بهترین همسایه آفلود می‌شود.

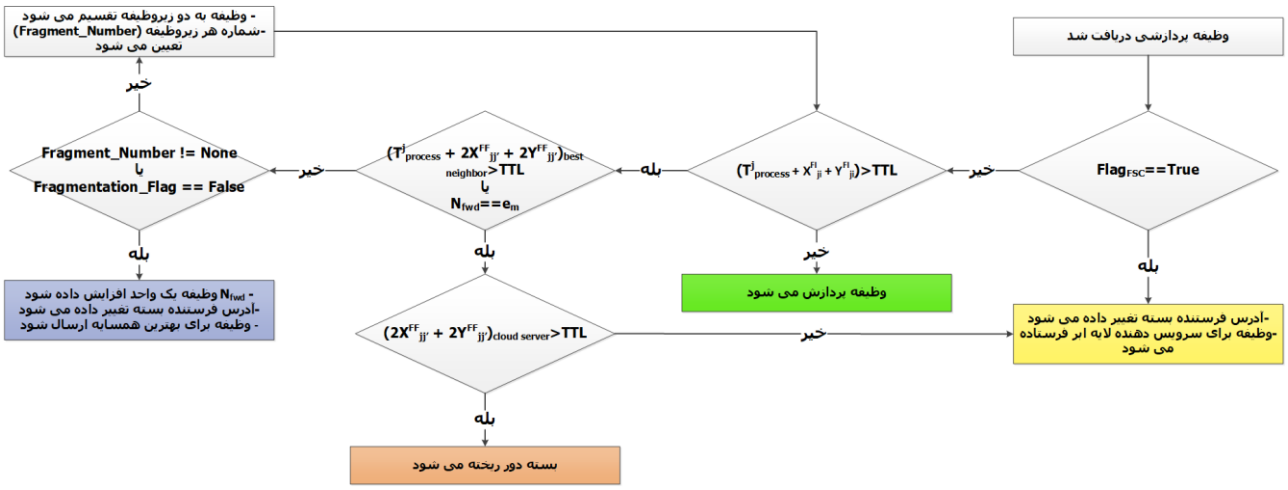
در صورت لزوم آفلود نمودن بسته برای سرویس‌دهنده لایه ابر، بررسی می‌شود که آیا سرویس‌دهنده قادر به پردازش وظیفه و بازگرداندن پاسخ آن به شیء مالک درزمانی کوتاه‌تر از TTL بسته هست یا خیر؟ اگر *دو برابر تأخیر انتشار + دو برابر تأخیر انتقال* ارسال بسته برای سرویس‌دهنده کوچک‌تر از TTL بسته بود، بسته برای سرویس‌دهنده آفلود می‌شود (پیش از ارسال عمل تغییر آدرس فرستنده بسته نیز انجام می‌شود). در غیر این صورت بسته دور ریخته می‌شود زیرا هیچ موجودیتی در شبکه وجود ندارد که بسته را پیش از منقضی شدنش پردازش و پاسخ را برای شیء مالک

[۱] Time-to-Live

[۲] Loop

ارسال کند. این عمل موجب پیشگیری از هدر رفت منابع شبکه به‌وسیله پردازش و انتقال بسته‌های منقضی می‌شود.

در پاراگراف بالا به علت قدرت پردازشی بالای سرویس دهندگان لایه ابر، هنگام تصمیم‌گیری در مورد آفلود کردن یا نکردن بسته به سرویس‌دهنده، تنها به تأخیر انتشار و تأخیر انتقال توجه شده است؛ زیرا مدت‌زمان پردازش هر وظیفه تو سط سرویس‌دهندگان ناچیز و قابل‌اغماض ا ست. این عمل باعث کاهش سربار سیگنالینگی میان لایه ابر و لایه مه می شود؛ چراکه دیگر نیاز به گزارش دائم و ضعیت صف سرویس‌دهندگان به نودهای متصل در لایه مه نیست.



شکل ۲. فلوچارت الگوریتم پیشنهادی

عمل دیگری که برای بالا بردن سطح امنیت و محافظت از حریم خصوصی اشیاء انجام می شود، جایگزینی آدرس شیء فرستنده بسته با آدرس نود پیش از آفلود است. این عمل تنها یک‌بار توسط نودهای متصل به اشیاء انجام می‌شود.

۴ شبیه‌سازی

۱,۴ محیط و پارامترهای شبیه‌سازی

محیط شبیه‌سازی شبکه‌ای سه سطحی، مشابه شکل ۱ فرض شده است. در این شبکه ۵۰۰ شیء، ۲۵ نود و ۶ سرویس‌دهنده لایه ابر حضور دارند. در شبیه‌سازی برای ایجاد محیط و نودها، مدیریت ارسال و دریافت بسته‌ها، زمان‌بندی صف‌ها و رخدادها از ابزار Simpy [۳۲] در پایتون استفاده‌شده است.

هر شیء وظایف پردازشی را با احتمال ۱۰ درصد خود پردازش کرده، با احتمال ۷۵ درصد وظایف را برای پردازش به لایه مه ارسال نموده و با احتمال ۱۵ درصد پردازش وظایف را به لایه ابر می‌سپارد. وظایف پردازشی بر دو نوع هستند: وظایف سبک و وظایف سنگین. هر شیء نیز یا وظایف سبک تولید می‌کند یا وظایف سنگین. تعیین این مسئله در ابتدای شبیه سازی و با احتمال مساوی برای هریک (تولید وظایف سبک یا سنگین) انجام می‌شود. نرخ تولید وظایف پردازشی سبک برای اشیاء ۰,۵ و نرخ تولید وظایف پردازشی سنگین ۰,۶ با توزیع پواسون فرض شده است.

نودهای لایه مه تنها راه ارتباطی اشیاء با شبکه هستند. هر شیء در لایه اشیاء به نزدیک‌ترین نود در لایه مه متصل می شود. در اینجا منظور از نزدیک ترین نود، نودی است که کمترین تأخیر انتشار میان شیء و آن نود وجود دارد. هر شیء تنها به یک نود متصل می شود اما هر نود می تواند به چندین شیء متصل شود. اتصال نودها به سرویس‌دهندگان لایه ابر نیز به همین صورت است. هر نود به نزدیک‌ترین سرویس‌دهنده لایه ابر متصل شده و اتصال آن‌ها یک به چند است (یک نود تنها به یک سرویس دهنده متصل می‌شود اما هر سرویس‌دهنده به چند نود).

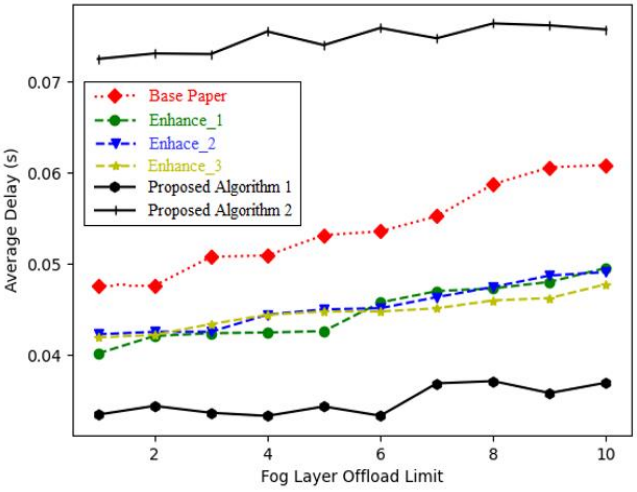
هر نود می تواند به یک یا چند نود دیگر متصل شود. توپولوژی اتصال نودها در لایه مه در هر دور از شبیه سازی توسط یک تولیدکننده تصادفی گراف با میانگین درجه ۳ ایجاد می شود. در شبیه‌سازی‌ها برای ایجاد چنین گرافی، از ابزار NetworkX [۳۳] استفاده‌شده است.

نرخ لینک‌های بین اشیاء و نودها، در صورتی‌که شیء از نوع سبک باشد ۲۵۰ kbps (NFC، NB-IoT، BLE یا ZigBee) و درصورتی‌که شیء از نوع سنگین باشد ۵۴ Mbps (LTE-M، WiFi-ah و IEEE ۸۰۲,۱۱ a/g) فرض شده است. همچنین نرخ لینک‌های بین نودها ۱۰۰ Mbps و نرخ لینک‌های میان نودها و سرویس‌دهندگان ۱۰ Gbps (فیبر نوری) فرض شده است.

تأخیر انتشار لینک‌های میان اشیاء و نودها، نودها با یکدیگر، نودها و سرویس دهندگان با توزیع یکنواخت و به ترتیب با U[۱,۲]، U[۰,۵،۱,۲] و U[۱۵،۳۵] میلی‌ثانیه تعین شده است.

اندازه وظایف پردازشی تولیدی توسط اشیاء بدین‌صورت فرض شده است: اندازه وظایف سبک با توزیع نمایی و میانگین ۱۰۰ بایت و اندازه وظایف سنگین با توزیع نمایی و میانگین ۸۰ کیلوبایت است. همچنین میانگین اندازه پاسخ وظایف پردازشی با اندازه وظایف پردازشی برابر فرض شده است.

برای نزدیک‌تر شدن محیط شبیه‌سازی به واقعیت، قدرت پردازشی اشیاء مشابه میکروکنترولر Arduino Uno R۳ و قدرت پردازشی نودهای لایه مه مشابه پردازنده Intel dual-core i۷ فرض شده است. در بدترین حالت قدرت پردازشی یک نود حدود ۳۰۰۰ برابر بیشتر از یک شیء تولیدکننده وظیفه سبک (U[۵۰۰،۴۰۰۰]) و ۲۰۰ بار بیشتر از یک شیء تولیدکننده وظیفه سنگین است (U[۱۰۰،۴۰۰]). همچنین فرض بر این است که سرویس‌دهندگان



شکل ۳. مقایسه میانگین تأخیر سرویس وظایف پردازش‌شده اشیاء با تغییر آستانه تعداد آفلود مجاز برای هر نود لایه مه

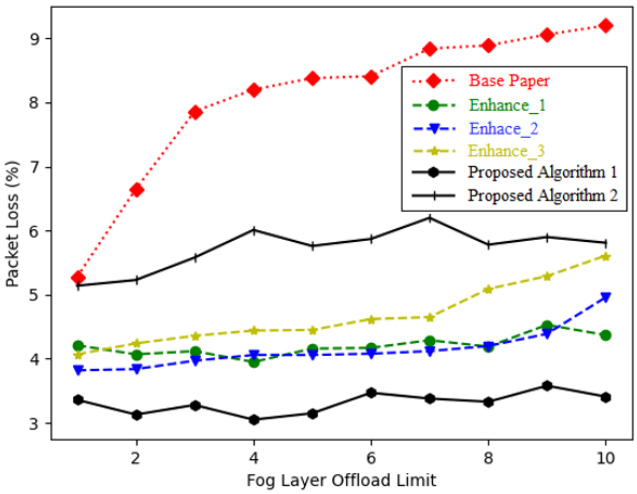
به‌طور میانگین ۱۰۰ برابر سریع‌تر از نودهای لایه مه هستند (U[۵۰،۲۰۰]). در شبیه‌سازی میانگین مدت‌زمان پردازش وظایف سبک و سنگین توسط اشیاء به ترتیب ۳۰ و ۴۰۰ میلی‌ثانیه فرض شده است.

برای حداقل نمودن تأثیر شرایط شبیه‌سازی (مانند توزیع اشیاء سبک و سنگین، نرخ تولید بسته، توپولوژی شبکه و غیره) بر نتایج شبیه‌سازی، با اتکا بر روش مونت‌کارلو، در هر دور از شبیه‌سازی دو میلیون وظیفه سبک و سنگین توسط اشیاء تولید شده و نتیجه پردازش آن‌ها به دست اشیاء رسیده است.

پارامترهای مربوط به رمزنگاری و زنجیره بلوکی نیز به‌صورت زیر فرض شده است:

- با توجه به ارزیابی‌های انجام‌شده در [۳۴-۳۶]، سربار رمزنگاری و رمزگشایی بسته‌ها برابر با ۱۰ درصد با انحراف معیار ۳ فرض شده است. سربار محاسبه چکیده دفترکل (IL) نیز ۰٫۲ میلی‌ثانیه در نظر گرفته شده است.
- تعداد وظیفه در هر بلوک ۱۵۰۰ فرض شده است.
- تکه‌تکه کردن بسته‌های پردازشی موجب افزایش مصرف حافظه در نودهای شبکه می‌شود. اما ازآنجایی‌که حافظه نودهای شبکه بی‌نهایت فرض شده است، عمل تکه‌تکه کردن سربار ازنظر مصرف حافظه در شبیه‌سازی ایجاد نمی‌کند.
- به علت ناچیز بودن سربار پردازشی عمل تکه‌تکه کردن وظایف، از سربار پردازشی آن صرف‌نظر شده است.

سایر پارامترهای شبیه‌سازی مطابق جدول زیر است. در این جدول



شکل ۴. درصد وظایف پردازشی که مقدار تأخیر سرویسشان بیشتر از آستانه بیشینه زمان انتظار شده است

θ_j آستانه زمان انتظار وظایف پردازشی مورداستفاده برای نودها در الگوریتم AFP [۱۰]، b_i احتمال سبک بودن نوع یک وظیفه پردازشی تولیدشده و TTL_j مدت‌زمانی است که یک شیء پس از ارسال وظیفه پردازشی jام به لایه مه یا لایه ابر، منتظر دریافت پاسخ آن می‌ماند.

جدول ۱ سایر پارامترهای مورداستفاده در شبیه‌سازی

p_i^I	p_i^F	b_i	θ_j	TTL_j
۰٫۱	۰٫۷۵	۰٫۵	۰٫۰۰۰۲	۰٫۲

۲٫۴ نتایج شبیه‌سازی

در این قسمت نتایج شبیه‌سازی بحث می‌شود. نتایج شبیه‌سازی الگوریتم پیشنهادی با الگوریتم پیشنهادی در [۱۰] (AFP) مقایسه شده است. ازآنجایی‌که تعداد ایده‌های پیشنهادی برای افزایش کارایی و بهبود سطح امنیت حریم خصوصی زیاد است، برای درک بهتر نتیجه هریک، نتیجه پیاده‌سازی هر ایده به‌صورت مجزا نمایش داده شده و سپس تأثیر پیاده‌سازی تمام ایده‌ها در قالب الگوریتم پیشنهادی ارائه می‌شود. در شکل‌های ۳ و ۴ منظور از AFP، Enhance_۱، Enhance_۲، Enhance_۳، Proposed Algorithm ۱ و Proposed Algorithm ۲ به شرح زیر است:

- منظور از AFP، الگوریتم ارائه‌شده در [۱۰] است.
- در Enhance_۱ آستانه زمان انتظار وظایف پردازشی برای نودهای لایه مه حذف شده و تصمیم‌گیری هر نود برای پردازش یا آفلود کردن وظایف با مقایسه مقدار تاخیر صف پردازشی نود و *تاخیر صف پردازشی*

+ دو برابر تأخیر انتشار + دو برابر تأخیر انتقال نودهای همسایه و *دو برابر تأخیر انتشار + دو برابر تأخیر انتقال* سرویس‌دهنده لایه ابر انجام می‌شود. بدین‌صورت هر نود بررسی می‌کند که آیا اگر خود بسته را پردازش کند بهتر است یا اینکه آن را به یکی از نودهای همسایه یا سرویس‌دهنده لایه ابر آفلود کند؟

- در Enhance_۲ تکه‌تکه کردن وظایف پردازشی سنگین پیش از آفلود کردن آن‌ها پیاده‌سازی شده است. این عمل باهدف افزایش سطح امنیت و بهبود حریم خصوصی انجام می‌شود.
- در Enhance_۳ سیاست صف‌های پردازش و ارسال نودها از FIFO به صف با اولویت تغییر داده شده است: هرچه مقدار TTL یک وظیفه کوچک‌تر، اولویت پردازش آن بالاتر. همچنین هر نود پیش از آفلود کردن هر وظیفه بررسی می‌کند که آیا امکان پردازش آن وظیفه در شبکه (توسط یک نود دیگر یا توسط سرویس‌دهنده لایه ابر) وجود دارد یا خیر؟ اگر جواب منفی بود، وظیفه مذکور دور ریخته می‌شود تا منابع شبکه بیهوده مصرف نشود. پاسخ این سؤال نیز از مقایسه تاخیر صف پردازشی نود با *تاخیر صف پردازشی + دو برابر تأخیر انتشار + دو برابر تأخیر انتقال* نودهای همسایه و *دو برابر تأخیر انتشار + دو برابر تأخیر انتقال* سرویس‌دهنده لایه ابر مشخص می‌شود.
- در Proposed Algorithm ۱ تمامی بهبودهای فوق پیاده‌سازی شده است.
- در Proposed Algorithm ۲ موارد امنیتی (رمزنگاری متقارن و زنجیره بلوکی) به Proposed Algorithm ۱ افزوده شده است.

در ۰ میزان تغییر میانگین تأخیر سرویس اشیاء با افزایش پارامتر e_m نشان‌داده شده است. همان‌طور که مشاهده می‌شود، هریک از الگوریتم‌های Enhance_۱، Enhance_۲، Enhance_۳ و Proposed Algorithm ۱ بین۱۰ الی ۵۰ درصد کاهش در میانگین تأخیر سرویس برای اشیاء نسبت به AFP به ارمغان آورده است. همچنین با افزایش e_m، میزان بهبود در عملکرد الگوریتم‌های فوق نسبت به AFP بیشتر شده است. این مشاهده نشان‌دهنده هوشمندی بیشتر الگوریتم‌های فوق در هنگام تصمیم‌گیری برای آفلود نمودن وظایف پردازشی نسبت به AFP است. در AFP لازم است مقدار e_m همواره مقدار کوچکی باشد، زیرا با افزایش آن تعداد دفعات آفلود بی‌مورد بسته‌ها افزایش یافته و درنتیجه میانگین تأخیر پردازش اشیاء نیز افزایش می‌یابد.

همان‌طور که مشاهده می‌شود، در Proposed Algorithm ۱ با ادغام ایده‌های فوق، میزان بهبود بین ۳۰ الی ۵۰ درصدی حاصل شد. اما در Proposed Algorithm ۲ ادغام تمامی بهبودها نیز نتوانسته تأثیر منفی رمزنگاری و ثبت تراکنش‌ها در زنجیره بلوکی بر میانگین تأخیر سرویس را خنثی کند. میانگین تأخیر سرویس اشیاء در Proposed Algorithm ۲ حدود ۵۰ درصد بیشتر از AFP است. این افزایش در میانگین تأخیر سرویس هزینه‌ای است که درازای به دست آوردن امنیت در برابر انواع حملات مطرح در شبکه اینترنت اشیاء و حفظ حریم خصوصی پرداخت شده است.

در حال حاضر پژوهش‌های بسیاری برای پیشنهاد الگوریتم‌های سبک‌وزن رمزنگاری، محاسبه چکیده پیام، اعتماد، احراز هویت، مدل اعتبار و تشکیل زنجیره بلوکی کم‌هزینه در حال انجام است. درصورت به‌ثمر رسیدن این تلاش‌ها، جایگزینی الگوریتم‌ها و مکانیسم‌های فوق با الگوریتم‌ها و مکانیسم‌های امنیتی مورداستفاده در Proposed Algorithm ۲، هزینه ایمن‌سازی در راهکار پیشنهادی کاهش پیدا می‌کند.

به‌جز میانگین تأخیر سرویس، پارامتر مهم دیگری که بر کیفیت سرویس دریافتی اشیاء تأثیرگذار می‌گذارد، عبور تأخیر سرویس وظایف از حد آستانه است. علت اهمیت این پارامتر آن است که وظایفی که میزان تأخیرشان بیش‌ازحد آستانه شود، دیگر ارزشی نداشته و دور ریخته می‌شوند؛ بنابراین پردازش و انتقال آن‌ها تنها موجب تحمیل سربار پردازش و انتقال بر شبکه اینترنت اشیاء است.

در ۰ تغییر پارامتر درصد وظایفی که تأخیر سرویسشان از حد آستانه عبور کرده (منقضی شده‌اند) نسبت به افزایش پارامتر e_m نشان‌داده شده است. همان‌طور که مشاهده می‌شود، درصد از دست دادن بسته‌ها الگوریتم‌های Enhance_۱، Enhance_۲، Enhance_۳، Proposed Algorithm ۱ و Proposed Algorithm ۲ نسبت به AFP مقدار کمتری است. این امر به معنی بیشتر بودن کارایی و کمتر بودن سربار شبکه -به علت عدم اتلاف پهنای‌باند ارتباطی و منابع پردازشی شبکه برای پردازش و انتقال وظایف متأخر و همچنین عدم نیاز به ارسال مجدد وظایف متأخر به لایه مه و ابر برای پردازش- است.

نتیجه دیگری که از مقایسه میان نتایج حاصل از الگوریتم‌های AFP، Proposed Algorithm ۱ و Proposed Algorithm ۲ در شکل‌های ۳ و ۴ حاصل می‌شود، بهبود رعایت عدالت میان وظایف پردازشی اشیاء

مختلف به لحاظ تأخیر سرویس توسط Proposed Algorithm ۱ و Proposed Algorithm ۲ است. الگوریتم AFP بااینکه میانگین تأخیر سرویس پایین‌تری نسبت به Proposed Algorithm ۲ برای اشیاء ایجاد کرده است، موجب عبور تأخیر سرویس تعداد بیشتری از وظایف از حد آستانه شده است. این بدان معناست که واریانس تأخیر سرویس وظایف پردازش‌شده در AFP بالاتر از Proposed Algorithm ۲ بوده و تأخیر سرویس وظایف پردازش‌شده در Proposed Algorithm ۲ پراکندگی کمتری از میانگین دارند. به‌این‌ترتیب در Proposed Algorithm ۱ و Proposed Algorithm ۲ از منظر کیفیت سرویس، عدالت میان اشیاء مالک وظایف بهتر رعایت شده است.

۵. نتیجه‌گیری

در این پژوهش یک راهکار باهدف کمینه نمودن میانگین تأخیر سرویس و تضمین امنیت و حفظ حریم خصوصی اشیاء پیشنهاد شد. ایده‌های مطرح‌شده در راهکار پیشنهادی برای کاهش تأخیر سرویس اشیاء موفق به ایجاد کاهشی قابل‌توجه نسبت به سایر پژوهش‌های انجام‌شده در این زمینه شده است. اما اضافه نمودن مکانیسم‌های امنیتی (شامل رمزنگاری ارتباطات، احراز هویت متقابل اشیاء و نودها و سرویس‌دهندگان، تشکیل زنجیره بلوکی در لایه مه، ماینینگ و ثبت تراکنش‌ها توسط نودهای لایه مه، محاسبه و ذخیره نمودن چکیده دفتر کل)، موجب تحمیل هزینه سنگینی ازنظر افزایش میانگین تأخیر سرویس اشیاء شد.

مقدار میانگین تأخیر سرویس اشیاء در راهکار پیشنهادی تابعی از تأخیر ایجادشده به دلیل پیاده‌سازی مکانیسم‌های امنیتی است. بدین ترتیب در آینده با بهبود کارایی این روش‌ها، عملکرد راهکار پیشنهادی نیز بهبود خواهد یافت.

علاوه بر موارد فوق، دستاوردهای دیگر این پژوهش رعایت عدالت میان اشیاء از منظر سطح کیفیت سرویس دریافتی و کاهش سربار ترافیکی و پردازشی بسته‌های منقضی (بسته‌هایی که قابلیت پردازش و بازگشت پاسخشان درزمانی کوتاه‌تر از حد آستانه وجود ندارد) درکل شبکه اینترنت اشیاء است.

مراجع

[۱] D. Raggett, “The web of things: Challenges and opportunities,” *Computer*, vol. ۴۸, no. ۵, pp. ۲۶-۳۲, ۲۰۱۵.

[۲] A. Whitmore, A. Agarwal, and L. Da Xu, “The Internet of Things—A survey of topics and trends,” *Information Systems Frontiers*, vol. ۱۷, no. ۲, pp. ۲۶۱-۲۷۴, ۲۰۱۵.

[۳] K. L. Lueth. "The ۱۰ most popular Internet of Things applications right now," iot-analytics.com/۱۰-internet-of-things-applications/, ۲۰۱۷.

[۴] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE communications surveys & tutorials*, vol. ۱۷, no. ۴, pp. ۲۳۴۷-۲۳۷۶, ۲۰۱۵.

[۵] K. Kumar, and Y.-H. Lu, “Cloud computing for mobile users: Can offloading computation save energy?,” *Computer*, vol. ۴۳, no. ۴, pp. ۵۱-۵۶, ۲۰۱۰.

[۶] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud." pp. ۳۰۱-۳۱۴, ۲۰۱۱.

[۷] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, “Saving portable computer battery power through remote process execution,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. ۲, no. ۱, pp. ۱۹-۲۶, ۱۹۹۸.

[۸] G. C. Hunt, and M. L. Scott, "A guided tour of the Coign automatic distributed partitioning system." pp. ۲,۲۶۲-۵۲, ۱۹۹۸

[۹] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading." pp. ۹۴۵-۹۵۳, ۲۰۱۲.

[۱۰] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog Computing: Towards Minimizing Delay in the Internet of Things." pp. ۱۷-۲۴, ۲۰۱۷.

[۱۱] A. Demers, S. Keshav, and S. Shenker, “Analysis and simulation of a fair queueing algorithm,” *ACM SIGCOMM Computer Communication Review*, vol. ۱۹, no. ۴, pp. ۱-۱۲, ۱۹۸۹.

[۱۲] H. Zhang, and J. C .Bennett, "WF۲Q: worst-case fair weighted fair queueing." pp. ۱۲۰-۱۲۸, ۱۹۹۶.

[۱۳] D. M. Dakshayini, and D. H. Guruprasad, “An optimal model for priority based service scheduling policy for cloud computing environment,” *International journal of computer applications*, vol. ۳۲, no. ۹, pp. ۲۳-۲۹, ۲۰۱۱.

[۱۴] J. Jang, J. Jung, and J. Hong, “K-LZF: An efficient and fair scheduling for Edge Computing servers,” *Future Generation Computer Systems*, vol. ۹۸, pp. ۴۴-۵۳, ۲۰۱۹.

[۱۵] T. Choudhari, M. Moh, and T.-S. Moh, "Prioritized task scheduling in fog computing." pp. ۱-۸, ۲۰۱۸.

Parallel and Distributed Systems, vol. ۲۷, no. ۱۰, pp. ۲۹۶۷-۲۹۸۳, ۲۰۱۶.

[۲۷] L. Zhang, Z. Cai, and X. Wang, "Fakemask: A novel privacy preserving approach for smartphones," *IEEE Transactions on Network and Service Management*, vol. ۱۳, no. ۲, pp. ۳۳۵-۳۴۸, ۲۰۱۶.

[۲۸] M. A. Ferrag, L. Maglaras, and A. Ahmim, "Privacy-preserving schemes for ad hoc social networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. ۱۹, no. ٤, pp. ۳۰۱۵-۳۰٤۵, ۲۰۱۷.

[۲۹] Y. Hong, W. M. Liu, and L. Wang, "Privacy preserving smart meter streaming against information leakage of appliance status," *IEEE transactions on information forensics and security* ,vol. ۱۲, no. ۹, pp. ۲۲۲۷-۲۲٤۱, ۲۰۱۷.

[۳۰] H. Delfs, H. Knebl, and H. Knebl, *Introduction to cryptography*: Springer, ۲۰۰۲.

[۳۱] "F-secure," https://campaigns.f-secure.com/total/pm/en_us/.

[۳۲] "Simpy," April ۲۵, ۲۰۱۹; https://simpy.readthedocs.io/en/latest/contents.html.

[۳۳] "NetworkX," April ۲۵, ۲۰۱۹; https://networkx.github.io/documentation/stable/index.html.

[۳٤] Y. Xiao, H.-H. Chen, B. Sun, R. Wang, and S. Sethi, "MAC security and security overhead analysis in the IEEE ۸۰۲,۱۵. ٤ wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. ۲۰۰۶, no. ۲, pp. ۸۱-۸۱, ۲۰۰۶.

[۳۵] O. Barahtian, M. Cuciuc, L. Petcana, C. Leordeanu, and V. Cristea, "Evaluation of Lightweight Block Ciphers for Embedded Systems." pp. ٤۹-۵۸, ۲۰۱۵.

[۳۶] S. Maitra, and K. Yelamarthi, "Rapidly Deployable IoT Architecture with Data Security: Implementation and Experimental Evaluation," *Sensors*, vol. ۱۹, no. ۱۱, pp. ۲٤۸٤, ۲۰۱۹.

[۱۶] E. S. Gama, R. Immich, and L. F. Bittencourt, "Towards a Multi-Tier Fog/Cloud Architecture for Video Streaming." pp. ۱۳-۱٤, ۲۰۱۸.

[۱۷] C.-F. Lai, D.-Y. Song, R.-H. Hwang, and Y.-X. Lai, "A QoS-aware streaming service over fog computing infrastructures." pp. ۹٤-۹۸, ۲۰۱۶.

[۱۸] B. Oniga, S. H. Farr, A. Munteanu, and V. Dadarlat, "IoT Infrastructure Secured by TLS Level Authentication and PKI Identity System." pp. ۷۸-۸۳, ۲۰۱۸.

[۱۹] J. Won, A. Singla, E. Bertino, and G. Bollella, "Decentralized public key infrastructure for internet-of-things." pp. ۹۰۷-۹۱۳, ۲۰۱۸.

[۲۰] I. Stojmenovic, and S. Wen, "The fog computing paradigm: Scenarios and security issues." pp. ۱-۸, ۲۰۱٤.

[۲۱] K. Christidis, and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, vol. ٤, pp. ۲۲۹۲-۲۳۰۳, ۲۰۱۶.

[۲۲] P. K. Sharma, S. Y. Moon, and J. H. Park, "Block-VN: A distributed Blockchain based vehicular network architecture in smart city," *Journal of information processing systems*, vol. ۱۳, no. ۱, ۲۰۱۷.

[۲۳] D. Shift, "Technology tipping points and societal impact.", ۲۰۱۵.

[۲٤] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactionson Dependable and Secure Computing*, vol. ۱۵, no. ٤, pp. ۵۷۷-۵۹۰, ۲۰۱۸.

[۲۵] X. Ren, X. Yang, J. Lin, Q. Yang, and W. Yu, "On scaling perturbation based privacy-preserving schemes in smart metering systems." pp. ۱-۷, ۲۰۱۳.

[۲۶] X. Yang, X. Ren, J. Lin, and W. Yu, "On binary decomposition based privacy-preserving aggregation schemes in real-time monitoring systems," *IEEE Transactions on*