بسمه‌تعالی

گزارش و تحلیل مقاله

A software-defined caching scheme for the Internet of Things

تعریف مسئله و هدف اصلی مقاله

اینترنت اشیا پارادایمی است که در آن میلیاردها دستگاه ناهمگن به هم متصل و ارتباط برقرار می‌کنند. با این حال، پیاده‌سازی سرویس‌های جدید با استفاده از اینترنت اشیا ساده نیست و با چالش‌های بسیاری روبرو است. از جمله این چالش‌ها، ایجاد ارتباط بین گره‌های ناهمگن و ماهیت گذرا داده‌ها، است. یک جنبه مهم در مورد اینترنت اشیا، ماهیت محتوای داده‌های اینترنت اشیا است، به این معنی که بیشتر برنامه‌های کاربردی طراحی شده برای این اکوسیستم، صرف نظر از این‌که در کجا قرار دارند یا توسط چه کسی تولید می‌شود، به محتوا علاقه‌مند هستند.

یکی از رویکردهایی که می‌تواند برای کاهش چالش‌های مربوط به اجرای اینترنت اشیا استفاده شود، CCN (Network Named Content) است. روش جدیدی که معماری و مفهوم اینترنت فعلی را تغییر می دهد. یکی از مشخصه‌های اصلی CCN نام‌گذاری داده‌ها است که باعث می‌شود هر تکه داده به یک واحد شناسایی تبدیل شود.

بازیابی داده‌ها با نام آن‌ها به جای مکان آن‌ها یکی دیگر از ویژگی های CCN است که می‌تواند برای پشتیبانی از تحرک[1] مورد استفاده قرار گیرد و از مهم‌ترین قابلیت CCN کش‌شدن اطلاعاتی است که می‌تواند برای حل بسیاری از مشکلات در حوزه اینترنت اشیا مورد استفاده قرار گیرد. داده‌های کش‌شده[2] می‌توانند مزایای زیادی به همراه داشته باشند. به عنوان مثال، پاسخ به درخواست‌های نزدیکترین گره کش‌کننده داده‌ها می‌تواند مزایای زیادی از جمله کاهش دسترسی به گره‌هایی با منابع محدود، صرفه جویی در مصرف انرژی و طولانی شدن چرخه زندگی گره‌ها را به همراه داشته باشد.

کش می‌تواند با افزایش شانس دسترسی به داده‌ها از طریق محتوای ذخیره شده، حتی اگر تولید کننده اصلی در دسترس نباشد، قابلیت دسترسی را بهبود بخشد. افزون بر این، کش در شبکه می‌تواند برای مقابله با چالش‌های ناشی از داده‌های بزرگ و کاهش پهنای باند با ذخیره مطالب مناسب در دستگاه‌های مناسب استفاده شود.

برای استفاده کامل از کش و مدیریت بهتر ذخیره‌سازی در کش، این مقاله پیشنهاد می‌دهد که سنسورها را به خوشه[3]ها و سپس قدرتمندترین گره در هر خوشه را به عنوان سر خوشه انتخاب کنیم. سر خوشه وظیفه

---

[1] Mobility

[2] Cached

[3] Cluster

مدیریت ارتباط با سایر خوشه‌ها را دارد و هم‌چنین می‌تواند به عنوان سرور کش برای هماهنگی تصمیم‌گیری‌های کش درون خوشه عمل کند.
این روش می‌تواند محتوای کش را بر اساس سطح انرژی، ظرفیت ذخیره‌سازی و چندین معیار دیگر بین دستگاه‌های مختلف توزیع کند و منجر به استفاده بهتر از منابع دستگاه‌ها در داخل یک خوشه شود. با این حال، برای استفاده کلی از منابع شبکه، موجودیت دیگری به نام کنترلر SDN/Cache (GSCC)، در نظر گرفته شده است که مسئول تنظیم تصمیمات کش بین خوشه‌های مختلف و انتخاب یک خوشه مناسب برای ذخیره محتوا است. به این منظور، GSCC معیارهای مختلفی مانند منابع جمع شده در یک خوشه و محل خوشه در نمودار شبکه، را در نظر می‌گیرد.
بنابراین، یک رویکرد جامع برای بهینه‌سازی تصمیمات کش‌کردن اطلاعات برای اینترنت اشیا ضروری است و این هدف اصلی این مقاله است.
در این مقاله، یک مکانیسم کش جدید، کش مشارکتی چند معیاره توسعه یافته (EM3C)، پیشنهاد شده است. در EM3C، با در نظر گرفتن چندین ویژگی مرتبط با اینترنت اشیا و تعیین اهمیت نسبی آن‌ها با استفاده از روش‌هایی مانند AHP و روش آنتروپی شانون، تصمیم گرفته می‌شود که آیا محتوا باید کش شود یا نه و هم‌چنین مناسب‌ترین گره/گره‌ها برای کش اطلاعات انتخاب می‌شود.
در دیگر مقالات روش‌هایی پیشنهاد شده که به برخی از آن‌ها اشاره می‌کنیم:

- بهبود کارایی CCN با کش کردن داده‌هایی که فقط بارها درخواست می‌شوند.
- استراتژی Socially-aware Caching که از CCN در شبکه‌های اجتماعی استفاده می‌کند. در این شرایط اطلاعات کش‌شده کاربرانی که معروف‌تر هستند توسط دیگر کاربران بیش‌تر استفاده می‌شود.
- حل مشکل کش داده‌ها در روترهای ناهمگون که حجم حافظه متفاوت دارند با استفاده از مسئله knapsack و برخی از پارامترها
- روش پیشنهادی OFAM-CCN که در شبکه‌هایی با پروتکل OpenFlow، سوییچ‌ها را به دو دسته سوییچ‌های ارتباطی سرور و کاربر و سوییچ‌های ارایه کننده محتوا، تقسیم می‌کند.
- روش MACD که با در نظر گرفتن تازگی داده‌ها، تعداد hop و انرژی دستگاه، روش on-path caching را بهبود می‌بخشد.
- روشی که با در نظر گرفتن تازگی داده‌ها، سطح انرژی دستگاه و حجم حافظه، بهبود ایجاد می‌کند.

روش CCN به عنوان مبنای رفع چالش‌های موجود به کار گرفته می‌شود. قابلیت‌های اصلی CCN عبارت هستند از:

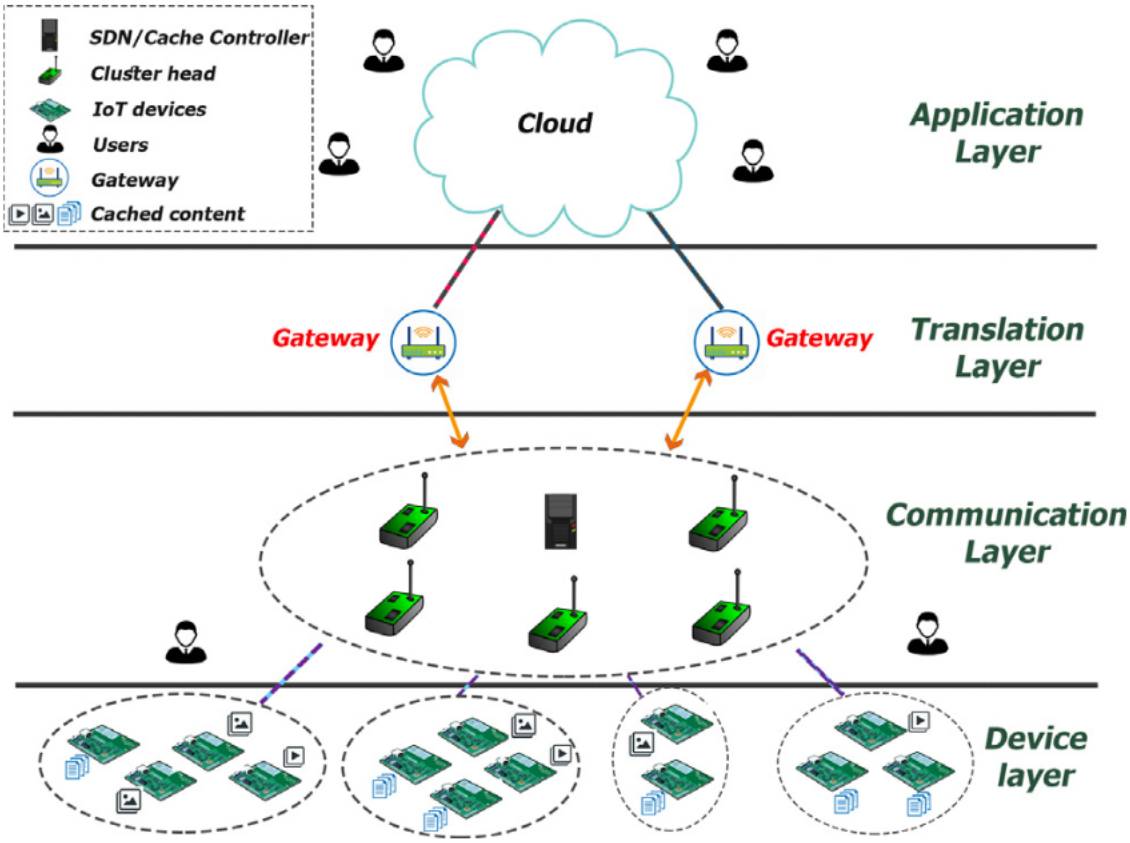
- On-path caching: گره اطلاعاتی که از آن عبور می‌کند را ذخیره می‌کند.
- On-path cache hit: درخواست‌های دسترسی به محتوا فقط زمانی از کش بهره می‌برند که گره مسیر بسته دارای اطلاعات کش‌شده باشد.

- Disunity in path selection and caching decisions: هر گره به صورت مستقل تصمیم می‌گیرد که اطلاعات را کش کند یا نه. گره‌ها از وضعیت و قابلیت‌های محیط اطلاعی ندارند.

روش CCN دارای ایراداتی خصوصا در اینترنت اشیا است:

- High level of content redundancy: بر اساس on-path caching تمامی اطلاعات در هر hop کش شده و حجم بالایی از داده‌های تکراری می‌سازند.
- Unbalanced utilization of resources: بر اساس محل قرارگیری گره، برخی زیر فشار و برخی بدون فشار خواهند بود.
- Cached fragmentation: تکه‌های مختلف یک داده ممکن است در گره‌های متفاوتی کش شود.

راه‌حل پیشنهادی مقاله برای مسئله



تصویر- ۱

روش پیشنهادی این مقاله، مشکلات CCN را که از عدم تمرکز در مدیریت کش به وجود می‌آیند حل می‌کند. در این روش چهار لایه در نظر گرفته می‌شود (تصویر- ۱):

- لایه Application: سرورها و کاربران در این لایه هستند. سرورها ذخیره بلند مدت داده‌ها را برعهده دارند.

- لایه Translation: این لایه درگاه‌ها را شامل می‌شود که لایه Application را به Communication متصل می‌کنند.
- لایه Communication: کنترلر SDN/Cache در این لایه جهت انتخاب بهترین مسیر درخواست و محتوا، وجود دارد. سرخوشه‌ها که با پروتکل OpenFlow قوانین را از کنترلر SDN/Cache دریافت می‌کنند و به صورت محلی یا در ارتباط با دیگران کش می‌کنند، در این لایه هستند. سرخوشه می‌تواند تصمیم بگیرند که درخواست‌های کنترلر SDN/Cache را جهت کش اطلاعات انجام دهند یا نه.
- لایه Device: در لایه تجهیزات و سنسورهای اینترنت اشیا وجود دارند. تجهیزات دارای حافظه هستند که توانایی کش محتوا جهت استفاده آتی را دارند.

در ادامه به روش تصمیم‌گیری برای کش داخل خوشه، بین خوشه‌ها و محتوا می‌پردازیم.

Inter-cluster global decision making

جهت انتخاب خوشه مناسب توسط کنترلر SDN/Cache، شاخص‌هایی در نظر گرفته می‌شوند:

- جمع حافظه‌ها: جمع حافظه تمامی گره‌های درون یک خوشه
- جمع انرژی‌ها: جمع باقی‌مانده انرژی تمامی گره‌های دورن یک خوشه
- فاصله از خوشه مبدا: فاصله کنترلر تا مبدا محتوا
- PLSI: شاخصی بر اساس اتصالات و موقعیت یک خوشه در مقایسه با دیگر خوشه‌ها

شاخص PLSI

اگر اینترنت اشیا را یک گراف درنظر بگیریم، هر خوشه یک گره آن است. میزان بین‌بودن[1] یک گره نشان دهنده میزان مرکزیت آن گره است.

میزان محبوبیت یک خوشه به میزان درخواست محتوای آن توسط کاربران بستگی دارد که البته در طول زمان تغییر می‌کند.

شاخص PLSI به شیوه زیر محاسبه می‌شود:

$$PLSI = \frac{N - |p - b|}{N}$$

که N تعداد خوشه‌ها، p میزان محبوبیت و b میزان بین‌بودن خوشه است.

در نبود دیگر شاخص‌ها، انتخاب خوشه مناسب نیازمند سنجش اهمیت نسبی شاخص‌ها است. با استفاده از روش آنتروپی شانون، به صورت پویا به هر شاخص وزن می‌دهیم (m تعداد خوشه‌ها):

$$K = \frac{1}{\ln(m)}$$

به این ترتیب وزن هر شاخص را به دست می‌آوریم:

[1] Betweenness

$$E_j = -k(\sum_{i=1}^{m} a_{ij} \times \ln(a_{ij}))$$

$$D_j = 1 - E_j$$

$$W_j = \frac{D_j}{\sum_{j=1}^{n} D_j}$$

کنترلر SDN/Cache یک ماتریس که برای هر خوشه یک سطر و در ستون‌ها شاخص‌ها قرار دارند ایجاد کرده و مقادیر آن شاخص را قرار می‌دهد:

$$\begin{array}{c} \\ cl_1 \\ cl_2 \\ \vdots \\ cl_4 \end{array} \begin{array}{c} \begin{array}{cccc} m_1 & m_2 & m_3 & m_4 \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} \end{bmatrix} \end{array}$$

به دلیل تفاوت مقیاس هر m، نیاز به نرمال‌سازی مقادیر داریم. به جای استفاده از روش استاندارد نرمال‌سازی TOPSIS از این روش استفاده می‌کنیم:

$$b_i^+ = \frac{Max\ a_{ij} - Min\ a_{ij}}{Max\ a_{ij}}$$

$$b_i^- = \frac{Min\ a_{ij}}{Max\ a_{ij}}$$

$$n_{ij} = b_i^- + \left[\frac{\left((a_{ij} - Min\ a_{ij})(b_i^+ - b_i^-)\right)}{\left(Max\ a_{ij} - Min\ a_{ij}\right)}\right],$$
$$where\ i = 1 \dots 4, j = 1 \dots n$$

در گام بعدی، ماتریس نرمال‌شده را در ماتریس وزن‌ها، ضرب می‌کنیم.

$$t_{ij} = n_{ij} \times W_j$$

در نهایت با محاسبه نزدیکی نسبی، معیار ایده‌آل را به دست می‌آوریم:

$$D_j^+ = \sqrt{\sum_{i=1}^{4} \left(t_{ij} - b_i^+\right)^2} \quad , where\ j = 1 \dots n$$

$$D_j^- = \sqrt{\sum_{i=1}^{4} \left(t_{ij} - b_i^-\right)^2} \quad , where\ j = 1 \dots n$$

$$R_j = \frac{D_j^-}{D_j^- + D_j^+}, where\ j = 1 \dots n\ and\ 0 \le R_j \le 1$$

هرچه میزان R_j نزدیک به ۱ و بیشتر باشد، آن خوشه بهتر است. البته خوشه می‌تواند این پیشنهاد کنترلر را بپذیرد یا نه.

Intra-cluster local decision machining

خوشه در صورتی که درخواست کش‌کردن اطلاعات توسط کنترلر را بپذیرد با این الگوریتم، محتوا را دریافت می‌کند:

Algorithm 1: performed by a cluster head on receiving a new content

```
1.  Function content_recieved(content, free_storage) do
2.    if (content→suggested_by_SDN_for_caching()) do
3.      candidate_nodes = determine_candidate_nodes();
4.      content_value = calculate_content_value(content);
5.      if (free_storage >= content→required_storage) do
6.        cache_the_content(content, candidate_nodes);
7.      else
8.        cache_replacement(content, content_value);
9.      end
10.   end
11.   forward_content_to_next_cluster_on_the_path(content)
12. end
```

انتخاب گره مناسب

هم‌چنین، انتخاب گره مناسب جهت کش محتوا با چند شاخص انجام می‌شود:

- میزان حافظه خالی
- سطح انرژی
- فاصله تا سر خوشه: فاصله گره تا سر خوشه که بر اساس round-trip time محاسبه می‌شود.
- تعداد سنسورهای یکسان: در صورت تعدد سنسورهای یکسان، معمولا به دلیل افزونگی، این تعداد به عنوان یک شاخص در نظر گرفته می‌شود.

با تجمیع همه شاخص‌ها، شاخص وزن کش به دست می‌آید. با استفاده از الگوریتم تحلیل فرآیند سلسله مراتبی (AHP)، اهمیت نسبی شاخص‌ها را نسبت به هم به دست آورده و شاخص وزن کش به دست می‌آید:

$$\begin{array}{c} \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{array} \begin{array}{c} \begin{array}{cccc} m_1 & m_2 & m_3 & m_4 \end{array} \\ \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix} \end{array}$$

$$W_m = \frac{1}{4} \sum_{i,j=1}^{4} \frac{r_{ij}}{\sum_{i=1}^{4} r_{ij}}$$

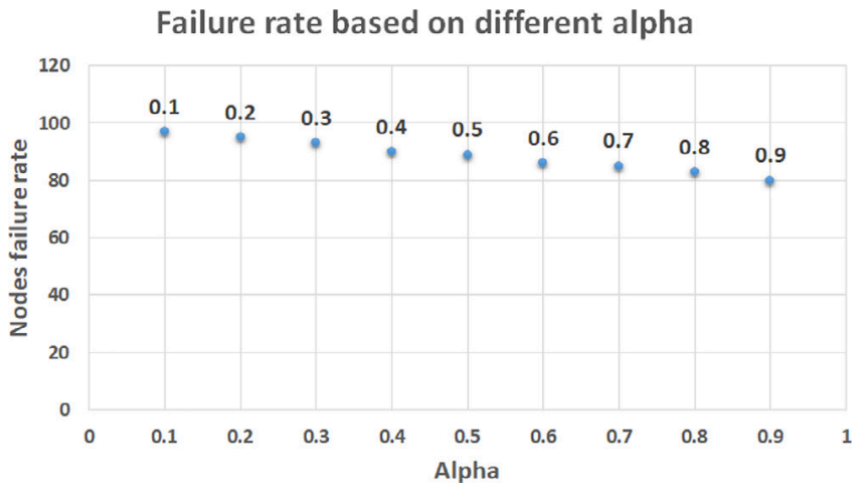
بر اساس این وزن، امتیاز هر گره را به دست می‌آوریم:

$$NS_j = \sum_{i=1}^{4} W_i \times x_i(j)$$

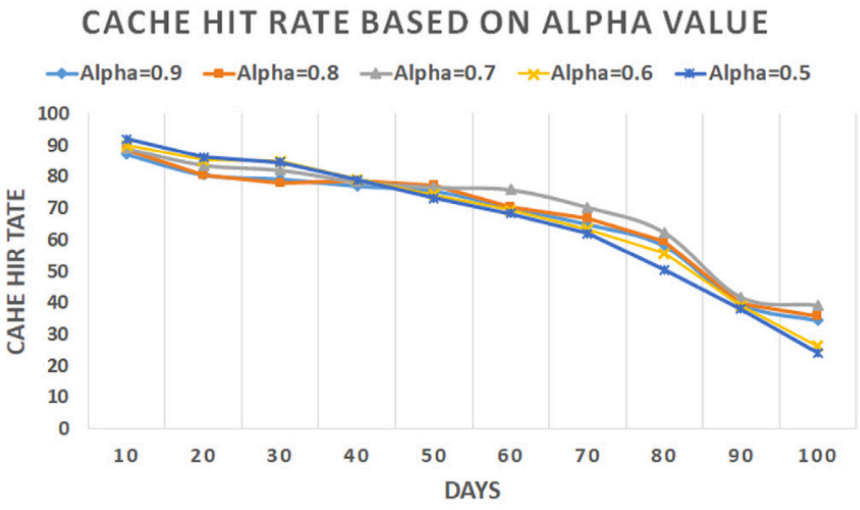
اگر M N S حداکثر میزان امتیاز باشد، T N S یک آستانه است که هر گره با وزن بالاتر از آن، به عنوان کاندیدا مطرح است:

$$TNS = \alpha \times MNS \ \ where \ 0 < \alpha \leq 1$$

پارامتر α، عمر گره و میزان نرخ ضربه کش را تحت تاثیر قرار می‌دهد. در تصویر- ۲ برای مقادیر مختلف این پارامتر در طول ۱۰۰ روز، میزان دستگاه‌های از کار افتاده مشخص شده است. هم‌چنین میزان نرخ ضربه بر اساس روز برای این پارامتر در تصویر- ۳ آمده است. در این مقاله مقدار ۰.۷ برای این پارامتر در نظر گرفته شده است.



تصویر- ۲



تصویر- ۳

انتخاب محتوای کاندید

معیارهای انتخاب محتوای کاندید برای کش شدن:

- تازگی
- محبوبیت
- سایز محتوا

در صورت وجود محتوا در کش، این الگوریتم جهت بازنشانی محتوا به کار می‌رود:

Algorithm 2: Cache replacement algorithm

1. **Function** *Cache_replacement* (w_i, CW[], Sum_of_PDS_storage, required_Storage, new_content) **do**
2. **while** ($sum_of_PDS_storage < required_storage$) **do**
3. $SCW = \sum CW_i$
4. **end**
5. **if** ($\boldsymbol{SCW < W_i}$) **do**
6. *discard_PDS*();
7. *cache_content*(new_content);
8. **else**
9. should_consult_with_SDN = $\left(rand() < \frac{w_i}{SCW}\right) ? true : false$;
10. **if** (should_consult_with_SDN) **do**
11. *Find_next_suitable_cluster*(new_content);
12. **else**
13. *Forward_content_to_cloud*();
14. **end**
15. **end**
16. **end**

نقاط قوت و ضعف مقاله

نقاط قوت این مقاله:

- بررسی دقیق و راه‌کاری همه جانبه
- ارائه شاخص‌های قابل اندازه‌گیری

نقاط ضعف این مقاله:

- چالش Single-point-of-failure برای کنترلر SDN/Cache
- عدم ارائه راه‌کار برای خوشه‌های در حال حرکت

جمع‌بندی و پیشنهادات برای کارهای آتی

در این مقاله روشی به نام EM3C برپایه روش CCN ارائه شده است. در روش ارائه شده، چهار لایه در نظر گرفته شده است و با قراردادن خوشه ها و سر خوشه در ارتباط با کنترلر SDN/Cache، شیوه کش شدن محتوا شکل گرفته است. در این شیوه انتخاب خوشه، گره و محتوای مناسب به صورت جامع و بر اساس شاخص‌های متفاوت طرح‌ریزی شده است.

در این راه‌کار مواردی در نظر گرفته نشده است و به صورت کار آتی می‌توان در نظر گرفت:

- حل چالش Single-point-of-failure برای کنترلر SDN/Cache

- تکمیل راه‌کار برای خوشه‌های در حال حرکت مثل سنسورهایی که در وسایل متحرک هستند.

شبیه‌سازی

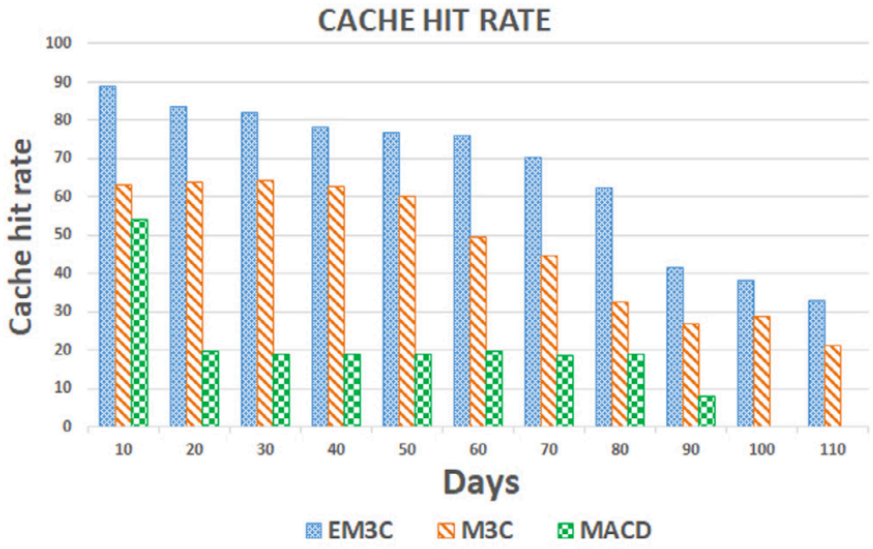
شبیه‌سازی این مقاله در Omnet++ انجام شده است. جهت مقایسه سه روش:

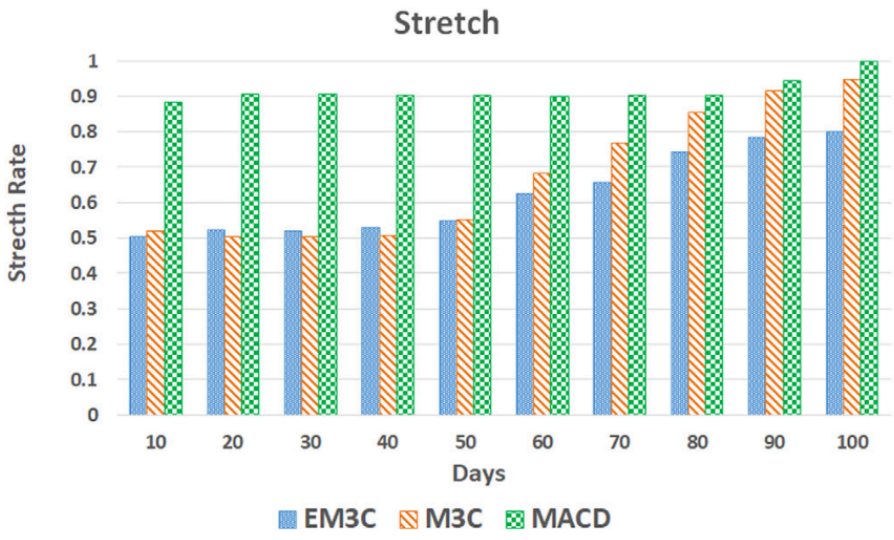
- M3C
- MACD
- EM3C (روش پیشنهادی مقاله)

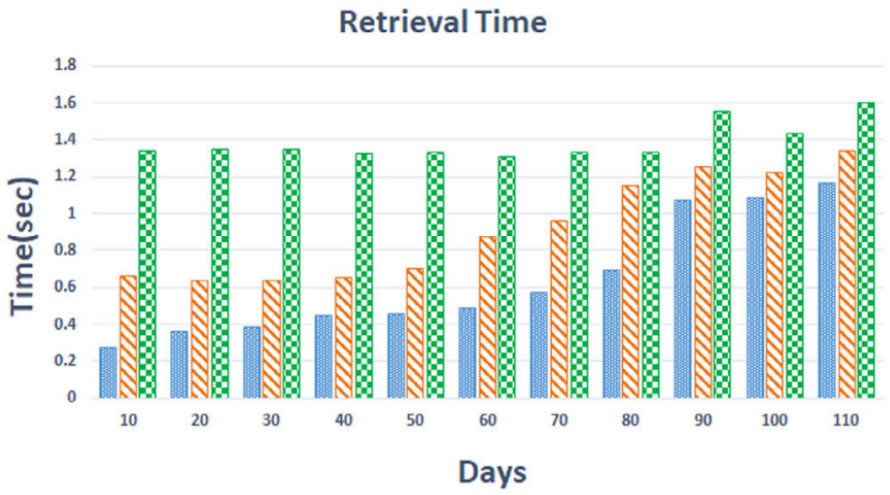
پیاده‌سازی و شبیه‌سازی شده‌اند.

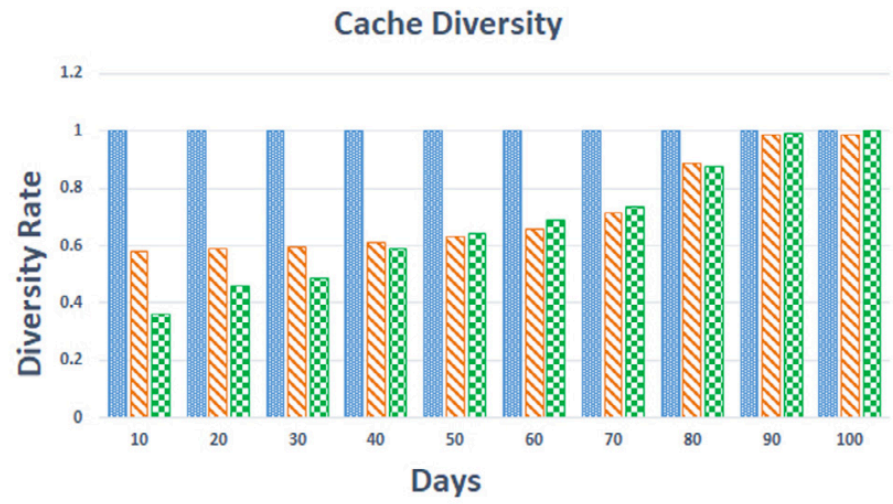
این شبیه‌سازی با حضور ۱۰۰ گره از انواع مختلف و سه نوع حافظه ۱۷، ۲۲ و ۲۷ مگابایتی انجام شد است.

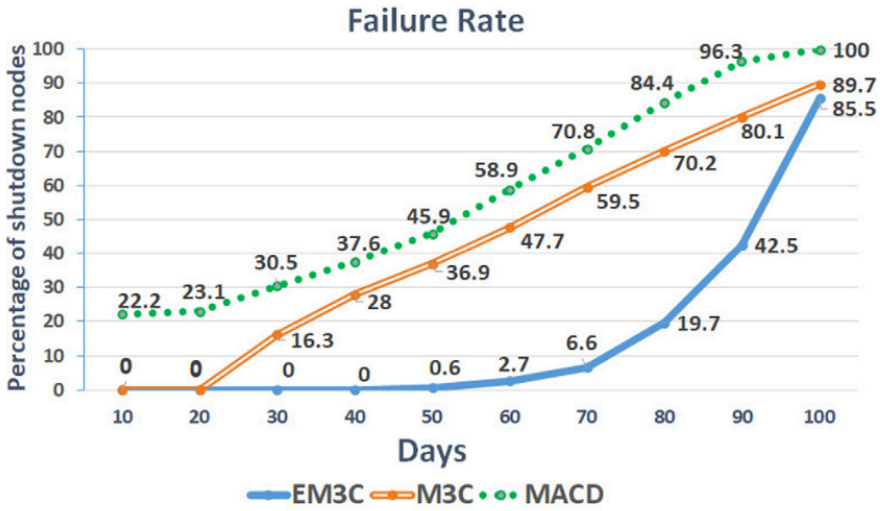
نتایج شبیه سازی در نمودارهای مختلف بر حسب پارامترهای مختلف آورده شده است.





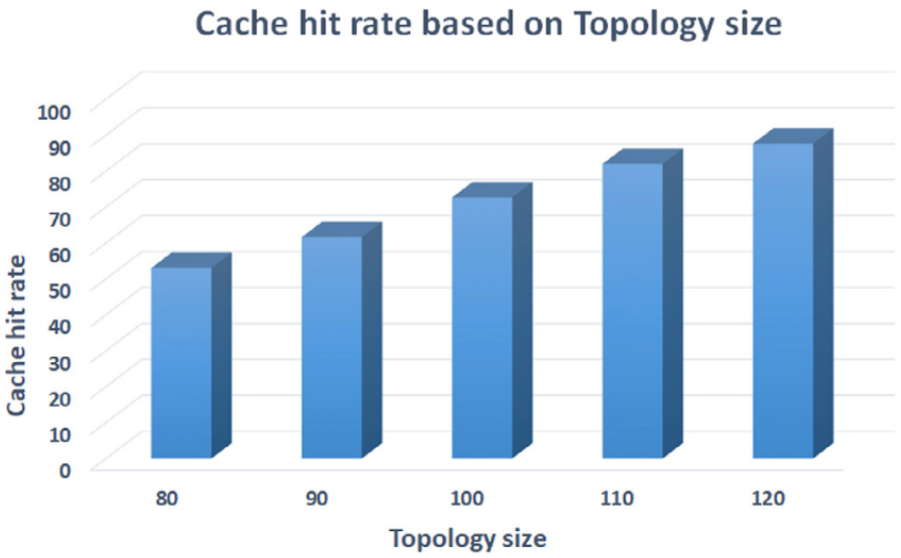


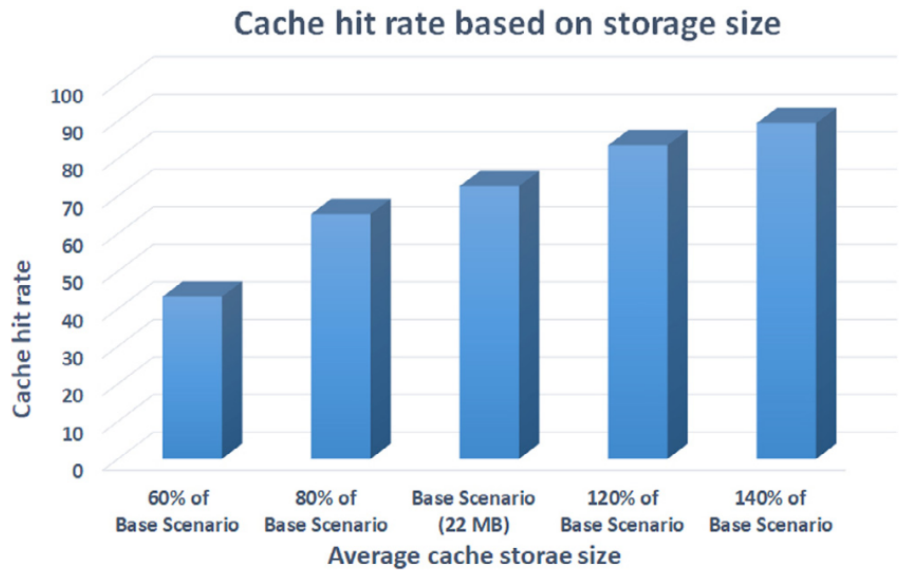




نتایج مقایسه نشان از ۳۵ درصد بهبود نرخ ضربه کش و ۶۰ درصد کاهش زمان دریافت محتوا، در مقابل دیگر روش‌ها دارد.

هم‌چنین تاثیر میزان حافظه و سایز توپولوژی بر نرخ ضربه کش، در نمودارها مقایسه شده است.





در مورد جزییات دقیق شبیه‌سازی، دیتاست و کدهای آن صحبتی نشده است و در این شرایط امکان شبیه‌سازی و اعتبارسنجی خروجی‌های مقاله امکان‌پذیر نیست.

مشخصات دقیق مقاله

Khodaparas, Sahand, Abderrahim Benslimane, and Saleh Yousefi, "A software-defined caching scheme for the Internet of Things", Computer Communications, 2020