

# گزارش ارزیابی کارایی سیستم‌های اینترنت اشیا پزشکی و اینترنت اشیا براساس مجموعه داده‌های CICIoMT2024 و مدلینگ ریاضیاتی استاد ناظر: آقای دکتر مهدی امینیان

علیرضا سلطانی نشان

۲۴ دی ۱۴۰۳

## مجوز

به فایل license همراه این برگه توجه کنید. این برگه تحت مجوز GPLv3 منتشر شده است که اجازه نشر و استفاده (کد و خروجی/pdf) را رایگان می‌دهد.

## فهرست مطالب

۸	۱ شاخش‌های محاسبه و ارزیابی عملکرد
۸	۱.۱ فرمول شانون
۸	۲.۱ فرمول محاسبه بار سیستم یا System load
۹	۳.۱ تاخیر سرویس‌دهی یا Service latency
۹	۱.۳.۱ بخش‌هایی که زمان سرویس‌دهی دارند
۱۰	۲.۳.۱ زمان ارتباطی
۱۰	۳.۳.۱ زمان پردازشی
۱۱	۴.۳.۱ زمان پردازش محلی $t_L$ یا زمان پردازش در هر زیر سیستم $t_{pi}$
۱۱	۵.۳.۱ تابع محاسبه CPU time
۱۲	۶.۳.۱ زمان پردازش محلی با توجه به اندازه داده (D)
۱۲	۴.۱ مصرف انرژی
۱۲	۱.۴.۱ مجموع مصرف انرژی $E_{dev}$
۱۳	۲ مدل‌های ارزیابی کارایی

## فهرست تصاویر

۳	۱ ۴ دسته‌بندی دامنه استفاده از سیستم‌های IoT
۴	۲ حوزه‌های تخصصی بخش اپلیکیشن در IoT
۵	۳ حوزه‌های بخش میان‌افزار در IoT
۶	۴ حوزه‌های بخش شبکه در IoT

## فهرست جداول

۱	تعریف ثابت‌های مورد استفاده در فرمول‌ها . . . . .	۸
مهم‌ترین انگیزه برای توسعه این تحقیق وجود کمبود در داده‌های موجود ارزیابی کارایی تجهیزات اینترنت اشیا پزشکی و پیشرفت امنیتی تمام شبکه‌هایی که در خصوص جریان‌های داده‌ای و پردازش داده‌های پزشکی کار می‌کنند، می‌باشد بخصوص برای دستگاه‌های اینترنت اشیا پزشکی به دلیل اطلاعات حیاتی‌ای که می‌توان به واسطه آن‌ها از بیماران با بیماری‌های مختلف مانیتور و دریافت کرد.		

یکپارچه‌سازی سیستم‌های IoT با سیستم‌های ابری چالش‌های زیادی داشته مثل:

- تاخیرهای شبکه‌ای
- گذردهی
- مصرف انرژی
- قابلیت اطمینان

یه سری مفاهیم جدیدی در حوزه پردازش‌ها مطرح شده که حتی می‌تواند کاربردهای مختلفی در استفاده از اینترنت اشیا باشد. این مفاهیم جدید مثل Fog computing، edge computing، mobile edge computing، mobile cloud computing و Cloudlet ها هستند. در این مقاله یک مدل ریاضیاتی برای توصیف رسمی سیستم‌های IoT ارائه داده شده است. علاوه بر این یک ارزیابی آنالیز شده برای طراحی این سیستم‌ها با استفاده از مطابقت با معماری، تکنولوژی‌ها، پروتکل‌ها و مدل‌های یکپارچه‌سازی برای بهینه‌سازی عملکرد نیز ارائه می‌دهد.

Approach of this article:

بعد از خواندن این مقاله به یک روش بهینه برای بهینه‌سازی کارایی مبتنی بر فرایندهای offloading مانند load balancing آشنا می‌شیم. مدلینگ ریاضیاتی سیستم‌های IoT یک نمایی از سیستم ایجاد می‌کنند که به فهمیدن المان‌ها، تعاملاتشون، و رفتارهاشون کمک می‌کند.

۱. مدل مفهومی یا conceptual model یک ساختار سطح بالایی برای توصیف عملیاتی است که در سیستم‌های IoT انجام می‌شود.

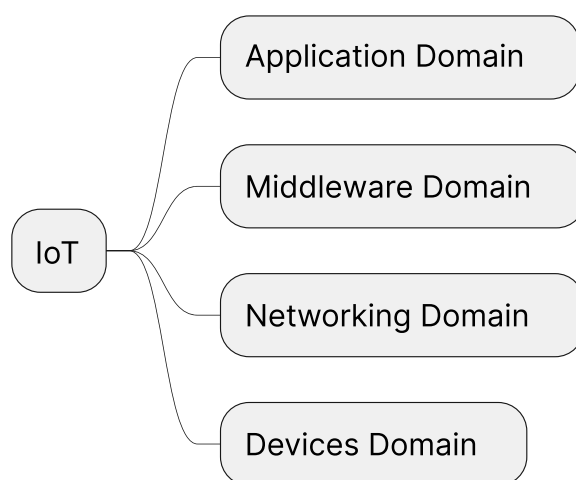
۲. مدل رفتاری یا behavioral model ممکنه شامل جزئیات باشد. مثل جریان داده بین المان‌ها.

به طور کلی مودلینگ به مشخص شدن و پاسخ به مسائل مربوط به کارایی کمک بسزایی می‌کنه و اجازه میده که سیستم‌ها بهینه‌تر، کاراتر و مطمئن‌تر باشن.

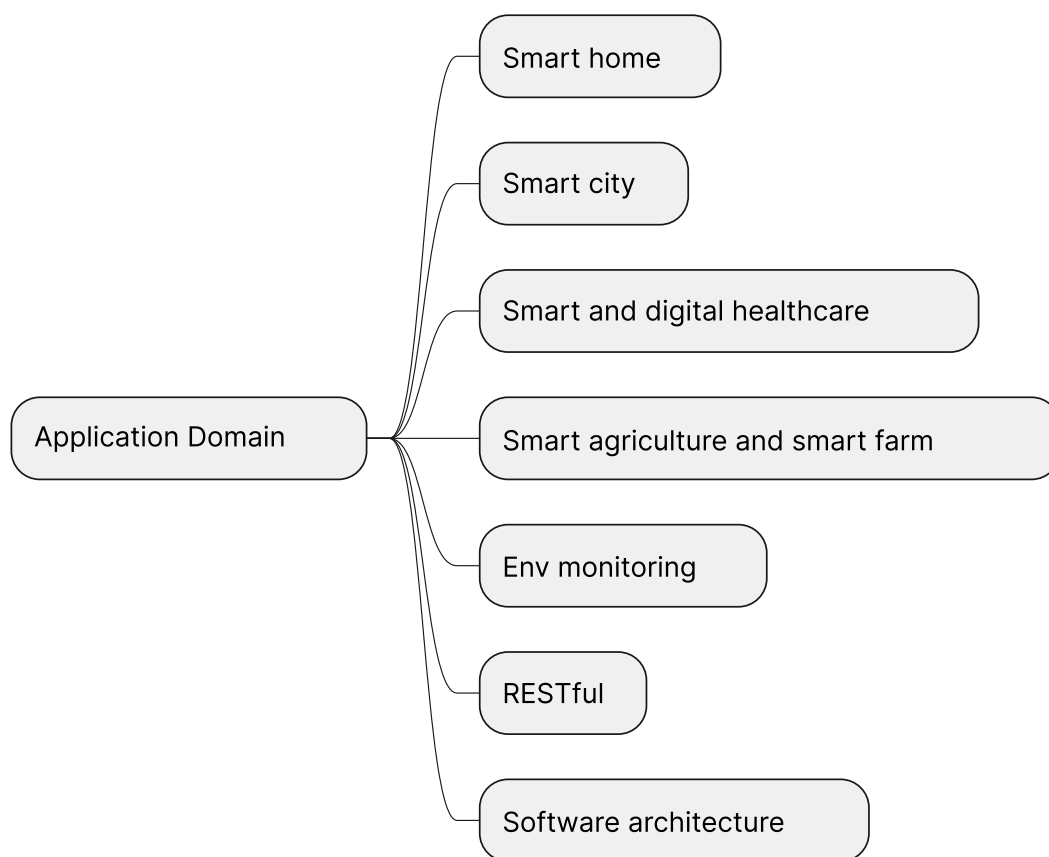
هر موقع در مورد مودلینگ یک سیستم IoT صحبت میشه در حقیقت قراره یه چهارچوبی درست بشه که بتونیم باهاش تست کنیم، تایید یا validation انجام بدیم و یا بتوانیم سیستم را به تقاضاهایی که داریم optimize کنیم.

فرایند سیستم‌های IoT معمولاً شامل شناسایی، دریافت اطلاعات، (Sensing) فعالیت‌های تحت شبکه، و محاسبات کوچک هستند که باعث میشه با محیط فیزیکی و هر اشیا یی ارتباط برقرار کند.

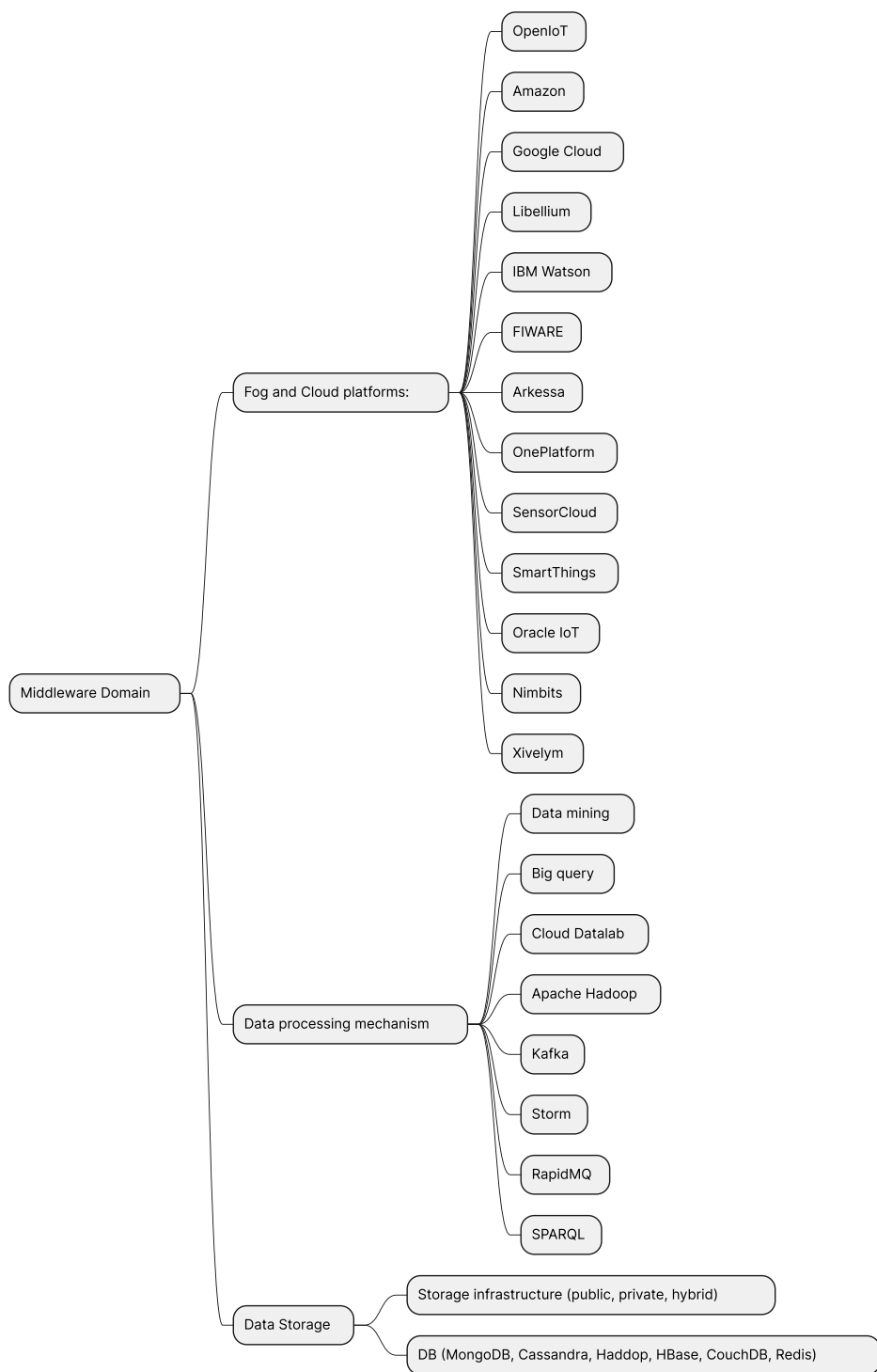
در این مقاله سیستم‌های IoT با کاربردی که دارند به ۴ دسته تقسیم می‌شوند:



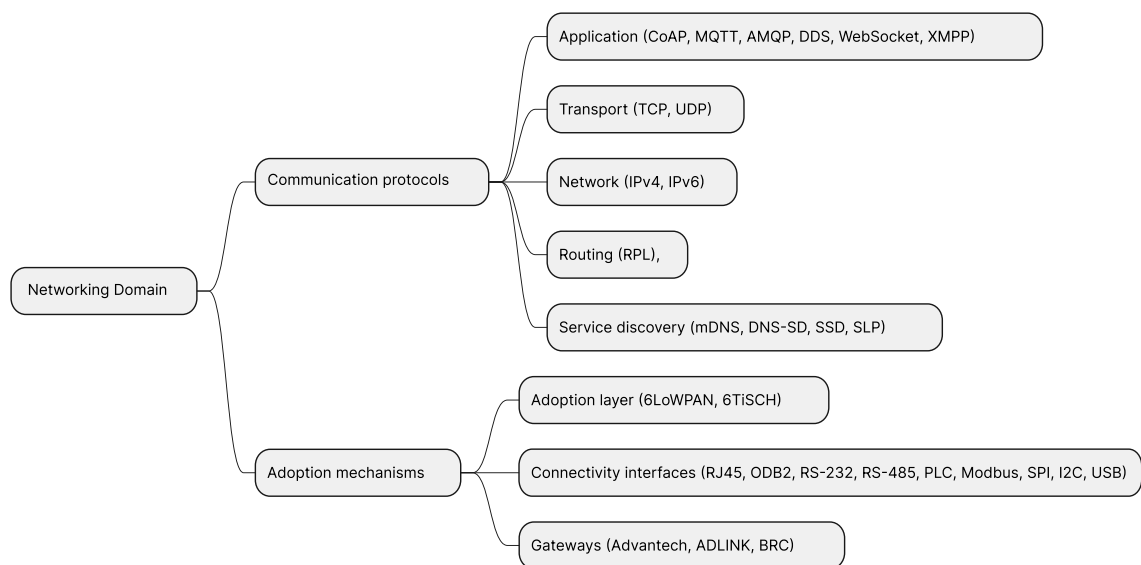
شکل ۱: ۴ دسته‌بندی دامنه استفاده از سیستم‌های IoT



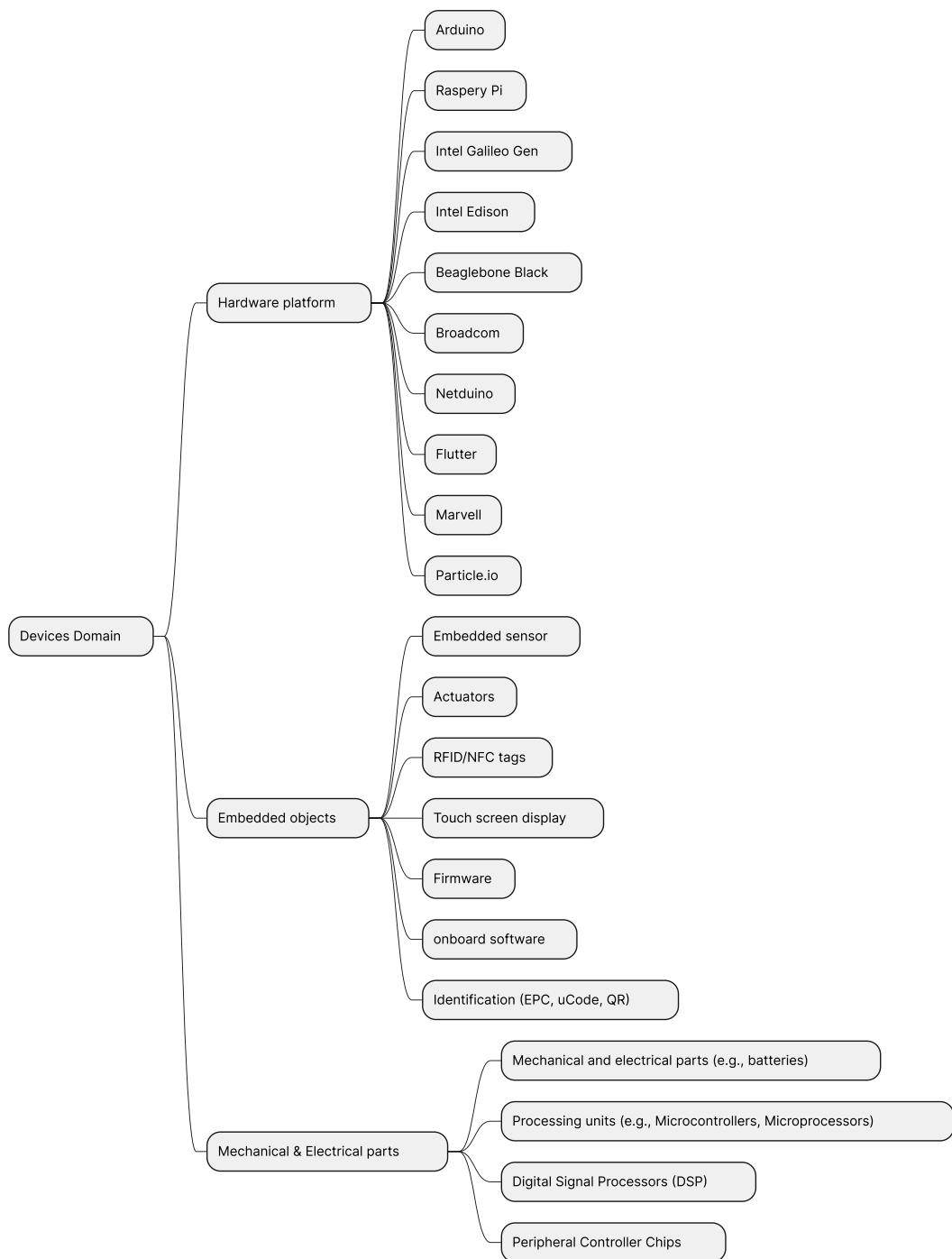
شکل ۲: حوزه‌های تخصصی بخش اپلیکیشن در IoT



شکل ۳: حوزه‌های بخش میان‌افزار در IoT



شکل ۴: حوزه‌های بخش شبکه در IoT



شکل ۵: حوزه‌های بخش Embedded در IoT

عمومیت جریان داده در سیستم‌های IoT:

- دریافت داده
- انتقال داده
- پردازش داده

- ذخیره‌سازی داده
- آنالیز و معنادار کردن داده

## ۱ شاخش‌های محاسبه و ارزیابی عملکرد

تعاریف	ثابت‌ها
نرخ ورود اطلاعات	$D_{rate}$
مصرف انرژی	$E_{dev}$
تاخیر سرویس‌دهی	$T_{exe.}$
مشخصات سیستم IoT	$IoT_{sys_{sp}}$
زمان ورکلودها	$t_{ws}$
تعداد دستگاه‌ها	$k$
نیازمندی مشخص کارایی $P_{ireq}$ جایی که $P_i$ مقدار از کارایی است	$IoT_{sys}(P) = \{P_{ireq}, P_i\}$
مدت زمان سرویس‌دهی	$P_i = \langle T_{exe.}, E_{dev} \rangle$

جدول ۱: تعریف ثابت‌های مورد استفاده در فرمول‌ها

### ۱.۱ فرمول شانون

نرخ داده‌ها می‌تواند از طریق فرمول شانون محاسبه شود.

$$D_{rate} = B_{i,j} \log_2 \left( 1 + \frac{|h_{ij}|^2 \cdot P_{tx}}{P_{Nj}} \right) \quad (1)$$

- $B_{ij}$ : پهنای باند
- $h_{ij}$ : بهره کانال بین دستگاه مبدا و مقصد که نشان می‌دهد سیگنال چگونه در مسیر بین فرستنده و گیرنده تقویت یا تضعیف می‌شود.
- $P_{tx}$ : توان ارسال
- $P_N$ : میزان نویز مقصد

کاربرد زیادی در سناریوهایی دارد که در آن یک ارسال کننده و یک دریافت کننده وجود دارد.

### ۲.۱ فرمول محاسبه بار سیستم یا System load

مجموع بار سیستم از طریق فرمول زیر بدست می‌آید:

$$D = \sum_{k=1}^N D_{rate,k} \times T_{w,k} \quad (2)$$

- $D_{rate,k}$ : نرخ تولید داده توسط دستگاه IoT
- $t_{w,k}$ : مدت زمان ورودی در دستگاه IoT یا به عبارتی دیگر، مدت زمانی که طول می‌کشد یک دستگاه IoT ورودی را دریافت و سپس آن را پردازش و هندل کند.



### ۳.۱ تاخیر سرویس‌دهی یا Service latency

مسئله service latency یا service execution time ( $T_{exe.}$ ) مدت زمانی است که طول می‌کشد سیستم IoT تمام درخواست‌های پردازشی و ارتباطی را اجرا کند (the total application exe time). مدت زمان کل مصرف شده از، مدت زمان سرویس یک ریکوئست به مدت زمان تمام تسک‌هایی که با موفقیت پردازش شده‌اند. بنابراین، این فاصله زمانی بین درخواست برنامه و بدست آوردن نتایج می‌باشد.

#### ۱.۳.۱ بخش‌هایی که زمان سرویس‌دهی دارند

- مدت زمان انتقال داده از دستگاه IoT به زیر ساخت فاگ
- مدت زمان انتقال داده از دستگاه IoT به سرورهای ابری
- مدت زمان انتقال داده از فاگ به کلاد
- مدت زمان انتقال اعلانات از کلاد به فاگ
- مدت زمان انتقال اعلانات از کلاد به دستگاه‌های IoT
- مدت زمان انتقال اعلانات از فاگ به دستگاه IoT
- مدت زمان محاسبات در دستگاه IoT
- مدت زمان محاسبات در لایه فاگ
- مدت زمان محاسبات در سرورهای ابری

نکته: مدت زمانی که برای هر کاری در سیستم‌های IoT سپری می‌شود به نوع و شیوه پیاده‌سازی معماری دستگاه‌ها و نرم‌افزار بخش‌ها بستگی دارد و می‌تواند کاملاً متفاوت باشند. عموماً سرویس لیتنسی بین المان‌های سیستم IoT توزیع شده هستش و شامل دستگاه‌های اینترنت اشیا، شبکه‌ها و سیستم‌های پردازشی می‌شود.  
تأخیر سرویس‌دهی:

$$T_{exe.} = T_{cm} + T_{cp} \quad (۳)$$

•  $T_{cm}$ : مدت زمان تأخیر در ارتباطات

•  $T_{cp}$ : مدت زمان تأخیر در پردازش

مدت زمان اجرا بایستی کمتر از زمان‌بندی تسک‌ها در فاگ یا کلاد باشد. یعنی سرویس تایم باید کمتر از نیازمندی‌های IoT application ( $T_{req}$ ) باشد.

برای کاهش service latency از فرمول زیر بایستی پیروی کند:

$$Objective : \min(T_{exe.}) = T_{cm} + T_{cp} \leq T_{req} \quad (۴)$$

موقعی که داری در مورد Execution time می‌نویسی فرمول communication latency رو باید داشته باشی که بگی از کجا بدست میاد. و همچنین فرمول computation latency رو هم بعدش. یعنی ثابت‌ها خودشون از زیر ثابت‌های اصلی بدست میان که جمع میشن و میشه  $T_{exe}$ .

### ۲.۳.۱ زمان ارتباطی

$$T_{cm} = \sum_{i=1}^N (d_{proc} + d_{queue} + d_{trans} + d_{prop}) \quad (۵)$$

•  $d_{prop}$ : تاخیر پردازشی

•  $d_{queue}$ : تاخیر در صف

•  $d_{trans}$ : تاخیر انتقال

•  $d_{prop}$ : تاخیر توزیع

تاخیر مربوط به Propagation مجموع زمان مورد نیاز برای داده جهت ارسال از منبع به مقصد که مبتنی بر طول لینک فیزیکی و سرعت رسانی می باشد.

$$d_{trans} = \frac{P_s}{R_L} \quad (۶)$$

که در آن:

•  $P_s$ : اندازه بسته در واحد bits

•  $R_L$ : سرعت لینک ارتباطی bps

$$d_{prop} = \frac{l_{ij}}{c} \quad (۷)$$

•  $l_{ij}$ : لینک فیزیکی

•  $c$ : سرعت توزیع media

### ۳.۳.۱ زمان پردازشی

$$T_{cp} = T_L + \sum_{i=1}^k t_{offi} \quad (۸)$$

که در آن:

•  $t_L$ : اجرا و پردازش های داخلی

•  $t_{offi}$ : اجرا و پردازش های خارج از دستگاه IoT مانند برنامه هایی که در سیستم های ابری یا Fog مستقر شده اند که وظیفه پردازش تسک های Offloading را دارند.

به بیان دیگر می توان آن را به صورت مدل زیر محاسبه کرد:

$$T_{cp} = t_L + t_F + t_C \quad (۹)$$

$$T_{cp} = t_L + \max_{i=1, \dots, k} t_{Fi} + \max_{j=1, \dots, n} t_{Cj} \quad (۱۰)$$

که در آن:

- $t_L$ : مدت زمان پردازش‌های داخلی

- $t_{Fi}$ : مدت زمان پردازش در نود  $i^{th}$  در Fog

- $t_{Ci}$ : مدت زمان پردازش در سرور  $i^{th}$  ابری

عموماً مصرف پردازشی بستگی به سرعت و معماری پردازنده مرکزی (CPU)، حافظه رم (RAM)، سرعت حافظه ذخیره‌ساز (HDD) یا (SSD)، سرعت پردازنده گرافیکی یا (GPU) و غیره دارد.

#### ۴.۳.۱ زمان پردازش محلی $t_L$ یا زمان پردازش در هر زیر سیستم $t_{pi}$

برای بدست آوردن زمان پردازش در هر زیر سیستم از فرمول زیر استفاده می‌شود:

$$t_{pi} = \frac{ICC_i}{f_{cpu,i}} \quad (11)$$

- $t_{pi}$ : زمان پردازشی در زیر سیستم  $i$

- $ICC_i$ : تعداد سایکل‌های CPU که برای اجرای یک برنامه نیاز است.

- $f_{cpu,i}$ : نرخ کلاک (فرکانس کاری CPU) زیر سیستم  $i$

$$t_{Pi} = t_{CPU_i} + t_{I/O_i} \quad (12)$$

#### ۵.۳.۱ تابع محاسبه CPU time

مدت زمانی که در CPU برای اجرا برنامه در نظر گرفته می‌شود به دو دسته تقسیم می‌شود:

۱. User CPU time

۲. System CPU time:  $t_{OS}$

محاسباتی که در CPU time انجام می‌شود خالصانه در قسمت پردازشگر مرکزی صورت می‌گیرد و هیچ محاسبه جانبی مانند مدت زمان I/O و مدت زمان اجرای دیگر برنامه‌ها در نظر گرفته نمی‌شود.

$$t_{cpu_i} = \frac{ICC_i}{f_{cpu_i}} + t_{OS} = ICC_i \times t_{cc_i} + t_{OS} \quad (13)$$

حاصل این تابع معمولاً بسیار کوچک است و می‌تواند نادیده گرفته شود زیرا به سمت صفر میل می‌کند ( $t_{OS} \rightarrow 0$ ). به همین خاطر بیشتر روی User CPU time تمرکز می‌کند که توسعه‌دهنده بر روی آن کدهای خود را اجرا می‌کند و سیستم IoT را راه‌اندازی می‌کند.

تابع مطرح شده بر اساس قدرت محاسباتی دستگاه ( $f_{cpu_i}$ ) و تعداد کلاک CPU برای اجرای یک برنامه ( $ICC_i$ ) می‌باشد. مقدار  $f_{cpu_i}$  بر واحد (Hz) می‌باشد و  $t_{cc_i}$  زمان چرخه کلاک است. لازم به ذکر است که  $ICC_i$  به نوع دستورالعمل که شامل اندازه داده ورودی، زبان برنامه نویسی، میزان پیچیدگی الگوریتم نرم‌افزاری مورد استفاده، و دیگر موارد می‌باشد.

بخش CPI (Clock cycles per instruction) به عنوان میانگین تعداد چرخه کلاک است که هر دستورالعمل به آن نیاز دارد. اگر  $I_{app_j}$  تعداد دستورالعمل‌ها برای یک برنامه باشد آن وقت  $ICC_i$  از طریق معادله زیر بدست می‌آید.

$$ICC_i = \sum_{j=1}^k I_{app_j} \times CPI_j \quad (14)$$

### ۶.۳.۱ زمان پردازش محلی با توجه به اندازه داده (D)

$$I_{CC_i} = X \times D \quad (15)$$

- $D$ : اندازه ورودی داده بر حسب بیت
  - $X$ : شدت پردازش (تعداد چرخه‌های مورد نیاز برای هر بیت داده)
- بنابراین خواهیم داشت:

$$t_{pi} = \frac{\beta_i \times D \times X}{f_{cpu,i}} \quad (16)$$

که در آن  $\beta_i$  درصد داده‌ای است که در سیستم  $i$  پردازش می‌شود.

## ۴.۱ مصرف انرژی

### ۱.۴.۱ مجموع مصرف انرژی $E_{dev}$

برای محاسبه مصرف کل انرژی در سیستم‌های IoT می‌توان از فرمول زیر استفاده کرد:

$$E_{dev} = F(E_{cp}, E_{cm}, E_{idle}, E_{other}) \quad (17)$$

- $E_{cp}$ : انرژی مصرف شده طی محاسبات
- $E_{cm}$ : انرژی مصرف شده در طی ارتباطات
- $E_{idle}$ : انرژی مصرف شده در حالت نرمال و بیکار سیستم
- $E_{other}$ : انرژی مصرف شده توسط بقیه فرایندها مانند سنسورها، صفحه نمایش، کارت گرافیک و غیره.

در یکی از کارها [۱] مدلی برای مصرف انرژی نسبت به انتقال و جا به جایی داده‌ها مطرح شده که سطوح مختلف مصرف باتری را برای uplink و downlink شامل می‌شود:

$$P_{tx} = p_u \tau_u + \beta \quad (18)$$

$$P_{rx} = p_d \tau_d + \beta \quad (19)$$

گذردهی uplink،  $\tau_u$  می‌باشد و گذردهی downlink،  $\tau_d$  است. در حالی که  $p_u$  و  $p_d$  میزان انرژی مورد نیاز برای انتقال داده‌ها در uplink و downlink می‌باشد. ثابت  $\beta$  میزان مصرف انرژی در حالت idle می‌باشد. این مقادیر کاملاً به تکنولوژی ارتباطی، پروتکل‌ها و دستگاه‌هایی که روی آن برنامه مستقر شده است وابسته می‌باشد. برای انتقال همزمان uplink و downlink سطح انرژی می‌تواند با فرمول زیر محاسبه شود:

$$P_{trx} = p_u \tau_u + p_d \tau_d + \beta \quad (20)$$

نسبت معادله uplink بر روی downlink به همراه پهنای باند، می‌تواند فرمول را برای محاسبه انرژی بهینه شبکه برای انتقال یک مقدار داده معین (energy per bit). مقدار انرژی مورد نیاز برای ارسال از طریق فرمول  $E(D)_{rx}$  میزان انرژی مورد نیاز برای دریافت داده‌ها از طریق فرمول  $E(D)_{rx}$  حاصل می‌شود.

$$E(D)_{tx} = p_u + \beta\tau_u^{-1} \quad (21)$$

$$E(D)_{rx} = p_d + \beta\tau_d^{-1} \quad (22)$$

مقدار داده‌هایی که بر واحد بیت توسط دستگاه‌های IoT ارسال و دریافت می‌شوند به ترتیب  $D_{tx}$  و  $D_{rx}$  می‌باشند. با در نظر گرفتن محاسبات پیشین، می‌توان در نهایت میزان مصرف انرژی توسط دستگاه‌های IoT را به شکل زیر بدست آورد:

$$E_{dev} = (P_L \times t_L) + (P_{tx} \times t_{tx}) + (P_{rx} \times t_{rx}) + (P_{id} \times t_{off}) \quad (23)$$

$$E_{dev} = (P_L \times t_L) + ((p_u\tau_u + \beta) \times t_{tx}) + ((p_d\tau_d + \beta) \times t_{rx}) + (P_{id} \times t_{off}) \quad (24)$$

$$E_{dev} = (P_L \times t_L) + ((p_u + \beta\tau_u^{-1}) \times D_{tx}) + ((p_d + \beta\tau_d^{-1}) \times D_{rx}) + (P_{id} \times t_{off}) \quad (25)$$

یکی از مهم‌ترین چالش‌های دستگاه‌های IoT مربوط به مصرف باتری آن‌ها می‌باشد. در بعضی مواقع دستگاه‌های IoT از باتری‌هایی استفاده می‌کنند که شرایط جایگزین کردن آن‌ها وجود ندارد. هر دستگاه IoT حتی در حالت بیکار انرژی بابت، پردازش داده‌ها، ارسال و دریافت داده‌ها مصرف می‌کنند. انرژی موجود  $E_{dev}(t)$  در طی زمان کاهش پیدا می‌کند. به همین ترتیب انرژی باقی‌مانده  $E_{dev}(r)$  یا مدت زمانی که سیستم می‌تواند روشن بماند از طریق فرمول زیر بدست می‌آید.

$$E_{dev}(r) = E_{dev}(i) - E_{dev}(t) \quad (26)$$

•  $E_{dev}(i)$ : مقدار اولیه انرژی دستگاه

مقدار باتری باقی‌مانده  $T(sys)$  به میزان ظرفیت باطری یا انرژی باقی‌مانده و انرژی مورد نیاز دستگاه برای انجام تمام سرویس‌های دستگاه، بستگی دارد. انرژی مصرفی وابسته به قدرت مورد نیاز برای پردازش‌های داخلی  $P_{cp}$  انتقال داده‌ها  $P_{cm}$  و بقیه فرایندها  $P_{other}$  می‌باشد.

$$T(sys) = \frac{E_{dev}(r)}{P_{cp} + P_{cm} + P_{other}} \quad (27)$$

یکی دیگر از چالش‌های دستگاه‌های IoT مربوط به منبع تغذیه آن‌ها می‌باشد. در مواقعی که دستگاه‌های IoT از باتری استفاده می‌کنند و به دلیل محیط‌های مختلف شرایط به گونه‌ای است که امکان تعویض باتری وجود ندارد، محدودیت‌های باتری اغلب به عنوان شاخص طول عمر دستگاه‌های IoT استفاده می‌شود و می‌تواند به عنوان یکی از مهم‌ترین معیارهای QoS مورد استفاده قرار گیرد. هدف اصلی در این دستگاه‌ها این است که مصرف انرژی به حداقل برسد و طول عمر کلی سیستم به بیشترین حد ممکن.

## ۲ مدل‌های ارزیابی کارایی

- [1] Huang, Junxian, Qian, Feng, Gerber, Alexandre, Mao, Z Morley, Sen, Subhabrata, and Spatscheck, Oliver. A close examination of performance and power characteristics of 4g lte networks. in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pp. 225–238, 2012.