

Маковецький С. О., Каук В. І.

## МЕТОД ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КРАЙОВИХ ВУЗЛІВ ІНТЕРНЕТ-РЕЧЕЙ MESH-МЕРЕЖ В УМОВАХ ХИБНИХ СПРАЦЮВАНЬ

**Предметом вивчення** є моделі й методи розроблення програмного забезпечення крайових вузлів (*edge nodes*) розподілених сенсорних мереж Інтернету речей з топологією *mesh*, що функціонують у нестационарному шумовому середовищі. **Мета дослідження** – розробити архітектурну модель і метод проєктування програмного забезпечення крайових вузлів, які поєднують спектральне виявлення подій із адаптивним відстеженням рівня шуму й локальним прийняттям рішень без залучення хмарного оброблення. **Завдання:** формалізувати трирівневу декомпозицію програмного забезпечення вузла; запропонувати шаблон триступеневого придушення хибних спрацювань як ортогональну композицію функціональних захистів; проаналізувати обчислювальну складність і обсяг пам'яті для двох альтернативних реалізацій шаблону; отримати аналітичну модель зростання трафіку. **Методи дослідження:** архітектурне моделювання програмних систем, теорія цифрового оброблення сигналів (швидке перетворення Фур'є, медіанна й експоненціальна фільтрація), аналітичне моделювання мережевого трафіку. **Досягнуті результати.** Сформовано трирівневу архітектурну модель, що ортогонально поєднує селективну спектральну фільтрацію, темпоральну персистентну фільтрацію та адаптивне відстеження рівня шуму. Розроблено шестиступінний метод проєктування з критеріями верифікації кожного компонента. Аналітично продемонстровано, що обидва варіанти реалізації шаблону задовольняють обмеження мікроконтролерів класу ARM Cortex-M0/M0+ із тактовою частотою 64 МГц у межах розрахункових 21,5 мс на кадр і 2,7 КБ оперативної пам'яті. **Висновки.** Запропоновані модель і метод подано на рівні шаблону проєктування, що допускає повторне використання в різних апаратних і доменних інстанціях. Шаблон ідентифікує три взаємодоповнювальні захисні механізми та встановлює, що відсутність будь-якого з них породжує клас хибних спрацювань, не покритий рештою. Результати слугують методологічною основою розроблення програмного забезпечення для розподілених систем визначення руху об'єктів і периметральної безпеки.

**Ключові слова:** розподілені системи; *mesh*-мережі; архітектура програмного забезпечення; шаблон проєктування; Інтернет речей; визначення руху об'єктів; адаптивне порогове детектування; цифрове оброблення сигналів; вбудовані системи; мікроконтролери.

### 1. Вступ

Розподілені сенсорні системи на основі архітектури Інтернету речей із *mesh*-топологією дедалі частіше застосовуються для завдань, у яких визначальну роль одночасно відіграють три вимоги: висока ймовірність правильного виявлення цільових подій, низька частота хибних спрацювань і обмежений енергетичний бюджет вузла. Серед прикладних доменів – периметральна безпека критичної інфраструктури, екологічний моніторинг, логістика безпілотних транспортних засобів, виявлення несанкційних переміщень об'єктів у промислових і оборонних застосуваннях.

Незважаючи на значний прогрес апаратних платформ, зокрема 32-розрядних мікроконтролерів класу ARM Cortex-M і низькоспоживчих радіоінтерфейсів LoRa, NB-IoT, Zigbee, BLE, програмне забезпечення для таких систем досі розробляється переважно за допомогою *ad-hoc*-методів. Архітектурні рішення часто виносяться

в одну монолітну прошивку без декомпозиції на рівні відповідальностей, що ускладнює верифікацію поведінки в реальному часі, перешкоджає повторному використанню коду між апаратними платформами й також обмежує можливості незалежного тестування окремих захисних механізмів.

Дослідження останніх років розрізняють дві ортогональні стратегії боротьби з мережевими обмеженнями. Перша з них спрямована на скорочення завантаження каналу через часодомений адаптивний поріг і селективну передачу, проте робить це ціною прорідження даних і, як наслідок, ризику пропуску транзйентних (короткочасних) подій. Друга стратегія забезпечує швидке прийняття рішень безпосередньо на вузлі через крайовий спектральний аналіз із фіксованим порогом, але втрачає стійкість у нестационарному шумовому середовищі. Поєднання переваг обох підходів у межах єдиної архітектурної моделі програмного забезпечення потребує методологічного оформлення на рівні шаблонів проєктування.

Об'єктом дослідження є процеси оброблення та передачі інформації в розподілених сенсорних мережах Інтернету речей із крайовою архітектурою, а предметом – моделі, методи й шаблони проектування програмного забезпечення крайових вузлів, що сприяють цілісності детектування за мінливих умов зовнішнього шуму.

## 2. Аналіз літературних джерел і визначення проблеми

Сучасні дослідження у сфері крайових обчислень для Інтернету речей систематизовано в роботах [1, 2], де детально розглянуто методи скорочення обсягу даних на сенсорному рівні. Серед них переважають часодоменні підходи, що використовують прогнозування або адаптивне відсікання повторюваних значень. Так, у дослідженні [3] продемонстровано підхід автономної адаптивної порогової редуції трафіку IoT-вузла, проте притаманне прорідження даних створює ризик пропуску транзйентних подій. У роботі [4] поєднано розподілене прогнозування з компресією та встановлено скорочення передачі до 96%. Автори статті [5] розширюють цю ідею на туманний (*fog*) рівень, виконуючи дворівневу агрегацію в періодичних сенсорних мережах. Окрему лінію досліджень формують роботи [6, 7], присвячені огляду методів стиснення із втратами й застосуванню глибокого навчання з підкріпленням до стиснення IoT-даних. Спільним недоліком усього цього класу методів є оптимізація для ефективності каналу зв'язку, а не цілісність детектування [8, 9].

Альтернативний клас методів становить спектральне виявлення подій безпосередньо на крайовому вузлі. У роботі [10] впроваджено STFT-аналіз для контролю якості електроенергії в IoT-системах моніторингу промислових моторів; у дослідженні [11] використано двійкові згорткові мережі для кодування часових рядів у спектральні ознаки на крайовому вузлі. У публікації [12] запропоновано систематичний огляд методів виявлення аномалій на основі штучного інтелекту в IoT, що підтверджує застосовність FFT-оброблення на ARM-сумісних мікроконтролерах. Ефективність такого оброблення зумовлена класичним результатом [13] (алгоритм Кулі – Тьюкі), який зменшує асимптотичну складність із  $O(L^2)$  до  $O(L \times \log(L))$ , що виявилось передумовою практичної реалізації спектрального оброблення на ARM-платформах [14].

В окремій групі робіт подано методи на базі TinyML. У дослідженні [15] описано неконтрольований автоенкодерний підхід до виявлення аномалій міського шуму на ресурсно-обмежених вузлах; у роботі [16] продемонстровано застосування TinyML для виявлення аномальних вібрацій у моторах. Проте таким методам властива принципова вада фіксованої референсної моделі: за повільного дрейфу шумового середовища необхідне перенавчання, що практично неможливе на мікроконтролерах класу Cortex-M0/M0+ без хмарної підтримки.

Адаптивні порогові методи з радіолокаційного кореня (CFAR), детально розглянуті в класичних роботах [17, 18], застосовують просторове усереднення сусідніх дистанційних комірок. У контексті вузлів Інтернету речей припущення про просторову однорідність порушується, оскільки вузли розташовані несинхронно, з різними локальними завадами, що робить пряме перенесення CFAR-схеми проблематичним. Натомість фундаментом стійкого непараметричного згладжування є медіанна фільтрація, запропонована в монографії [19] і формально досліджена в праці [20]. Її 50-відсоткова точка пробою забезпечує властивість, за якої до половини відліків у вікні можуть бути довільно спотворені без впливу на вихід.

Окремим напрямом досліджень є архітектурні моделі для крайових систем. У дослідженні [21] обґрунтовано доцільність попереднього локального оброблення перед хмарною передачею. У роботі [22] виокремлено інтелектуальне керування крайовими *mesh*-системами як ключову нерозв'язану проблему автономних сенсорних мереж. Огляд праць [8, 9] демонструє, що переважна більшість архітектурних робіт оперує абстракцією рівня системи (*cloud-fog-edge*), не опускаючись до рівня шаблону проектування, що описує конкретну композицію функціональних компонентів усередині прошивки крайового вузла. У студіях [23, 24] розглянуто колаборативні *cloud-edge*-схеми передбачення, проте без виокремлення повторно використовуваних шаблонів проектування програмного забезпечення. У своїй попередній роботі [25] автор аналізує енергоефективне застосування LPWAN-технологій у мережах резервного зв'язку для умов обмеженої інфраструктури, що корелює з тематикою *mesh*-надбудов поверх SX1276, обговорених у підрозд. 4.5 цієї публікації.

Аналіз літератури дає змогу виокремити прогалину в методології розроблення програмного

забезпечення крайових вузлів, що містить три взаємопов'язаних компоненти. По-перше, відсутній архітектурний шаблон, який ортогонально поєднував би адаптивний поріг для стійкості до нестационарного шуму, спектральну фільтрацію для відсікання вузькосмугових завад і темпоральну персистентність для відсікання миттєвих імпульсів. По-друге, відсутній метод проектування, що уможливував би верифікування кожного компонента захисту незалежно. По-третє, відсутня кількісна модель впливу хибних спрацювань крайового вузла на сукупний трафік багатовузлової транзитної *mesh*-мережі.

### 3. Мета й завдання дослідження

Мета дослідження – розробити архітектурну модель і метод проектування програмного забезпечення крайових вузлів *mesh*-мереж Інтернету речей, що дадуть змогу поєднувати спектральне виявлення подій з адаптивним відстеженням рівня шуму й локальним прийняттям рішень без залучення хмарного оброблення.

Для досягнення окресленої мети необхідно виконати такі завдання:

- формалізувати тривірневу декомпозицію програмного забезпечення вузла за рівнями відповідальності з явним розшаруванням компонентів спектрального оцінювання, адаптивного відстеження шуму й прийняття рішення;

- запропонувати шаблон триступеневого придушення хибних спрацювань як ортогональну композицію функціональних захистів, де кожен ступінь нейтралізує окремий клас механізмів спрацювання;

- проаналізувати обчислювальну складність і обсяг оперативної пам'яті для двох альтернативних реалізацій шаблону (на основі усереднення та медіанної фільтрації) для платформ ARM Cortex-M0/M0+;

- отримати аналітичну модель зростання трафіку в багатовузловій транзитній *mesh*-топології, що пов'яже локальну частоту хибних спрацювань вузла з обсягом сукупного навантаження мережі;

$$Q = \alpha \times FNR + \beta \times FPR \times A_{\text{traffic}}, \text{ якщо } C_{\text{proc}} \leq C_{\text{max}}, M_{\text{RAM}} \leq M_{\text{max}}, \quad (3)$$

де  $FNR$  – частка пропущених подій;  $FPR$  – частка хибних спрацювань;  $A_{\text{traffic}}$  – коефіцієнт зростання трафіку в мережі (визначено далі в розд. 5);  $C_{\text{proc}}$  – обчислювальне навантаження на мікроконтролер у тактах на кадр;  $M_{\text{RAM}}$  – обсяг оперативної пам'яті;

- продемонструвати на прикладі синтезу параметрів реалізації шаблону для типового сценарію розгортання сенсорної мережі та оцінити ефективність запропонованого підходу.

### 4. Матеріали й методи дослідження

Дослідження ґрунтується на методах архітектурного моделювання програмних систем, формальному аналізі обчислювальної складності, теорії цифрового оброблення сигналів і аналітичному моделюванні мережевого трафіку.

Розглянемо мережу з  $N_{\text{nodes}}$  сенсорних вузлів, об'єднаних у багатовузлову транзитну *mesh*-топологію із типовою середньою кількістю стрибків між вузлом-джерелом і шлюзом. Кожен вузол неперервно дискредитує сигнал свого первинного перетворювача (радар-датчик ближньої дії, акселерометр, мікрофон тощо) із частотою дискретизації  $f_s$  та обробляє його непересіченими кадрами завдовжки  $L$  відліків. Сигнал, що надходить на вхід обробника, описується співвідношенням

$$x[n] = s[n - \tau] + w_{\text{th}}[n] + w_{\text{EMI}}[n] + w_{\text{dig}}[n], \quad (1)$$

де  $s[n]$  – корисний сигнал події (спадний синусоїдальний імпульс у смузі очікуваного сигналу);  $w_{\text{th}}[n]$  – теплові гауссівські завади;  $w_{\text{EMI}}[n]$  – структурована електромагнітна завада від мережі живлення (50/60 Гц і гармоніки);  $w_{\text{dig}}[n]$  – імпульсні цифрові завади у високочастотній смузі. Потужність шуму  $P$  є нестационарною функцією часу, що дрейфує в межах  $\pm A$  дБ із періодом  $T_{\text{cycle}}$ , відтворюючи сукупний вплив температури, погодних умов і експлуатаційних чинників:

$$P(t) = P_0 \times 10^{(A/10) \times \sin(2\pi t / T_{\text{cycle}})}. \quad (2)$$

Завдання проектування програмного забезпечення вузла полягає в синтезі такої архітектури обробника, для якої сукупна функція якості  $Q$  є мінімальною за заданих ресурсних обмежень:

$\alpha, \beta$  – вагові коефіцієнти, що позначають доменно-специфічну ціну помилок. Для застосувань периметральної безпеки добуток  $\beta \times A_{\text{traffic}}$  має той самий порядок або перевищує  $\alpha$ , оскільки кожне

хибне спрацювання поширюється в мережі та споживає енергію всіх ретрансляторів.

#### 4.1. Трирівнева архітектурна модель програмного забезпечення крайового вузла

Запропонована модель програмного забезпечення крайового вузла побудована на принципі ортогонального розшарування за рівнями відповідальності. На відміну від типового монолітного підходу, у якому весь алгоритм оброблення реалізований єдиним блоком прошивки, модель явно виокремлює три рівні: спектральне оцінювання, адаптивне відстеження шуму й прийняття рішення.

На першому рівні – спектральне оцінювання – вхідний кадр із часової області перетворюється в частотну й селекцію інформативних спектральних бінів. Дискретне перетворення Фур'є визначається класичним співвідношенням

$$X[k] = \sum_{n=0}^{L-1} x[n] e^{-j2\pi kn/L}, \quad k = 0, 1, \dots, L-1. \quad (4)$$

Для обчислювально ефективної реалізації на 32-розрядному мікроконтролері застосовується алгоритм Кулі – Тьюкі радіус-2 із асимптотикою  $O(\log(L))$ . Замість повного спектра зберігається лише вектор бінів інтересу, що відповідає очікуваній смузі сигналу події:

$$f_t = [|X[k_1]|, |X[k_2]|, \dots, |X[k_M]|]^T. \quad (5)$$

Така редукція з  $L/2$  унікальних бінів до  $M$  бінів інтересу одночасно зменшує обсяг пам'яті, потрібний для подальшого оброблення, фокусує чутливість детектора на смузі очікуваного сигналу та явно відсікає енергію всіх позасмугових шумових механізмів. У межах запропонованої архітектурної моделі цей етап реалізує перший ступінь захисту – селективну спектральну фільтрацію.

На другому рівні – адаптивне відстеження шуму – підтримується оцінка поточного фонового спектрального рівня, що еволюціонує в часі разом зі зміною зовнішніх умов. На відміну від просторових CFAR-схем, які усереднюють по сусідніх просторових комірках, запропонована модель здійснює усереднення темпоральне, у межах кожного біна окремо, оскільки припущення про просторову однорідність шуму між сенсорними вузлами в типовому розгортанні не виконується.

Архітектура цього рівня є каскадом двох фільтрів. Перший каскад має коротке вікно  $\gamma_d \in [3, 5]$  кадрів і

відсікає миттєві імпульсні викиди тривалістю меншою за  $\gamma_d$  кадрів – це другий ступінь захисту, темпоральна персистентна фільтрація. Другий каскад має довге вікно  $\gamma_a \in [64, 128]$  кадрів і відстежує повільний дрейф шумового рівня на масштабах десятків секунд або хвилин – це третій ступінь захисту, адаптивне відстеження рівня шуму. Каскадна структура є принциповою: вхід третього каскаду вже очищений другим каскадом від транз'єнтів, тому медіана або експоненціальна оцінка довгого каскаду не контамінується миттєвими піками, що виникають під час реальних подій.

На третьому рівні – прийняття рішення – формується бінарний індикатор події на основі мультиплікативного порогового правила, нормованого на поточну оцінку рівня шуму:

$$D[t] = \begin{cases} 1, & \text{якщо } |X[k][t]| > \zeta \times N[t] \\ 0, & \text{інакше,} \end{cases} \quad (6)$$

де  $\zeta \geq 1$  – мультиплікативний коефіцієнт детекторного запасу, конкретне значення якого обирається на етапі калібрування системи відповідно до співвідношення сигнал / шум очікуваної події та доменно-специфічних вимог до робочої точки на ROC-кривій. Для застосувань із низьким очікуваним SNR (порядку 3–6 дБ) типові значення  $\zeta \in [1; 2]$ , тоді як для задач периметральної безпеки з вищими SNR (15+ дБ) і критичною вимогою низької частоти хибних спрацювань калібрування дає значення  $\zeta$  у межах 5–8 (детальніше – у підрозд. 4.3). Мультиплікативна форма забезпечує сталу чутливість у дБ-шкалі, оскільки добуток  $\zeta \times N$  масштабується разом із фоновим рівнем шуму, тоді як адитивна схема потребувала б переналаштування в разі зміни рівня  $P_0$ . Для системно-рівневої події застосовується OR-агрегація по бінах:

$$E[t] = 1, \quad \text{якщо } \exists k \in K : D_k[t] = 1. \quad (7)$$

Принциповою властивістю запропонованої моделі є ортогональність трьох ступенів захисту покриття простору механізмів хибних спрацювань. Перший нейтралізує позасмугові завади (вузькосмугові 50/60 Гц і гармоніки, цифрові імпульси в смузі 0,8–2 кГц, широкосмуговий тепловий шум поза смугою події); другий ступінь нейтралізує внутрішньосмугові імпульсні транз'єнти меншої тривалості за  $\gamma_d$  кадрів; третій нейтралізує повільний дрейф фонового рівня на масштабах десятків секунд і вище.

## 4.2. Метод проєктування

### на основі шаблону триступеневого придушення

Запропонований метод формалізує розроблення програмного забезпечення крайового вузла як послідовність шести етапів, кожен з яких інстанціює відповідний компонент архітектурної моделі.

*Етап 1.* Доменна специфікація. Розробник задає очікувану смугу сигналу  $[f_{low}, f_{high}]$ , типову тривалість події  $T_{event}$ , очікуваний динамічний діапазон фонового шуму  $\Delta P$  та цільові показники якості  $(FNR_{max}, FPR_{max})$ . Перелічені параметри визначаються природою прикладної задачі й слугують вхідними обмеженнями для решти етапів.

*Етап 2.* Синтез селективної спектральної фільтрації. На основі  $f_{low}, f_{high}$  і доступної частоти дискретизації  $f_s$  обираються розмір кадру  $L$ , набір бінів інтересу  $K = \{k_1, \dots, k_M\}$  і скалярна редукція спектра. Накладаються три обмеження. Перше полягає в обмеженні довжини кадру:  $L = 2^p$  (вимога алгоритму FFT radix-2). Друге обмеження – це умова дискретизації Найквіста, що пов'язує частоту дискретизації з найвищою частотою корисного сигналу:  $f_s \geq 2 \times f_{high}$ . Третє обмеження – умова спектральної роздільної здатності: частотна роздільна здатність  $\Delta f = f_s/L$  має бути достатньо малою для подання смуги  $[f_{low}, f_{high}]$  як множини щонайменше  $M_{min}$  бінів, тобто  $\Delta f \leq (f_{high} - f_{low})/M_{min}$ . Значення  $M_{min}$  обирається доменно: типово  $M_{min} \geq 4-8$ , що забезпечує енергетичну стійкість оцінки смугової магнітуди до неточностей частотної прив'язки сигналу подій.

*Етап 3.* Синтез темпоральної персистентної фільтрації. Розмір вікна  $\gamma_d$  обирається з умови  $\gamma_d \times T_b \leq T_{event}$ , де  $T_b = L/f_s$  – тривалість кадру. Це гарантує, що персистентний фільтр пропустить корисну подію (вона тривала понад  $\gamma_d$  кадрів), але відсіче поодинокі імпульсні шумові викиди.

*Етап 4.* Синтез адаптивного відстежування рівня шуму. Розмір вікна  $\gamma_a$  (для медіанного варіанта) або коефіцієнт згладжування  $\alpha = 1 - 1/\gamma_a$  (для експоненціального) обирається з умови  $\gamma_a \times T_b \gg T_{event}$  і  $\gamma_a \times T_b \ll T_{drift}$ , де  $T_{drift}$  – характерний час зміни фонового рівня.

*Етап 5.* Параметризація рішення. Коефіцієнт  $\zeta$  обирається на базі вимірюваного у фоновому режимі відношення  $R = X/\theta$  і необхідного запасу  $SNR_{min} = 20 \log(\zeta)$ .

*Етап 6.* Інтегральна верифікація. Кожен ступінь захисту верифікується незалежно за оцінкою його доменно-специфічної властивості: перший – за коефіцієнтом придушення позасмугової енергії; другий – відповідно до здатності відсікання одиночних кадрів-викидів; третій ступінь – за здатністю відстеження повільного дрейфу.

## 4.3. Приклад синтезу параметрів для типового сценарію

Розглянемо приклад синтезу параметрів реалізації шаблону для типового сценарію периметрального моніторингу з очікуваною смугою сигналу подій [1, 5] Гц, типовою тривалістю події  $T_{event} = 5$  с і характерним часом зміни фонового рівня  $T_{drift} \approx 1$  год.

*Етап 2:* за  $f_s = 100$  Гц обираємо  $L = 128$  (radix-2).

Перевіряємо обмеження.

а) Найквіст:

$f_s = 100$  Гц  $\geq 2 \times f_{high} = 10$  Гц виконується із 10-кратним запасом.

б) Спектральна роздільна здатність:

$\Delta f = f_s/L = 0.78$  Гц, що дає  $M = 6$  бінів інтересу  $K = \{1, 2, 3, 4, 5, 6\}$ , які покривають смугу [0,78; 4,69] Гц і відповідають умові  $\Delta f \leq (f_{high} - f_{low})/M_{min}$  за  $M_{min} = 6$ . Тривалість кадру  $T_b = L/f_s = 1.28$  с.

*Етап 3:*  $\gamma_d = 3(\gamma_d \times T_b = 3.84$  с  $\leq T_{event} = 5$  с).

*Етап 4:*

$\gamma_a = 64(\gamma_a \times T_b = 81.9$  с  $\gg T_{event} = 5$  с і  $\ll T_{drift} = 3600$  с).

Для експоненціального варіанта  $\alpha = 1 - 1/64 \approx 0.984$ .

*Етап 5:* для типового сценарію периметрального моніторингу обираємо  $\zeta = 6$ , що відповідає мінімальному виявленому  $SNR_{min} = 20 \log(6) \approx 15.6$  дБ. Це значення обґрунтовано результатами апаратного калібрування у фоновому режимі: відношення  $R = \bar{X}/\theta$  під час відсутності події не перевищує одиниці й має середнє значення приблизно 0.17, що залишає достатній запас для надійного спрацювання для подій зі  $SNR \geq 18$  дБ. Гейтова умова

$R_{gate} = 0.8$  запобігає проникненню енергії події в оцінку шуму.

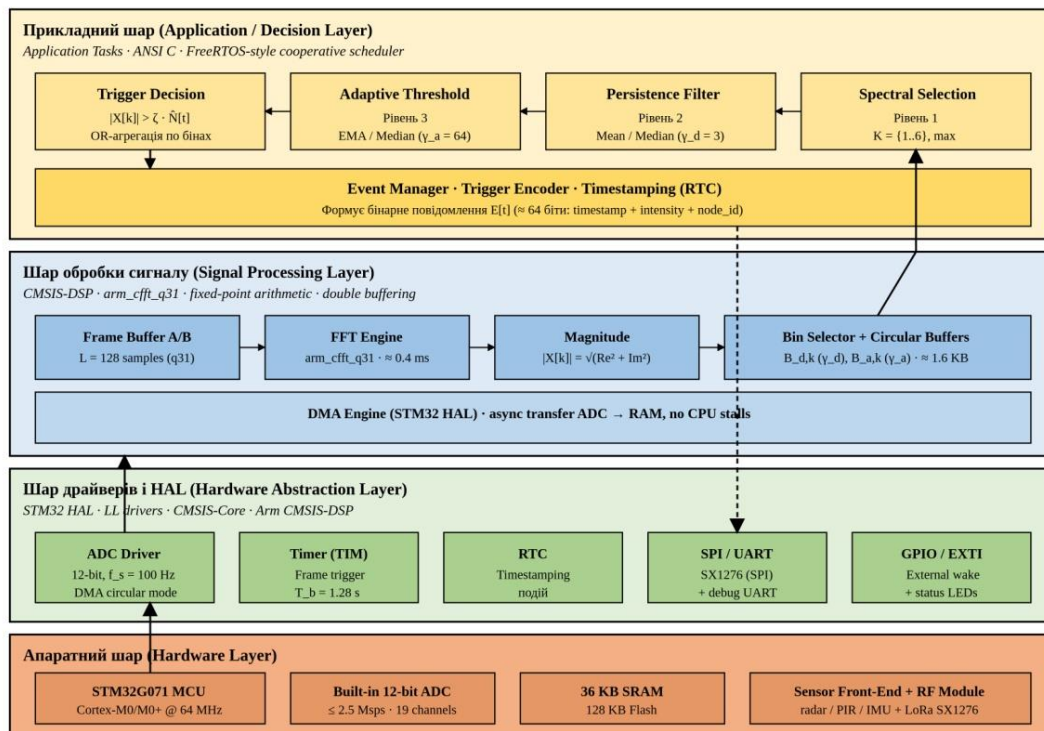
Розрахунок ресурсів для платформи STM32G071 (ARM Cortex-M0/M0+, 64 МГц): сумарний обсяг оперативної пам'яті для медіанного варіанта становить приблизно 2,7 КБ (буфери кадрів, стан медіанних фільтрів, коефіцієнти  $\zeta_k$ ), що вдвічі менший за мінімальний обсяг ОЗП такого мікроконтролера. Час оброблення одного кадру не перевищує 21,5 мс і становить менше ніж 2% періоду кадру за  $f_s = 100$  Гц. Цей приклад підтверджує реалістичність синтезованих параметрів і ефективність запропонованого методу для практичних завдань.

#### 4.4. Декомпозиція програмного забезпечення на функціональні шари

У підрозд. 4.1–4.3 запропоновано трирівневу архітектурну модель, що описує функціональну логіку детектування як ортогональну композицію

трьох взаємодоповнювальних захисних механізмів (рівень 1 – селективна спектральна фільтрація; рівень 2 – темпоральна персистентна фільтрація; рівень 3 – адаптивне відстеження рівня шуму). Натомість для практичної реалізації цієї логіки в прошивці крайового вузла необхідна окрема, програмно-інженерна декомпозиція, яка структурує код за принципами абстракції апаратного забезпечення, повторного використання й незалежного тестування. Така декомпозиція подана у вигляді чотиришарової структури й зображена на рис. 1. Принципова різниця між двома моделями полягає в тому, що трирівнева модель є логічною (визначає, що саме обробляється), тоді як чотиришарова – інженерною (з'ясує, де саме в коді реалізується це оброблення). Чотиришарова декомпозиція є типовим інженерним підходом до розроблення вбудованих систем і дає змогу розділити апаратно-залежні компоненти від платформи-незалежної прикладної логіки, що зі свого боку спрощує перенесення коду між апаратними платформами й незалежне модульне тестування.

Архітектура ПЗ крайового вузла IoT mesh-мережі  
(STM32G071, ARM Cortex-M0/M0+, ANSI C, STM32CubeIDE)



Стрілки: суцільні — синхронний потік даних; пунктирні — асинхронні події; колір блоку — шар декомпозиції.  
Червоний — апаратний; зелений — драйвери/HAL; блакитний — обробка сигналу; жовтий — прикладна логіка.  
Рівень 1, Рівень 2, Рівень 3 — три ступені придушення хибних спрацювань (див. розд. 4.1).

Рис. 1. Архітектура програмного забезпечення крайового вузла IoT mesh-мережі: декомпозиція на чотири шари (апаратний, драйверів, оброблення сигналу, прикладний) і розташування трьох ступенів захисту від хибних спрацювань

Найнижчий апаратний шар містить мікроконтролер STM32G071 із вбудованим 12-розрядним АЦП послідовного наближення (SAR), сенсорний фронт-енд і радіочастотний модуль трансивера LoRa SX1276, під'єднаний до мікроконтролера через інтерфейс SPI. Шар драйверів і апаратної абстракції інкапсулює доступ до периферії через STM32 HAL і Low-Layer (LL) драйвери, що уніфікують програмний інтерфейс ADC, DMA, таймерів, RTC, SPI (для трансивера SX1276), UART (для діагностичного інтерфейсу) та GPIO. Шар оброблення сигналу реалізує власне обчислювальний конвеєр: подвійну буферизацію кадрів  $L = 128$  відліків, швидке перетворення Фур'є з фіксованою точкою (CMSIS-DSP `arm_cfft_q31`), обчислення магнітуд бінів, селекцію бінів інтересу й підтримку циркулярних буферів для медіанних фільтрів. Прикладний шар реалізує триступеневий шаблон придушення хибних спрацювань (рівень 1, рівень 2, рівень 3), правило прийняття рішення й формування події з мінімальною метайнформацією для подальшої передачі.

Принциповим архітектурним рішенням є виконання DMA-передач між АЦП й оперативною пам'яттю в режимі циркулярного буфера без участі ядра процесора. Це дає змогу розділити етапи набору кадру й етапи обчислення FFT у часі: у процесі обчислення FFT для одного буфера його друга половина наповнюється новими відліками від АЦП. Така подвійна буферизація є необхідною умовою детермінованого виконання реального часу й за умови коректної реалізації забезпечує стабільну тривалість кадру  $T_b$  незалежно від часу обчислення.

#### 4.5. Управління потоком даних і відповідність рівням моделі OSI

Управління потоком даних на крайовому вузлі організовано як послідовність детермінованих кроків, що ініціюються апаратними перериваннями DMA. Узагальнена блок-схема потоку продемонстрована на рис. 2 в лівій частині. Після ініціалізації клокінгу, периферії та запуску АЦП у DMA-режимі ядро процесора переходить у стан очікування переривання (WFI), у якому забезпечує мінімальне енергоспоживання. Після надходження переривання DMA *half/full-complete* активне завдання оброблення виконує FFT на неактивному буфері, послідовно застосовує три ступені захисту й у разі позитивного рішення формує бінарне повідомлення для передачі

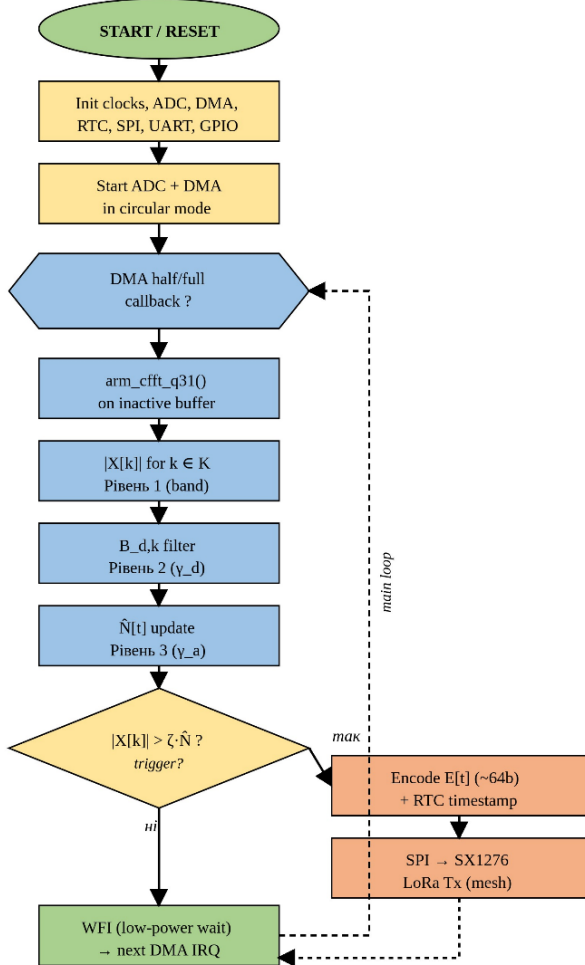
через радіотрансивер. За умови негативного рішення вузол повертається до сну, очікуючи на наступне переривання.

Запропонована архітектура природно мапиться на еталонну модель взаємодії відкритих систем (OSI), що дає змогу чітко відокремити функції різних рівнів і сприяє інтегрованим рішенням з типовими IoT-стеками. Правий блок рис. 2 присвячений такому мапінгу. Важливо зазначити, що в цій статті під "LoRa mesh" розуміємо не стандартний LoRaWAN, який за специфікацією LoRa Alliance має *star-of-stars*-топологію, а кастомна *mesh*-надбудова побудована безпосередньо поверх фізичного рівня LoRa-трансивера SX1276 з власною реалізацією нижніх рівнів стеку. На прикладному рівні (L7) функціонує менеджер подій із кодуванням повідомлення у форматі TLV або CBOR; на рівні подання (L6) виконується серіалізація та опціональне шифрування AES-128. Сесійний рівень (L5) реалізує власну стейт-машину з підрахунком кадрів і керуванням *duty cycle* з огляду на регуляторні вимоги діапазону EU868. Транспортний (L4) і мережевий (L3) рівні реалізують, відповідно, надійну доставку й власну *mesh*-маршрутизацію (наприклад, на основі RPL-подібного протоколу) між сусідніми вузлами в багатовузловій транзитній топології. Канальний рівень (L2) реалізує власний LoRa MAC із механізмами CSMA-CA / Listen-Before-Talk, а не LoRaWAN MAC. Фізичний рівень (L1) подано модуляцією LoRa Chirp Spread Spectrum на трансивері SX1276, який під'єднано до мікроконтролера через інтерфейс SPI.

Класична модель OSI описує лише комунікаційну частину стеку, тому для повноти опису IoT-вузла на рис. 2 впроваджено додатковий рівень  $L_0^*$  (*Sensing*), що охоплює сенсорний фронт-енд і вбудований АЦП мікроконтролера. Цей рівень не належить до канонічної моделі OSI, проте є фізичним фундаментом будь-якої сенсорної системи й має бути явно описаний у проєктній документації.

Важливою властивістю запропонованого мапінгу є те, що компоненти різних рівнів є слабкопов'язаними між собою через чітко визначені програмні інтерфейси (API). Це означає, що, наприклад, заміна нижніх рівнів (зокрема перехід з LoRa на NB-IoT) потребує модифікації лише компонентів  $L_1-L_2$ , тоді як вища логіка детектування залишається незмінною.

а) Управління потоком даних на вузлі



б) Відповідність рівням моделі OSI

|                    |  |
|--------------------|--|
| L7<br>Application  | Event Manager + Trigger Decision<br>JSON/CBOR-кодування подій (timestamp, intensity, node_id), формат TLV для mesh-payload   |
| L6<br>Presentation | Серіалізація + AES-128 (опц.)<br>Перетворення внутрішніх структур у байтовий потік; шифрування корисного навантаження за необхідності  |
| L5<br>Session      | Custom session / state machine<br>Frame counters, ACK/NACK handling, duty-cycle accounting, retransmission policy  |
| L4<br>Transport    | Reliability layer (опц. UDP/CoAP over LoRa)<br>Confirmed/unconfirmed delivery, fragmentation для повідомлень > MTU, FEC опційно  |
| L3<br>Network      | Custom LoRa mesh routing (RPL-like)<br>Маршрутизація через ретранслятори до шлюзу; середнє $\bar{h}$ скачків — параметр у формулі (9)  |
| L2<br>Data Link    | Custom LoRa MAC + CSMA-CA / LBT<br>CRC, framing, MIC; collision avoidance — впливає на $p_{getx}$ у формулі (9)  |
| L1<br>Physical     | LoRa CSS modulation, SX1276 transceiver<br>SF7-SF12, BW 125/250/500 kHz, EU868 band; Tx ~100 mA (домінує в енергобюджеті)  |
| L0*<br>Sensing     | Sensor + STM32 built-in 12-bit ADC + DMA<br>Радар-датчик / акселерометр / мікрофон → ADC → кадр L = 128 відліків. Поза стандартом OSI, але є фізичною основою sensing-частини стека. |

Зуваження про застосування моделі OSI:

Класична модель OSI описує стек комунікації, тоді як архітектура IoT-вузла потребує доповнення sensing-рівнем (L0\*) і прикладною логікою детектування. Запропонована модель ПЗ покриває L0\*-L7 у єдиній прошивці; роздільне виділення компонентів кожного рівня сприяє повторному використанню коду між апаратними платформами.

Лівий блок: блок-схема обробника від ADC-переривання до Tx-переривання. Правий блок: відповідність L0\*-L7 OSI.

Рис. 2. Управління потоком даних на крайовому вузлі (а) та відповідність функцій програмного забезпечення рівням еталонної моделі OSI (б)

5. Результати дослідження

5.1. Порівняльний аналіз двох реалізацій шаблону

Запропонована модель допускає дві альтернативні реалізації шару адаптивного відстеження: на основі

усереднення (далі – варіант А) й на основі медіанної фільтрації (далі – варіант Б). Обидва варіанти відповідають апаратним обмеженням мікроконтролерів класу ARM Cortex-M0/M0+, але різняться за вимогами до обчислювальних ресурсів. Зведене порівняння подано в табл. 1.

Таблиця 1. Порівняння двох реалізацій шаблону триступеневого придушення хибних спрацювань

| Характеристика                             | Варіант А (усереднення)                 | Варіант Б (медіана)                    |
|--|---|--|
| Агрегація за бінами                        | Максимум за 6 бінами → скаляр           | Побінне зберігання, логічне АБО (OR)   |
| Фільтр короточасних завад                  | Середнє за $\gamma_d = 3$ кадрами       | Медіана за $\gamma_d = 3-5$ кадрами    |
| Відстеження фонового рівня шуму            | ЕМА із гейтуванням ( $\alpha = 0,984$ ) | Медіана за $\gamma_a = 64-128$ кадрами |
| Потреба у гейтуванні оновлень              | Так ( $R_{gate} = 0,8$ )                | Ні (50% точка пробую)                  |
| Збереження спектральної форми              | Ні (редукція за максимумом)             | Так (побінне рішення)                  |
| Обсяг ОЗП, байт                            | $\approx 24$                            | $\approx 1608$                         |
| Обчислювальне навантаження                 | $\approx 10$ операцій / кадр            | $\approx 9\,600$ порівнянь / с         |
| Стійкість (робастність) до сильних викидів | Помірна                                 | Висока (до $ \gamma_{a/2} $ викидів)   |

## 5.2. Аналіз обчислювальної складності та ресурсних обмежень

Сукупне обчислювальне навантаження на крайовий вузол містить три складники: час накопичення кадру (що виконується паралельно з обробленням завдяки подвійній буферизації), час обчислення ШПФ і час фільтраційно-вирішального етапу:

$$\tau_E = \tau_{acquire} + \tau_{FFT} + \tau_{filter} + \tau_{decision} \quad (8)$$

Для типових параметрів ( $L = 128$ ,  $M = 6$  бінів,  $\gamma_d = 3$ ,  $\gamma_a = 64$ ) і платформи ARM Cortex-M0/M0+ із тактовою частотою 64 МГц оцінено обчислювальну складність кожного складника. ШПФ із фіксованою комою (функція `arm_cfft_q31` бібліотеки CMSIS-DSP) обчислюється приблизно за 0,4 мс. Обчислення модуля спектра для 6 бінів триває менш ніж 0,1 мс. Варіант А виконується за лічені мікросекунди (10 операцій додавання та зсуву). Варіант Б із частковим сортуванням до медіанної позиції потребує приблизно  $|\gamma_d/2| \times \gamma_d$  порівнянь у короткому каскаді та  $|\gamma_a/2| \times \gamma_a$  порівнянь у довгому. Для  $\gamma_a = 64$  це становить приблизно  $64 \times 32 = 2048$  порівнянь на один бін або орієнтовно 9 600 порівнянь за секунду для 6 бінів за частоти кадрів 0,78 Гц, що охоплює менш ніж 0,02% процесорного часу.

Сумарний час оброблення одного кадру не перевищує 21,5 мс. За частоти дискретизації  $f_s = 100$  Гц тривалість кадру  $T_b = 1.28$  с, тому витрати процесорного часу становлять менше ніж 2%; за  $f_s = 1$  кГц вони дорівнюють приблизно 17%, що забезпечує достатній резерв часу для виконання інших операцій вузла. Для частот дискретизації понад 10 кГц необхідна платформа Cortex-M4 з апаратним блоком FPU.

Загальний обсяг ОЗП моделі (для варіанта Б) дорівнює сумі обсягів подвійних буферів кадру ( $2L$  відліків  $\times 4$  байти = 1 024 байти), станів медіанних фільтрів  $M \times (\gamma_d + \gamma_a) \times 4$  байти = 1608 байтів) і коефіцієнтів  $\zeta_k$  (24 байти), що загалом не перевищує 2,7 КБ. Це становить менш ніж половину обсягу ОЗП мікроконтролера STM32G071 у мінімальній конфігурації (8 КБ), що забезпечує достатній резерв пам'яті для стеку, мережевих буферів і додаткових прикладних модулів.

## 5.3. Аналітична модель зростання мережевого трафіку

У багатовузлових транзитних *mesh*-мережах хибні спрацювання локального детектора мають кумулятивний ефект на сукупний трафік мережі: кожна хибна подія потребує передачі через усі стрибки маршруту до шлюзу, до того ж в умовах перевантаження каналу можуть виникати ретрансляції внаслідок *collision backoff*. Для кількісного оцінювання додаємо коефіцієнт зростання трафіку:

$$A_{traffic} = \bar{h} \times R_{FA} \times S \times \rho_{retx} \quad (9)$$

де  $\bar{h}$  – середня кількість стрибків маршруту;  $R_{FA}$  – частота хибних спрацювань вузла (подій/год);  $S$  – розмір корисного навантаження пакета (біт);  $\rho_{retx}$  – коефіцієнт ретрансляції ( $\geq 1$ , залежить від завантаженості каналу).

Запропонована архітектурна модель мінімізує  $A_{traffic}$  одразу за двома незалежними напрямками. По-перше, тричасткова композиція захистів зменшує  $R_{FA}$  на кілька порядків порівняно з однокомпонентними альтернативами. По-друге, локалізація прийняття рішення на крайовому вузлі редукує  $S$  до двійкового індикатора з мінімальними метаданими (мітка часу, оцінка інтенсивності) – приблизно 64 біти на подію, тоді як передача необробленого спектрального вектора з 16 бінів і 16-бітною роздільністю становить 256 бітів на кадр.

Розглянемо чисельний приклад: для мережі з  $h = 4$  стрибками, частотою кадрів 0,78 Гц і потенційним FPR на рівні 1% (наближення для системи, у якій відсутній шар адаптивного відстеження шуму) сукупний трафік хибних подій становив би  $R_{FA} = 0.01 \times 0.78 \times 3600 \approx 28$  хибних спрацювань за годину на вузол. У мережі з  $N_{nodes} = 200$  вузлів за 24 год це 134 400 хибних подій сумарно, що за  $S = 64$  біти й  $h = 4$  дає 34,4 МБіт корисного трафіку та принаймні стільки ж пропускної здатності, спожитих на ретрансляції. У межах прийнятої аналітичної моделі запропонована архітектура з трьома ортогональними механізмами захисту наближає  $R_{FA}$  до нуля. Кількісне підтвердження цього очікування для конкретних умов розгортання потребує окремої експериментальної валідації, що не є завданням цієї роботи.

## 6. Обговорення результатів

Запропоновані модель і метод подано на рівні шаблону проектування програмного забезпечення, що допускає повторне використання в різних апаратних і доменних інстанціях. На відміну від наявних робіт, в яких зосереджено або на алгоритмічних результатах для конкретного апаратного стенда, або на загальних *cloud-fog-edge*-абстракціях, запропонований підхід явно ідентифікує три взаємодоповнювальні захисні механізми та встановлює, що відсутність будь-якого з них породжує клас хибних спрацювань, не покритий рештою.

Описані в статті архітектурна модель і метод проектування слугують методологічною основою для розподілених систем визначення руху об'єктів, у яких крайовий вузол виявляє наявність динамічної цілі за спектральною сигнатурою її переміщення в локальному середовищі без передачі сирих сенсорних даних на хмарний рівень. Сфери застосування запропонованих результатів охоплюють такі промислові домени. Перший – системи периметральної безпеки об'єктів критичної інфраструктури (енергетичні підстанції, сховища, складські комплекси), де ціна хибних спрацювань і ціна пропусків мають співмірний порядок, а виявлення несанкційного переміщення об'єктів у захищеній зоні є основною функціональною вимогою. Другий – системи моніторингу промислового обладнання в режимі предиктивного обслуговування, де крайові вузли мають детектувати ранні ознаки відмов на тлі робочого шуму обладнання. Третій – оборонні застосування, що потребують автономної роботи без хмарної підтримки в умовах радіоелектронної протидії, зокрема виявлення руху живої сили й техніки за акустичною або сейсмічною сигнатурою. Четвертий – екологічний моніторинг (детектування браконьєрської діяльності в заповідниках, аномальних шумів у середовищі проживання охоронюваних видів), де енергетична автономність вузла є критичною.

Зв'язок із промисловістю реалізується завдяки можливості прямої інтеграції запропонованого шаблону в процес розроблення прошивки сенсорних вузлів на базі ARM-сумісних мікроконтролерів. Двофазна структура методу (синтез параметрів – інтегральна верифікація) узгоджується із промисловими процесами розроблення програмного забезпечення для систем реального часу за стандартами IEC 61508

та ISO 26262 у частині декомпозиції на верифіковані компоненти. Запропонована аналітична модель зростання трафіку дає змогу кількісно оцінювати мережеву й енергетичну вигоду локалізації прийняття рішення на крайовому вузлі ще на етапі архітектурного проектування, що знижує ризик дорогих архітектурних перероблень на пізніх стадіях виробництва.

### Обмеження запропонованого підходу

По-перше, результати, подані в статті, основані на аналітичній моделі, архітектурному обґрунтуванні та розрахунковому прикладі; повноцінна експериментальна валідація на стенді або польових даних виходить за межі цієї роботи та є предметом подальших досліджень. По-друге, модель припускає, що смугу сигналу події можна зафіксувати на етапі проектування, що не завжди справедливо для застосувань зі змінними доменними характеристиками (мобільні джерела з доплерівським зсувом). По-третє, аналітична модель зростання трафіку не бере до уваги ефекти MAC-протоколу нижнього рівня (LBT, CSMA-CA), які можуть нелінійно модифікувати  $\rho_{retx}$  у режимах насичення каналу. По-четверте, формальна верифікація властивостей реального часу шаблону потребує окремого дослідження за допомогою методів модельної перевірки (TLA+, NuSMV).

## 7. Висновки

У процесі дослідження створено тривірневу архітектурну модель програмного забезпечення крайового вузла розподіленої сенсорної мережі Інтернету речей, що ортогонально композицією поєднує селективну спектральну фільтрацію, темпоральну персистентну фільтрацію й адаптивне відстежування рівня шуму. Запропоновано метод проектування, який формалізує процес ініціалізації цієї моделі у вигляді послідовності шести етапів із чітко визначеними вхідними обмеженнями й критеріями верифікації кожного компонента.

Аналітична оцінка обчислювальної складності свідчить, що обидва варіанти реалізації шаблону відповідають обмеженням мікроконтролерів класу ARM Cortex-M0/M0+ із тактовою частотою 64 МГц у межах розрахункових 21,5 мс на кадр і 2,7 КБ оперативної пам'яті. Варіант на базі усереднення оптимальний за обсягом пам'яті (приблизно 24 байти

стану), варіант на базі медіанної фільтрації – за стійкістю до сильних викидів і збереженням спектральної форми (1,6 КБ стану). Запропонована аналітична модель зростання трафіку дає змогу кількісно оцінити мережеву й енергетичну вигоду локалізації прийняття рішення на крайовому вузлі ще на етапі архітектурного проектування.

Наукова новизна досягнутих результатів полягає в трьох позиціях. По-перше, запропоновано трирівневу архітектурну модель програмного забезпечення крайового вузла, що відрізняється явним розшаруванням компонентів виявлення, фільтрації та прийняття рішення. По-друге, створено шаблон триступеневого придушення хибних спрацювань як самостійну одиницю проектування програмного забезпечення для вузлів Інтернету речей, що допускає незалежну верифікацію кожного ступеня. По-третє, отримано аналітичну модель зростання трафіку в багатовузловій транзитній *mesh*-топології, що дає змогу кількісно зіставляти альтернативні архітектурні рішення на ранніх етапах розроблення.

Практичне значення отриманих результатів полягає в можливості системного створення програмного забезпечення для розподілених систем визначення руху об'єктів і периметральної безпеки, де ціна хибних спрацювань і ціна пропусків мають співмірний порядок. Запропоновані модель і метод можуть бути безпосередньо інтегровані в розроблення прошивки вузла Інтернету речей на базі ARM-сумісних мікроконтролерів і використані як методологічна основа подальшої верифікації та валідації за стандартами якості програмного забезпечення для систем реального часу.

Подальші дослідження автора будуть присвячені таким питанням: розширенню моделі для мобільних

вузлів-джерел (адаптивне розширення смуги бінів інтересу за оцінкою доплерівського зсуву); верифікації шаблону методами модельної перевірки на коректність часових властивостей; дослідженню *mesh*-кооперативних варіантів, у яких сусідні вузли обмінюються оцінками поточного рівня шуму для крос-валідації локальних детекторів.

---

### Конфлікт інтересів

Автори декларують, що не мають конфлікту інтересів, зокрема фінансового, особистого, авторського чи будь-якого іншого характеру, який міг би вплинути на дослідження, а також на результати, опубліковані в цій статті.

---

### Фінансування

Дослідження проводилося без фінансової підтримки.

---

### Доступність даних

Рукопис не має пов'язаних даних.

---

### Використання засобів штучного інтелекту

Автори підтверджують, що не використовували технології ШІ для написання змістових частин цієї роботи. Допоміжні інструменти штучного інтелекту застосовувалися виключно для технічних завдань форматування тексту й перевірки граматики, що було повторно перевірено авторами вручну й не вплинуло на наукові висновки дослідження.

### References

1. Pioli, L., Dorneles, C. F., de Macedo, D. D. J., Dantas, M. A. R. (2022), "An overview of data reduction solutions at the edge of IoT systems: a systematic mapping of the literature", *Computing*, Vol. 104, No. 8, pp. 1867–1889. DOI: <https://doi.org/10.1007/s00607-022-01073-6>
  2. Sadri, A. A., Rahmani, A. M., Saberikamarposhti, M., Hosseinzadeh, M. (2022), "Data reduction in fog computing and Internet of Things: a systematic literature survey", *Internet of Things*, Vol. 20, art. 100629. DOI: <https://doi.org/10.1016/j.iot.2022.100629>
  3. Zhang, H., Na, J., Zhang, B. (2023), "Autonomous Internet of Things (IoT) Data Reduction Based on Adaptive Threshold", *Sensors*, Vol. 23, No. 23, art. 9427. DOI: <https://doi.org/10.3390/s23239427>
  4. Hussein, A. M., Idrees, A. K., Couturier, R. (2022), "Distributed energy-efficient data reduction approach based on prediction and compression to reduce data transmission in IoT networks", *International Journal of Communication Systems*, Vol. 35, No. 9, e5282. DOI: <https://doi.org/10.1002/dac.5282>
-

5. Idrees, A. K., Al-Qurabat, A. K. M. (2021), "Energy-Efficient Data Transmission and Aggregation Protocol in Periodic Sensor Networks Based Fog Computing", *Journal of Network and Systems Management*, Vol. 29, No. 1, art. 4. DOI: <https://doi.org/10.1007/s10922-020-09567-4>
6. Correa, J. D. A., Pinto, A. R., Montez, C. (2022), "Lossy Data Compression for IoT Sensors: A Review", *Internet of Things*, Vol. 19, art. 100516. DOI: <https://doi.org/10.1016/j.iot.2022.100516>
7. Chen, X., Yu, Q., Dai, S., Sun, P., Tang, H., Cheng, L. (2024), "Deep Reinforcement Learning for Efficient IoT Data Compression in Smart Railroad Management", *IEEE Internet of Things Journal*, Vol. 11, No. 14, pp. 25494–25504. DOI: <https://doi.org/10.1109/JIOT.2023.3348487>
8. Hafeez, T., Xu, L., Mcardle, G. (2021), "Edge Intelligence for Data Handling and Predictive Maintenance in IIoT", *IEEE Access*, Vol. 9, pp. 49355–49371. DOI: <https://doi.org/10.1109/ACCESS.2021.3069137>
9. GabAllah, N. A., Farrag, I., Nawawy, O., Khalil, R., Sharara, H., Elbatt, T. (2021), "Towards Enabling IoT Systems with Edge Intelligence", *Proc. 2021 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 271–277. DOI: <https://doi.org/10.1109/SmartIoT52359.2021.00050>
10. Bhoi, S. K., Faseehuddin, M., Ravulakollu, K. K., Suthar, B., Sahoo, K. S., Pradhan, B. (2022), "Advanced Edge Computing Framework for Grid Power Quality Monitoring of Industrial Motor Drive Applications", *Proc. 2022 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, IEEE, pp. 455–459. DOI: <https://doi.org/10.1109/speedam53979.2022.9841966>
11. Xie, C., Tao, W., Zeng, Z., Dong, Y. (2023), "Binary-Convolution Data-Reduction Network for Edge-Cloud IIoT Anomaly Detection", *Electronics*, Vol. 12, No. 15, art. 3229. DOI: <https://doi.org/10.3390/electronics12153229>
12. Trilles, S., Hammad, S. S., Iskandaryan, D. (2024), "Anomaly detection based on Artificial Intelligence of Things: A Systematic Literature Mapping", *Internet of Things*, Vol. 25, art. 101063. DOI: <https://doi.org/10.1016/j.iot.2024.101063>
13. Cooley, J. W., Tukey, J. W. (1965), "An algorithm for the machine calculation of complex Fourier series", *Mathematics of Computation*, Vol. 19, No. 90, pp. 297–301. DOI: <https://doi.org/10.1090/S0025-5718-1965-0178586-1>
14. Wickert, M. A. (2015), "Using the ARM Cortex-M4 and the CMSIS-DSP library for teaching real-time DSP", *Proc. 2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE)*, IEEE, pp. 283–288. DOI: <https://doi.org/10.1109/DSP-SPE.2015.7369567>
15. Hammad, S. S., Iskandaryan, D., Trilles, S. (2023), "An unsupervised TinyML approach applied to the detection of urban noise anomalies under the smart cities environment", *Internet of Things*, Vol. 23, art. 100848. DOI: <https://doi.org/10.1016/j.iot.2023.100848>
16. Arciniegas, S., Rivero, D., Piñan, J., Diaz, E., Rivas, F. (2025), "IoT device for detecting abnormal vibrations in motors using TinyML", *Discover Internet of Things*, Vol. 5, art. 142. DOI: <https://doi.org/10.1007/s43926-025-00142-4>
17. Rohling, H. (1983), "Radar CFAR Thresholding in Clutter and Multiple Target Situations", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-19, No. 4, pp. 608–621. DOI: <https://doi.org/10.1109/TAES.1983.309350>
18. Finn, H. M., Johnson, R. S. (1968), "Adaptive detection mode with threshold control as a function of spatially sampled clutter level estimates", *RCA Review*, Vol. 29, No. 3, pp. 414–464.
19. Tukey, J. W. (1977), *Exploratory Data Analysis*, Addison-Wesley, Reading, MA, 688 p.
20. Gallagher, N. C., Wise, G. L. (1981), "A theoretical analysis of the properties of median filters", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 29, No. 6, pp. 1136–1141. DOI: <https://doi.org/10.1109/TASSP.1981.1163708>
21. Sittón-Candanedo, I., Corchado, J. M. (2019), "An Edge Computing Tutorial", *Oriental Journal of Computer Science and Technology*, Vol. 12, No. 2, pp. 34–38. DOI: <https://doi.org/10.13005/OJCST12.02.02>
22. Oikonomou, P., Karanika, A., Anagnostopoulos, C., Kolomvatsos, K. (2021), "On the Use of Intelligent Models towards Meeting the Challenges of the Edge Mesh", *ACM Computing Surveys*, Vol. 54, No. 6, pp. 1–42. DOI: <https://doi.org/10.1145/3456630>
23. Putra, M. A. P., Hermawan, A. P., Kim, D.-S., Lee, J.-M. (2023), "Data Prediction-Based Energy-Efficient Architecture for Industrial IoT", *IEEE Sensors Journal*, Vol. 23, No. 14, pp. 15856–15866. DOI: <https://doi.org/10.1109/JSEN.2023.3280485>
24. Wu, Y., Dai, H.-N., Wang, H., Xiong, Z., Guo, S. (2023), "To Transmit or Predict: An Efficient Industrial Data Transmission Scheme With Deep Learning and Cloud-Edge Collaboration", *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 11, pp. 11322–11332. DOI: <https://doi.org/10.1109/TII.2023.3245673>
25. Lykov, Yu. V., Oleynikov, A. M., Lykova, H. O., Savenko, S. O., Makovetskyi, S. O. (2023), "Organization of Backup Communication Channels for Emergencies in Environments with Limited Infrastructure Using LPWAN Technology", *Radioelectronics and Communications Systems*, Vol. 66, No. 6, pp. 287–304. DOI: <https://doi.org/10.3103/S0735272723090054>

Received (Надійшла) 01.04.2026

Accepted for publication (Прийнята до друку) 12.05.2026

Publication date (Дата публікації) 29.05.2026

*Відомості про авторів / About the Authors*

**Маковецький Сергій Олександрович** – Харківський національний університет радіоелектроніки, аспірант кафедри програмної інженерії, Харків, Україна;

**Serhii Makovetskyi** – Kharkiv National University of Radio Electronics, Postgraduate Student of the Department of Software Engineering, Kharkiv, Ukraine;

e-mail: serhii.makovetskyi@nure.ua

ORCID ID: <https://orcid.org/0000-0002-9814-9465>

**Каук Віктор Іванович** – кандидат технічних наук, доцент, Харківський національний університет радіоелектроніки, доцент кафедри програмної інженерії, Харків, Україна;

**Viktor Kauk** – Candidate of Technical Sciences, Associate Professor, Kharkiv National University of Radio Electronics, Associate Professor of the Department of Software Engineering Kharkiv, Ukraine;

e-mail: viktor.kauk@nure.ua

ORCID ID: <https://orcid.org/0000-0002-2780-2666>

## ARCHITECTURAL MODEL AND DESIGN METHOD OF SOFTWARE FOR EDGE NODES OF IoT MESH NETWORKS WITH THREE-STAGE FALSE-TRIGGER SUPPRESSION

**The subject** of the study is models and methods for software development of edge nodes of distributed sensor networks of the Internet of Things with mesh topology operating in non-stationary noise environments. **The aim** is to develop an architectural model and a design method for edge-node software that combine spectral event detection with adaptive noise-floor tracking and local decision-making without cloud processing. **Tasks:** to formalise a three-layer decomposition of the node software; to propose a three-stage false-trigger suppression pattern as an orthogonal composition of functional defences; to perform analysis of computational complexity and memory footprint for two alternative pattern instantiations; to derive an analytical model of traffic amplification quantifying the network benefit of localised decision-making. **Methods:** architectural modelling of software systems, formal computational complexity analysis, digital signal processing theory (Fast Fourier Transform, median and exponential filtering), analytical modelling of network traffic in multi-hop topologies. **Results.** A three-layer architectural model is formulated that orthogonally combines selective spectral filtering, temporal persistence filtering and adaptive noise-floor tracking. A six-phase design method is developed with verification criteria for each component. It is proven that both pattern instantiations satisfy the constraints of ARM Cortex-M0/M0+-class microcontrollers operating at 64 MHz within 21.5 ms per frame and 2.7 KB of RAM. The proposed analytical model of traffic amplification allows quantitative comparison of alternative architectural solutions at early development stages. **Conclusions.** The proposed model and method are presented at the level of a design pattern that allows reuse in different hardware and domain instances. The pattern identifies three complementary defence mechanisms and establishes that the absence of any of them produces a class of false triggers not covered by the rest. The results may serve as a methodological foundation for software development of distributed object-motion detection and perimeter-security systems.

**Keywords:** distributed systems; mesh networks; software architecture; design pattern; Internet of Things; object motion detection; adaptive thresholding; digital signal processing; embedded systems; microcontrollers.

*Бібліографічні описи / Bibliographic descriptions*

Маковецький С. О., Каук В. І. Метод проектування програмного забезпечення крайових вузлів інтернет-речей mesh-мереж в умовах хибних спрацювань. *Автоматизовані системи управління та прилади автоматики*. 2026. № 2 (189). С. 211–223. DOI: <https://doi.org/10.30837/0135-1710.2026.189.211>

Makovetskyi, S., Kauk, V. (2026), "Architectural model and design method of software for edge nodes of IoT mesh networks with three-stage false-trigger suppression", *Management Information System and Devices*, No. 2 (189), P. 211–223. DOI: <https://doi.org/10.30837/0135-1710.2026.189.211>