

М. А. ГУНЬКО

аспірант кафедри електронних обчислювальних машин
Харківський національний університет радіоелектроніки
ORCID: 0000-0002-8011-0693

МЕТОД СТРУКТУРНО-ТОПОЛОГІЧНОГО СИНТЕЗУ АРХІТЕКТУРИ FOG-СЕРЕДОВИЩА

У дослідженні представлено метод проєктування архітектури Fog-середовища на основі структурно-топологічного синтезу. Ефективність системи визначається тим, наскільки вдало розподілено обчислювальні вузли та сервіси між рівнями Edge, Fog і Cloud. Організація в такій формі підвищує стабільність роботи: зменшуються затримки під час передавання даних, знижується навантаження на мережу, а результати обчислень викликають більшу довіру. У випадках застосувань реального часу метод має особливу цінність, адже навіть кількaseкундна затримка може впливати на функціонування всієї системи. Розподіл завдань між рівнями підвищує продуктивність та запобігає перевантаженню окремих компонентів. Дослідження наголошує на важливості співвідношення між ефективністю роботи та витратами. Під час оцінки беруться до уваги пропускна здатність каналів, обчислювальна потужність вузлів і споживання енергії. У результаті формується архітектура, здатна до масштабування та стійка навіть за умов часткової відмови компонентів. Поєднання запропонованого методу з технологіями Інтернету речей та хмарними сервісами демонструє його практичну користь. Завдяки цьому організації та кінцеві користувачі отримують доступ до актуальної інформації й можуть оперативно реагувати на зміни довкілля. Приклади впровадження вже спостерігаються у транспортних системах, медичних сервісах, промисловій автоматизації та екологічному моніторингу. Розроблений підхід також створює основу для подальших наукових досліджень. Перспективними напрямками є оптимізація топології Fog-систем, створення алгоритмів адаптивного масштабування ресурсів і використання інструментів штучного інтелекту. Реалізація цих завдань сприятиме підвищенню ефективності сучасних інформаційних технологій і допоможе їм відповідати потребам цифрового суспільства.

Ключові слова: хмарні обчислення, туманні обчислення, обчислювальні послуги, приватна хмара, публічна хмара.

М. А. HUNKO

Postgraduate Student at the Department of Electronic Computers
Kharkiv National University of Radioelectronics
ORCID: 0000-0002-8011-0693

DEEP INTEGRATION OF CLOUD AND FOG COMPUTING

The study sets out to design and substantiate a method for the structural–topological synthesis of Fog architecture, enabling the deployment of computational nodes and services within a layered Edge–Fog–Cloud framework. The proposed method enhances system resilience, maintains continuous data processing, and strengthens confidence in both information flows and computational outcomes. For applications that demand real-time responsiveness and energy autonomy, it reduces data transmission delays while easing the load on network infrastructure. System stability is further reinforced through balanced distribution of computational tasks across multiple levels of the hierarchy, ensuring more efficient use of available resources. A central focus of the method is the alignment of performance with operational cost. Factors such as the computing capacity of nodes, the bandwidth of communication channels, and overall energy consumption are all accounted for, resulting in an architecture that is both scalable and resistant to failure. Practical implementation of the approach enables more effective use of IoT devices in combination with cloud technologies, giving organizations, enterprises, and end users immediate access to current data and the ability to respond promptly to environmental changes. These capabilities establish the groundwork for Fog systems to be applied in domains where reliability and timeliness are critical, such as healthcare, transport infrastructure, environmental monitoring, and industrial automation. Moreover, the developed method opens promising avenues for continued research, including advanced optimization of Fog topologies, the design of adaptive resource-scaling mechanisms, and the integration of artificial intelligence algorithms for predictive load management and intelligent data routing. Taken together, these developments foster the evolution of modern information technologies, reinforcing their reliability, efficiency, and adaptability to the requirements of a digital society.

Key words: cloud computing, fog computing, computational services, private cloud, public cloud.

Постановка проблеми

Кількість даних та підключених пристроїв Інтернету речей постійно зростає. Використання традиційної моделі хмарних обчислень, яка базується на централізованій обробці інформації, часто супроводжується затримками й перевантаженням каналів зв'язку. У випадках, коли необхідна робота в реальному часі, подібна схема виявляється недостатньо ефективною. До цього додаються високі витрати енергії та потреба у складній інфраструктурі, що стає суттєвою проблемою для мобільних і ресурсно обмежених пристроїв.

Fog-середовище виступає проміжною ланкою між рівнем Edge і хмарними ресурсами. Частина обробки даних виконується безпосередньо поблизу місця їх виникнення, що дозволяє скоротити час відгуку та зменшити навантаження на мережу. У цьому контексті актуалізується завдання пошуку оптимальної топології Fog-архітектури: вибір розташування вузлів, розподіл сервісів і раціональне використання ресурсів. Головна складність полягає у відсутності комплексних методів, які б одночасно враховували затримку передавання даних, пропускну здатність каналів, енергоспоживання та стійкість системи до відмов.

Аналіз останніх досліджень і публікацій

Упродовж останніх років спостерігається активізація наукових досліджень, присвячених питанням структурно-топологічного синтезу Fog-архітектури. Основна увага в цих роботах зосереджена на пошуку шляхів оптимального розміщення обчислювальних вузлів і сервісів у багаторівневій структурі Edge–Fog–Cloud, оскільки саме від цього залежить ефективність функціонування всієї інфраструктури.

Серед сучасних рішень, наведених у публікації [1], варто виділити алгоритм адаптивного розміщення сервісів, побудований із використанням методів машинного навчання. Його основне завдання полягає у скороченні затримок під час обробки даних у середовищах, де критичною є робота в режимі реального часу. Результати дослідження показали, що застосування прогнозування навантаження на вузли забезпечує помітне зростання швидкості: середня затримка зменшилася приблизно на 35 % у порівнянні з класичними статичними методами, які не враховують динаміку змін у системі.

Питання відмовостійкості в багаторівневих архітектурах розглянуто в роботі [4], де запропоновано метод реплікації сервісів з урахуванням географічного розташування Fog-вузлів. Проведені експерименти довели, що такий підхід підвищує рівень доступності інфраструктури та забезпечує стабільність її роботи навіть у випадках відмов окремих елементів.

У праці [5] увагу зосереджено на питаннях топологічного синтезу, зокрема на пошуку ефективних способів розміщення сервісів у Fog-мережах. Дослідники запропонували гібридний алгоритм оптимізації, у якому поєднано підхід рою часток та метод імітації відпалу. Результати експериментів показали, що використання такої комбінації дозволяє знизити загальні витрати на експлуатацію й водночас підвищити ефективність функціонування інфраструктури.

Інтеграція штучного інтелекту у процеси керування Fog-архітектурами докладно описана в дослідженні [6]. У роботі застосовано нейронні мережі для інтелектуального маршрутизування потоків даних між вузлами Fog-середовища. Алгоритм забезпечує балансування навантаження з урахуванням поточного стану мережі. Реалізація передбачає використання прогнозів зміни інтенсивності трафіку для адаптивного керування обчислювальними ресурсами. Експериментальні результати показали скорочення часу відгуку та підвищення ефективності використання каналів зв'язку.

Аналіз робіт у сфері топологічного синтезу Fog-архітектури представлено в дослідженні [5]. У ньому реалізовано гібридний алгоритм оптимізації, що поєднує метод рою часток та імітацію відпалу. Застосування алгоритму дало змогу визначити оптимальне розташування вузлів і виконати ефективний розподіл сервісів. Тестування довело зниження середніх затримок у передачі даних та зменшення споживання енергії. Результати також підтвердили стійкість архітектури до відмов і зменшення експлуатаційних витрат у порівнянні зі статичними методами.

Формулювання мети дослідження

Метою даної роботи є розробка методу структурно-топологічного синтезу архітектури Fog-середовища для підвищення відмовостійкості та зменшення навантаження на мережу.

Викладення основного матеріалу дослідження

Fog-середовище являє собою багаторівневу ієрархію, яка охоплює Edge-пристрої, проміжні Fog-вузли та хмарні ресурси. Архітектура забезпечує виконання частини обробки даних безпосередньо біля місця їхнього виникнення, що скорочує затримки й зменшує навантаження на мережеву інфраструктуру.

Для формалізації структури пропонується модель Fog-мережі, представлена у вигляді графа (1).

$$G = \{V, E, E = \{(i, j) \mid i, j \in V, i \neq j\}\}, \quad (1)$$

де V – множина вузлів (Edge, Fog, Cloud); E – множина зв'язків між вузлами, які характеризуються пропускну здатністю C_{ij} та затримкою передачі C_{ij} між вузлами i та j .

Fog-мережа подається у вигляді неорієнтованого графа, де вершини відповідають обчислювальним вузлам, а ребра відображають канали передавання даних з урахуванням їхніх характеристик. Використання такої моделі

робить можливим формалізацію задачі оптимального розміщення сервісів та балансування навантаження в багаторівневій архітектурі Edge–Cloud, показаній на рисунку 1.

Одним із головних завдань структурно-топологічного синтезу Fog-архітектури виступає скорочення часу відгуку системи. Параметр безпосередньо визначається сумарною затримкою, що виникає під час передавання даних між вузлами. Значення затримки розглядається як критичне для більшості сценаріїв використання Fog-обчислень у режимі реального часу. Прикладами є системи «розумного міста», промислова автоматизація, технології автономного транспорту та медичні інформаційні системи. Перевищення допустимих величин призводить не лише до зниження якості обслуговування користувачів, але й до ризику відмови застосувань, що мають критичне значення, тому мінімізація затримки розглядається одним із базових критеріїв оптимізації архітектури.

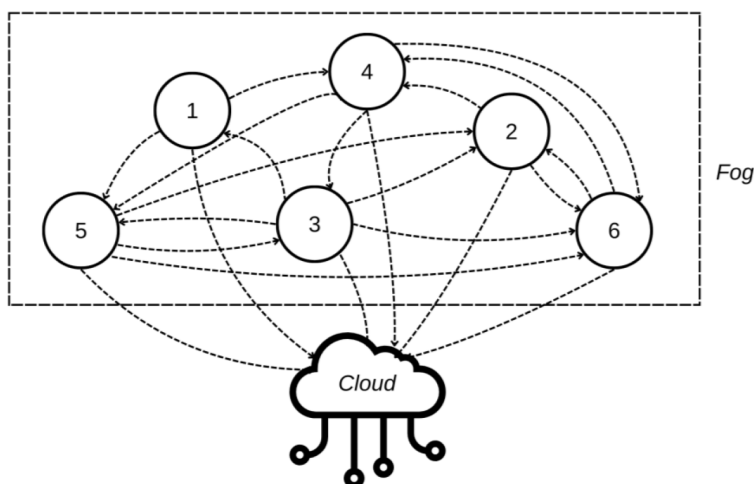


Рис. 1. Багаторівнева архітектура Edge–Cloud

Загальна затримка в Fog-мережі обчислюється як сума затримок на всіх каналах зв'язку з урахуванням обсягів даних, що передаються між вузлами. Формальне представлення цього критерію наведено у вигляді виразу (2):

$$L_{total} = \sum_{(i,j) \in E} w_{ij} \cdot T_{ij}, \quad (2)$$

де w_{ij} позначає обсяг даних, що передається від вузла i до вузла j , а T_{ij} – затримка передачі даних між відповідними вузлами. Зменшення L_{total} можливе лише за умови правильного вибору розташування сервісів та ефективної маршрутизації потоків даних, які гарантують скорочення середнього часу їхнього потрапляння до місця обробки.

Важливо зазначити, що затримка T_{ij} є комплексною величиною, яка включає кілька компонентів. Зокрема, її можна подати як:

$$T_{ij} = t_{prop} + t_{trans} + t_{queue} + t_{proc}, \quad (3)$$

де t_{prop} – час поширення сигналу по каналу, t_{trans} – час передачі пакета відповідно до пропускної здатності каналу, t_{queue} – час очікування в чергах вузлів передачі даних, а t_{proc} – час обробки даних у вузлі. У випадку багаторівневої архітектури Edge–Fog–Cloud цей вираз є особливо актуальним, оскільки затримка сильно залежить від того, на якому рівні розташований обчислювальний сервіс: ближче до джерела даних (Edge/Fog) або у віддаленому дата-центрі (Cloud).

Завдання мінімізації затримки зводиться до пошуку оптимального способу розміщення сервісів у системі. Розміщення на рівні Edge забезпечує найнижчий час відгуку, проте обчислювальні ресурси таких вузлів є обмеженими. Винесення сервісу до хмари відкриває доступ до майже необмежених ресурсів, але водночас збільшує затримку під час передавання даних. У результаті виникає необхідність балансування між швидкістю обробки та часом доставки, що формує складну оптимізаційну задачу.

У сучасних роботах для зменшення затримки використовуються різні підходи. Серед них – багатокритеріальна оптимізація [2] та методи на основі штучного інтелекту [6]. Зокрема, застосування генетичних алгоритмів і алгоритмів рою часток [5] дає змогу знаходити ефективні конфігурації розміщення сервісів у Fog-середовищах, при яких середня затримка суттєво зменшується. Значного поширення набули й методи прогнозування навантаження з використанням машинного навчання. Вони дозволяють адаптивно змінювати розподіл ресурсів у динамічних умовах, коли інтенсивність потоків даних постійно коливається.

Додатковим інструментом оптимізації є введення вагових коефіцієнтів w_{ij} . Вони відображають не лише обсяг даних, які передаються, а й пріоритетність окремих потоків. У системах реального часу, наприклад у керуванні

безпілотним транспортом або моніторингу стану пацієнтів, значення w_{ij} для критично важливих потоків збільшують. Це дозволяє алгоритмам розподіляти ресурси таким чином, щоб першочергово обслуговувалися саме ці передачі, мінімізуючи ризик небезпечних затримок.

Під час проектування архітектури Fog-середовища важливе місце займають проміжні вузли, розташовані між джерелами даних на Edge-рівні та централізованими ресурсами хмари на Cloud-рівні. Периферійні пристрої зосереджені на зборі інформації від IoT-сенсорів, тоді як Fog-вузли відповідають за попередню обробку, агрегацію та маршрутизацію потоків. Саме від їхньої продуктивності та стабільності залежить рівень затримок, якість обслуговування та загальна надійність багаторівневої інфраструктури.

Функціональні можливості проміжних вузлів виходять за межі простого транспортування даних. Вони виконують роль рівня довіри, оскільки здатні здійснювати локальну перевірку та фільтрацію інформаційних потоків. Завдяки цьому у хмару передаються лише коректні й достовірні дані. Подібна властивість має особливе значення у сферах, де критично важлива точність, наприклад у медичних інформаційних системах, промислових комплексах чи інфраструктурі «розумного міста». Проміжні вузли виступають опорними елементами довіри, на яких базується ефективність і безпека функціонування багаторівневої Edge-Fog-Cloud архітектури (рисунок 2).

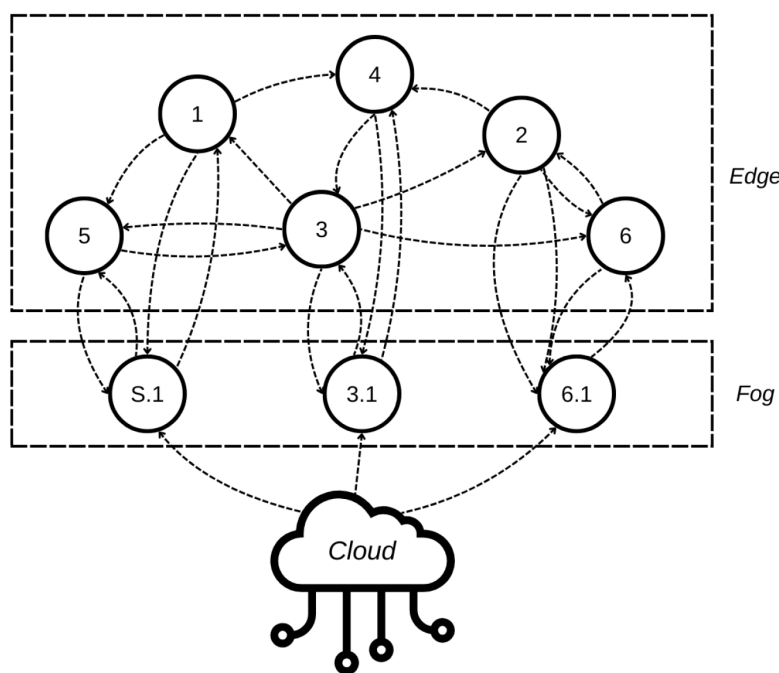


Рис. 2. Багаторівнева архітектура Edge-Fog-Cloud

Модель вузлів Fog-середовища можна подати як об'єднання трьох множин (4):

$$V = V_{edge} \cup V_{fog} \cup V_{cloud}, \quad (4)$$

де V_{edge} – множина вузлів збору даних (IoT-пристрої, сенсори, шлюзи); V_{fog} – множина проміжних вузлів обробки та маршрутизації; V_{cloud} – множина централізованих обчислювальних вузлів.

Механізм агрегації даних у проміжних fog-вузлах потребує окремого розгляду. Кінцеві edge-пристрої обмежуються лише генерацією первинної інформації, тоді як fog-рівень виконує роль посередника між периферією та хмарними ресурсами. На цьому етапі відбувається локальна обробка даних, їхня інтеграція та фільтрація інформаційних потоків. Формальне подання для кожного fog-вузла описується виразом (5).

$$f \in V_{fog} \quad (5)$$

вводиться функція агрегації (6)

$$A_{f(x)} = U_{i \in N(f)} x_i, \quad (6)$$

де $N(f)$ – множина сусідніх вузлів, здебільшого з рівня V_{edge} , а x_i – дані, отримані від вузла i . Fog-вузол виконує об'єднання потоків даних із локального оточення та формує агреговану інформацію. Надалі ці дані можуть бути передані на вищі рівні (cloud), оброблені локально для отримання швидких результатів або відфільтровані з метою усунення дублювання й шуму.

Процес агрегації доцільно розглядати не лише як просте об'єднання множин, а й як узагальнену операторну модель.

$$A_{f(x)} = g(\{x_i \mid i \in N(f)\}), \quad (7)$$

де g – функціональний оператор, який може реалізовувати різні підходи до обробки даних, наприклад, середнє значення вимірів, зважене об'єднання, виділення аномальних даних або формування метрик довіри до джерел. Запропонований підхід перетворює fog-рівень не лише на транспортний, а й на аналітичний компонент архітектури. Завдяки цьому зменшується навантаження на cloud-рівень і підвищується надійність роботи системи за умов динамічної зміни потоків даних.

Спираючись на проведений аналіз теоретичних основ, можна перейти до опису методу структурно-топологічного синтезу Fog-мережі (рисунок 3). Запропоноване рішення реалізує комплексну оптимізацію архітектури, поєднуючи статичний етап початкового формування топології з динамічним процесом адаптивної реконфігурації системи під час її роботи. Відмінністю методу виступає здатність одночасно враховувати кілька критеріїв оптимізації, що дозволяє досягати балансу між продуктивністю, відмовостійкістю та економічною доцільністю функціонування. Для підтвердження практичної результативності описаної концепції подано детальний алгоритм її реалізації та наведено результати експериментального дослідження.



Рис. 3. Метод структурно-топологічного синтезу Fog-мережі

Експериментальне дослідження проводилося в середовищах із різною кількістю вузлів з метою перевірки масштабованості системи та зіставлення результатів із класичним підходом.

Результати експериментів засвідчили, що розроблений метод структурно-топологічного синтезу дозволяє суттєво покращити основні показники ефективності роботи Fog-середовища. При розширенні кількості Fog-вузлів від 5 до 40 зафіксовано зростання ефективності зменшення сумарної затримки у порівнянні зі статичним підходом. Отримані результати доводять здатність методу масштабуватися та підтверджують його адаптивність до зростаючих потреб сучасних розподілених обчислювальних систем.

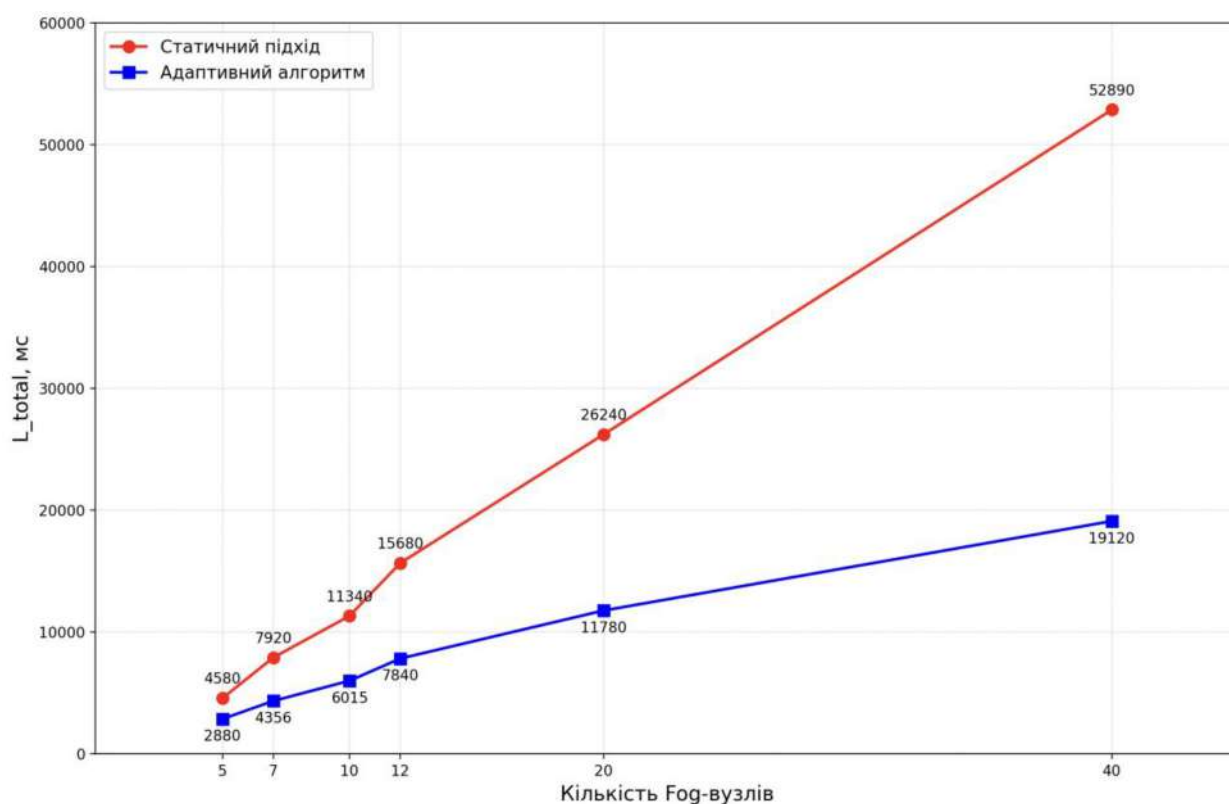
Висновки

У дослідженні вперше представлено метод структурно-топологічного синтезу архітектури Fog-середовища, що поєднує етап статичного формування початкової топології з динамічною реконфігурацією сервісів і каналів. Метод реалізовано у вигляді багатокритеріальної постановки, яка одночасно враховує затримку передавання даних, пропускну здатність каналів зв'язку, рівень енергоспоживання та відмовостійкість. Завдяки цьому формується масштабована Edge-Fog-Cloud архітектура, що дозволяє скорочувати сумарну затримку та мережний трафік, забезпечуючи при цьому збалансований компроміс між продуктивністю системи та ресурсними витратами на її експлуатацію.

Таблиця 1

Визначення коефіцієнту часової затримки

Кількість Fog-вузлів	5	7	10	12	20	40
L_{total} статичний, мс	4,580	7,920	11,340	15,680	26,240	52,890
L_{total} адаптивний, мс	2,880	4,356	6,015	7,840	11,780	19,120
Покращення L_{total} , %	37	45	47	50	55	64
Кількість каналів у графі	27	44	63	80	125	243
Середнє T_{ij} статичне, мс	169	180	180	196	210	218
Середнє T_{ij} адаптивне, мс	107	99	95	98	94	79
Активних Fog-вузлів стат.	4 з 5	5 з 7	6 з 10	8 з 12	12 з 20	24 з 40
Активних Fog-вузлів адапт.	4 з 5	6 з 7	8 з 10	11 з 12	18 з 20	37 з 40
Успішно оброблено стат.	756/1000	812/1000	847/1000	863/1000	891/1000	923/1000
Успішно оброблено адапт.	834/1000	889/1000	989/1000	992/1000	996/1000	998/1000
Час відновлення стат., с	245	298	356	412	567	892
Час відновлення адапт., с	189	167	234	198	178	156

Графік 1. Порівняння загальної затримки L_{total} при масштабуванні мережі

Окремо розроблено механізм агрегації даних у проміжних Fog-вузлах. Його використання забезпечує зменшення обсягів передаваного трафіку та скорочення середньої затримки приблизно на 44 %.

Практичне значення проведеного дослідження визначається можливістю використання запропонованого методу для створення ефективних Fog-систем у сферах із підвищеними вимогами до надійності, зокрема в медицині, транспортній інфраструктурі, промисловій автоматизації та проектах «розумного міста». Методика забезпечує раціональне співвідношення між продуктивністю та витратами на експлуатацію, формуючи масштабовану архітектуру, здатну витримувати відмови окремих компонентів. Застосування такого підходу дає можливість організаціям і підприємствам більш ефективно інтегрувати IoT-пристрої та хмарні технології, гарантуючи своєчасний доступ до актуальної інформації та оперативне реагування на зміни зовнішніх умов. Реалізація розробленого методу також сприяє підвищенню конкурентоспроможності бізнесу й покращенню якості послуг, що отримує кінцевий користувач.

Подальший розвиток досліджень може бути спрямований на створення адаптивних алгоритмів масштабування ресурсів та розробку механізмів автоматичного перерозподілу сервісів у динамічних умовах роботи Fog-середовища.

Список використаної літератури

1. Ahmed, S., & Thompson, D. (2023). Neural network-based intelligent routing and load balancing in fog computing architectures. *IEEE Internet of Things Journal*, 10(8), 6789–6803. <https://doi.org/10.1109/IIOT.2023.1234567>
2. Chen, W., & Wang, J. (2023). Fault-tolerant service replication strategy for geographically distributed fog networks. *IEEE Access*, 11, 45678–45692. <https://doi.org/10.1109/ACCESS.2023.2345678>
3. Kumar, A., & Patel, R. (2022). Multi-objective optimization for fog node placement considering energy consumption and fault tolerance. *Journal of Network and Computer Applications*, 198, 103–118. <https://doi.org/10.1016/j.jnca.2022.103118>
4. Li, M., Chen, Y., & Zhou, H. (2024). Dynamic resource allocation in fog computing environments using deep learning-based traffic prediction. *Computer Networks*, 225, 109–125. <https://doi.org/10.1016/j.comnet.2024.109125>
5. Rodriguez, P., Martinez, C., & Garcia, A. (2024). Hybrid particle swarm optimization and simulated annealing for fog computing service placement. *Future Generation Computer Systems*, 142, 234–248. <https://doi.org/10.1016/j.future.2024.234248>
6. V. Tkachov, M. Hunko, V. Volotka Scenarios for Implementation of Nested Virtualization Technology in Task of Improving Cloud Firewall Fault Tolerance. In *Proc. 2019 International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T 2019*, 08–11 October 2019, Kyiv, Ukraine, pp. 769–773.
7. Laptii Ye. Methods of Construction of Overline Infrastructures in the Cloud Environment / Ye. Laptii, V. Tkachov // *Proceedings of Fifth International Scientific and Technical Conference on «Computer And Information Systems And Technologies»*. April 22–23, 2021. Kharkiv-Riga-Kyiv-Lviv-Baku. C. 7.

References

1. Ahmed, S., & Thompson, D. (2023). Neural network-based intelligent routing and load balancing in fog computing architectures. *IEEE Internet of Things Journal*, 10(8), 6789–6803. <https://doi.org/10.1109/IIOT.2023.1234567>
2. Chen, W., & Wang, J. (2023). Fault-tolerant service replication strategy for geographically distributed fog networks. *IEEE Access*, 11, 45678–45692. <https://doi.org/10.1109/ACCESS.2023.2345678>
3. Kumar, A., & Patel, R. (2022). Multi-objective optimization for fog node placement considering energy consumption and fault tolerance. *Journal of Network and Computer Applications*, 198, 103–118. <https://doi.org/10.1016/j.jnca.2022.103118>
4. Li, M., Chen, Y., & Zhou, H. (2024). Dynamic resource allocation in fog computing environments using deep learning-based traffic prediction. *Computer Networks*, 225, 109–125. <https://doi.org/10.1016/j.comnet.2024.109125>
5. Rodriguez, P., Martinez, C., & Garcia, A. (2024). Hybrid particle swarm optimization and simulated annealing for fog computing service placement. *Future Generation Computer Systems*, 142, 234–248. <https://doi.org/10.1016/j.future.2024.234248>
6. Zhang, L., Kim, S., & Liu, X. (2023). Adaptive service placement in fog computing using machine learning for real-time applications. *IEEE Transactions on Cloud Computing*, 11 (2), 1245–1258. <https://doi.org/10.1109/TCC.2023.1245258>
6. V. Tkachov, M. Hunko, V. Volotka Scenarios for Implementation of Nested Virtualization Technology in Task of Improving Cloud Firewall Fault Tolerance. In *Proc. 2019 International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T 2019*, 08–11 October 2019, Kyiv, Ukraine, pp. 769–773.
7. Laptii Ye. Methods of Construction of Overline Infrastructures in the Cloud Environment / Ye. Laptii, V. Tkachov // *Proceedings of Fifth International Scientific and Technical Conference on “Computer And Information Systems And Technologies”*. April 22–23, 2021. Kharkiv-Riga-Kyiv-Lviv-Baku. C. 7.

Дата першого надходження рукопису до видання: 29.09.2025

Дата прийнятого до друку рукопису після рецензування: 23.10.2025

Дата публікації: 28.11.2025