

**МЕТОД УГРУПУВАННЯ ТЕМПОРАЛЬНИХ МУЛЬТИМОДАЛЬНИХ  
ПОТОКОВИХ ДАНИХ ПРИСТРОЇВ ІОТ  
НА ОСНОВІ РЕАКТИВНОГО ПРОГРАМУВАННЯ**

*Анотація. Стаття присвячена розробленню методу угруповання темпоральних потоків даних в мережах пристроїв Інтернету речей (ІоТ). Актуальність пояснюється необхідністю ефективної обробки даних пристроїв ІоТ у реальному часі. Мета полягає в розробці методу, що враховує складність та специфіку поточкових даних ІоТ. Запропоновано використовувати реактивне програмування для угруповання даних. У результаті показано можливість побудови ієрархії угруповання з урахуванням особливостей пристроїв ІоТ та масштабованість запропонованого методу.*

*Ключові слова: угруповання, ІоТ, поточкові дані, реактивне програмування, мультимодальні дані.*

**Постановка проблеми.** Мережі пристроїв Інтернету речей (ІоТ) стають все більш розповсюджені у теперішній час. Мережі можуть складатися з різноманітних пристроїв, що збирають дані про погоду, трафік, знімають відео або звук тощо. Ці дані мають різні формати, включаючи тексти, зображення, відео та числові значення, та збираються з різних джерел у реальному часі.

Для оброблення даних і представлення результатів користувачу, для деяких задач дані необхідно попередньо згрупувати за ознакою часу їх генерації. Різноманіття пристроїв та можливих даних ускладнює процес угруповання. Також ускладнюють задачу можливі розбіжності у часі, спричиненні технічними особливостями пристроїв та мереж, що передають відповідні дані.

Таким чином, існує необхідність у розробці ефективних методів угруповання темпоральних мультимодальних поточкових даних пристроїв ІоТ, які б забезпечували швидку та точну обробку цих даних в реальному часі, враховуючи їхню складність та специфіку.

**Аналіз останніх досліджень і публікацій.** Для угруповання даних їх необхідно визначити за певним параметром як наявні до однієї групи. У випадку

угруповання темпоральних даних критерієм є саме час, відповідно постає задача синхронізації даних.

Задача синхронізації поточкових даних є відомою та має значну кількість способів вирішення. До створення мереж IoT задача синхронізації існувала і для інших даних. Широкого розповсюдження набули методи синхронізації мультимедійних даних.

Серед основних проблем синхронізації наявні наступні: затримки мережі, мережеві флуктуації, флуктуації на кінцевих системах, розбіжність годинників, дрейф годинників, дрейф швидкості, розбіжність мережі, розбіжність відображення [1]. Відповідно існують способи синхронізації часу за допомогою NTP або GPS. Також існують алгоритмічні способи вирішення проблеми синхронізації [1] та способи компенсації затримок при відображенні мультимедійних даних [2]. Відповідні способи синхронізації існують і для мереж пристроїв [3].

Обробка даних IoT розподіляється на дві категорії: пакетну та подійну обробку. Пакетна обробка ефективна при роботі з великим обсягом даних, які збираються протягом певного періоду часу, у той час як подійна обробка виконує аналіз даних без їх зберігання та прийняття рішень для кожного окремого запису системи [4].

Для реалізації вимог до обробки даних в реальному часі для IoT, нещодавно були запропоновані рішення на основі концепції архітектури, орієнтованої на події (EDA), що призвело до розвитку нової дослідницької області, відомої як обробка складних подій (CEP) [4]. CEP включає обробку, аналіз та кореляцію потоків подій з різних джерел даних за допомогою розподілених систем на основі повідомлень з метою вилучення високорівневих знань майже в реальному часі.

Для пакетної обробки існують методи синхронізації по подіям [5], проте вони орієнтовані на визначення однієї події, такої як сплеск, стрибок чи однаковий звук на багатьох датчиках одночасно. Через це синхронізація по подіям в загальному випадку не може бути застосована до різноманітних мереж IoT, а лише до деяких з них.

**Мета дослідження.** Метою дослідження є розроблення методу угруповання темпоральних поточкових даних пристроїв IoT, що дозволив би врахувати при угрупованні такі особливості, як можливі мережеві затримки при доставці даних та різницю синхронізації годинників у різних пристроях.

**Викладення основного матеріалу дослідження.** Різні системи IoT побудовані на різноманітних протоколах, і створення єдиної та глобальної сис-

теми для мережі IoT, яка може працювати разом, є складним завданням [4]. У цьому контексті рішення для роботи з даними IoT повинно відповідати декільком вимогам.

Запропоноване рішення повинне бути достатньо загальним, щоб зчитати дані, що надходять з різних систем IoT у необхідних форматах. Воно повинно мати здатність обробляти гетерогенні потоки даних з мінімальними модифікаціями наявних компонентів. Воно повинно мати можливість виконувати аналітику в реальному часі для застосунків, які потребують оперативності. Нарешті, воно повинно враховувати можливі часові відмінності, породжені особливостями роботи розподілених мереж пристроїв.

Обробка потоків зазвичай використовується для роботи з великим обсягом даних, де навантаження має бути розподілене між кількома ядрами процесора або навіть кількома обчислювальними вузлами. Тому підхід, який виконує агрегації на цих потоках, повинен забезпечувати масштабованість та надійність, щоб не стати головною проблемою методу в цілому [6]. Також метод має забезпечувати ієрархічну побудову, яка б репрезентувала фізичну або логічну структуру (пристрої можуть бути розділені за типами або за географічним розташуванням).

Вирішення проблеми синхронізації часу може бути здійснено за допомогою протоколів синхронізації часу в мережі (наприклад, використовуючи протокол мережевого часу (NTP) або глобальної системи позиціонування (GPS) [1]. Проте, для багатьох пристроїв IoT дане рішення може не підійти через неможливість встановлення часу на деяких пристроях, зависоку ціну приймача GPS, порівняно з ціною самого пристрою, зламаній приймач або відсутність сигналу на деяких територіях (приймач закритий будівлею, сам сигнал подавлений або ж підмінений). Відповідно встановлення часу має відбуватися на пристроях також і іншими способами, які дають меншу точність синхронізації, що вимагає можливості врахування потенційних розбіжностей в часі, спричинених налаштуваннями годинників пристроїв. Також має бути врахована затримка при передачі даних від пристроїв до серверу (наприклад, затримка при використанні супутникового терміналу може складати декілька секунд).

Реалізувати дані вимоги запропоновано за допомогою реактивного програмування, що пристосоване до роботи з потоками даних та може бути використано для їх об'єднання [7]. Також даний спосіб підходить і для роботи з потоками у реальному часі. Прикладом способу з реактивного програмування є GroupJoin [8], що дозволяє об'єднувати потоки від двох джерел для пошуку груп

між ними. Також GroupJoin має можливість налаштування очікування для даних з двох джерел, що дозволяє врахувати різницю у часі надходження від пристроїв по мережі.

На рис.1 зображений запропонований метод угруповання темпоральних поточкових даних з використанням реактивного програмування, а саме на основі оператора GroupJoin. На початку необхідно ініціалізувати угруповання, а саме передати до GroupJoin відповідні потоки даних від двох пристроїв та налаштування для роботи. Використання лише двох потоків для єдиної ланки ієрархії зумовлене саме особливістю GroupJoin, який працює з двома потоками даних. Надалі відбувається очікування отримання даних від головного пристрою. При отриманні даних, можливо почати порівняння за певним критерієм з подальшим формуванням групи. У разі, якщо пройшов час очікування формування групи, можливо сформувати групу лише з даними від головного пристрою для подальшої передачі цих даних. Критерієм формування групи у запропонованому методі є саме час отримання з дозволеною похибкою. Проте, за необхідності, цю частину методу можливо модифікувати для формування групи також за додатковими критеріями. Час очікування формування та максимально дозволена похибка часу можуть встановлюватися відповідно до очікуваних умов роботи системи, таких як максимальний час розбіжності годинників пристроїв та час надходження даних. Також можлива модифікація алгоритму, що буде формувати групу однократно для кожного елемента даних головного пристрою. Оскільки метод заснований на реактивному програмуванні, після формування групи викликається подія формування групи, що дозволяє надалі використовувати дану групу.

Для подальшого угруповання даних відбувається аналогічне об'єднання даних зі сформованих груп. Також можливо створити реалізацію, що буде угруповувати дані між потоком сформованих груп від декількох пристроїв та потоком отримання даних від пристрою.

Для зменшення витрат пам'яті та часу при подальшому угрупованні дані додаються до списку вже сформованої групи замість формування нової. Таким чином можливо угрупувати дані від довільної кількості пристроїв, сформувавши ієрархію угруповання. Завдяки цьому можливо розподілити процес угруповання між різними процесами, забезпечивши масштабованість та розподілення ресурсів системи для обчислення як між ядрами процесора, так можливо і різними серверами.

Для перевірки роботи запропонованого методу була створена реалізація з використанням бібліотеки Rx для .NET. Оскільки бібліотека Rx, що імплементує різноманітні методи для реактивного програмування, має реалізації для багатьох мов програмування [9], реалізація запропонованого методу можлива на інших платформах.

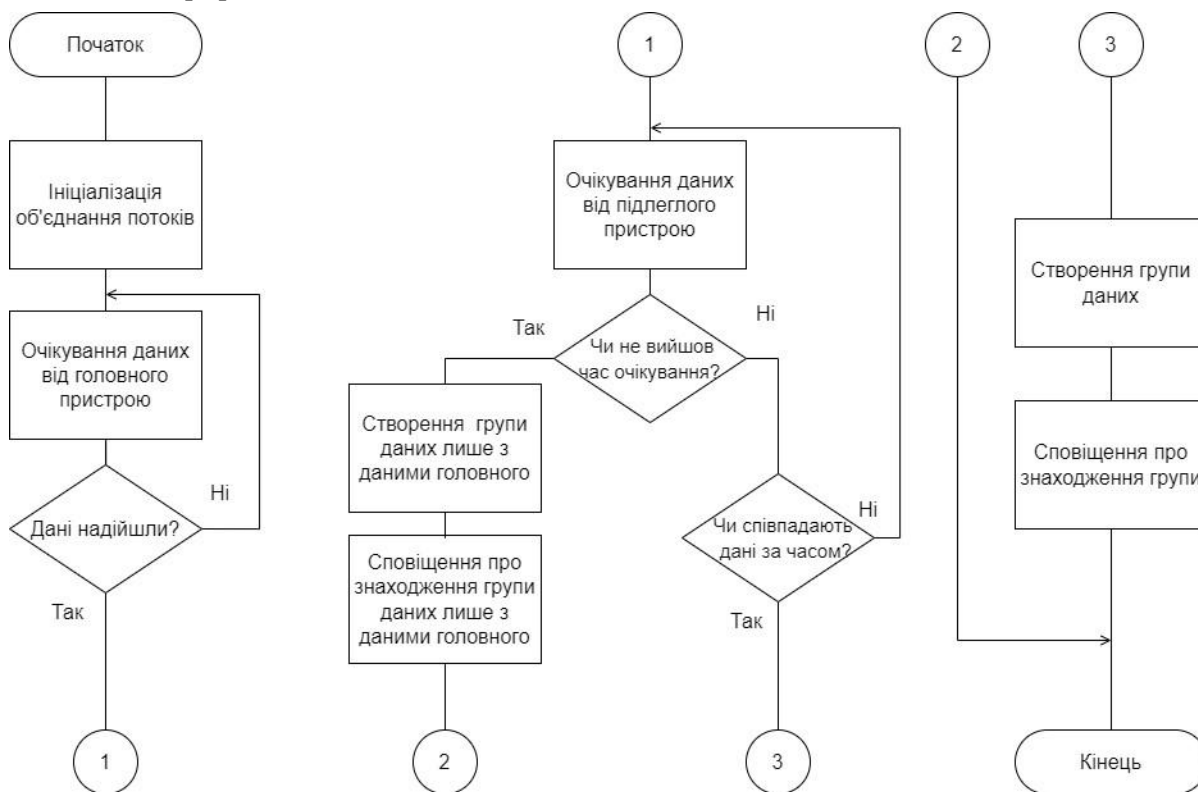


Рисунок 1 – Метод угруповання поточкових даних з використанням реактивного програмування

Для порівняння роботи методу також була створена наївна реалізація, що виконує угруповання між списками даних замість угруповання поточкових даних. На рис.2 зображений результат бенчмарку, швидкодія для невеликої кількості є схожою, проте при зростанні кількості даних для угруповання швидкодія для угруповання поточкових даних зменшується. Цей та наступні бенчмарки були запущені на даній конфігурації: win10-x64, .NET 8, intel i7-7700HQ.

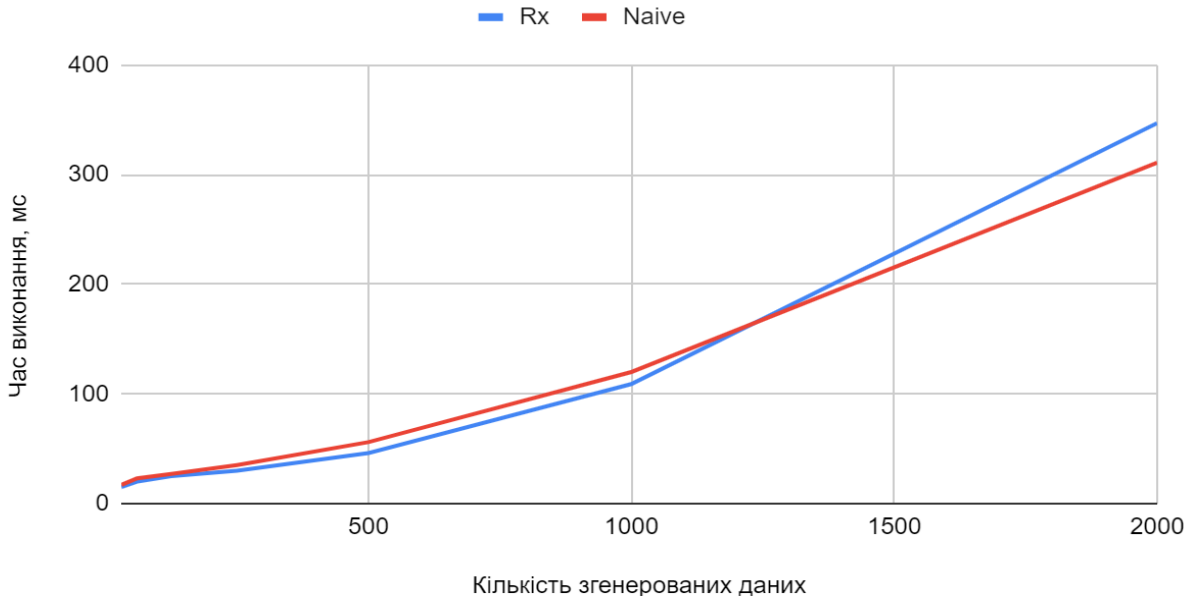


Рисунок 2 – Результати перевірки роботи реалізацій методів з використанням Rx та наївної реалізації

На рис.3 зображений результат бенчмарку для запропонованої реалізації для різної кількості емуляторів пристроїв, що генерують дані. Швидкість генерування даних сягає 40 тисяч за секунду сумарно для всіх пристроїв. Очікувано, при зростанні кількості пристроїв зростає і використана пам'ять. Проте, при подвоєнні кількості пристроїв коефіцієнт збільшення використаної менше ніж у два рази.

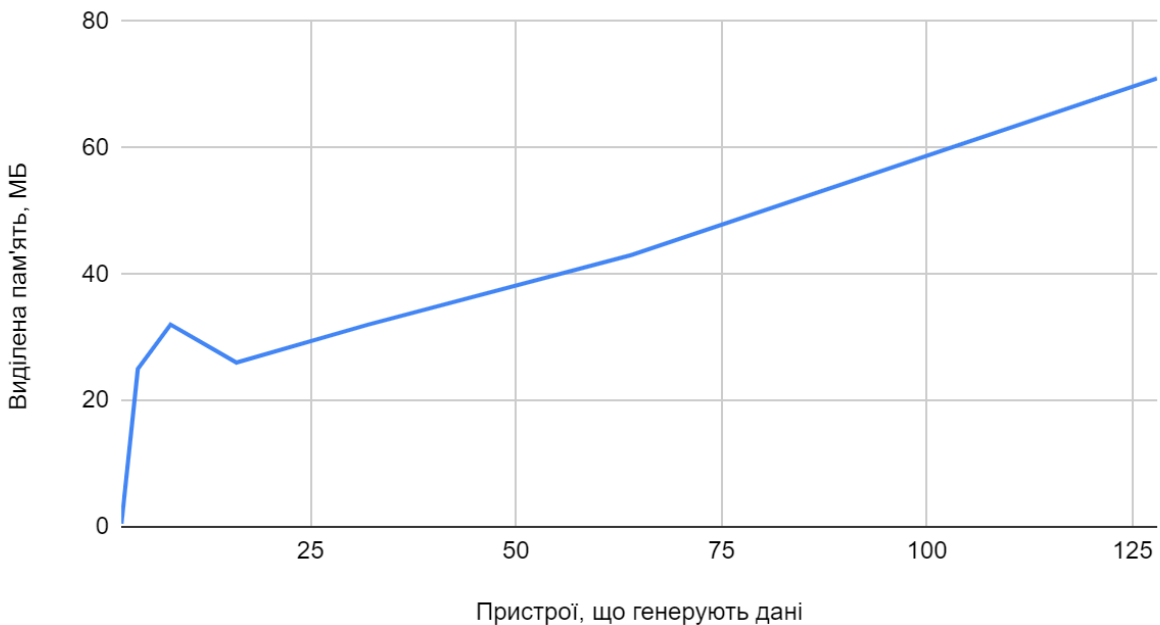


Рисунок 3 – Результати перевірки роботи реалізації методу з використанням Rx по виділеній пам'яті

На рис.4 зображений аналогічний бенчмарк до проведеного на рис.3, але вимірювався час виконання угруповання. Помітне аналогічне зростання витраченої пам'яті до попереднього бенчмарку, тобто при подвоєнні кількості пристроїв час виконання зростає менше ніж у два рази.

Серед особливостей запропонованого методу варто відзначити можливість побудови ієрархії угруповання, що дозволяє гнучко реалізовувати необхідні вимоги угруповання з урахуванням особливостей як самих пристроїв, так і їх підключення до мережі. Відповідно, дане налаштування вимагає часу і ресурсів для аналізу та встановлення необхідних параметрів для конфігурації ланок методу.

Для спрощення налаштування можливе створення модифікації методу, що на кожній ланці буде угруповувати дані одразу від багатьох пристроїв, при цьому зменшивши можливість налаштування кожної окремої ланки та, відповідно, спростивши побудову ієрархії.

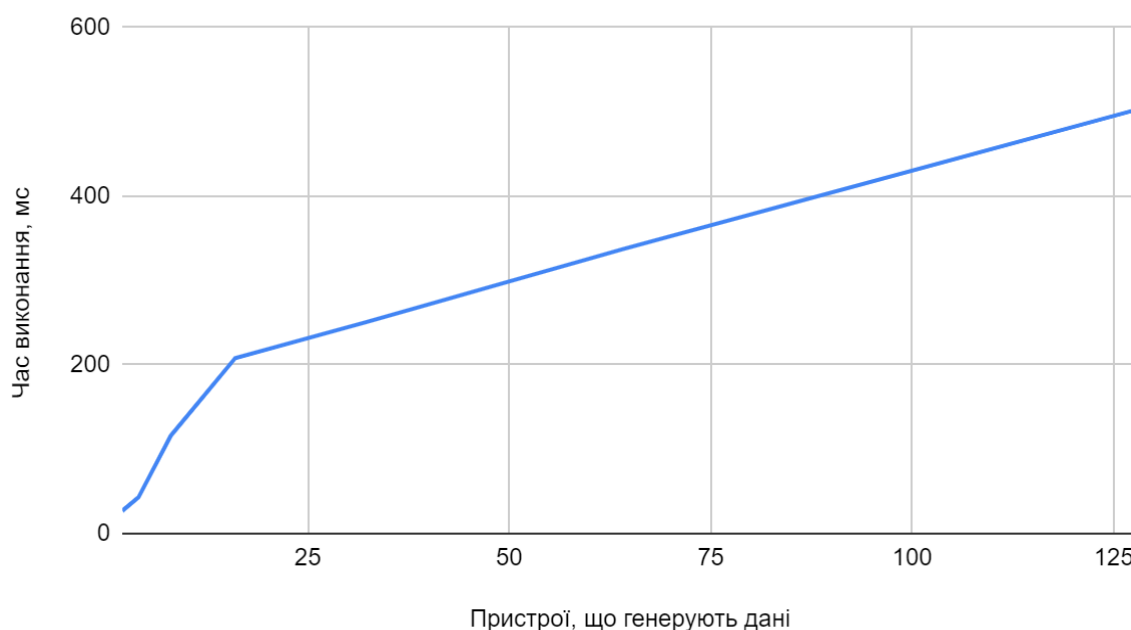


Рисунок 4 – Результати перевірки роботи реалізації методу з використанням Rx по часу виконання

**Висновки.** Отже, запропоновано метод, що дозволяє будувати ієрархію угруповання мультимодальних темпоральних потокових даних від пристроїв IoT з можливістю налаштування часу очікування та максимального часу відмінності між часовими мітками у даних від різних пристроїв. З особливостей варто відмітити необхідність побудови ієрархії угруповання з виділенням головних пристроїв, що зумовлена принципом роботи GroupJoin у Rx.

Також, завдяки розділенню угруповання на окремі ланки, запропонований метод може бути розділений між окремими процесами, що будуть відбуватися у окремих потоках програми або, можливо, й між різними серверами, забезпечуючи масштабованість. Тобто, поставлені цілі для методу були загалом досягнуті. Створена реалізація на .NET підтвердила можливість виконання поставлених завдань запропонованим методом.

З особливостей запропонованого методу варто відзначити можливість налаштування кожної окремої ланки в ієрархії угруповання, що є універсальним рішенням для багатьох можливих систем пристроїв IoT. У разі відсутності необхідності глибокого налаштування кожної окремої ланки можливо продублювати налаштування для усіх ланок, проте перспективним вбачається створення методу угруповання, що дозволить одночасно угруповувати дані від багатьох пристроїв.

#### ЛІТЕРАТУРА

1. Boronat F., Lloret J., Garcia M. Multimedia group and inter-stream synchronization techniques: A comparative study //Information Systems. – 2009. – Т. 34. – №. 1. – С. 108-131.
2. Ravindran K., Bansal V. Delay compensation protocols for synchronization of multimedia data streams //IEEE Transactions on Knowledge and Data Engineering. – 1993. – Т. 5. – №. 4. – С. 574-589.
3. Sivrikaya F., Yener B. Time synchronization in sensor networks: a survey //IEEE network. – 2004. – Т. 18. – №. 4. – С. 45-50.
4. Akbar A. et al. Real-time probabilistic data fusion for large-scale IoT applications //Ieee Access. – 2018. – Т. 6. – С. 10015-10027.
5. Bannach D., Amft O., Lukowicz P. Automatic event-based synchronization of multimodal data streams from wearable and ambient sensors //Smart Sensing and Context: 4th European Conference, EuroSSC 2009, Guildford, UK, September 16-18, 2009. Proceedings 4. – Springer Berlin Heidelberg, 2009. – С. 135-148.
6. Henning S., Hasselbring W. Scalable and reliable multi-dimensional sensor data aggregation in data streaming architectures //Data-Enabled Discovery and Applications. – 2020. – Т. 4. – №. 1. – С. 5.
7. Schueller G., Behrend A. Stream fusion using reactive programming, LINQ and magic updates //Proceedings of the 16th International Conference on Information Fusion. – IEEE, 2013. – С. 1265-1272.
8. Griffiths I., Campbell L. Introduction to Rx.NET. .NET Foundation, 2024. 256 с.



9. Languages. *ReactiveX*. URL: <https://reactivex.io/languages.html> (дата звернення: 15.3.2024).

#### REFERENCES

1. Boronat F., Lloret J., Garcia M. Multimedia group and inter-stream synchronization techniques: A comparative study //Information Systems. – 2009. – Т. 34. – №. 1. – pp. 108-131.
2. Ravindran K., Bansal V. Delay compensation protocols for synchronization of multimedia data streams //IEEE Transactions on Knowledge and Data Engineering. – 1993. – Т. 5. – №. 4. – pp. 574-589.
3. Sivrikaya F., Yener B. Time synchronization in sensor networks: a survey //IEEE network. – 2004. – Т. 18. – №. 4. – pp. 45-50.
4. Akbar A. et al. Real-time probabilistic data fusion for large-scale IoT applications //Ieee Access. – 2018. – Т. 6. – pp. 10015-10027.
5. Bannach D., Amft O., Lukowicz P. Automatic event-based synchronization of multimodal data streams from wearable and ambient sensors //Smart Sensing and Context: 4th European Conference, EuroSSC 2009, Guildford, UK, September 16-18, 2009. Proceedings 4. – Springer Berlin Heidelberg, 2009. – pp. 135-148.
6. Henning S., Hasselbring W. Scalable and reliable multi-dimensional sensor data aggregation in data streaming architectures //Data-Enabled Discovery and Applications. – 2020. – Т. 4. – №. 1. – p. 5.
7. Schueller G., Behrend A. Stream fusion using reactive programming, LINQ and magic updates //Proceedings of the 16th International Conference on Information Fusion. – IEEE, 2013. – pp. 1265-1272.
8. Griffiths I., Campbell L. Introduction to Rx.NET. .NET Foundation, 2024. 256 p.
9. Languages. *ReactiveX*. URL: <https://reactivex.io/languages.html> (access date: 15.3.2024).

Received 13.05.2024.

Accepted 15.05.2024.

#### ***Method of aggregating temporal multimodal streaming data from iot devices based on reactive programming***

*Recent advancements in Internet of Things (IoT) technologies have led to an exponential increase in the generation of temporal multimodal streaming data from various IoT devices. Aggregating and processing this data efficiently poses significant challenges, including scalability, reliability, and synchronization issues. This paper proposes a novel method for aggregating temporal multimodal streaming data from IoT devices based on reactive programming principles.*

*We begin by analyzing recent research and publications in the field to identify existing challenges and limitations in aggregating temporal data from IoT devices. The key problem addressed in this study is the efficient aggregation of data streams while ensuring scalability, reliability, and synchronization across different devices.*

*The primary objective of this research is to develop a method that can hierarchically organize and aggregate temporal data streams from IoT devices while addressing the challenges of scalability and reliability. To achieve this, we leverage reactive programming techniques, specifically the GroupJoin operation, which allows for the seamless integration of data streams from multiple sources.*

*Our method involves initializing the aggregation process by passing relevant data streams from two devices to the GroupJoin operation and configuring the settings for operation. Additionally, we introduce flexibility in the aggregation process by allowing for the hierarchical structuring of data streams based on physical or logical criteria.*

*Furthermore, we discuss the implementation of our proposed method using the Rx library for .NET, which provides implementations for various programming languages, ensuring its adaptability across different platforms. We benchmark the performance of our method against a naive implementation, demonstrating its efficiency and scalability in aggregating temporal data streams from IoT devices.*

*In conclusion, our proposed method offers a scalable and reliable solution for aggregating temporal multimodal streaming data from IoT devices. By leveraging reactive programming principles, we address key challenges in data aggregation and provide a flexible method for organizing and processing data streams efficiently.*

**Лук'янець Михайло Олександрович** – аспірант кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

**Сулєма Євгенія Станіславівна** – д.т.н., доцент, завідувач кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

**Lukianets Mykhailo Oleksandrovych** - Postgraduate student of the Computer Systems Software Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

**Sulema Yevgeniya Stanislavivna** – Doctor of Technical Sciences, Associate Professor, Head of the Computer Systems Software Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".