

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

Іванчук Олексій Вікторович

УДК 004.75:004.02:004.724

ДИСЕРТАЦІЯ
ДОСЛІДЖЕННЯ МОДЕЛЕЙ ТА МЕТОДІВ ОПТИМІЗАЦІЇ
ЕНЕРГОВИТРАТ СИСТЕМ ІНТЕРНЕТУ РЕЧЕЙ

126 Інформаційні системи та технології

12 Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело



О.В. Іванчук

Науковий керівник (консультант) Козел Віктор Миколайович - кандидат технічних наук, доцент

Хмельницький 2025

АНОТАЦІЯ

Івачнук О.В. Дослідження моделей та методів оптимізації енерговитрат у протоколах Інтернету речей. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 126 Інформаційні системи та технології, Херсонський національний технічний університет, Міністерство освіти і науки України, Хмельницький, 2024.

У дисертаційній роботі проведено аналіз існуючих протоколів для систем Інтернету речей на методи і моделі оптимізації енерговитрат, оскільки велика кількість пристроїв застосовують у якості джерела живлення акумулятори або одноразові батареї, які необхідно регулярно заряджати або міняти. Через це можлива ситуація коли пристроїв стане багато і витрати часу на заміну джерела живлення нівелюватимуть корисність пристрою у спрощенні житті людини.

Існуючі протоколи Інтернету речей можна розділити на дві категорії: протоколи передачі сигналів та протоколи структури даних.

До протоколів передачі сигналів відносяться: Wi-Fi, Bluetooth, NB-IoT, ZigBee, 6LoWPAN, WirelessHART та Matter. Через те, що Matter не має власних особливостей для передачі сигналів, а використовує мережі Wi-Fi та ZigBee, то він виключений з цього дослідження, оскільки результати його ефективності можна отримати з результатів протоколів Wi-Fi та ZigBee.

Протоколами структури даних є: HTTP, SOAP, XMPP, STOMP, CoAP, MQTT.

В процесі аналізу для протоколів передачі сигналів визначено структуру типового пакету даних та усі наявні в них методів оптимізації енерговитрат. Аналіз структури пакету даних дозволяє визначити обсяги технічних даних, що дозволить провести аналіз споживання енергії на основі загальної кількості інформації, що передається.

На основі аналізу протоколів передачі сигналів виявлено методи оптимізації, що наявні в них та розподілені за категоріями: керування часом активності

пристрою; керування частотними характеристиками передачі; керування потужністю передавача; керування пакетами даних; керування маршрутизацією.

Протоколи ZigBee та WirelessHART мають методи оптимізації за всіма наявними категоріями. Протоколи Wi-Fi, 6LoWPAN мають по 4 категорії методів оптимізації. Bluetooth та NB-IoT мають по 3 категорії оптимізації. Найкращі показники у методів оптимізації, що відносяться до критерію «Керування часом активності пристрою», що зумовлено повним відключенням пристрою на певні проміжки часу, поки не настає момент збору даних та відправки їх до центру обробки.

Над протоколами структури даних був проведений аналіз впливу форматування інформації на загальний розмір пакету даних. Для аналізу були сформовані три пакети даних з різним наповненням: числові дані, текстові дані, змішані дані. Ці пакети були приведені до форматів встановленні протоколами структури даних та порівняно їх розмір. Найкращі показники має протокол MQTT, але оскільки він вимагає постійної активності пристрою для підтримки каналу зв'язку, то оптимальніше використати протокол CoAP, що не має такої вимоги та має найближчі показники до MQTT.

Оскільки протоколи структури даних мають працювати у парі з протоколами передачі сигналів, то була виконана перевірка загальних розмірів пакетів у парах кожен-з-кожним.

Найгірші результати має протокол Zigbee у комбінації з SOAP або XMPP через те що у протоколах SOAP та XMPP є надмірність через подвійний опис змінних, через що дані не вміщуються в одному пакеті ZigBee і необхідно застосовувати декілька пакетів між якими розділяться усі данні.

Найкращі результати виявлені у комбінації NB-IoT з MQTT. Вважаючи обмеження MQTT доцільніше застосування NB-IoT у парі з CoAP.

Оскільки усі протоколи передачі сигналів мають у собі вимогу до захисту інформації при передачі, то було проведено аналіз впливу шифрування кожного протоколу на кінцеві розміри пакету даних. Згідно з результатами в кожному протоколі необхідно проводити доповнення технічними даними інформації, аби

вона розміри повідомлення для шифрування були кратними до розміру ключа. В середньому обсяги доповнення складають близько 26% відносно обсягів інформації, що передається. Вплив на загальний розмір пакету складає близько 10% для протоколів Wi-Fi, ZigBee, WirelessHART, 6LoWPAN. Для Bluetooth складає 16%, а для NB-IoT – 20%. Більший обсяг доповнень пов'язаний з меншим обсягом технічних даних у протоколах Bluetooth та NB-IoT.

На основі математичного рішення розроблене програмне забезпечення, що дозволяє обрати протоколи вказавши характеристики системи, що необхідно побудувати. Результати роботи програми є рекомендаційними і не обов'язковими для застосування, якщо є невраховані розбіжності між можливостями протоколів та системою Інтернету речей.

Ключові слова: протоколи, Інтернет речей, енергоефективність, оптимізація, трафік, системи.

Список публікацій:

1. Kozel, V., Ivanchuk, O., Drozdova, I., Prykhodko, O. Automation of the Protocol Selection Process for IoT Systems. International Journal of Computing. 2022. 21(2). Pp. 251-257. DOI: 10.47839/ijc.21.2.2594
2. Козел В.М., Іванчук О.В., Дроздова Є.А. Розробка системи збору інформації від IoT пристроїв. Вісник Херсонського національного технічного університету. 2019. № 3(70). С. 126-132. DOI: 10.35546/kntu2078-4481.2019.3.14
3. Козел В.М., Іванчук О.В., Дроздова Є.А., Завгородній В.В. Аналіз протоколів обміну даними для керування системами Інтернету речей. Вчені записки Таврійського національного університету імені В.І. Вернадського. 2020. № 31(70). С. 99-104. DOI: 10.32838/2663-5941/2020.2-1/15
4. Іванчук О.В., Козел В.М. Дослідження впливу захисту інформації на обсяги пакетів даних протоколів Інтернету речей. Науковий журнал "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" Луцьк, 2024. Випуск № 57. с. 43-50. DOI: 10.36910/6775-2524-0560-2024-57-06
5. Ivanchuk O., Kozel V., Drozdova Ie., Prykhodko O. Research of the Methods for optimizing energy consumption in IEEE 802.15.4 protocols. Вчені записки

Таврійського національного університету імені В.І. Вернадського. 2024. Том 35 (74) №6. С. 112-117. DOI: 10.32782/2663-5941/2024.6.1/19

6. Іванчук О.В. Дослідження методів оптимізації енерговитрат у протоколах Інтернету речей. Вісник Херсонського національного технічного університету. 2024. № 4(91). с. 273-279. DOI: 10.35546/kntu2078-4481.2024.4.35

Апробації результатів дисертації:

1. Іванчук О.В., Козел В.М. Проблеми використання IoT-пристроїв. Молодь у світі сучасних технологій за тематикою: Використання інформаційних технологій в системах управління : матеріали міжнар. наук.- практ. конф. м. Херсон, 6-7 червня 2019 р.) / за заг. ред. Г.О. Райко – Херсон : Видавництво ФОП Вишемирський В. С., 2019. – 216 с.

2. Іванчук О.В., Козел В.М., Дроздова Є.А. Аналіз методів шифрування протоколів інтернету речей . Сучасні комп'ютерні системи та мережі в управлінні: матеріали VI Всеукраїнської наук.-практ. Інтернет-конф. здобувачів вищої освіти та молодих вчених (30 листопада 2023 р., м. Хмельницький, м. Херсон) / за ред. А.А. Григорової. – Херсон: Книжкове видавництво ФОП Вишемирський В. С., 2023. –с 245-247

3. Іванчук О.В., Козел В.М., Зоря Р.І. Дослідження протоколів та стандартів IoT. Сучасні комп'ютерні системи та мережі в управлінні: матеріали VI Всеукраїнської наук.-практ. Інтернет-конф. здобувачів вищої освіти та молодих вчених (30 листопада 2023 р., м. Хмельницький, м. Херсон) / за ред. А.А. Григорової. – Херсон: Книжкове видавництво ФОП Вишемирський В. С., 2023. –с 173-174

4. Іванчук О.В., Козел В.М., Дроздова Є.А. Інтернет речей у післявоєнному відновленні Херсонщини. Синергія науки і бізнесу у повоєнному відновленні Херсонщини : матеріали Міжнародної науково-практичної конференції (ХНТУ, 26–28 квітня 2023 р.) у 2-х т. ; Т. 1 / за ред. О. В. Чепелюк. – Одеса : Олді+, 2023. – 456 с.

5. Іванчук О.В., Козел В.М., Дроздова Є.А. Методи оптимізації енерговитрат у протоколах Інтернету речей. Сучасні комп'ютерні системи та мережі в

управлінні: матеріали VII Всеукраїнської наук.-практ. Інтернет-конф. здобувачів вищої освіти та молодих вчених (29 листопада 2024 р., м. Хмельницький, м. Херсон) / за ред. А.А. Григорової. – Херсон: Книжкове видавництво ФОП Вишемирський В. С., 2024.

ABSTRACT

Ivachnuk O.V. Research on models and methods for optimizing energy consumption in Internet of Things protocols. – Qualification scientific work in the form of a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 126 Information Systems and Technologies, Kherson National Technical University, Ministry of Education and Science of Ukraine, Khmelnytskyi, 2024.

The dissertation analyzes existing protocols for Internet of Things systems for methods and models for optimizing energy consumption, since a large number of devices use accumulators or disposable batteries as a power source, which must be regularly charged or changed. Because of this, a situation is possible when there will be many devices and the time spent on replacing the power source will negate the usefulness of the device in simplifying human life.

Existing Internet of Things protocols can be divided into two categories: signal transmission protocols and data structure protocols.

Signal transmission protocols include: Wi-Fi, Bluetooth, NB-IoT, ZigBee, 6LoWPAN, WirelessHART and Matter. Since Matter does not have its own features for signal transmission, but uses Wi-Fi and ZigBee networks, it is excluded from this study, since the results of its efficiency can be obtained from the results of Wi-Fi and ZigBee protocols.

The data structure protocols are: HTTP, SOAP, XMPP, STOMP, CoAP, MQTT.

During the analysis, the structure of a typical data packet and all the energy optimization methods available in them were determined for the signal transmission protocols. Analysis of the data packet structure allows you to determine the volume of technical data, which will allow you to analyze energy consumption based on the total amount of information transmitted.

Based on the analysis of the signal transmission protocols, the optimization methods available in them were identified and distributed by categories: device activity time management; transmission frequency characteristics management; transmitter power management; data packet management; routing management.

The ZigBee and WirelessHART protocols have optimization methods for all available categories. The Wi-Fi and 6LoWPAN protocols have 4 categories of optimization methods each. Bluetooth and NB-IoT have 3 categories of optimization each. The best indicators in optimization methods related to the criterion "Device activity time management", which is due to the complete shutdown of the device for certain periods of time until the moment of data collection and sending it to the processing center.

The data structure protocols were analyzed for the impact of information formatting on the total size of the data packet. For the analysis, three data packets with different contents were generated: numeric data, text data, mixed data. These packets were converted to the formats established by the data structure protocols and their sizes were compared. The MQTT protocol has the best indicators, but since it requires constant device activity to maintain the communication channel, it is more optimal to use the CoAP protocol, which does not have such a requirement and has the closest indicators to MQTT.

Since data structure protocols must work in pairs with signaling protocols, the total packet sizes were checked in pairs of each-with-each.

The worst results are achieved by the Zigbee protocol in combination with SOAP or XMPP because the SOAP and XMPP protocols have redundancy due to double description of variables, which means that the data does not fit in one ZigBee packet and it is necessary to use several packets between which all the data will be divided.

The best results were found in the combination of NB-IoT with MQTT. Considering the limitations of MQTT, it is more expedient to use NB-IoT in combination with CoAP.

Since all signaling protocols have a requirement for information protection during transmission, an analysis of the impact of encryption of each protocol on the final size of the data packet was conducted. According to the results, in each protocol it is necessary to supplement the technical data of the information so that the size of the message for encryption is a multiple of the key size. On average, the amount of supplementation is about 26% relative to the amount of information transmitted. The impact on the total size of the packet is about 10% for the Wi-Fi, ZigBee, WirelessHART, 6LoWPAN protocols.

For Bluetooth, it is 16%, and for NB-IoT – 20%. The larger volume of additions is associated with a smaller volume of technical data in the Bluetooth and NB-IoT protocols.

Based on the mathematical solution, software has been developed that allows you to select protocols by specifying the characteristics of the system that needs to be built. The results of the program are recommendatory and not mandatory for use if there are unaccounted for discrepancies between the capabilities of the protocols and the Internet of Things system.

Keywords: protocols, Internet of Things, energy efficiency, optimization, traffic, systems.

List of publications:

1. Kozel, V., Ivanchuk, O., Drozdova, I., Prykhodko, O. Automation of the Protocol Selection Process for IoT Systems. *International Journal of Computing*. 2022. 21(2). Pp. 251-257. DOI: 10.47839/ijc.21.2.2594

2. Kozel V.M., Ivanchuk O.V., Drozdova E.A. Development of a system with of information from IoT devices. *Bulletin of the Kherson National Technical University*. 2019. No. 3(70). P. 126-132. DOI: 10.35546/kntu2078-4481.2019.3.14

3. Kozel V.M., Ivanchuk O.V., Drozdova E.A., Zavgorodniy V.V. Analysis of data exchange protocols for controlling Internet of Things systems. *Scientific notes of the Tavria National University named after V.I. Vernadsky*. 2020. No. 31(70). P. 99-104. DOI: 10.32838/2663-5941/2020.2-1/15

4. Ivanchuk O.V., Kozel V.M. Study of the impact of information protection on the volumes of data packets of Internet of Things protocols. *Scientific journal "Computer-integrated technologies: education, science, production"* Lutsk, 2024. Issue No. 57 p. 43-50. DOI: 10.36910/6775-2524-0560-2024-57-06

5. Ivanchuk O., Kozel V., Drozdova Ie., Prykhodko O. Research of the Methods for optimizing energy consumption in IEEE 802.15.4 protocols. *Scientific notes of the Tavria National University named after V.I. Vernadsky*. 2024. Volume 35 (74) №6. DOI:

6. Ivanchuk O.V. Research of methods for optimizing energy consumption in Internet of Things protocols. *Bulletin of the Kherson National Technical University*. 2024. № 4(91).

Approbations of the dissertation results:

1. Ivanchuk O.V., Kozel V.M. Problems of using IoT devices. Youth in the world of modern technologies on the topic: Using information technologies in management systems: materials of the international scientific and practical conference (Kherson, June 6-7, 2019) / edited by G.O. Raiko – Kherson: Publishing house FOP Vyshemyrskyi V.S., 2019. – 216 p.

2. Ivanchuk O.V., Kozel V.M., Drozdova E.A. Analysis of encryption methods for Internet of Things protocols. Modern computer systems and networks in management: materials of the VI All-Ukrainian scientific-practical Internet conference of higher education students and young scientists (November 30, 2023, Khmelnytskyi, Kherson) / edited by A.A. Grigorova. – Kherson: Book publishing house FOP Vyshemyrskyi V. S., 2023. –p. 245-247

3. Ivanchuk O.V., Kozel V.M., Zorya R.I. Research on IoT protocols and standards. Modern computer systems and networks in management: materials of the VI All-Ukrainian scientific-practical Internet conference. of higher education students and young scientists (November 30, 2023, Khmelnytskyi, Kherson) / edited by A.A. Grigorova. – Kherson: Book Publishing House FOP Vyshemyrskyi V. S., 2023. –p. 173-174

4. Ivanchuk O.V., Kozel V.M., Drozdova E.A. Internet of Things in the Post-War Restoration of the Kherson Region. Synergy of Science and Business in the Post-War Restoration of the Kherson Region: Materials of the International Scientific and Practical Conference (KhNTU, April 26–28, 2023) in 2 vols. ; Vol. 1 / edited by O.V. Chepelyuk. – Odesa: Oldi+, 2023. – 456 p.

5. Ivanchuk O.V., Kozel V.M., Drozdova E.A. Methods for optimizing energy consumption in Internet of Things protocols. Modern computer systems and networks in management: materials of the VII All-Ukrainian scientific-practical Internet conference of higher education students and young scientists (November 29, 2024, Khmelnytskyi, Kherson) / edited by A.A. Grigorova. – Kherson: Book publishing house FOP Vyshemyrskyi V. S., 2024.

ЗМІСТ

ВСТУП	13
РОЗДІЛ 1	18
Аналіз протоколів інтернету речей	18
1.1 Види протоколів Інтернету речей.....	19
1.2 Протокол Wi-Fi.....	21
1.3 Протокол Bluetooth	26
1.3 Протокол NB-IoT.....	31
1.4 Протокол ZigBee.....	35
1.5 Протокол 6LoWPAN	39
1.6 Протокол WirelessHART	41
1.7 Протокол Matter.....	45
1.8 Протокол HTTP	49
1.9 Протокол SOAP	52
1.10 Протокол XMPP	53
1.11 Протокол STOMP	55
1.12 Протокол CoAP.....	57
1.13 Протокол MQTT та MQTT-SN.....	58
1.14 Висновки до першого розділу.....	60
РОЗДІЛ 2	62
Дослідження протоколів передачі сигналів	62
2.1 Аналіз протоколу Wi-Fi	62
2.2 Аналіз протоколу Bluetooth.....	65
2.3 Аналіз протоколу NB-IoT	66
2.4 Аналіз протоколу ZigBee	69
2.5 Аналіз протоколу 6LoWPAN	71
2.6 Аналіз протоколу WirelessHART	74
2.7 Аналіз розмірів технічних даних у пакетах.....	77
2.8 Аналіз методів оптимізації енергоспоживання	77

	12
2.9 Висновки до розділу.....	82
РОЗДІЛ 3	83
Дослідження протоколів структури даних	83
3.1 Формула розрахунку об'єму даних	83
3.2 Формування пакету даних для передачі	85
3.3 Формування тіла повідомлення	86
3.4 Розрахунок об'ємів даних у тілі повідомлення.....	89
3.5 Розрахунок об'ємів даних у тілі повідомлення.....	91
3.6 Аналіз пакетів у протоколах структури даних.....	97
3.7 Розрахунок об'ємів технічних даних у повному пакеті даних.....	99
3.8 Розрахунок обсягів повного пакету даних	100
3.9 Аналіз обсягів повних пакетів даних	101
3.10 Висновки до розділу	105
РОЗДІЛ 4	106
Дослідження впливу шифрування на розміри пакетів	106
4.1 Аналіз шифрування у протоколах передачі сигналів.....	106
4.2 Аналіз впливу шифрування на пакети даних	108
4.3 Висновки до розділу	112
РОЗДІЛ 5	114
Автоматизація процесу вибору протоколів Інтернету речей	114
5.1 Математична основа вибору протоколів Інтернету речей.....	114
5.2 Автоматизація процесу вибору протоколів Інтернету речей	120
5.3 Висновки до розділу	122
ВИСНОВКИ.....	123
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	126
ДОДАТОК А.....	136

ВСТУП

Розвиток технологій призвів до появи систем розумного будинку, що мають підключення до мережі Інтернет. Всі пристрої у таких системах відносяться до Інтернету речей. Інтернет речей (Internet of Things, IoT) - це система, яка забезпечує зв'язок між електронними пристроями та датчиками через Інтернет, щоб полегшити життя людини.

Актуальність дослідження. Зазвичай пристрої мають власне джерело живлення, таке як акумулятор або одноразові батареї. Проблемою таких пристроїв є час, за який джерело живлення буде виснажене. Це накладає обмеження на протоколи, що використовуватимуться у системі.

З кожним роком збільшується кількість пристроїв інтернету речей. Разом і з ними збільшується кількість нових протоколів для проєктування систем, або модернізованих вже існуючих протоколів для забезпечення вимог функціонування систем інтернету речей. Головною вимогою при оновленні існуючих протоколів зазвичай є підвищення енергоефективності протоколів для можливості використання пристроїв Інтернету речей протягом років без супутнього обслуговування.

Аналіз моделей та методів оптимізації енерговитрат у існуючих протоколах Інтернету речей дозволить визначити їх загальний вплив на споживання енергії та в майбутньому оптимізувати існуючі протоколи, або створити нові.

Зв'язок роботи з науковими програмами, планами, темами. Наукові дослідження виконувались в рамках тематичного плану науково-дослідних робіт відповідно теми «Моделі, методи та інформаційні технології комп'ютеризованого навчання» кафедри інформаційних технологій Херсонського національного технічного університету (державний реєстраційний № 0119U103822). Автор приймав участь у цій роботі як виконавець. Роль автора полягала в дослідженні протоколів систем Інтернету речей для визначення моделей та методів оптимізації енергоспоживання та застосування їх в системах комп'ютеризованого навчання.

Мета та завдання дослідження. Метою дисертаційної роботи є удосконалення теоретичних і науково-практичних положень, розроблення науково-методичних основ і практичних рекомендацій щодо вибору протоколів для побудови систем Інтернету речей базуючись на їх оптимізації енерговитрат.

Для досягнення поставленої мети в роботі необхідно було визначити та вирішити такі завдання:

- сформулювати список актуальних протоколів для систем Інтернету речей;
- визначити та систематизувати методи оптимізації енерговитрат у протоколах Інтернету речей;
- проаналізувати моделі енерговитрат у протоколах Інтернету речей;
- визначити критерії для вибору протоколів при побудові системи Інтернету речей;
- сформулювати залежність впливу критеріїв на вибір протоколів та вивести математичну функцію залежності, що має визначати оптимальні протоколи за заданими вимогами системи та рівнем оптимізації енерговитрат протоколів;
- визначити оптимальні протоколи згідно математичної функції на основі базових прикладів та підтвердити доцільність її застосування;
- сформулювати практичні рекомендації щодо формування критеріїв та використання математичної функції.

Методи дослідження. В рамках дисертаційної роботи було використано методи системного аналізу – для дослідження наявних протоколів Інтернету речей; теорію та методів роботи мереж – для аналізу пакетів даних у протоколах; метод експерименту та порівняння – для перевірки впливу типів даних на загальних трафік при передачі інформації у мережі Інтернету речей; візуальні методи, методи системного аналізу, математичні та експериментальні методи – для побудови графіків залежності доповнень технічними при захисті пакету даних за допомогою шифрування у протоколах передачі сигналів відносно обсягів інформації, що передається; методи математичної логіки та теорії операторів, методи системного аналізу – для визначення математичних функцій вибору протоколів за критеріями при побудові систем Інтернету речей; методи математичної логіки, об'єктно-

орієнтованого програмування, функціонального програмування, теорію алгоритмів, методи наближених обчислень – для розробки програмного забезпечення автоматичного вибору протоколів для систем Інтернету речей за наявними вимогами до системи.

Наукова новизна полягає в наступному:

- Проведено комплексний аналіз методів оптимізації енерговитрат протоколів Інтернету речей, що має у собі розподіл протоколів на дві категорії та аналіз на основі цієї категорії: протоколи передачі сигналів та протоколи структури даних.
- Визначено вплив на енергоспоживання протоколів базуючись на розмірах пакетів даних в залежності від типу інформації та протоколу структури даних, що задіяно для формування пакету даних.
- Проведено аналіз впливу шифрування на обсяги пакету даних при доповненні технічними даними до розміру ключа.
- Визначено математичні функції вибору протоколів для побудови систем Інтернету речей за критеріями на основі булевої алгебри, що дозволяє спростити процес вибору.
- На основі булевих функцій розроблено програмне забезпечення, яке дозволяє автоматизувати процес вибору протоколів для побудови систем Інтернету речей на основі їх оптимізації енерговитрат та можливостей, що вони надають.

Практичне значення одержаних результатів. Результати роботи дозволяє обирати протоколи для систем Інтернету речей на основі їх можливостей та енергоефективності. Також результати дозволяє обирати методи для підвищення енергоефективності при розробці нових протоколів для систем Інтернету речей базуючись на результатах вже існуючих протоколів.

Апробація результатів дисертації відбулися на конференціях:

1. Міжнародна науково-практична конференція «Молодь у світі сучасних технологій за тематикою: Використання інформаційних технологій в системах управління» (м. Херсон 6-7 червня 2019 р.).

2. Міжнародна науково-практична конференція «Молодь у світі сучасних технологій за тематикою: Сучасні інформаційні технології: стан та перспективи розвитку» (м. Херсон 2021 р.).

3. 13 Міжнародна науково-практична конференція «Eurasian scientific discussions» (м. Берселона, 2023 р.); 1 Міжнародна науково-практична конференція «Innovations and prospects in modern science» (м. Стокгольм, 2023 р.).

4. VI Всеукраїнська науково-практична конференція «Сучасні комп'ютерні системи та мережі в управлінні» (м. Херсон – м. Хмельницький, 30 листопада 2023 р.).

5. V Всеукраїнська науково-практична конференція «Сучасна молодь в світі інформаційних технологій» (м. Херсон – м. Кропивницький, 17 травня 2024 р.).

6. Міжнародна науково-практична конференція «Синергія науки і бізнесу у повоєнному відновленні Херсонщини» (м. Хмельницький, 26–28 квітня 2023 р.).

7. II Міжнародна науково-практична конференція «Синергія науки і бізнесу у повоєнному відновленні Херсонщини» (м. Хмельницький, 24–26 квітня 2024 р.).

8. VII Всеукраїнська науково-практична конференція «Сучасні комп'ютерні системи та мережі в управлінні» (м. Херсон – м. Хмельницький, 2024 р.).

Особистий внесок здобувача полягає в тому, що результати дослідження отримані самостійно.

Публікації

Основні результати дослідження опубліковані в 5 наукових працях, усі у наукових фахових виданнях, рекомендованих МОН України для публікації наукових досліджень:

1. International Journal of Computing. 2022. 21(2). Pp. 251-257.

2. Вчені записки Таврійського національного університету імені В.І. Вернадського. 2020. № 31(70). С. 99-104.

3. Науковий журнал "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" Луцьк, 2024. Випуск № 57. С. 43-50.

4. Вчені записки Таврійського національного університету імені В.І. Вернадського. 2024.

5. Вісник Херсонського національного технічного університету. №4(91). ХНТУ. 2024.

Структура та обсяг дисертації

Дисертація складається зі вступу, п'яти розділів, висновків для кожного розділу, висновку, бібліографічного списку зі 91 найменувань, 1 додатку. Обсяг дисертації складає 137 сторінок, з яких 135 сторінок основного тексту, в тому числі 32 рисунки і 11 таблиць.

РОЗДІЛ 1

АНАЛІЗ ПРОТОКОЛІВ ІНТЕРНЕТУ РЕЧЕЙ

Термін «Інтернет речей» (IoT) був введений Кевіном Ештоном на презентації в Proctor & Gamble у 1999 році. Інтернет речей (IoT) – це мережа фізичних об’єктів, які називаються «речами», у яких вбудовано програмне забезпечення, електроніка, мережа та датчики, що дозволяє цим об’єктам збирати та обмінюватися даними для полегшення існування людини [1, 2]. Інтернет речей можна назвати Інтернетом усього. Його також можна розглядати як промисловий Інтернет. В системі інтернету речей окремим пристроєм може бути людина, що має в своєму тілі електронні імпланти, які можуть як і підтримувати існування людини, так і керувати іншими пристроями.

Пристрій IoT (також відомий як кінцева точка) можна розуміти як гаджет, який збирає інформацію під час моніторингу певної цілі. Це здійснюється за допомогою датчиків, які дозволяють збирати дані про цікаві змінні, такі як місцезнаходження та інші мультимедійні дані. Пристрій просто видає інформацію для підключення до Інтернету та передає інформацію, спрямовану в хмару. Кожен пристрій має власну ідентифікацію в системі, відому як унікальний ідентифікатор (UID), який може допомогти визначити походження будь-яких конкретних даних. Отже, IoT забезпечує не тільки комфорт громадян, але й промислові потреби міста.

Більшість архітектур, розроблених для IoT, були створені через бездротові мережі, оскільки вони пропонують ефективну передачу даних із низьким споживанням ресурсів. Спеціальні архітектури були розроблені для великих площ або великогабаритних пристроїв для роботи через радіоканал. Це спрямовано на диверсифікацію методів транспортування даних IoT. Значною проблемою для розробників пристроїв є створення пристроїв, які можна поєднати з різними гілками Інтернету речей.

Пристрої у системі Інтернету речей можуть бути трьох типів:

- пристрої збору даних;

- пристрої виконання;
- комбіновані.

Пристрої збору даних виконують зчитування інформації з датчиків та передачу отриманої інформації у центр обробки даних або напряму на інші пристрої. Це можуть бути пристрої зчитування мікроклімату, пристрої безпеки та інші.

Пристрої виконання отримують команди від центру керування або інших пристроїв та забезпечують виконання дії з команди. Це можуть бути пристрої керування освітленням, пристрої керування системою опалення, пристрої сповіщення о небезпеці та інші.

Комбіновані пристрої мають в собі можливості попередніх двох типів: збирають данні, передають у центр обробки даних або на іншим пристроєм, отримують команди, виконують команди.

1.1 Види протоколів Інтернету речей

Протоколи регламентують передачу даних у системі. Вони описують метод зв'язку у системі, сигнали та структуру даних. Протоколи можна розділити на два види:

- протоколи передачі сигналів;
- протоколи структури даних.

Протоколи ділять між собою рівні моделі OSI. На рисунку **Ошибка! Текст указанного стиля в документе отсутствует..1** відображено розділення їх по рівням OSI.



Рисунок **Ошибка! Текст указанного стиля в документе отсутствует..1** –

Розподілення протоколів за рівнями у моделі OSI

Згідно з розподіленням рівнів моделі OSI протоколи передачі сигналів визначають середовище для передачі даних, адресацію та розділення на пакети [3].

Протоколи структури даних визначають формат ідентифікаторів та даних у пакеті. Також пакет визначає наявність пакетів підтвердження про успішність передачі даних.

До протоколів передачі сигналів належать:

- Wi-Fi;
- Bluetooth;
- NB-IoT;
- ZigBee;
- 6LoWPAN;
- WirelessHART.

До протоколів структури даних відносяться:

- HTTP;
- SOAP;
- XMPP;
- STOMP;
- CoAP;
- MQTT або MQTT-SN.

Для формування пакету даних необхідно комбінувати протоколи передачі сигналів з протоколами структури даних. Таке комбінування визначає середовище

передачі та сигнали якими передаватимуться дані закодовані протоколами структури даних.

1.2 Протокол Wi-Fi

Wi-Fi – це технологія бездротових мереж, яка дозволяє пристроям взаємодіяти з мережею Інтернет [4]. У якості пристроїв мережі можуть бути комп'ютери, ноутбуки, смартфони, переносні пристрої, принтери, відеокамери, пристрої розумного будинку. Це дозволяє усім пристроям обмінюватися даними між собою, утворюючи локальну мережу між собою.

Для обміну даними у мережі необхідний маршрутизатор, який має підключення до мережі Інтернет формуючи шлюз між пристроями та мережею Інтернет. Для своєї роботи Wi-Fi використовує радіоканал на частоті 2.4, 5 та 6 ГГц.

В основі протоколу використовується набір стандартів IEEE 801.11, що регламентує комунікацію бездротової локальної мережі. Оскільки середовищем передачі згідно стандарту є радіохвилі, то для корекції помилок при впливі перешкод, або через одночасну передачу то необхідний метод для пошуку помилок. Для цього було обрано метод перевірки на колізії CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) [5].

Коли клієнт хоче надіслати дані, він спочатку перевіряє, чи є в середовищі активна передача. Якщо активність виявлено, вузол переходить у режим очікування на певний проміжок часу, після чого здійснює повторну спробу. Коли середовище передачі стає вільним, клієнт надсилає запит на передачу (RTS) до бездротової точки доступу (WAP). Якщо WAP зайнятий, запит RTS відхиляється, і клієнт продовжує очікування. Коли WAP стає доступним для обміну даними, він підтверджує RTS, надсилаючи сигнал Clear to Send (CTS). WAP припиняє обмін з іншими вузлами та очікує завершення передачі від вузла, який отримав підтвердження. Після завершення передачі даних WAP відновлює обмін з іншими клієнтами або очікує нових запитів на передачу.

У мережах Wi-Fi використовується напівдуплексна передача даних між пристроями. При такій системі всі пристрої у мережі використовують один спільний радіоканал. При цьому під час передачі пристрій блокує отримання даних, що не дозволяє самому пристрою дізнатися про колізію. Для цього використовується розподілена функція корекції DCF (Distributed Coordination Function) [9]. Згідно цієї функції пристрій виконує передачу лише у випадку якщо канал вільний. Також пристрій має отримати підтвердження про отримання даних. Якщо він відсутній, то буде виконана повторна відправка пакету.

Для підключення до мережі пристрою необхідно з'єднатися з точкою бездротового доступу, що є маршрутизатором у мережі. Він перевіряє дані авторизації для підключення до мережі. Також при запиті від пристрою передає дані мережі, що утворена через нього для можливості подальшого підключення.

Перша версія протоколу була розроблена 1997 році. За період з 1997 по 2019 роки були розроблені нові версії протоколу. У таблиці 1.1 відображені версії протоколу, час їх виходу та основні характеристики.

Таблиця 1.1 – Характеристики версій протоколів Wi-Fi

Версія	Рік виходу	Частота передачі сигналу	Швидкість передачі
802.11	1997	2.4 ГГц	1 Мбіт/с
802.11b	1999	2.4 ГГц	11 Мбіт/с
802.11a	1999	2.4 ГГц	54 Мбіт/с
802.11g	2003	2.4 ГГц	54 Мбіт/с
802.11n	2009	2.4/5 ГГц	600 Мбіт/с
802.11ad	2009	60 ГГц	7 Гбіт/с
802.11ac	2013	5 ГГц	7 Гбіт/с
802.11af	2014	790 МГц	26 Мбіт/с
802.11ah	2016	900 МГц	347 Мбіт/с
802.11ax	2019	2.4/5 ГГц	11 Гбіт/с

На рисунку 1.2 відображено розподіл версій за частотним спектром та відносними зонами покриття.

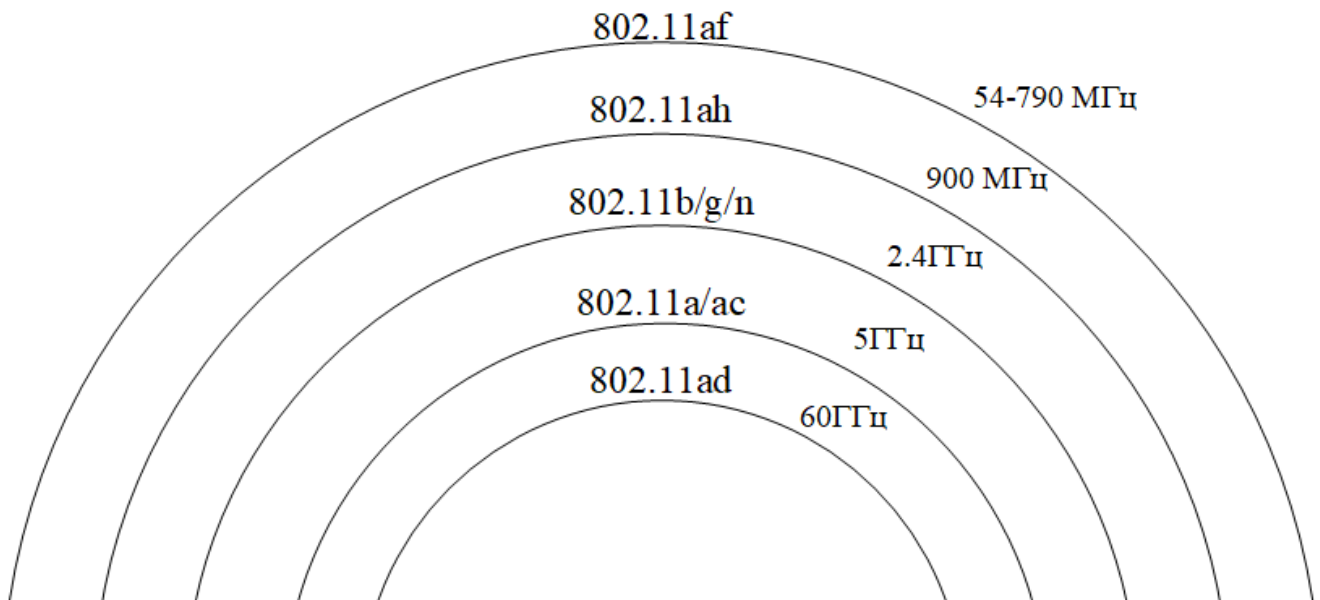


Рисунок 1.2 – Зони покриття версій Wi-Fi

Протокол на частотах у 2.4 ГГц має 14 каналів. Кожен канал має ширину у 20 МГц. При необхідності дозволено використання каналів у 40 МГц. На рисунку 1.5 відображено частоти каналів.

Діапазон дії Wi-Fi маршрутизатора зі стандартом 802.11n і типовою антеною (з ампліфікацією близько 5 дБі) становить приблизно 150 метрів на відкритій місцевості та до 50 метрів у приміщенні. Однак перешкоди, як-от цегляні стіни та металеві конструкції, можуть зменшити цей радіус більш ніж на 25%. Стандарти 802.11a/ac працюють на вищих частотах порівняно з 802.11b/g, що робить їх більш чутливими до перешкод. Крім того, мікрохвильові печі суттєво впливають на зону покриття мереж Wi-Fi стандартів 802.11b та 802.11g. Навіть листя дерев є значною перешкодою, оскільки містить воду, яка поглинає мікрохвильове випромінювання цього діапазону. Наприклад, сильний дощ послаблює сигнал у діапазоні 2,4 ГГц на 0,05 дБ/км, густий туман — на 0,02 дБ/км, а ліс (густе листя, гілки) — на 0,5 дБ/м.

Кожен маршрутизатор працює в певному діапазоні частот. Тобто, Wi-Fi не функціонує суворо з частотою 2,4 або 5 ГГц. Для версій протоколу 802.11b/g/n є діапазон від 2412 МГц до 2484 МГц, в якому виділено 14 каналів (рис 1.3).

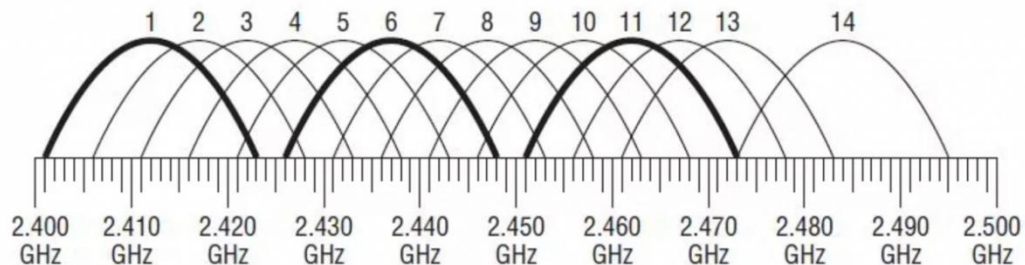


Рисунок 1.3 – Розподіл каналів Wi-Fi на частоті 2.4 ГГц

На частоті 5 ГГц доступні 33 канали, і всі вони є неперекривними. Типова ширина каналів становить 80 МГц, а в деяких випадках може досягати 160 МГц. У Wi-Fi на частоті 2,4 ГГц ширина каналу зазвичай дорівнює 20 МГц (з можливістю розширення до 40 МГц). Це забезпечує вищу пропускну здатність і, відповідно, більшу швидкість передачі даних.

Протокол Wi-Fi підтримує три варіанти топології для організації мережі:

- Епізодична мережа (Ad-Hoc або IBSS — Independent Basic Service Set).
- Базова зона обслуговування (BSS) або Infrastructure Mode.
- Розширена зона обслуговування (ESS — Extended Service Set).

Режим Ad-Hoc — це найпростіший спосіб налаштування локальної мережі, де клієнти обмінюються даними безпосередньо між собою. Така структура ідеально підходить для швидкого розгортання мережі та потребує мінімального обладнання — кожен клієнт має бути оснащений WLAN-адаптером. На рисунку 1.4 відображена топологія Ad-Hoc.

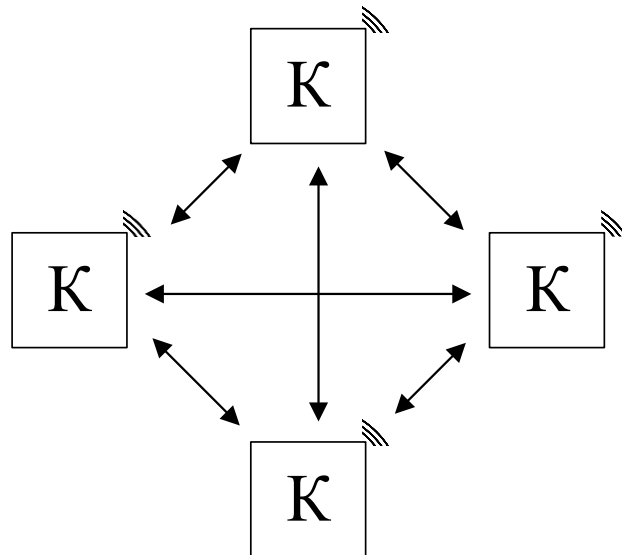


Рисунок 1.4 – Топологія Ad-Нос Wi-Fi мережі

У режимі BSS вузли мережі взаємодіють не напряму, а через точку доступу (Access Point - AP). На рисунку 1.5 відображено топологія BSS.

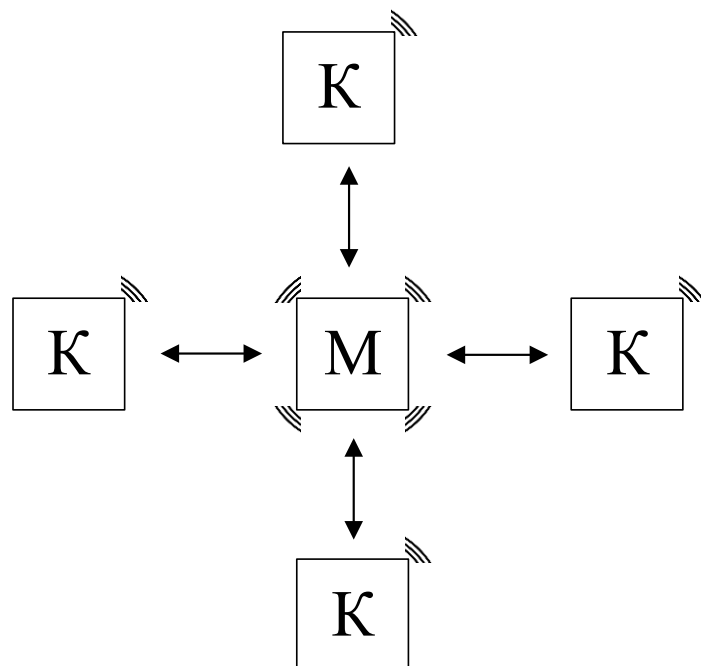


Рисунок 1.5 –Топологія BSS Wi-Fi мережі

Режим ESS дає змогу об'єднати кілька точок доступу, тобто з'єднати кілька мереж BSS. У цьому випадку точки доступу також можуть взаємодіяти між собою. Розширений режим зручно використовувати, коли потрібно об'єднати в одну

мережу численних користувачів або підключити кілька дротових чи бездротових мереж. Топологія ESS представлена на рисунку 1.6.

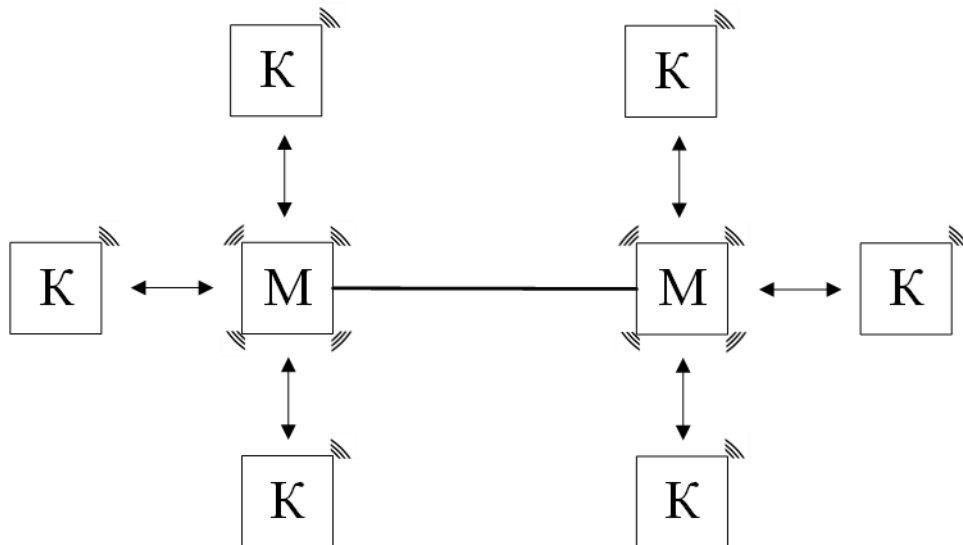


Рисунок 1.7– Wi-Fi топологія ESS

Для адресації вузлів та маршрутизаторів використовується стандарти IPv4 та IPv6. Окрім цього в мережі може функціонувати NAT для відокремлення мережі від зовнішнього доступу.

Для забезпечення захисту у протоколі використовується методи шифрування: WEP, WPA та WPA2. Методи WEP та WPA не є достатньо захищеними для реального використання, через це рекомендується лише використання WPA2 [6].

1.3 Протокол Bluetooth

Протокол Bluetooth — це стандарт для бездротової передачі даних на короткі відстані, представлений у 1994 році розроблений компанією Ericsson [7]. Зараз він керується як відкрита специфікація Bluetooth Special Interest Group.

Bluetooth було задумано як технологію для заміни дротового обміну даними, наприклад, стандарту RS-232. Найбільш поширений приклад його використання — це «бездротовий комп'ютер», що складається з кількох пристроїв: персонального

комп'ютера, ноутбука, клавіатури, миші, джойстика, принтера, сканера тощо, кожен із яких оснащений Bluetooth-модулем. Ці пристрої взаємодіють без кабелів, і Bluetooth забезпечує безперервний зв'язок між ними, фактично замінюючи послідовні та паралельні кабелі й інфрачервоний зв'язок.

У таблиці **Ошибка! Источник ссылки не найден.** відображені характеристики основних версій протоколів [8].

Таблиця 1.2 – Характеристики основних версій протоколу Bluetooth

Версія протоколу	Рік виходу	Максимальна швидкість	Енергоефективна версія
Bluetooth 1.0	1998	768 Кбіт/с	Ні
Bluetooth 1.2	2005	768 Кбіт/с	Ні
Bluetooth 2.0	2004	3 Мбіт/с	Ні
Bluetooth 3.0	2009	3 Мбіт/с	Ні
Bluetooth 4.0	2010	24 Мбіт/с	Так
Bluetooth 4.2	2014	24 Мбіт/с	Так
Bluetooth 5.0	2016	48 Мбіт/с	Так

Останні версії протоколу мають підвищену захищеність та покращену енергоефективність, через що рекомендується використання лише нових версій у пристроях.

Адреса Bluetooth є 48-бітним ідентифікатором, де перші 3 байти вказують на виробника (згідно з правилами IEEE), а останні 3 байти можуть вільно визначатися компанією-виробником. Наприклад, всі адреси Bluetooth-пристроїв Sony Ericsson починаються з 00:0A, оскільки цей номер зареєстровано за компанією в IEEE.

Для обміну даними між пристроями використовується частота 2.4 ГГц [9]. Наявність великої кількості протоколів, що використовують таку частоту, а також мала потужність сигналу зменшує реальну дальність передачі до декількох метрів. Самим стандартом регламентується відстань до 10 метрів.

Для своєї роботи використовується принцип FHSS (Frequency-Hopping Spread Spectrum) [9]. Згідно нього принципу дані під час передачі мають розбиватися на невеликі пакети, які передаються на різних частотах. Це маскує передачу під вигляд шуму для пристроїв, що не мають шаблону за якими обираються частота для передачі.

Для передачі виділено 79 каналів. Вибір каналів визначається псевдовипадковим алгоритмом, за яким формується шаблон передачі. Далі шаблон синхронізується між пристроями. Після синхронізації виконується обмін даними в процесі якого пристрої змінюють частоти передачі. На рисунку 1.8 відображено приклад зміни частоти пристроями під час передачі [10].

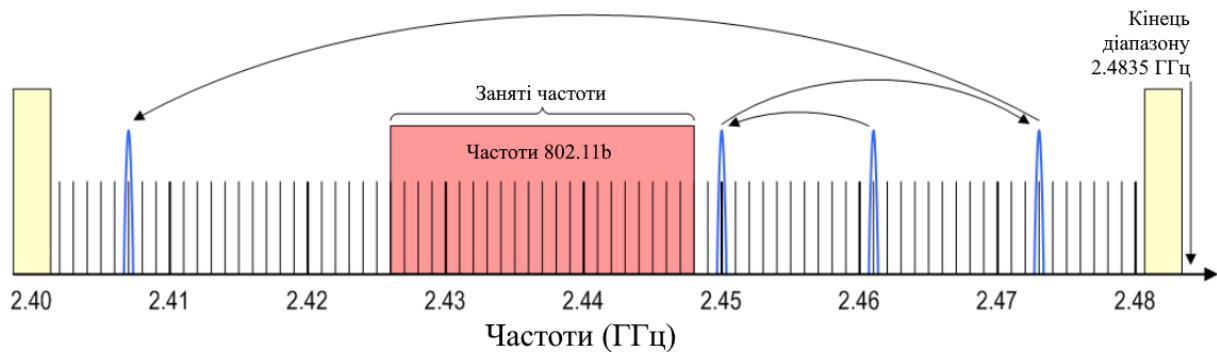


Рисунок 1.8 – Зміна частот у процесі передачі

Усі пристрої мережі розділяються на головні (master) і підлеглі (slave). Обмін даними відбувається лише між головним та підлеглими йому пристроями. Кожен пристрій може виконувати як роль головного, так і підлеглого. На рисунку 1.9 показано зв'язки між головним пристроєм і підлеглими.

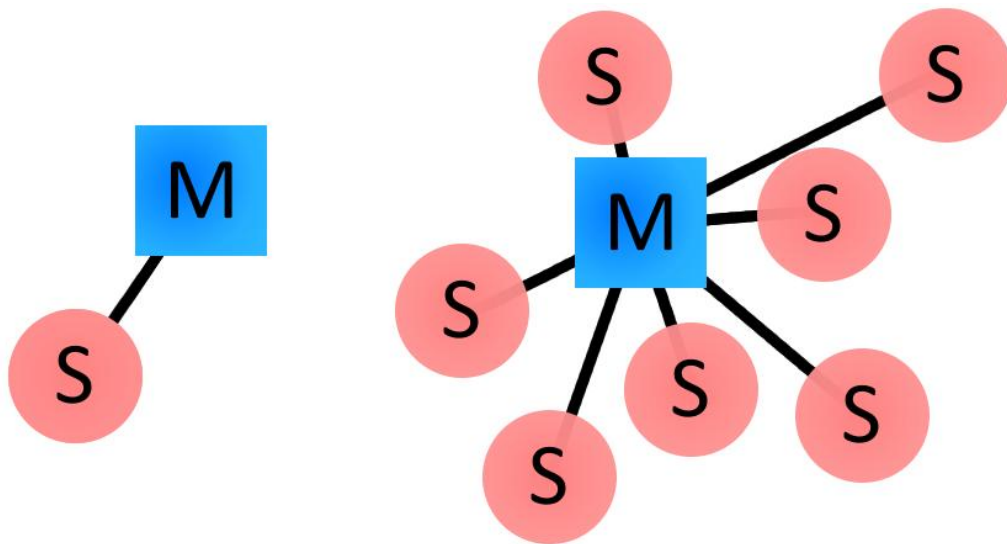


Рисунок 1.9 – Зв'язки між головним та дочірніми пристроями

Мережі загалом поділяються на два типи: пікомережі та розсіяні мережі (рис. 1.10).

Пікомережа:

- В одній пікомережі є лише один головний пристрій, а всі інші — підлеглі.
- Максимальна кількість пристроїв у пікомережі, що можуть одночасно брати участь у передачі даних, становить не більше 8.
- Загальна кількість пристроїв, підключених до головного пристрою в одній пікомережі, не має обмежень.
- У будь-який момент часу обмін даними може здійснюватися лише між двома пристроями в одному напрямку.

Розсіяна мережа:

- Створюється шляхом перекриття кількох пікомереж.
- Кожен пристрій у пікомережі може входити до іншої пікомережі як підлеглий або головний.

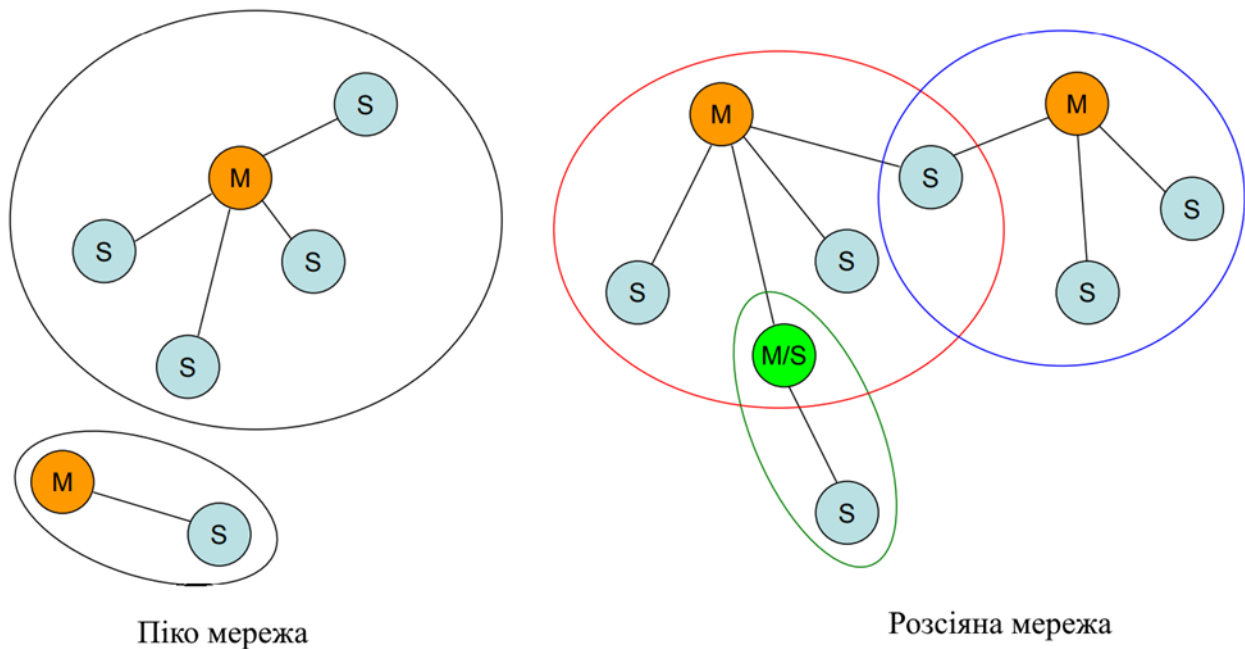


Рисунок 1.10 – Мережі протоколу Bluetooth

Наразі існує два основних типи пристроїв з підтримкою Bluetooth:

Bluetooth Classic (BR/EDR) - використання в бездротових гучномовцях, автомобільних інформаційно-розважальних системах та навушниках, тобто там передаються великі обсяги даних.

Bluetooth Low Energy (BLE), тобто Bluetooth з низьким споживанням енергії, який з'явився у версії стандарту Bluetooth 4.0. Він виступає в пристроях, яким необхідне низьке енергоспоживання (наприклад, у пристроях з батарейним живленням) або в пристроях, що передають невеликі обсяги даних з великими перервами між передачами (наприклад, сенсорні параметри навколишнього середовища або бездротові вимикачі).

Протокол Bluetooth складається з низькорівневого стеку протоколів, що управляють мережею та передають дані.

Протокол керування з'єднаннями (Link Manager Protocol). Відповідає за встановлення підключень між пристроями Bluetooth. Він також охоплює питання безпеки, такі як автентифікація та шифрування, що пов'язані з генерацією ключів шифрування та підключення, а також обміном ключами і їх перевіркою. LMP має вищий пріоритет у порівнянні з іншими протоколами (наприклад, L2CAP), тому

якщо канал зайнятий, він негайно звільняється для передачі повідомлень LMP, коли це необхідно.

Адаптований протокол керування логічними зв'язками (L2CAP) адаптує протоколи верхнього рівня для роботи з Baseband. L2CAP є основним протоколом передачі в Bluetooth. Протокол Baseband забезпечує можливість встановлення з'єднань типу SCO та ACL, проте L2CAP функціонує лише з ACL-з'єднаннями. Багато протоколів і сервісів верхнього рівня використовують L2CAP у якості транспортного протоколу.

Протокол виявлення послуг (Service Discovery Protocol). Визначає який сервіс може бути доступним для використання користувачем. Використовує L2CAP як транспортний протокол.

Завдяки протоколу SDP можна запитувати інформацію про сам пристрій, його послуги та характеристики цих послуг. Після отримання цієї інформації можна встановити з'єднання між двома або кількома пристроями Bluetooth.

Протокол заміни кабелю RFCOMM є одним із протоколів, що використовують L2CAP як транспортний. Цей протокол емулює з'єднання PPP (point-to-point) через послідовний порт (RS-232, більш відомий як COM-порт). Він забезпечує транспортування даних для послуг верхнього рівня, які використовують послідовну лінію як механізм передачі. Завдяки RFCOMM, наприклад, можливо забезпечити доступ до локальної мережі (LAN).

У технології Bluetooth протокол "точка-точка" функціонує "поверх" RFCOMM. З'єднання PPP слугує засобом для переміщення IP-пакетів з рівня PPP у локальних мережах.

1.3 Протокол NB-IoT

NB-IoT (Narrowband IoT) – це протокол вузькосмугової передачі даних для систем Інтернету речей, що працює у частотному діапазоні стільникової мережі LTE [11].

Більшу частину своєї специфікації протоколу отримав від протоколу LTE, що дозволяє розвернути систему на вже існуючому обладнанні стільникових мереж.

Для передачі даних необхідна базова станція, що керуватиме з'єднанням пристроїв у мережі. Завдяки однаковим методам передачі з протоколом LTE можливе використання однієї базової станції для стільникового зв'язку та системи Інтернету речей. На рисунку 1.11 відображено одночасне використання базової станції IoT-пристроями та пристроями стільникового зв'язку [12].

Використання протоколу в мережі LTE дозволяє виконувати передачу на відстань до 50 км, та отримувати в одній мережі до 50 тис. пристроїв.

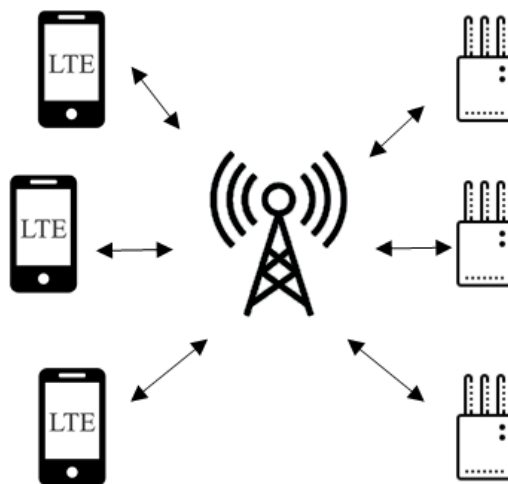


Рисунок 1.11 – Одночасне використання базової станції

Мережа NB-IoT, представлена на рис. 1.12, складається з п'яти основних частин:

Термінал NB-IoT. Кінцеві IoT-пристрої отримують доступ до мережі NB-IoT за умови встановлення відповідної SIM-карти.

Базова станція NB-IoT. Це базова станція, яка вже була розгорнута операторами зв'язку та підтримує всі три типи режимів розгортання, про які йшлося раніше.

Базова мережа NB-IoT. Базова станція NB-IoT підключається до базової мережі NB-IoT, що забезпечує зв'язок із хмарною платформою NB-IoT.

Хмарна платформа NB-IoT. Ця платформа обробляє різні послуги, а результати передаються у вертикальний бізнес-центр або до терміналу NB-IoT.

Сервери обробки даних. Вони можуть отримувати сервісні дані NB-IoT у своєму центрі та контролювати термінал NB-IoT.

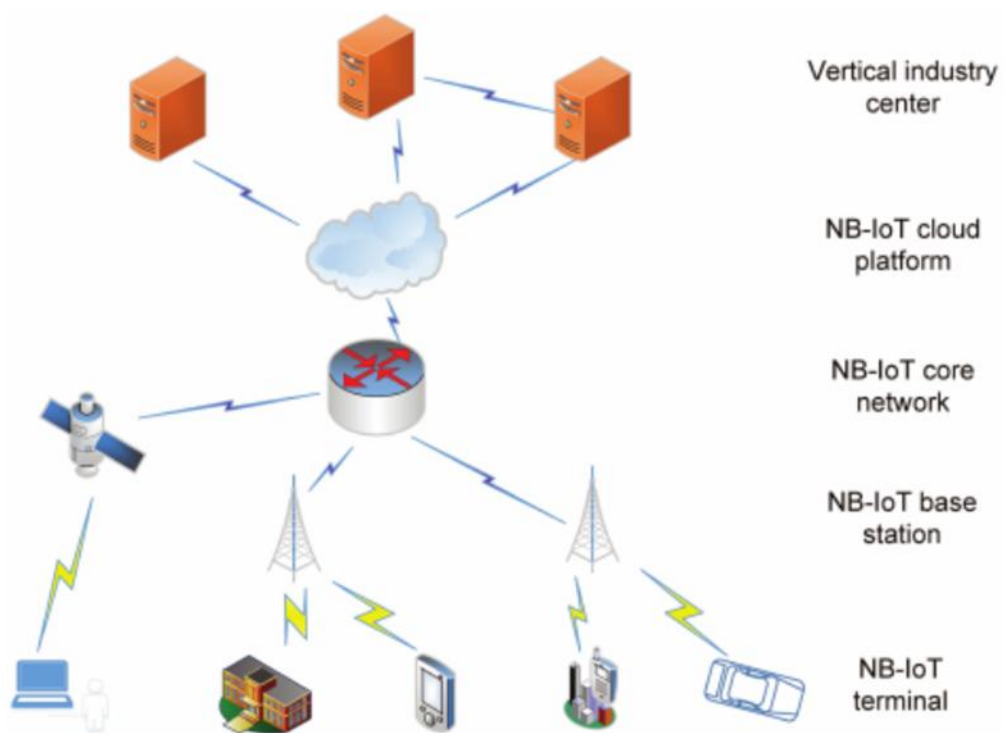


Рисунок 1.15 – Апаратна частина мережі NB-IoT

NB-IoT займає смугу частот 180 кГц, що відповідає одному блоку ресурсів у передачі LTE, забезпечуючи швидкість до 250 кбіт/с. В цьому частотному спектрі протокол може мати до трьох режимів роботи, залежно від розташування блоку в спектрі LTE, як показано на рис. 1.13.

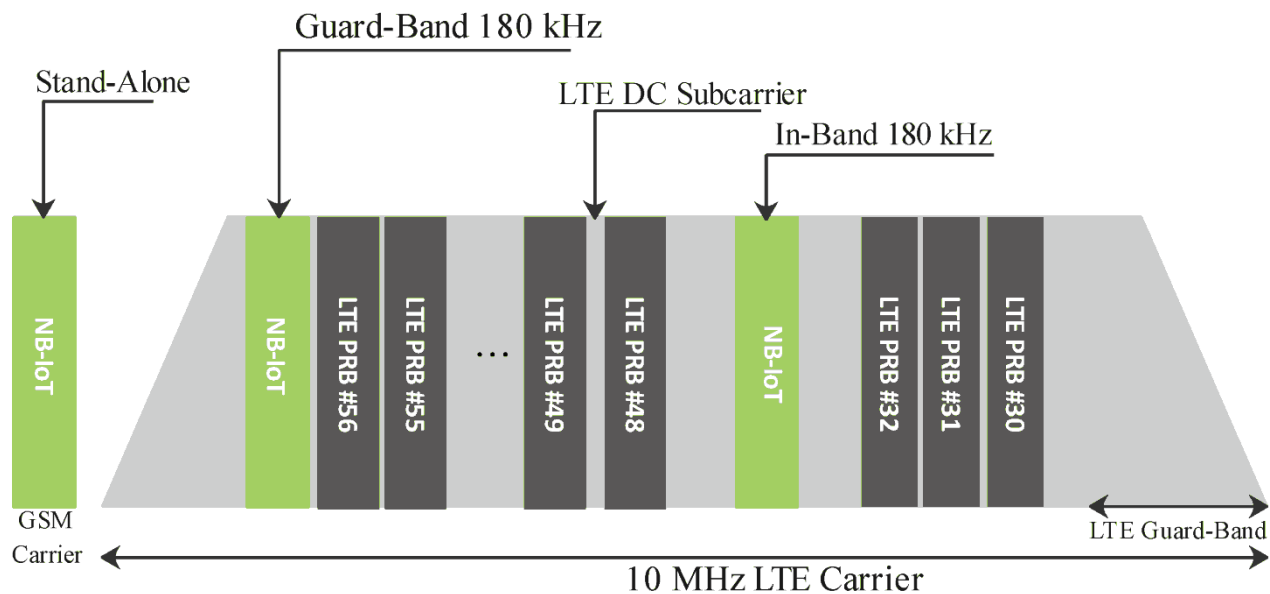


Рисунок 1.13 – Розподіл блоків NB-IoT у частотному спектрі

Більшість цільових методів використання NB-IoT передбачає передачу міні-пакетів, і через складність забезпечення довготривалої та постійної індикації змін якості каналу NB-IoT вводить рівні покриття замість динамічної адаптації зв'язку. Виділяють три класи покриття: нормальне, надійне та екстремальне, що відповідають мінімальним втратам зв'язку (MCL) у 144 дБ, 158 дБ та 164 дБ відповідно. Модуляція, режим кодування та час повторення передачі даних вибираються залежно від класу покриття терміналів, забезпечуючи таким чином напівстатичну адаптацію каналу. Базова станція NB-IoT налаштовує список RSPR з двома пороговими значеннями для розрізнення рівнів покриття.

У таблиці 1.3 відображено порівняння NB-IoT з іншими версіями протоколів, що базуються на протоколі LTE [13, 14W].

Таблиця 1.3 – Порівняння версій протоколів на основі LTE

Протокол	Рік виходу	Максимальна швидкість	Затримки передачі
NB-IoT	2016	250 Кбіт/с	до 10 с
LTE Cat 1	2008	10 Мбіт/с	до 100 мс
LTE Cat 0	2015	1 Мбіт/с	не тестувався

Протокол	Рік виходу	Максимальна швидкість	Затримки передачі
LTE Cat M1	2016	1 Мбіт/с	до 15 мс
EC-GSM-IoT	2016	474 Кбіт/с	до 2 с

1.4 Протокол ZigBee

Протокол ZigBee розроблений альянсом ZigBee, до якого входить багато компаній, таких як Ember, Freescale, Chipcon, Invenys, Mitsubishi, CompXs, AMI Semiconductors, ENQ Semiconductors. ZigBee — це технологія бездротового зв'язку, призначена для мереж на коротких відстанях, із низькою складністю, низьким енергоспоживанням і невеликою швидкістю передачі даних. Дані, які передаються за допомогою ZigBee, можуть включати періодичні, переривчасті дані або дані дублювання з мінімальним часом реакції. Стек протоколів ZigBee містить фізичний рівень (PHY), рівень керування доступом до середовища (MAC), мережевий рівень (NWK), рівень APS і рівень APL. Як показано на рис. 1.14, стандарти PHY та MAC розроблені групою IEEE 802.15.4, тоді як інші рівні — альянсом ZigBee.

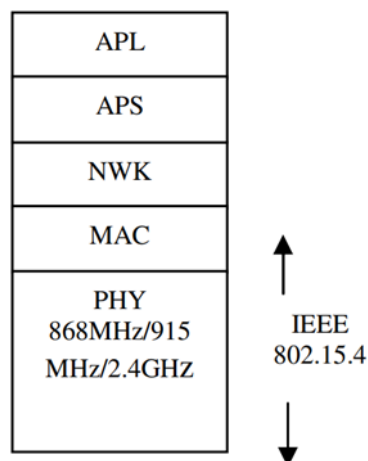


Рисунок 1.14 – Стек рівнів протоколу ZigBee

Мережа ZigBee спеціально розроблена для керування передачею даних. При створенні локальної мережі ZigBee, яка має підключення до віддалених

комп'ютерів через Інтернет, використовують кінцеві вузли з різними зонами покриття, бездротові трансивери та шлюзи обміну даними з інтернетом. Хоча мережа ZigBee є локальною, зона покриття може бути розширена завдяки додатковим маршрутизаторам або за допомогою сітчастої мережі. Залучаючи можливості Інтернету, мобільних мереж та інших каналів зв'язку, кілька локальних мереж ZigBee можна об'єднати для вирішення проблеми мертвих зон.

У мережі ZigBee є три основні категорії пристроїв:

Координатор:

- Відповідає за формування мережі.
- Діє як маршрутизатор при сітчастій мережі.
- Виступає центром довіри у мережі.

Маршрутизатори:

- Керує маршрутизацією у мережі.
- Може бути центром обробки даних.
- Не може переходити у сплячий режим для економії енергії.

Кінцеві пристрої:

- Може обмінюватися даними лише з вузлами вище за рівнями.
- Переходить у режим сну для економії енергії.
- Зберігають активним приймач сигналів, але не передають сигнали.
- Може адаптуватися до зміни розташування та підключатися до іншого сегменту мережі – режим мобільного кінцевого пристрою.

При побудові мережі ZigBee може мати одну з трьох топологій:

Топологія зірка. Це найпростіша та найбільш обмежена топологія в ZigBee-мережах. Вона складається з центрального координатора та інших пристроїв мережі (маршрутизаторів і кінцевих пристроїв) (рис. 1.15). Кожен пристрій у мережі може спілкуватися тільки з координатором. Тому для передачі пакета між двома пристроями він повинен пройти через координатора, який пересилає його до кінцевого пункту.

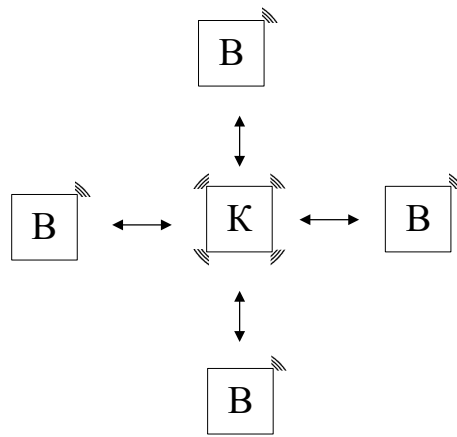


Рисунок 1.15 – ZigBee - топологія зірка

Топологія дерева. У деревовидній мережі координатор ініціалізує мережу та є вершиною (коренем) дерева. Координатор тепер може мати або маршрутизатори, або кінцеві пристрої, підключені до нього (рис. 1.16). До кожного підключеного маршрутизатора можуть підключитися інші маршрутизатори або кінцеві вузли. Кінцеві вузли не можуть підключитися до інших кінцевих вузлів, оскільки вони не мають можливості ретранслювати повідомлення. Ця топологія допускає різні рівні вузлів, причому координатор знаходиться на найвищому рівні. Щоб повідомлення передавались іншим вузлам у тій же мережі, вихідний вузол повинен передати повідомлення своєму батьківському, який вже передаватиме данні необхідному вузлу.

Топологія сітки. Має структуру, що схожа до деревовидної з координатором на вершині дерева. У цій топології координатор з'єднаний зі своїми дочірніми вузлами (маршрутизаторами, кінцевими пристроями), а також може бути пов'язаний з декількома маршрутизаторами та їх кінцевими пристроями (їх дочірніми вузлами) (рис. 1.17).

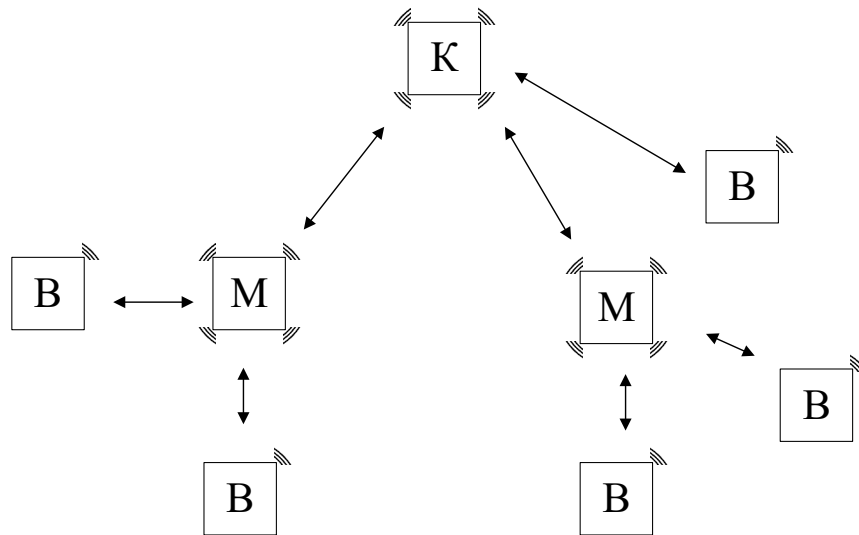


Рисунок 1.16 – ZigBee - топологія дерево

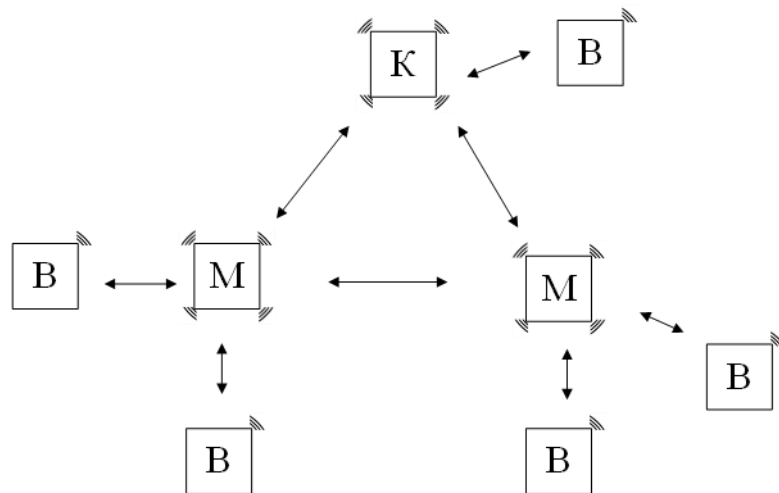


Рисунок 1.17– ZigBee - топологія сітки

ZigBee характеризується дуже низьким енергоспоживанням, зазвичай до 1 мВт або менше. Незважаючи на це, технологія забезпечує радіус дії до 150 метрів на відкритій місцевості завдяки застосуванню методу розширеного спектра з прямою послідовністю (DSSS), який також є енергоефективнішим порівняно з методом стрибкоподібного розширеного спектра (FHSS). ZigBee працює в частотних діапазонах ISM 868 МГц (Європа), 915 МГц (Північна Америка та Австралія) та 2,4 ГГц (глобальний доступ), що забезпечує швидкість передачі даних до 20, 40 та 250 Кбіт/с відповідно. Оскільки ці частотні діапазони не збігаються з діапазонами Wi-Fi, Bluetooth та інших популярних бездротових мереж,

між ними не виникає взаємних перешкод, що сприяє стабільній роботі системи. Стандарт IEEE 802.15.4 підтримує адресацію за допомогою 64-бітних та 16-бітних адрес, що теоретично дозволяє мережі підтримувати понад 65 000 вузлів, а максимальна кількість пристроїв у мережі ZigBee може досягати 653 356. У реальних умовах відстань між пристроями може варіюватися від 5 до 50 метрів.

1.5 Протокол 6LoWPAN

Протокол 6LoWPAN було розроблено на основі IEEE802.15.4 для побудови мережі на основі адресації IPv6 з самостійною організацією маршрутизації у мережі. Стандарт визначає рівні контролю доступу до бездротового середовища (MAC) та фізичний (PHY) при використанні низькошвидкісної бездротової мережі PAN (WPAN) (рис.1.18).

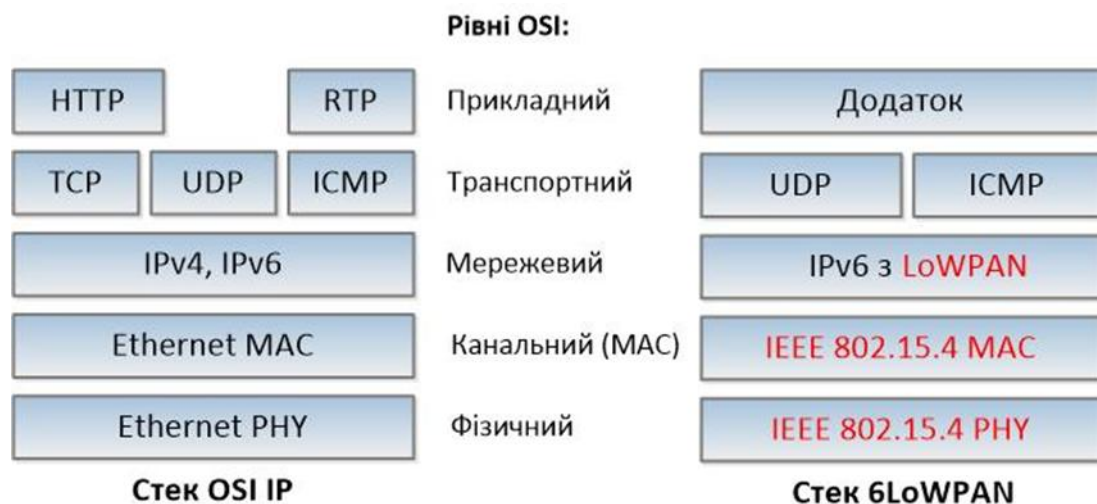


Рисунок 1.18 – Рівні моделі OSI у 6LoWPAN

Характеристики протоколу можуть відрізнятися від країну в якій він застосовується. У таблиці 1.4 зазначено характеристики за територіальним розподілом [15].

Таблиця 1.4 – Характеристика 6LoWPAN за регіонами

Регіон	Діапазон частот, МГц	Кількість виділених каналів	Швидкість передачі даних, кб/с
Європа	868-868,6	1	20, 100, 250
Америка	902-928	10	40, 250
Інші країни світу	2400-2483,5	16	250

На відміну від ZigBee, у 6LoWPAN для ідентифікації вузлів використовується IPv6-адресація. Проте повна підтримка IPv6 у бездротових сенсорних мережах (WSN) обмежена через невеликий розмір кадру, доступний у WSN, що становить 127 байт, тоді як IPv6 має заголовок фіксованого розміру у 40 байт. Це означає, що значні накладні витрати на заголовки залишають менше місця для корисного навантаження в WSN.

6LoWPAN складається з прикордонного маршрутизатора та вузлів датчиків. Прикордонний маршрутизатор є ядром мережі 6LoWPAN, забезпечуючи з'єднання мережі з IP-інтернетом, а також відповідає за маршрутизацію пакетів між 6LoWPAN і IPv6, призначення префіксів IPv6 у мережі. Існують два основні механізми маршрутизації: mesh-under і route-over. Mesh-under забезпечує маршрутизацію на рівні каналу, тоді як route-over використовує мережевий рівень.

Максимальний розмір пакету IPv6 у стандартних мережах становить 1024 байти, тоді як у 6LoWPAN він обмежений до 127 байт, що унеможливорює пряме вписування пакетів IPv6 у кадри IEEE 802.15.4. У мережі IPv6 40 байт зарезервовано для заголовка IPv6, 25 байт для MAC-заголовка, і 8 байт для заголовків UDP, залишаючи лише близько 54 байт для корисного навантаження. Для компенсації цього цільова група з розробки інтернету (IETF) створила адаптивний рівень, який зменшує розмір заголовків IPv6.

У протоколі 6LoWPAN використовуються сітчаста мережева архітектура (рис. 1.19). Ця архітектура включає вузли 6LoWPAN та межовий маршрутизатор, який працює без необхідності використання шлюзового програмного забезпечення,

що дозволяє вільний обмін даними між вузлами. Межовий маршрутизатор виконує роль простого шлюзу.

Підключення до Інтернету забезпечується точкою доступу (АР), яка діє як маршрутизатор IPv6. У типовій конфігурації до точки доступу можуть підключатися різні пристрої, такі як ПК, сервери тощо. У цій топології дальність передачі може досягати 10-15 метрів.

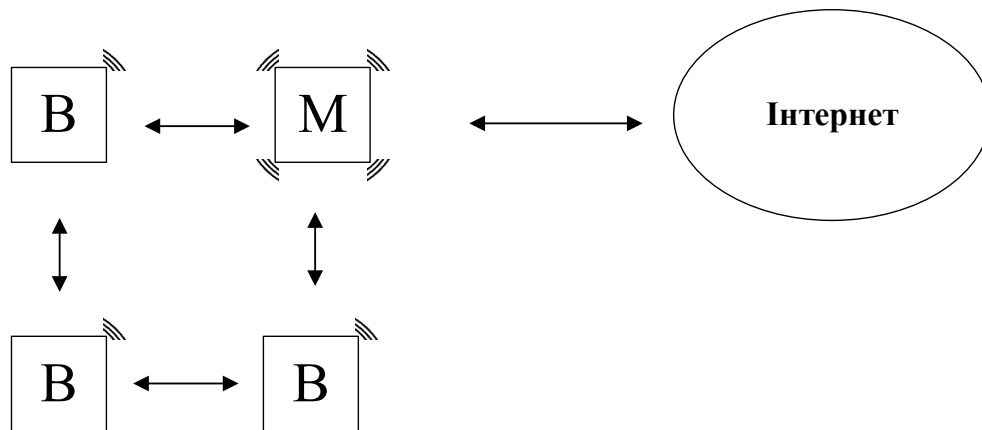


Рисунок 1.19 – Топологія мережі 6LoWPAN

1.6 Протокол WirelessHART

У середині 80-х років американська компанія Rosemount створила протокол Highway Addressable Remote Transducer (HART), який на початку 90-х років був удосконалений і став відкритим стандартом комунікації. Протокол HART є найпоширенішим стандартом для зв'язку з польовими приладами, а його специфікації керуються HART Communication Foundation (CF). Цей протокол розширює аналоговий сигнал від 4 до 20 мА завдяки змодульованій цифровій передачі.

WirelessHART, як додатковий фізичний рівень HART, забезпечує недороге бездротове з'єднання для HART-сумісних пристроїв (рис. 1.20) із відносно низькою швидкістю порівняно з, наприклад, IEEE 802.11g. WirelessHART працює в діапазоні 2,4 ГГц ISM, використовуючи метод множинного доступу з розділенням

часу (TDMA) для координації зв'язку між пристроями з інтервалом 10 мс, утворюючи суперкадр [16, 17].

WirelessHART також дозволяє стрибкоподібно перемикає канали, щоб уникати перешкод і зменшувати ефект завмирання через багатошляхове поширення сигналу. У визначеному слоті може бути запланований обмін даними між одним або декількома джерелами і пристроями призначення, з можливістю спільного зв'язку. HART вільно структурований відповідно до 7-рівневої моделі ISO/OSI для протоколів зв'язку. Впровадження бездротових технологій дозволило HART підтримувати два рівні каналу даних: передача маркерів і TDMA, що обидва працюють із загальним прикладним рівнем HART.

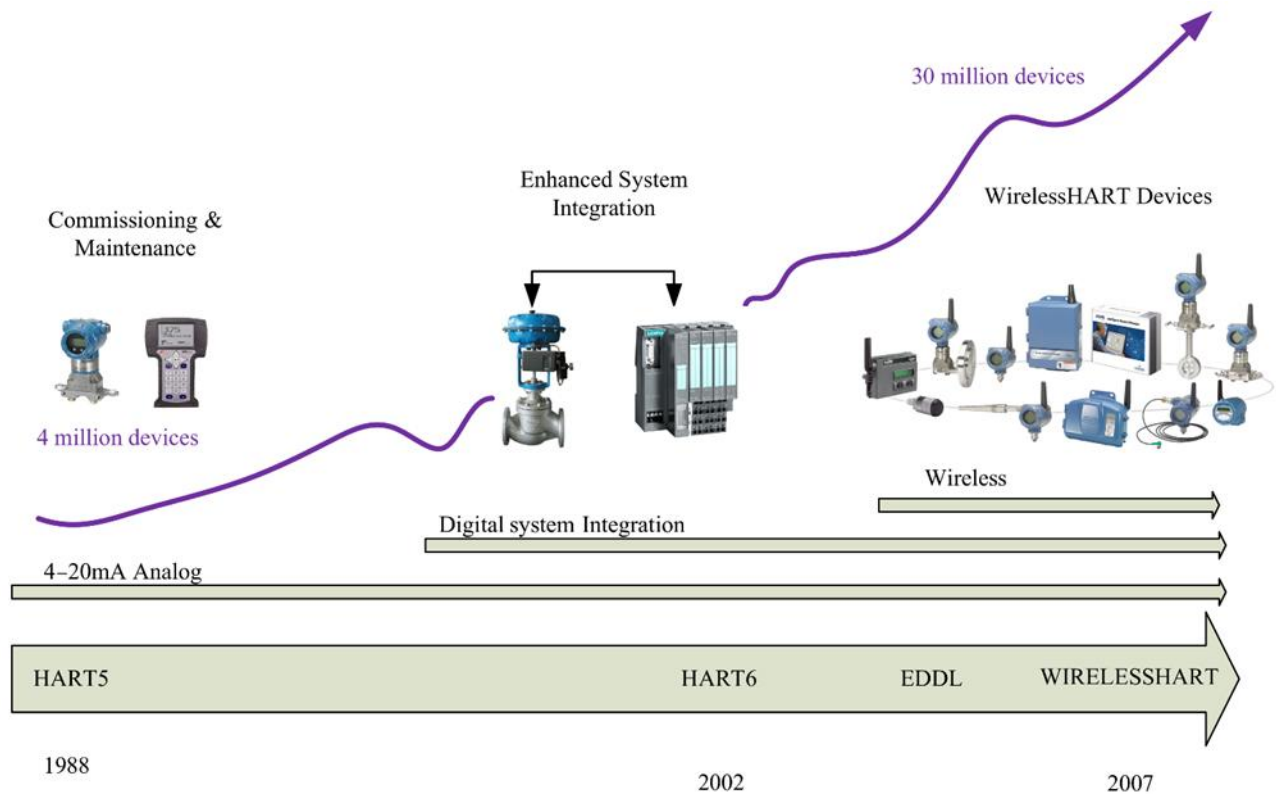


Рисунок 1.20 – Еволюція протоколів сімейства HART

Протокол WirelessHART описує п'ять рівнів моделі OSI із семи. На рисунку 1.21 показано використання рівнів моделі OSI звичайним дротовим протоколом HART та WirelessHART. П'ять рівнів OSI, які задіює WirelessHART, це фізичний

рівень, рівень каналу даних, мережевий рівень, транспортний рівень і прикладний рівень.

Усі аспекти маршрутизації, планування зв'язку та генерації відповідних сигналів управляються центральним менеджером мережі. Цей підхід забезпечує ефективну організацію зв'язку між пристроями в мережі WirelessHART, дозволяючи оптимізувати використання доступних ресурсів і зменшити ймовірність конфліктів у передачі даних.

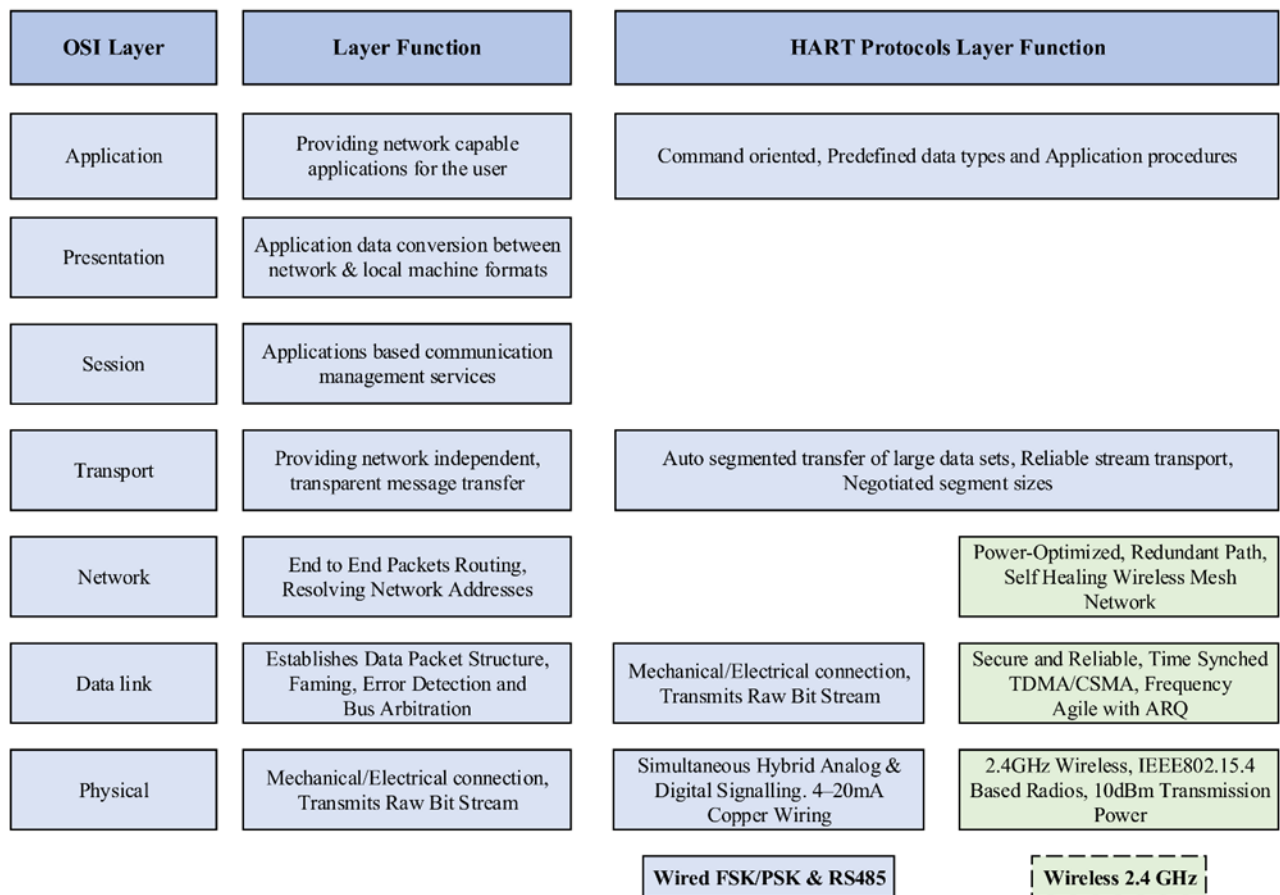


Рисунок 1.21 – Рівні моделі OSI задіяні у WirelessHART

У мережі HART кожен пристрій має унікальну адресу, яка дозволяє ідентифікувати його серед інших пристроїв. Всі підлеглі пристрої мають власні мережеві адреси, що включаються у повідомлення провідного пристрою. Маршрутизатор HART передає адреси всім учасникам мережі, і кожен пристрій

одночасно декодує їх, проте відповідає лише той, чия адреса збігається з отриманою.

Структура адреси в HART може бути короткою (4 біти) або довгою (38 біт). Можна застосовувати комбінації короткої та довгої адрес, причому довга адреса призначається виробником і є незмінною. Існує також альтернативний спосіб адресації на основі користувацьких ідентифікаторів.

Топологія мережі HART зазвичай має вигляд сітки, що забезпечує надійне з'єднання (рис. 1.22) [18].

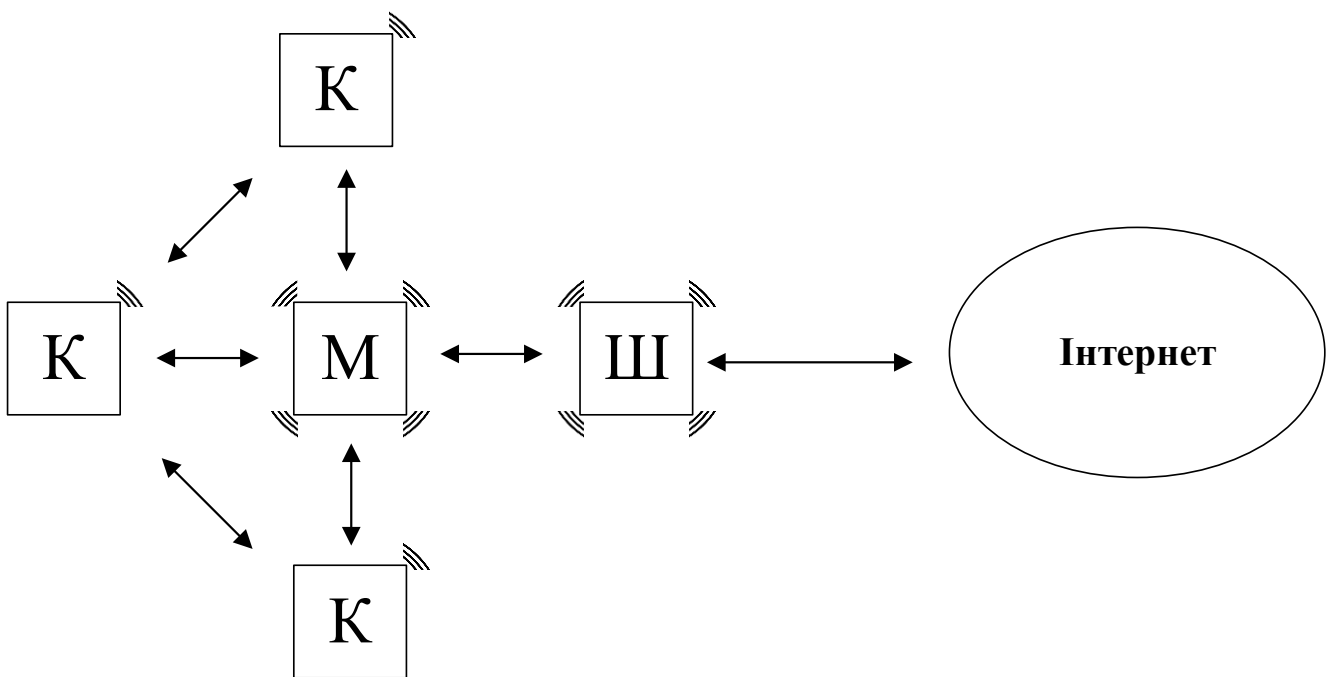


Рисунок 1.22 – Топологія мережі WirelessHART

В компонентах мережі є кінцеві пристрої, маршрутизатори, та шлюз обміну даними з інтернетом оскільки мережа не може напряду взаємодіяти з глобальною мережею без перетворення пакетів даних.

1.7 Протокол Matter

Протокол Matter – це новий стандарт підключення з відкритим кодом для пристроїв IoT, офіційно випущений у жовтні 2022 року [19]. Раніше відомий як Project CHIP (Connected Home over IP).

Ринок розумного дому швидко розвивається, і все більше домогосподарств використовують підключені пристрої для автоматизації своїх домівок. Однак пристрої IoT зазвичай працюють на різних протоколах зв'язку, що значно ускладнює взаємодію в системі розумного будинку.

Протокол розумного дому Matter був створений, щоб вирішити цю проблему, забезпечивши універсальне рішення з відкритим вихідним кодом для пристроїв розумного дому, щоб спілкуватися один з одним, незалежно від виробника [20].

Оскільки всі пристрої значного будинку будуть працювати за технологією Matter, ними можна буде керувати з одного додатка та видалити всі інші. При голосовому управлінні пристроями з підтримкою Matter можна буде використовувати будь-якого голосового помічника: Google Assistant, Alexa або Siri.

Стандарт Matter є прикладним рівнем для пристроїв, що використовують протоколи бездротового зв'язку. Стандарт побудований на IPv6 і використовує UDP і TCP для передачі даних. Оскільки IP є найпоширенішим мережевим рівнем, який клієнти використовують у своїх домівках, Matter став найкращим рішенням для індустрії розумних домівок. Стандарт працює на основі архітектури з нульовою довірою, а саме розширеного стандарту шифрування (AES) 128 для захисту даних користувачів від кібератак, забезпечуючи безпечні оновлення OTA для всіх пристроїв у сітчастій мережі. Matter працює через Wi-Fi та Ethernet (рис. 1.23).

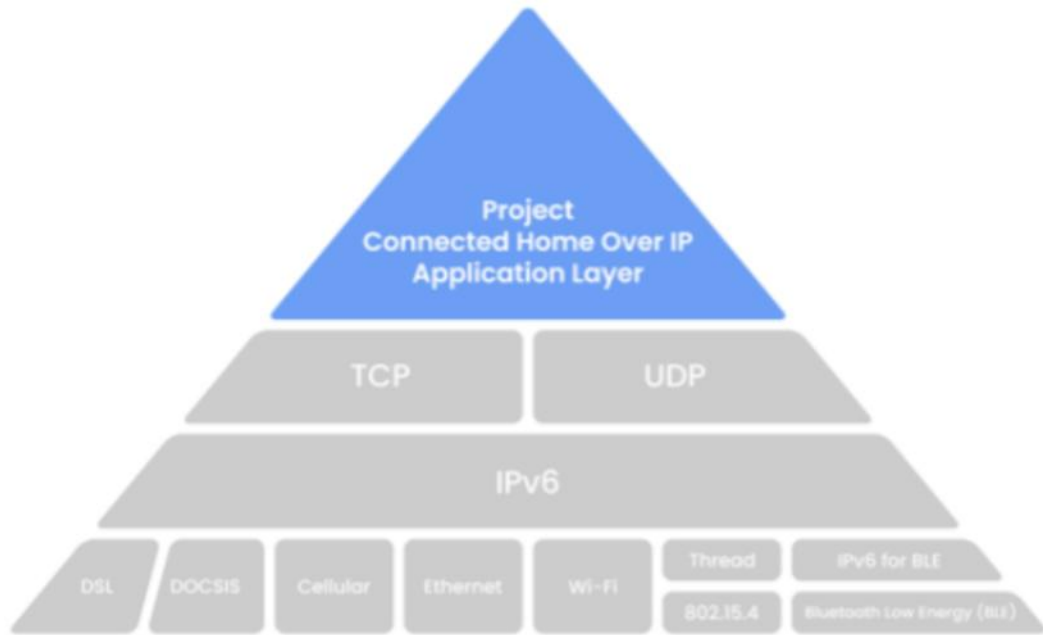


Рисунок 1.23 – Рівні роботи протоколу Matter

Топологія мережі Matter відноситься до структури з'єднань між пристроями Matter і мережами IPv6. Різні мережі IPv6 можуть спілкуватися одна з одною через центральні концентратори.

На рисунку 1.24 зображено з'єднання у змішаній мережі Matter.

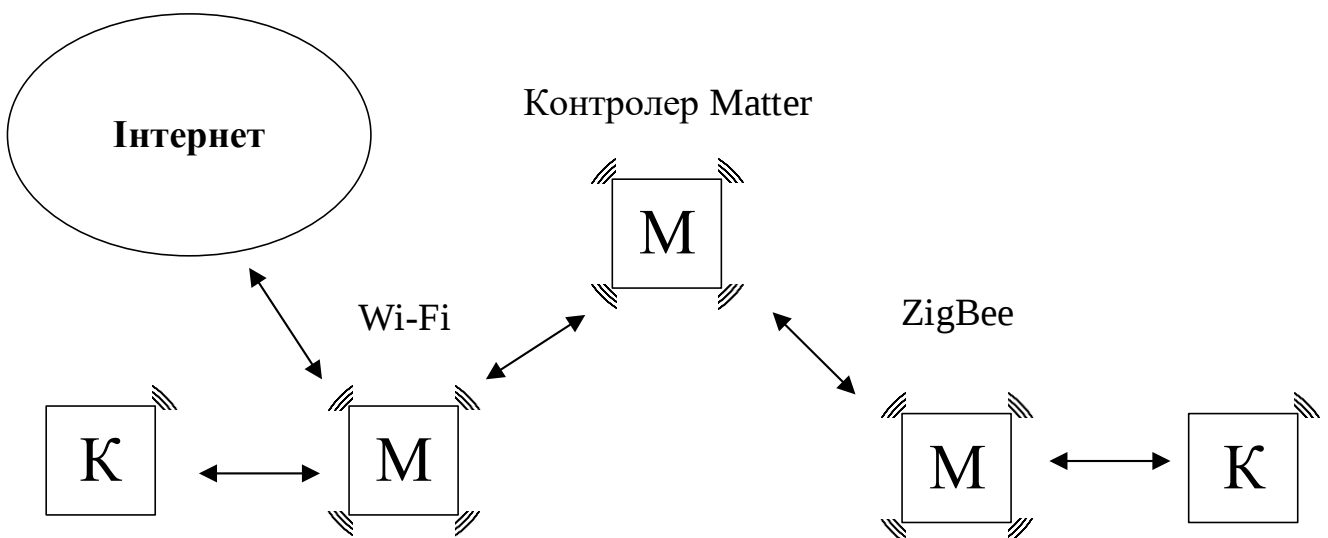


Рисунок 1.24 – Структура мережі Matter

Зв'язок – функція, яка забезпечує встановлення комунікації між кінцевими точками Matter на одному або двох окремих вузлах Matter. Зв'язки описуються прив'язками, які зберігаються постійно в пам'яті пристрою та керуються Binding Cluster. Прив'язки дають змогу призначати цілі для клієнтського кластера вузла, щоб пристрій знав, з яким віддаленим пристроєм необхідно взаємодіяти. Виконання прив'язки визначається додатком і не обмежується Matter Core, що створює можливості для різних сценаріїв. Наприклад, зв'язок між кнопкою вимикача світла та лампочками дозволяє змінювати стан окремих або групових пристроїв.

Міст – мережевий пристрій, який дозволяє підключення пристроїв із мереж, не сумісних з Matter, наприклад, Zigbee, до мережі Matter. Він забезпечує злагоджену роботу та обмін даними між пристроями Matter і пристроями інших мереж, гарантуючи безпеку з'єднання. Міст гарантує безпеку з'єднання між пристроями матерії.

Контролер у мережі Matter – це вузол, який відповідає за сполучення й віддалене керування додатковими пристроями Matter через мережу. Він використовує Bluetooth® LE для первинного налаштування та IPv6 для подальшої взаємодії. Контролер є ключовим елементом у середовищі розробки Matter, надаючи можливість управління підключеними пристроями.

Граничний маршрутизатор – мережевий пристрій, призначений для обміну даними між різними мережами IPv6, сумісними з Matter. Його основна функція полягає в координації зв'язку між мережами, запобігаючи створенню єдиної точки відмови завдяки можливості одночасної роботи кількох маршрутизаторів у мережі. Прикладами граничних маршрутизаторів є межовий маршрутизатор потоку та Wi-Fi точка доступу.

Фабрик – це логічний набір вузлів, які можуть бути частиною різних мереж, але мають спільний корінь довіри та єдиний стан конфігурації. Пристрої всередині фабрика ідентифікуються унікальним 64-розрядним ідентифікатором Fabric, що дозволяє їм взаємодіяти в межах однієї структури.

Мультифабрик – це функція Matter, яка дозволяє одному вузлу Matter бути підключеним до кількох фабрик, кожен із власним адміністратором. Це дає змогу взаємодіяти з пристроями різних мереж і сприяє оперативній сумісності кількох екосистем. Мультифабрик також відомий як мультиадміністратор і важливий для інтеграції різних програм і платформ.

Кінцевий пристрій – це окремий пристрій Matter, який доступний у мережі IPv6 після введення в експлуатацію та оснащений ідентифікатором вузла й обліковими даними безпеки. Завдяки функції Multiple Fabrics, один вузол може бути частиною кількох мереж і мати кілька адміністраторів, відповідно — кілька ідентифікаторів вузла. Пристрої Matter підтримують домашню автоматизацію, а також сумісність з іншими технологіями автоматизації й комунікаційними рівнями. Кожен вузол ідентифікується за 64-бітним ідентифікатором вузла, 16-бітним ідентифікатором постачальника (VID), ідентифікатором продукту (PID) і груповим ідентифікатором (GID), що дозволяє розрізняти їх у структурі.

Постачальник OTA – є вузлом Matter, що відповідає на запити вузла OTA Requestor щодо доступних оновлень програмного забезпечення та надає відповідні пакети під час оновлення по повітрю (OTA).

Запитувач OTA – вузол Matter, який запитує інформацію про доступні оновлення та отримує пакети від постачальника OTA, також обробляючи оголошення про доступних постачальників OTA у мережі Matter.

Фактично усі пристрої, що відповідають вимогам Matter відносяться до мереж інших протоколів. Головна особливість Matter це можливість працювати у режимі мосту між різними протоколами інтернету речей та бути єдиним центром керування. При цьому цей протокол не має власних з'єднань з пристроями і використовує з'єднання інших протоколів. Тому при дослідженні його енергоефективності необхідно дивитися на характеристики протоколів з якими взаємодіє Matter.

1.8 Протокол HTTP

HTTP (Hypertext Transfer Protocol) – це протокол для прикладного рівня для передачі гіпертекстових документів [21]. Він розроблявся для передачі даних від серверу до браузера. Для передачі він використовує стек протоколів TCP/IP для забезпечення підтвердження передачі даних до отримувача.

У роботі протоколу використовується схема «запит-відповідь», за якою клієнт виконує ініціацію підключення до серверу. Після передає URI запит, за яким сервер визначає до яких даних бажає отримати доступ клієнт. Далі сервер відправляє у відповідь дані та закриває активне підключення. Для кожного запиту клієнт має ініціювати нове підключення.

Хоч цей протокол спочатку розроблявся для отримання даних від серверу, але через нього також можна передати дані на сервер. Для цього під час URI запиту додаються дані, який сервер має обробити.

Для визначення дії, що має виконати сервер з даними використовується «HTTP метод». Список методів не є фіксованим, але сервер має в будь-якому випадку відповісти на запит. Якщо сервер не має можливості обробити метод, то він має відповісти з кодом помилки.

У таблиці 1.5 зазначені основні методи роботи з файлами серверу, що наявні у протоколі, та їх характеристики.

Таблиця 1.5 – Методи роботи з файлами у HTTP протоколі

Метод	Опис
GET	За цим методом клієнт запитує від серверу ресурс. Для визначення ресурсу, який має повернути сервер задається використовується URI ідентифікатор, який є шляхом. Додатково можна визначити параметри, які дозволять отримати частину даних, або використовувати параметри як фільтр, при нечіткому визначенню ресурсу.

Метод	Опис
	Важливою частиною цього методу є ідемпотентність, тобто однаковий результат у відповіді при однаковому запиті.
POST	Використовується для передачі даних від клієнта на сервер. Для визначення способу обробки даних використовується URI ідентифікатор. Усі дані для обробки передаються у тілі повідомлення. Після обробки даних сервер має повернути відповідний код результату. При цьому запиті можливе створення нового ресурсу. URI значення ресурсу має бути повернене у якості результату обробки.
PUT	Використовується для завантаження даних на сервер. Для визначення місця розміщення ресурсу задається через значення URI. Сервер має повернути результат в залежності від того чи був створений новий ресурс, чи він був оновлений.
DELETE	Видаляє ресурс з серверу. Ресурс для видалення має бути зазначеним за допомогою URI.
HEAD	Використовується для перевірки наявності ресурсу на сервері за заданим URI. Також може використовуватися для перевірки актуальності ресурсу у кеші клієнта. Може отримувати параметри так само як і метод GET.
OPTIONS	Запит на отримання можливостей Web-серверу. У якості результату сервер повертає список методів, що підтримує сервер та розширення, що підтримуються. При необхідності клієнт може вказати у тілі повідомлення які данні він хоче отримати.
PATCH	Метод схожий на PUT, але оновлює лише частину ресурсу. Для визначення ресурсу використовується URI.

Метод	Опис
TRACE	Використовується для отримання даних, які додають або оновлюють проміжні сервери.
CONNECT	Використовується для формування тунельного з'єднання. Використовується для створення SSL з'єднання.

Оскільки список методів не є фіксованим, то сервер може мати власний набір в залежності від потреб.

Разом з відповіддю сервер має повертати код результату. Для цього є фіксований список наборів кодів. У таблиці 1.6 зазначені набори кодів наявні у HTTP.

Таблиця 1.6 – Коди результатів у протоколі HTTP

Коди	Призначення	Опис
1XX	Інформаційні	Інформація про процес передачі.
2XX	Успіх	Відправляється у разі успішної обробки запиту від клієнта.
3XX	Перенаправлення	Використовується якщо ресурс із запиту був переміщений за новим URI. Нове розташування повертається разом з кодом.
4XX	Помилка клієнта	Якщо клієнт передав не існуючий URI, або якщо один із параметрів не є дійсним.
5XX	Помилка сервера	Якщо сервер під час обробки запиту не зміг правильно його обробити через проблеми у власному алгоритмі обробки.

На рисунку 1.25 відображено структуру пакету при запиті клієнта до серверу [21].

Метод	URI	Версія протоколу
Заголовки		
Роздільник		
Тіло повідомлення		

Рисунок 1.25 – Структура пакету протоколу HTTP

Кожен запит має обов’язково починатися з методу. Далі зазначається URI ресурсу до якого звертається клієнт. Версія протоколу дозволяє визначити які заголовки може мати клієнт та сервер, а також чи підтримує сервер обробку за цією версією.

Заголовки можуть визначити додаткові параметри при обробці запиту. Зазвичай вони визначаються у якому форматі передаються дані між клієнтом та сервером. Також в них міститься інформація про клієнт.

Тіло повідомлення має у собі дані, що передаються. Передаються у вигляді тексту. При передачі бінарних даних також використовується текстовий формат.

При передачі даних до заголовку додається інформація про розмір даних у тілі повідомлення.

При необхідності у заголовках можна передавати самі дані, не використовуючи тіло. Це можливо у випадку розробки власного серверу, який підтримуватиме обробку даних з заголовків.

1.9 Протокол SOAP

Протокол SOAP розроблений для обміну повідомленнями з чіткою структурою опису пакету даних [22]. Розроблений у 1998 році та підтримується консорціумом W3C.

Структура повідомлення не включає у себе опис структури даних при виконанні запиту, та не має структури відповіді на запит. Головне призначення це обмін повідомленнями між клієнтом та сервером. Усе інше визначає розробник під час проектування системи.

Для формування повідомлення використовується мова розмітки XML. Усі дані, що передаються мають власний ідентифікатор. Дані можуть збиратися у об'єкти з ієрархічною структурою даних.

Єдина структура даних дозволяє обмінюватися даними між клієнтом та сервером повідомленнями незалежно від технології на якій вони розроблені. Це дозволяє передавати дані з систем Інтернету речей на віддалені центри обробки даних без додаткової конвертації даних з одного типу в інший.

Структура пакету відображена на рисунку 1.26 [22].

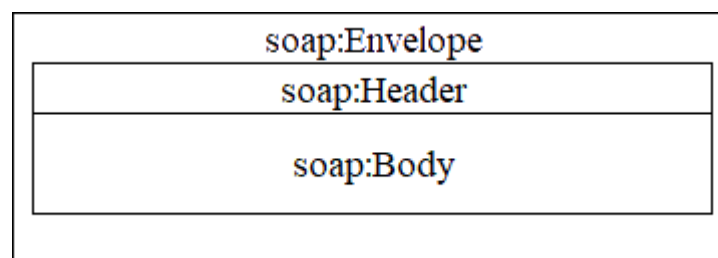


Рисунок 1.26 – Структура пакету протоколу SOAP

`soap:Envelope` – кореневий елемент, який визначає початок повідомлення за протоколом, та є контейнером для заголовку та тіла повідомлення.

`soap:Header` – містить параметри повідомлення такі як: захист та мережеву маршрутизацію.

`soap:Body` – тіло повідомлення, у якому кожне значення має власний ідентифікатор.

Мова розмітки потребує передачу текстових даних. Через це бінарні дані також необхідно представляти у текстовому форматі та слідкувати за наявністю зарезервованих значень у повідомленні.

1.10 Протокол XMPP

XMPP (Extensible Messaging and Presence Protocol) – протокол для обміну повідомленнями у реальному часі [23]. Для формування структури пакету використовує мову розмітки XML.

Спочатку цей протокол розроблявся для обміну повідомленнями у режимі реального часу між серверами та клієнтами. Це дозволяє розробляти власні сервери та об'єднувати їх у глобальну мережу. При цьому можлива інтеграція з серверами, що розроблені за іншим протоколом за допомогою розширень, що використовуватимуть таблиці відповідності облікових записів між різними протоколами.

Оскільки цей протокол також вимагає передачу даних у текстовому вигляді, то бінарні важко адаптувати в підходящій формат для передачі. Для цього або використовують власні рішення, або додають паралельно ще протоколи, за якими буде відбуватися передача даних у бінарному вигляді [23].

Структура пакету відображена на рисунку 1.27.

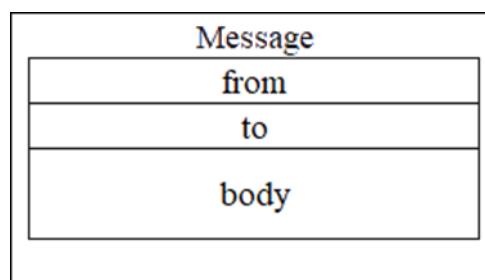


Рисунок 1.27 – Структура пакету протоколу XMPP

Message – головний контейнер пакету. Він визначає початок та кінець повідомлення. Оскільки мова розмітки дозволяє використовувати атрибути в ідентифікаторі, то розробниками було додано два атрибути: **from** та **to**.

from – ідентифікатор відправника пакет. В системі інтернету речей це буде ідентифікатор пристрою.

to – ідентифікатор отримувача. В системі інтернету речей це буде ідентифікатор центру обробки.

body – тіло повідомлення. Складається з наборів даних, що мають власні ідентифікатори. Можуть бути об'єднані у об'єкти при використанні ієрархічної структури.

Наявність ідентифікаторів відправника та отримувача є обов'язковою у протоколі XMPP, оскільки він може використовуватися при обміні даними у об'єднаній мережі. Тоді ідентифікатори допоможуть при маршрутизації пакету у мережі між серверами. Кожен ідентифікатор складається з двох частин. Перша частина вказує на клієнт, друга частина на сервер, за яким цей клієнт закріплено. При відправці даних на віддалений сервер аналізується ідентифікатор отримувача та виконується пересилка на відповідний сервер у мережі Інтернет.

1.11 Протокол STOMP

STOMP (Streaming Text Orientated Messaging Protocol) – протокол обміну текстовими повідомленнями [24]. Для забезпечення універсальності протоколу для обміну даними між клієнтом та сервером використовуються брокери. Вони можуть виступати в ролі мосту між мережами з різними протоколами.

Формат повідомлення схожий на формат протоколу HTTP, але має більш просту форму. Він також використовує текстову форму представлення даних, що накладає обмеження на передачу бінарних даних. Також тіло може бути відкинуто, якщо сервер дозволяє передавати усі дані у межах заголовків.

На рисунку 1.28 відображено структуру пакету протоколу STOMP.

Команда
Заголовки
Роздільник
Тіло повідомлення

Рисунок 1.28 – Структура пакету протоколу STOMP

Команда визначає тип повідомлення, що передається у мережі.

Заголовки є обов'язковими параметрами під час передачі пакету. Список параметрів залежить від команди.

Тіло повідомлення включають у собі дані, що передаються.

У таблиці 1.7 зазначені наявні види команд у протоколі STOMP.

Таблиця 1.7 – Команди у протоколі STOMP

Команда	Призначення
SEND	Використовується для відправки клієнтом даних на сервер. В заголовках мають бути присутніми параметри content-length та content-type.
SUBSCRIBE	Використовується для підписки на розсилку даних. При отриманні даних сервер виконує розсилку даних клієнтам, що виконали підписку на отримання цих даних. Обов'язково має бути зазначено ресурс для підписки.
UNSUBSCRIBE	Відміняє підписку на отримання даних. Обов'язково має бути зазначено ресурс для відміни підписки на нього.
BEGIN	Використовується для ініціалізації транзакції. Ідентифікатор транзакції обов'язково має бути зазначеним.
COMMIT	Підтверджує усі зміни виконані за транзакцію. Ідентифікатор транзакції обов'язково має бути зазначеним.
ABORT	Відміняє усі дії пов'язані з транзакцією. Ідентифікатор транзакції обов'язково має бути зазначеним.
ACK	Підтвердження отримання повідомлення клієнтом від серверу. Має бути зазначено ідентифікатор транзакції та ідентифікатор повідомлення.
NACK	Інформує сервер, що клієнт не отримав від серверу повідомлення.
CONNECT	Ініціалізація підключення клієнта до серверу.
DISCONNECT	Відключення клієнту від серверу та припинення будь-якої взаємодії між клієнтом та сервером.

1.12 Протокол CoAP

CoAP (Constrained Application Protocol) – протокол обміну даними схожий на протокол HTTP [25]. Протокол CoAP також використовує модель REST за якої клієнт робить запити на сервер коли необхідно передати або отримати актуальні дані. При цьому використовується асинхронна передача даних, що дозволяє виконувати актуальну інформацію без підтвердження отримання. Це дозволяє виконувати періодичну відправку даних на сервер не витрачаючи енергію на підтвердження отримання. Якщо дані не мають бути обов’язково актуальними, то така передача збільшує час використання джерела живлення.

При необхідності клієнт може відправити запит з підтвердженням отримання даних сервером.

Структура пакету протоколу CoAP відображена на рисунку 1.29.

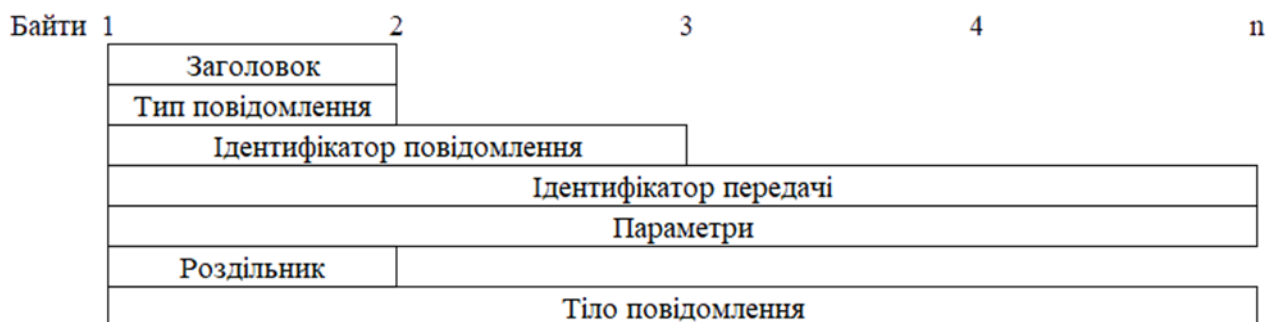


Рисунок 1.29 – Структура пакету даних протоколу CoAP

Структура пакету даних дозволяє передавати як бінарні дані, так і текстові.

Заголовок визначає версію протоколу за якою сформовано пакет даних, тип передачі та довжину ідентифікатору передачі. Тип передачі необхідний для визначення необхідності відправки підтвердження про успішне отримання пакету даних.

Ідентифікатор повідомлення необхідний для визначення повторення у випадку якщо повідомлення було передано декілька раз.

Ідентифікатор передачі визначає пов'язані між собою у рамках однієї сесії. При відправленні відповіді сервер має вказати такий самий ідентифікатор передачі, для того аби клієнт зміг визначати до якого з повідомлень відноситься відповідь. Це необхідно, оскільки протокол асинхронний, через що відповіді можуть бути отримані в різній послідовності.

Параметри – це набір текстових значень з власним ідентифікатором. Аналог заголовків з протоколу HTTP.

Роздільник – один байт зі значенням 255 для відділення параметрів від тіла повідомлення.

Тіло повідомлення зберігає у собі дані для передачі. Можуть передаватися як бінарні значення, так і текстові. При бінарних значеннях необхідно формувати роздільник між даними, який ніколи не зустрінеється у даних, або виділяти під бінарні дані фіксовані об'єми.

1.13 Протокол MQTT та MQTT-SN

MQTT (Message Queuing Telemetry Transport) – це протокол обміну повідомленнями по методу видавник-підписник [26]. У цьому протоколі клієнти це видавники, а сервери обробки це підписники. Сервери очікують дані від клієнтів та оброблюють їх.

MQTT-SN – це версія протоколу для інтернету речей у системі сенсорних мереж.

Між клієнтами та серверами обробки є центр маршрутизації – брокер. Він визначає отримувачів нових даних. Іноді клієнт може стати підписником, якщо має виконувати команди від сервера. У разі необхідності клієнт може поєднувати обидві ролі.

На рисунку 1.30 відображена схема взаємодії у мережі MQTT.

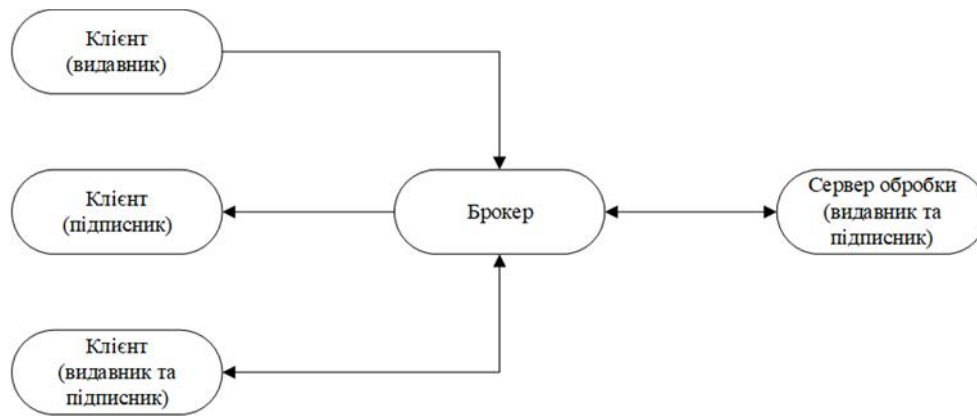


Рисунок 1.30 – Схема взаємодії у мережі MQTT

У якості ідентифікатора даних використовуються теми (Topics). Ідентифікатори тем схожі на URI ідентифікатори ресурсів в протоколі HTTP.

Протокол є бінарним, що дозволяє передавати будь-які дані без додаткової конвертації, як з протоколами, що використовують текстовий формат представлення даних.

На рисунку 1.31 відображено структуру пакету даних для протоколу MQTT.

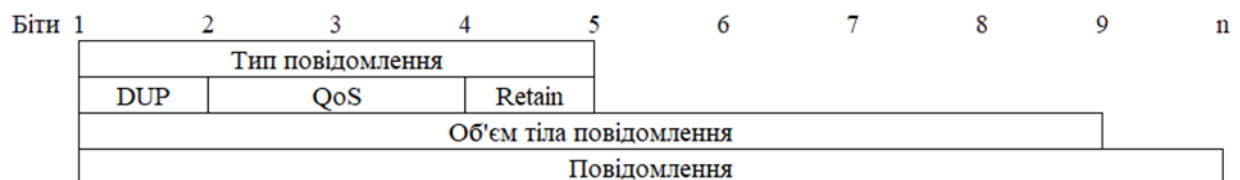


Рисунок 1.30 – Структура пакету MQTT

DUP – встановлюється в 1, якщо пакет є дублем уже існуючих даних.

QoS – рівень підтвердження отримання пакету.

Retain – флаг зберігання пакету у брокері для підписників. Якщо буде новий підписник доданий до системи, то він одразу отримає пакет, що зберігається у брокері.

У таблиці 1.8 зазначені види типу повідомлення та призначення [27].

Таблиця 1.8 – Повідомлення у MQTT

Тип повідомлення	Код	Опис
CONNECT	0001	Запит клієнту на підключення до серверу
CONNACK	0010	Підтвердження вдалого підключення
PUBLISH	0011	Передача повідомлення
PUBACK	0100	Підтвердження передачі
PUBREC	0101	Передачі отримана
PUBREL	0110	Дозвіл на видалення повідомлення
PUBCOMP	0111	Передача завершена
SUBSCRIBE	1000	Запит на підписку
SUBACK	1001	Запит на підписку отримано
UNSUBSCRIBE	1010	Запит на відписку
UNSUBACK	1011	Запит на відписку отримана
PINGREQ	1100	PING запит
PINGRESP	1101	PING відповідь
DISCONNECT	1110	Повідомлення про відключення від серверу

1.14 Висновки до першого розділу

У першому розділі було досліджено наявні протоколи Інтернету речей. Вони були розділені на дві категорії згідно їх призначення: протоколи структури даних та протоколи передачі сигналів.

До протоколів передачі сигналів відносяться: Wi-Fi, Bluetooth, NB-IoT, ZigBee, 6LoWPAN, WirelessHART та Matter. Через те, що Matter не має власних особливостей для передачі сигналів, а використовує мережі Wi-Fi та ZigBee, то він виключений з цього дослідження, оскільки результати його ефективності можна отримати з результатів протоколів Wi-Fi та ZigBee.

Протоколами структури даних є: HTTP, SOAP, XMPP, STOMP, CoAP, MQTT.

Для протоколів передачі сигналів розглянуто характеристики та топології мереж. Для протоколів структури даних розглянуто загальну структуру тіла повідомлення, наявні базові командні дані та коди результатів або помилок.

РОЗДІЛ 2

ДОСЛІДЖЕННЯ ПРОТОКОЛІВ ПЕРЕДАЧІ СИГНАЛІВ

В рамках дослідження було проведено аналіз пакетів даних та наявних методів оптимізації у протоколах передачі сигналів.

2.1 Аналіз протоколу Wi-Fi

На рисунку 2.1 відображено структуру пакету даних Wi-Fi [28].

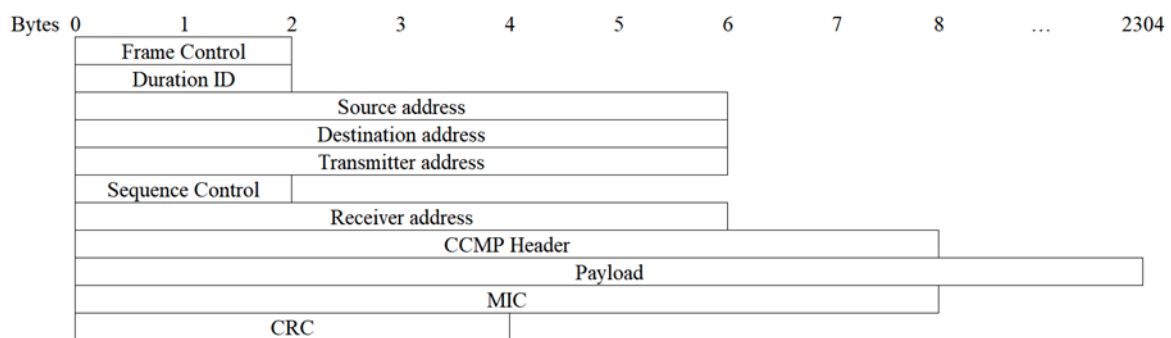


Рисунок 2.1 – Структура пакету даних Wi-Fi

Frame Control – визначає тип кадру, що передається.

Duration ID – може мати інформацію про час резервування базової станції для передачі, або ідентифікатор кадру під час буферизації загального пакету даних.

Source address – адреса відправника.

Destination address – адреса призначення.

Transmitter address – адреса передавача.

Sequence Control – унікальний ідентифікатор пакету для визначення передачі дублікатів.

Receiver address – адреса отримувача.

CCMP Header – параметри шифрування даних у пакету.

Payload – дані пакету.

MIC – контрольна сума даних у пакеті.

CRC – контрольна сума для заголовку та даних у пакеті.

Загальна кількість технічних даних складає 50 байтів. Для даних відведено 2304 байт.

Для оптимізації енерговитрат Wi-Fi використовує такі методи:

- Power Saving Mode (PSM) – це основний режим енергозбереження для пристроїв Wi-Fi, які відповідають стандарту 802.11. У PSM пристрій переходить у стан низького енергоспоживання (сон), коли в нього немає даних для відправки чи отримання, і періодично прокидається, щоб перевірити наявність вхідних пакетів [29]. Пристрій інформує точку доступу (AP), що вона знаходиться в PSM, надсилаючи кадр Power Save Poll (PSP), і AP буферизує пакети, призначені для пристрою, доки не отримає кадр PSP. PSM може заощадити значну кількість енергії, особливо для пристроїв, які мають низький або переривчастий трафік, але він також має деякі недоліки, такі як збільшення затримки, втрата пакетів і перевантаження мережі.

- Стандарт Wi-Fi HaLow. Wi-Fi HaLow, заснований на протоколі IEEE 802.11ah і представлений у 2016 році, працює на частотах нижче 1 ГГц, на відміну від традиційних частот 2,4 ГГц, 5 ГГц і 6 ГГц, які використовуються в Wi-Fi 5, Wi-Fi 6 і Wi-Fi 7 [30].

Низькочастотний діапазон дозволяє Wi-Fi HaLow забезпечувати підключення на великі відстані, підтримуючи з'єднання до трьох кілометрів і далі в сценаріях прямої видимості, а також проникаючи через щільні матеріали своїми вузькосмуговими сигналами. Незважаючи на те, що використання низьких частот призводить до вузьких каналів і меншої пропускної здатності, що забезпечує швидкість передачі даних від 150 Кбіт/с до 86,7 Мбіт/с залежно від відстані.

Wi-Fi HaLow не замінює традиційні стандарти Wi-Fi, а доповнює їх. Він розширює зв'язок на великі відстані без потреби у власних концентраторах, множинних точках доступу чи складних дротових з'єднаннях, що робить його особливо корисним для пристроїв Інтернету речей, проєктів розумного міста та сітчастих мереж.

У США Wi-Fi HaLow працює в діапазоні частот 900 МГц у суб-ГГц-спектрі, що не підлягає ліцензуванню, що дозволяє кожному безкоштовно користуватися. У всьому світі Wi-Fi HaLow може використовувати різні частоти нижче ГГц залежно від регіональних правил і доступного спектру, адаптуючись до різних міжнародних налаштувань.

- Керування частотою передачі – усі версії Wi-Fi до 802.11n (a, b, g, n) включно працюють на частотах від 2400 до 2500 МГц. Ці 100 МГц розділені на 14 каналів по 20 МГц кожен. Оскільки 14 каналів 20 МГц – це набагато більше, ніж 100 МГц, то кожен канал 2,4 ГГц перекривається принаймні двома або чотирма іншими каналами (рис. 1.3) [31].

Канали 1, 6 і 11 розташовані досить далеко один від одного, щоб вони не перекривалися. У налаштуваннях без МІМО (тобто 802.11 a, b або g) необхідно намагатися використовувати канал 1, 6 або 11. При використанні каналів 40 МГц, можливої, що ефір може бути перевантажений.

- Керування потужністю передавача – це технічний механізм, який використовується в деяких мережевих пристроях для запобігання надмірній кількості небажаних перешкод між різними бездротовими мережами [32].

Ідея механізму полягає в тому, щоб автоматично зменшувати використану вихідну потужність передачі, коли інші мережі знаходяться в зоні дії. Зменшення потужності означає зменшення проблем із перешкодами, що надає менше повторних передач, які витрачають енергію.

- Агрегація кадрів. Агрегація кадрів збільшує пропускну здатність, надсилаючи кілька кадрів даних за одну передачу. Це зменшує накладні витрати на протокол 802.11, оскільки кілька пакетів можна надсилати з одним заголовком РНУ і MAC замість того, щоб кожен пакет мав власні заголовки. Кількість АСК і міжкадрових проміжків (і періодів конфлікту, якщо не в ТХОР) також зменшується [33].

У 802.11 є два типи агрегації кадрів:

Агрегація блоку даних служби MAC (MSDU): пакети, отримані MAC від верхнього рівня, є MSDU. Кожен пакет отримує заголовок субфрейму MSDU. Два

або більше субфреймів об'єднуються разом і поміщаються в кадр MAC 802.11 (заголовок + трейлер). Отриманий кадр є агрегатним MSDU (a-MSDU). По радіостанції a-MSDU передаються з одним заголовком PHY.

Агрегація блоків даних протоколу MAC (MPDU): MPDU — це кадри, що передаються від MAC до рівня PHY. Кожен MPDU має MAC-заголовок і трейлер. Кілька MPDU об'єднуються разом, щоб створити сукупний MPDU (a-MPDU), який передається з заголовком PHY по радіо.

2.2 Аналіз протоколу Bluetooth

На рисунку 2.2 відображено структуру пакету даних Bluetooth [34].

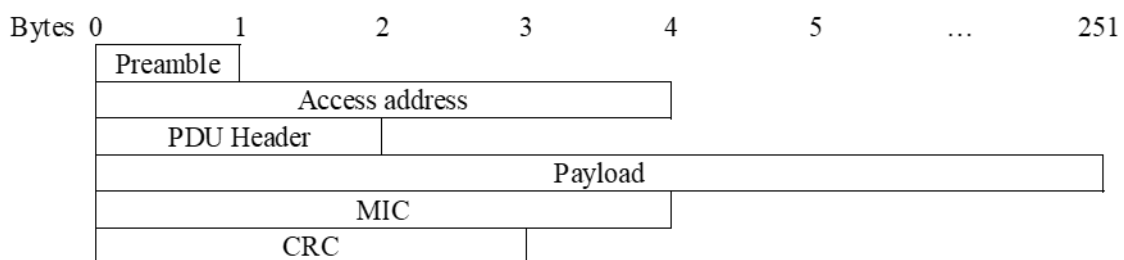


Рисунок 2.2 – Структура пакету даних Bluetooth

Preamble – використовується для синхронізації під час передачі кадру.

Access address – адреса отримувача.

PDU Header – визначає параметри кадру, що передається.

Payload – дані пакету.

MIC – контрольна сума даних у пакеті.

CRC – контрольна сума для заголовку та даних у пакеті.

Для передачі даних необхідно задіяти 14 байтів технічних даних для заголовку і контрольних сум (MAC рівень).

Оптимізація енерговитрат Bluetooth можлива такими методами:

- Режим сну (Sleep Mode) – пристрій переходить у сплячий режим, у якому більшість компонентів вимикається, за винятком радіоприймача, який залишається

частково активним для прослуховування певних тригерів пробудження. Варіанти тригерів для пробудження [35]:

Спрямовані рекламні пакети: пристрій виходить із режиму сну після отримання рекламних пакетів, націлених на його конкретну адресу.

Події підключення: пристрій виходить із режиму сну через заздалегідь визначені проміжки часу, очікуючи події підключення від підключеного пристрою.

Зовнішні події: деяке обладнання може дозволяти пробудження через зовнішні події, як-от натискання кнопок або показання датчиків.

- Режим рідкого пошуку пристроїв (Low Duty Cycle Advertising) – в процесі роботи стандарт розсилає пакети для підтримки зв'язку (реклама) [36]. Інтервал реклами регулюється від 20 мс до 10,24 с (без підключення: мінімум 100 мс). Збільшення рекламного інтервалу може значно зменшити середнє споживання струму пристроєм BLE. Наприклад, збільшення інтервалу реклами зі 100 мс до 1 с знижує середнє споживання струму на 93%.

- Керування потужністю передавача – потужність передачі регулюється від -26 дБм до +8 дБм (за замовчуванням 8 дБм). 0 дБм достатньо, щоб охопити діапазон від 10 до 15 м, на основі тестів, проведених на прикладі iBeacon і телефоні Android[36].

Зміна потужності передачі з 8 дБм до 0 дБм може зменшити споживання струму до 57% при використанні інтервалу реклами 100 мс.

- Керування параметрами пакетів – для оптимізації енерговитрат можна використати агрегацію пакетів даних. Це зменшить час на передачу декількох пакетів, оскільки заголовки потрібно буде передавати лише один раз. Винятком є дані, які обов'язково мають передаватися одразу та не можуть очікувати декілька циклів для агрегації в один пакет. Це можуть бути дані від яких залежить життя людини, або безпеки цілих міст.

2.3 Аналіз протоколу NB-IoT

На рисунку 2.3 відображено структуру пакету даних NB-IoT [37].

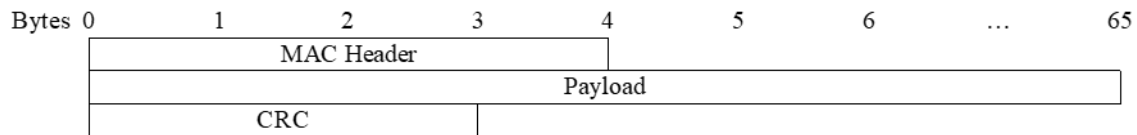


Рисунок 2.3 – Структура пакету даних NB-IoT

Пакет даних NB-IoT має малий розмір технічної інформації відносно Wi-Fi та Bluetooth.

MAC Header – зберігає усі параметри передачі.

Payload – дані пакету.

CRC – контрольна сума для заголовку та даних у пакеті.

Загалом під технічні дані виділено 7 байт.

Оптимізація енерговитрат NB-IoT виконується в такі методи:

- Power Saving Mode (PSM) – розроблений, щоб допомогти пристроям IoT економити заряд акумулятора. Хоча програма пристрою завжди могла вимкнути свій радіомодуль для економії батареї, пристрій згодом доведеться повторно підключити до мережі, коли радіомодуль буде знову ввімкнено.

Процедура повторного підключення споживає невелику кількість енергії, але кумулятивне енергоспоживання повторних підключень може стати значним протягом терміну служби пристрою. Тому термін служби батареї можна продовжити, якщо уникнути цієї процедури. Коли пристрій ініціює PSM з мережею, він надає два бажаних таймера (T3324 і T3412);

Час PSM – це різниця між цими таймерами (T3412-T3324). Мережа може приймати ці значення або встановити інші. Тоді мережа зберігає інформацію про стан, а пристрій залишається зареєстрованим у ній мережі. Якщо пристрій прокидається і надсилає дані до закінчення погодженого інтервалу часу мережі, процедура повторного підключення не потрібна.

- Режим розширеного періодичного оновлення (Extended Discontinuous Reception) – це метод енергозбереження, яка використовується в пристроях Інтернету речей і мінімізує енергоспоживання шляхом періодичного отримання

пакетів даних [38]. Цей підхід особливо корисний для пристроїв, розгорнутих у віддалених районах з обмеженим бездротовим покриттям, оскільки він дозволяє їм підключатися до стільникових мереж, зберігаючи час роботи батареї.

Цикл eDRX - це процес, за допомогою якого пристрої IoT, що використовують eDRX, періодично отримують пакети даних. Цикл складається з двох основних фаз:

Активний період: протягом цієї фази пристрій IoT підключено до мережі та активно отримує пакети даних. Він спілкується з мережею та виконує будь-які необхідні завдання, наприклад надсилання або отримання інформації.

Період сну: після активного періоду пристрій переходить у режим сну, від'єднується від мережі та зберігає енергію. Пристрій залишається в цьому стані до початку наступного активного періоду, після чого він знову підключається до мережі та відновлює прийом пакетів даних.

Тривалість періодів активності та сну може відрізнятися залежно від таких факторів, як потреби пристрою в енергії, умови мережі та конкретні випадки використання. Адаптувавши цикл eDRX до потреб кожного пристрою IoT, можна оптимізувати енергоефективність і продуктивність.

- Адаптивні схеми модуляції та кодування (Modulation and Coding Scheme) – як і в LTE, адаптація зв'язку NB-IoT передбачає адаптивну модуляцію та схеми кодування, а також адаптивний розподіл потужності [39]. Однак схеми модуляції обмежені QPSK, щоб забезпечити низьку складність і, отже, зменшити загальне енергоспоживання. Для розширення зони покриття і підвищення надійності зв'язку введено кількість повторень до 128 разів.

- Керування частотним спектром – масовий NB-IoT є однією з цілей оптимізації мережі 5G. Однак колізії можуть часто виникати в щільних передачах NB-IoT щоразу, коли два таких NB-IoT передають одночасно через той самий канал довільного доступу, що призводить до втрати однієї або обох передач [40].

Використовуючи різні частотні канали можна зменшити шанс того, що відбуватиметься колізія при передачі даних. Для NB-IoT є декілька десятків

частотних каналів, що при не великій кількості пристроїв можуть повністю прибрати виникнення колізій [41].

- Кешування даних – об'єднуючи декілька пакетів даних в один можна економити споживання енергії на з'єднаннях до мережі та на передачі заголовків пакетів даних, оскільки замість кількох заголовків буде лише один.

2.4 Аналіз протоколу ZigBee

На рисунку 2.4 відображено структуру пакету даних ZigBee [42].

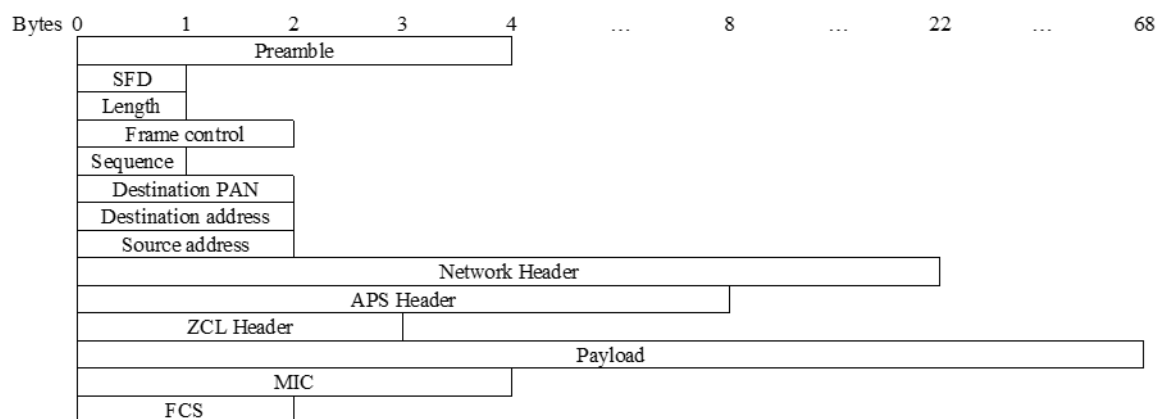


Рисунок 2.4 – Структура пакету даних ZigBee

Preamble – комбінація бітів для синхронізації передавача та приймача.

SFD – роздільник між бітами синхронізації та технічними даними пакету.

Length – довжина кадру.

Frame control – містить основні параметри пакету.

Sequence – зберігає номер кадру при передачі послідовності даних, що була розбита на декілька частин.

Destination PAN – ідентифікатор мережі в яку передається пакет даних.

Destination address – адреса пристрою призначення.

Source address – адрес пристрою відправника.

Network Header – містить заголовки мережі.

APS Header – дані для підтримки роботи програмного забезпечення вказуючи інформацію про профіль і кластер.

ZCL Header – визначає дані для сумісності між пристроями різних виробників.

Payload – дані пакету.

MIC – контрольна сума даних у пакеті.

FCS – контрольна сума для заголовку та даних у пакеті.

Загальний обсяг технічних даних складає 54 байти.

Оптимізація енерговитрат ZigBee виконується в такі способи:

- Режим сну (Sleep Mode) – існує два режими сну: сплячий режим та циклічний сплячий режим.

Сплячий режим використовується для пробудження та засинання модуля за допомогою апаратного контакту. Коли контакт активується, модуль переходить у режим сну після завершення будь-якої передачі чи прийому у процесі. Якщо модуль не приєднався до мережі, він перейде в режим сну після приєднання до мережі [43].

Коли приєднаний кінцевий пристрій прокидається зі сплячого режиму, він надсилає запит опитування базовій станції, щоб дізнатися, чи станція буферизувала дані для кінцевого пристрою. Кінцевий пристрій продовжуватиме надсилати запити опитування кожні 100 мс, поки він залишається в стані активності.

Циклічний сон дозволяє модулям періодично прокидатися для перевірки даних від станції та спати при бездіяльності. У циклічному режимі сну, якщо отримані послідовні або радіочастотні дані, модуль запустить таймер бездіяльності і залишиться в стані неспанья, поки таймер не закінчиться. Поки модуль не спить, він продовжуватиме надсилати запити опитування своїй станції для перевірки буферизованих даних кожні 100 мс. Таймер перезапускатиметься щоразу при отриманні послідовних або радіочастотних даних. Модуль відновить режим сну, коли закінчиться таймер.

- Агрегація та буферизація даних – як і для інших протоколів при агрегації даних надсилається кілька кадрів даних за одну передачу. Це зменшує накладні

витрати на протокол 802.11, оскільки кілька пакетів можна надсилати з одним заголовком замість того, щоб кожен пакет мав власні заголовки.

- Керування потужністю передавача – при побудові мережі можна налаштувати потужність передавачів на пристроях, аби зменшити споживання енергії при цьому не допускаючи занадто слабкого сигналу, що не зможе дістатися до станції.

- Контроль частотними характеристиками - Zigbee використовує 16 каналів (від 11 до 26) у діапазоні 2,4 ГГц по всьому світу, 13 каналів у діапазоні 915 МГц у Північній Америці та один канал у діапазоні 868 МГц у Європі. Деякі пристрої також використовують діапазон 784 МГц у Китаї для Zigbee [44].

У цих каналах кожен пристрій Zigbee використовує смугу пропускання до 2 МГц, тоді як будь-які два різні канали розділені захисною смугою 5 МГц для запобігання перешкодам, спричиненим іншими пристроями Zigbee. Це дозволяє уникнути виникнення колізії при передачі даних.

- Оптимізація маршрутів передачі – ZigBee має основні варіанти мережевої топології: Зірка, Дерево, Сітчата [45]. Налаштовуючи маршрути передачі можна зменшити кількість стрибків у маршруті або зменшити чи повністю прибрати передачу через проміжні вузли в сітчатій топології.

2.5 Аналіз протоколу 6LoWPAN

На рисунку 2.5 відображено структуру пакету даних 6LoWPAN.

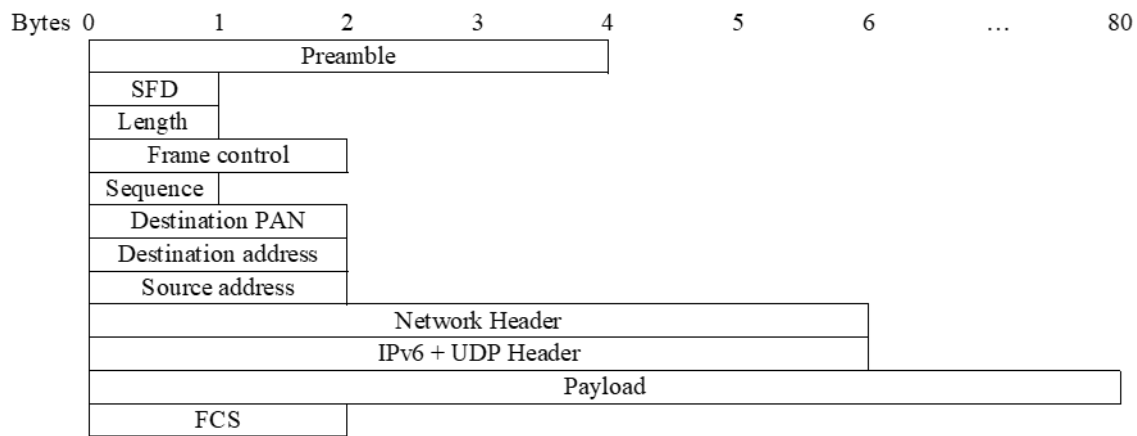


Рисунок 2.5 – Структура пакету даних 6LoWPAN

Preamble – комбінація бітів для синхронізації передавача та приймача.

SFD – роздільник між бітами синхронізації та технічними даними пакету.

Length – довжина кадру.

Frame control – містить основні параметри пакету.

Sequence – зберігає номер кадру при передачі послідовності даних, що була розбита на декілька частин.

Destination PAN – ідентифікатор мережі в яку передається пакет даних.

Destination address – адреса пристрою призначення.

Source address – адрес пристрою відправника.

Network Header – містить заголовки мережі.

IPv6 + UDP Header – адресація у системі IPv6 та заголовки пакету за стандартом UDP.

Payload – дані пакету.

FCS – контрольна сума для заголовку та даних у пакеті.

Оптимізація енерговитрат 6LoWPAN виконується в такі способи:

- Режим сну – щоб зменшити енергоспоживання вузла 6LoWPAN, пропонується механізм доступу до каналу, заснований на включеному стані сну вузла. У цьому механізмі вузли можуть належним чином спати як протягом періоду відстрочки суперкадру, так і періоду спокою суперкадру [46].

- Компресія заголовків IPv6 – 6LoWPAN підтримує стиснення заголовка у системі адресації IPv6 як одну зі своїх функцій для зменшення кількості бітів у заголовку за допомогою методів стиснення [47].

HC1 був першим методом стиснення заголовка IPv6 для 6LoWPAN, запропонованим у RFC 4944 у 2007 році. HC1 є акронімом від Header Compression 1. Замість 40 байтів заголовка IPv6 використовуються 2 байти, які вказують спосіб стиснення заголовка IPv6 і звідки його значення можна відновити під час декомпресії [48].

Техніка стиснення заголовків, яка може стискати локальні, глобальні одноадресні та багатоадресні IPv6-адреси - це стиснення заголовків Інтернет-протоколу (IPHC). Це кодування може складатися з 2 байтів (у локальному зв'язку каналу), 3 байтів (за наявності додаткового контекстного кодування).

S&SFHC - це аббревіатура від другого та наступного стиснення заголовків фрагментів. Цей метод стиснення заголовка пропонується, коли пакети потрібно фрагментувати. В ідеалі заголовок IPv6 повинен передаватись разом із усіма фрагментами, що належать одному пакету. У S&SFHC заголовок з першого фрагмента зберігається в словнику заголовків. Словник заголовків зберігає копію заголовка, отриманого в першому фрагменті. Одержувач надсилає унікальний ідентифікатор посилення (LUI) 8 біт відправнику зі свого списку безкоштовних унікальних ідентифікаторів (FUIList). Відправник замінює заголовок IPv6 цим ідентифікатором у наступних фрагментах. S&SFHC використовує два підходи, тобто автономний та інтегрований. У автономному підході заголовок у першому фрагменті надсилається без будь-якого стиснення. У наступних фрагментах заголовок замінюється стислим заголовком S&SFHC, тобто LUI, який надсилається одержувачем. У інтегрованому підході заголовок у першому фрагменті стискається за допомогою технології IPHC та за допомогою стиснутого заголовка S&SFHC у наступних фрагментах.

- Фрагментація та реасембля – наявні обмеження у 6LoWPAN є однією з головних проблем у методах агрегації даних. Однак традиційний CSMA/CA на рівні MAC може спричинити значні витрати на передачу та керування, а також

затримку під час прослуховування та конкуренції за канали. Це малоефективний спосіб передачі пакету IPv6 через великий заголовок пакета. Рішенням цього є агрегація даних з декількох пакетів в один, якщо це не матиме наслідків через запізнення передачі даних [49].

- Контроль частотними характеристиками – специфікація рівня РНУ визначає, як пристрої 6LoWPAN можуть спілкуватися один з одним через бездротовий канал. Всього на рівні РНУ визначено 27 каналів. Ці канали розподіляються в різних діапазонах частот із різною швидкістю передачі даних [50]. Керуючи розподілом каналів зменшується кількість колізій, що виникають при передачі даних. Це дозволяє зменшити кількість повторних передач, що мають відбуватися при колізії під час одночасної передачі.

- Оптимізація маршрутів передачі – як і ZigBee, протокол 6LoWPAN має підтримку декількох топологій мережі. Це дозволяє і будувати більш гнучкі мережі, і перенаправляти маршрути, аби зменшити навантаження на певні вузли мережі, що використовуються як маршрутизатор.

2.6 Аналіз протоколу WirelessHART

На рисунку 2.6 відображено структуру пакету даних WirelessHART.

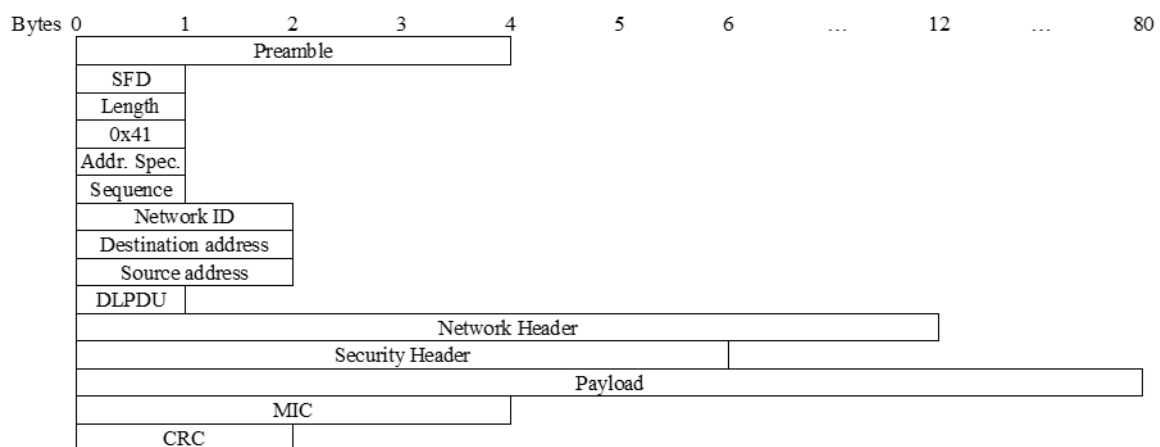


Рисунок 2.6 – Структура пакету даних WirelessHART

Preamble – комбінація бітів для синхронізації передавача та приймача.

SFD – роздільник між бітами синхронізації та технічними даними пакету.

Length – довжина кадру.

0x41 – роздільник пакету даних.

Addr. Spec. – містить параметри адресації пакету.

Sequence – зберігає номер кадру при передачі послідовності даних, що була розбита на декілька частин.

Network ID – ідентифікатор мережі в яку передається пакет даних.

Destination address – адреса пристрою призначення.

Source address – адрес пристрою відправника.

DLDPDU – визначає формат кадру, що передається у мережі.

Network Header – містить заголовки мережі.

Security Header – визначає параметри захисту даних пакету.

Payload – дані пакету.

MIC – контрольна сума даних у пакеті.

CRC – контрольна сума для заголовку та даних у пакеті.

Оптимізація енерговитрат WirelessHART виконується в такі способи:

- Режим сну та активації – щоб максимізувати термін служби батареї, електроніка АББ має наднизьке енергоспоживання, у 20 разів менше порівняно зі звичайним пристроєм HART 4–20 мА [51]. Режим сну дозволяє вимикаючи живлення, якщо вони не потрібно. Таким чином сам датчик вимикається між двома вимірюваннями. Якщо швидкість оновлення досить низька, пристрій якомога частіше впадатиме в «режим глибокого сну» між двома вимірюваннями.

- Керування потужністю передавача – WirelessHART дозволяє регулювати потужність передавача залежно від умов зв'язку та відстані до приймача [52]. Налаштування потужності передавача до мінімально необхідного рівня для забезпечення стабільного зв'язку дозволяє знизити енергоспоживання, особливо в умовах, де пристрої розташовані близько один до одного. За замовчуванням використовується передавач потужністю 10 дБм (10 міліват) [53].

- Частотна сітка – використовуючи Direct-sequence spread-spectrum (DSSS) зі стрибкоподібним перемиканням частоти між 15 каналами в цьому діапазоні

дозволяє забезпечувати безпеку та зменшення перешкод, які вимагали би повторної передачі пакету для його відновлення [53].

- Керування топологією мережі – топологія мережі WirelessHART може бути зіркою, кластером або решіткою, що забезпечує набагато кращу масштабованість. Це також дає можливість керувати розподілом навантаженням між базовими станціями, та обмежувати потужність передавача, що дозволяє зменшити загальні енерговитрати кінцевих пристроїв [54].

- Буферизація та агрегація даних – агрегація даних визначається як процес агрегування даних від кількох датчиків для усунення кількості передач і надання злитої інформації на базову станцію. Агрегація даних зазвичай передбачає об'єднання даних від кількох датчиків у проміжних вузлах і передачу агрегованих даних на базову станцію [55]. існуючі методи агрегації дат не можуть бути безпосередньо застосовані до мережі WirelessHART через її особливості, включаючи багатоканальний протокол синхронізованої сітки (TSMP) і планування слотів зв'язку на основі суперкадрів. Зокрема, агрегація даних зазвичай збільшує наскрізну затримку пакетів, оскільки їм, можливо, доведеться чекати, поки інші повідомлення будуть об'єднані на проміжних вузлах, що потенційно збільшує наскрізну затримку цих пакетів. Однак пакети WirelessHART зазвичай мають суворі вимоги до наскрізної затримки, яка гарантується призначенням часового інтервалу зв'язку в плануванні суперкадрів. Таким чином, ефективність між скороченням енергії та обмеженнями зв'язку в реальному часі є головною проблемою проектування для агрегації даних у мережах WirelessHART.

- Оптимізація маршрутів передачі – при використанні сіткової топології мережі можна формувати більш гнучкі маршрути передачі даних, що зменшувати споживання енергії у кінцевих пристроїв. Це може бути реалізовано через зменшення потужності передавача для передачі на менші відстані, зміну передачі у сітковій мережі з інших вузлів на хаб або на менш навантажені вузли.

2.7 Аналіз розмірів технічних даних у пакетах

Використовуючи інформацію про технічні дані у кадрах кожного протоколу сформовано гістограму їх розмірів (рис. 2.7).

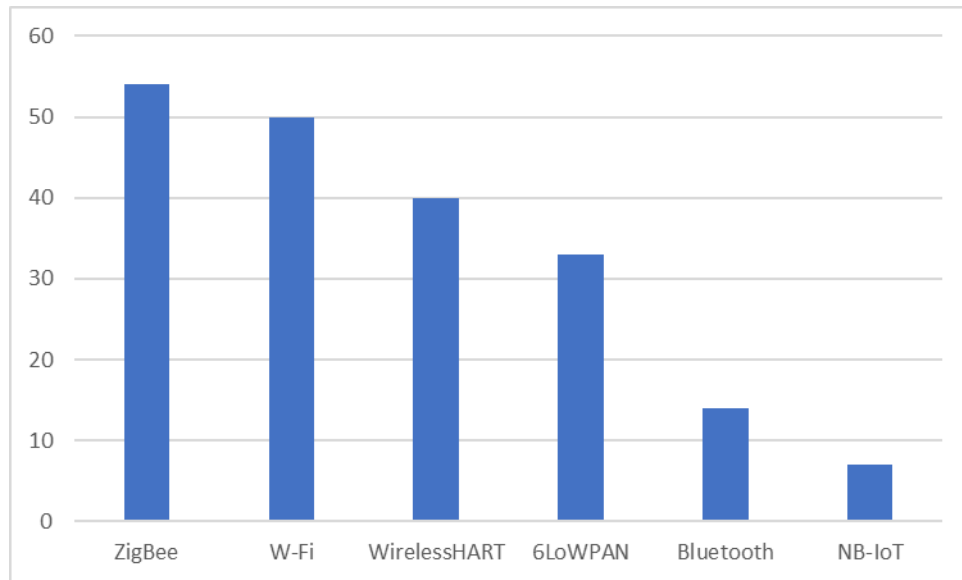


Рисунок 2.7 – Гістограма розмірів технічних даних у кадрах

За результатами гістограми отримано, що ZigBee має найбільший обсяг технічних даних у пакету. На 4 байти має менше Wi-Fi, що на 7.4% менше відносно ZigBee. Найкращі показники має NB-IoT. Обсяги його технічних даних складає 13% відносно розмірів ZigBee [70].

2.8 Аналіз методів оптимізації енергоспоживання

В рамках дослідження було обрано категорії за якими було розподілено кожний метод оптимізації в протоколах Інтернету речей:

- Керування часом активності пристрою;
- Керування частотними характеристиками передачі;
- Керування потужністю передавача;
- Керування пакетами даних;

- Керування маршрутизацією.

Використовуючи інформацію про наявні методи оптимізації енерговитрати було виконано розподілення їх за категоріями та проаналізовано їх вплив на витрати у кожному наявному протоколі та стандарті (табл. 2.1-2.6).

Таблиця 2.1 – Оптимізації за критеріями для Wi-Fi

Категорія	Метод оптимізації
Керування часом активності пристрою	Power Saving Mode (PSM) - до 45-90% [56]
Керування частотними характеристиками передачі	Стандарт Wi-Fi HaLow - до 76% [57]
	Керування частотою передачі - до 71.4% [58]
Керування потужністю передавача	Керування потужністю передавача - до 60% [59]
Керування пакетами даних	Агрегація кадрів - до 49%
Керування маршрутизацією	-

Таблиця 2.2 – Оптимізації за критеріями для Bluetooth

Категорія	Метод оптимізації
Керування часом активності пристрою	Режим сну (Sleep Mode) - до 90% [60]
	Режим рідкого пошуку пристроїв (Low Duty Cycle Advertising) - до 90% [61]
Керування частотними характеристиками передачі	-

Категорія	Метод оптимізації
Керування потужністю передавача	Керування потужністю передавача - до 57.1% [61]
Керування пакетами даних	Керування параметрами пакетів - до 46.7%
Керування маршрутизацією	-

Таблиця 2.3 – Оптимізації за критеріями для NB-IoT

Категорія	Метод оптимізації
Керування часом активності пристрою	Power Saving Mode (PSM) - до 40% [62]
	Режим розширеного періодичного оновлення (Extended Discontinuous Reception) - до 9.4% [63]
Керування частотними характеристиками передачі	Керування частотним спектром - до 25% [63]
Керування потужністю передавача	-
Керування пакетами даних	Адаптивні схеми модуляції та кодування (Modulation and Coding Scheme) - до 25% [64]
	Кешування даних - до 43.7%
Керування маршрутизацією	-

Таблиця 2.4 – Оптимізації за критеріями для ZigBee

Категорія	Метод оптимізації
Керування часом активності пристрою	Режим сну (Sleep Mode) - до 99% [65]
Керування частотними характеристиками передачі	Контроль частотними характеристик - до 37% [66]
Керування потужністю передавача	Керування потужністю передавача - до 70.5% [66]
Керування пакетами даних	Агрегація та буферизація даних - до 49.2%
Керування маршрутизацією	Оптимізація маршрутів передачі - до 100% [67]

Таблиця 2.5 – Оптимізації за критеріями для 6LoWPAN

Категорія	Метод оптимізації
Керування часом активності пристрою	Режим сну - до 99% [46]
Керування частотними характеристиками передачі	Контроль частотними характеристик - до 5%
Керування потужністю передавача	-
Керування пакетами даних	Компресія заголовків IPv6 - до 57.1% [68]
	Фрагментація та реасемблія - до 48.3%
Керування маршрутизацією	Оптимізація маршрутів передачі - до 100%

Таблиця 2.6 – Оптимізації за критеріями для WirelessHART

Категорія	Метод оптимізації
Керування часом активності пристрою	Режим сну та активації - до 99% [69]
Керування частотними характеристиками передачі	Частотна сітка - до 3% [70]
Керування потужністю передавача	Керування потужністю передавача - до 33% [69]
Керування пакетами даних	Буферизація та агрегація даних - до 48.8%
Керування маршрутизацією	Оптимізація маршрутів передачі - до 100%
	Керування топологією мережі - до 100%

Протоколи ZigBee та WirelessHART мають методи оптимізації за всіма наявними категоріями. Протоколи Wi-Fi, 6LoWPAN мають по 4 категорії методів оптимізації. Bluetooth та NB-IoT мають по 3 категорії оптимізації. Але при цьому ці протоколи мають декілька методів оптимізації, що відносяться до однієї категорії, що в цілому вирівнює їх показники загальної оптимізації енергоспоживання. Найкращі показники у методів оптимізації, що відносяться до критерію «Керування часом активності пристрою», що зумовлено повним відключенням пристрою на певні проміжки часу, поки не настає момент збору даних та відправки їх до центру обробки [71, 72].

2.9 Висновки до розділу

У другому розділі були проаналізовані протоколи передачі сигналів. Для кожного з них визначено типовий вигляд пакету даних, що дозволить провести аналіз впливу технічних даних на загальну енергоефективність під час передачі пакету інформації до центру обробки базуючись на обсягах даних.

Також було проведено аналіз наявних моделей та методів оптимізації енерговитрат для протоколів передачі сигналів. Наявні методи були розбиті за категоріями їх призначення: керування часом активності пристрою; керування частотними характеристиками передачі; керування потужністю передавача; керування пакетами даних; керування маршрутизацією.

Протоколи ZigBee та WirelessHART мають методи оптимізації за всіма наявними категоріями. Протоколи Wi-Fi, 6LoWPAN мають по 4 категорії методів оптимізації. Bluetooth та NB-IoT мають по 3 категорії оптимізації [71, 72].

.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ ПРОТОКОЛІВ СТРУКТУРИ ДАНИХ

3.1 Формула розрахунку об'єму даних

Для отримання числових значень об'ємів даних, що передаватимуться у системі, та відсортувати їх за типами. Якщо у системі більшість, або усі дані представлятимуть цілі числа, то можна відкинути протоколи зі строковим форматуванням даним. Цілі дані найкраще передавати у бінарному вигляді.

Якщо необхідна передача послідовності строкових команд, то можна розглянути протоколи, що мають найменший обсяг додаткових символів для кодування ідентифікатора даних та мінімальну кількість технічних даних.

Відкинувши протоколи, що не підходять за типами даних, необхідно сформувати пакети даних, що зазвичай будуть передаватися у системі та розрахувати їх обсяги. На основі цього, будуть отримані числові значення об'ємів.

Також до цього розрахунку можна додати додатковим коефіцієнтом частоту передачі пакету даних. Якщо у системі не має чітко визначено типу даних, то частотний коефіцієнт дозволить підвищити точність розрахунку. Можливо, що текстові дані будуть передаватися частіше, ніж числові, що потребує більш зручної передачі, ніж для чисел. Або навпаки.

У роботі [73] приведено аналіз розрахунку пакету даних для деяких протоколів структури даних. Для аналізу приведено приклад повідомлення, розрахунок об'ємів повідомлення та порівняння отриманих об'ємів даних.

Після необхідно виконати розрахунок пакетів даних протоколів передачі сигналів. Кожен з протоколів має власну структуру пакету під час передачі. В них міститься технічна інформація та корисні дані.

Технічна інформація – це данні, що визначають протокол за яким відбувається передача та параметри, що необхідні для передачі за даним протоколом.

Розрахунок пакетів даних виконується таким самим чином, як і протоколи структури даних. Формуються пакети даних, що передаватимуться у системі та розраховується їх розмір. У якості корисних даних необхідно вказати дані з протоколів структури даних.

У деяких випадках пакети розбиваються на кадри. Для кожного кадру дублюється технічна інформація. Розмір кадру залежить від протоколу. При малому розмірі кадру можлива ситуація, що протокол, у якому більший розмір технічної інформації та з більшим розміром кадру, буде передавати меншу кількість даних, оскільки кадр буде всього один.

Для виконання розрахунку числових значень об'ємів передаваних даних необхідно розрахувати суму даних протоколів передачі сигналів та протоколів структури даних.

На основі цього твердження отримана формула:

$$V_i = V_{ip} + V_{id} \quad (3.1)$$

де V_{ip} – обсяг даних протоколу передачі сигналів; V_{id} – обсяг даних після кодуванням протоколом структури даних.

Обсяг даних протоколу структури даних розраховується за формулою (3.2).

$$V_{ip} = \left\lceil \frac{V_{id}}{\Delta V_{ip}} \right\rceil V_{ipt} \quad (3.2)$$

де V_{id} – обсяг даних після кодуванням протоколом структури даних; ΔV_{ip} – об'єм даних на кадр; V_{ipt} – обсяг технічних даних на кадр.

За формулою **Ошибка! Источник ссылки не найден.** визначається кількість сум даних для протоколу структури даних. Оскільки дані можуть бути розбиті на фрейми, то необхідно визначати об'єм даних з урахуванням декількох фреймів. Тому спочатку визначається кількість фреймів. Після виконується округлення у більшу сторону. Це необхідно через те, що кількість фреймів не може мати дробову

кількість. Далі розраховується вже кількість даних через добуток кількості фреймів на обсяг технічних даних у кадрі.

Загальний обсяг даних для передачі формується з базових даних протоколу структури даних та самих даних. Самі дані зазвичай мають ідентифікатор розмір якого має також урахуватися. В залежності від протоколу можуть бути наявні додаткові символи для різних типів даних.

Виходячи з зауважень отримана формула обсягу структури даних:

$$V_{id} = \Delta V_{id} + \sum_{j=0}^m \Delta V_B + \Delta V_I + \Delta V_D \quad (3.3)$$

де ΔV_{id} – об'єм технічних даних; ΔV_B – об'єм технічних даних для типу даних; ΔV_I – розмір ідентифікатору даних; ΔV_D – об'єм даних.

3.2 Формування пакету даних для передачі

Для виконання порівняльного аналізу необхідно сформувати пакет даних, що передаватиметься у системі. З пакету даних сформувати пакети для передачі за кожним з протоколів.

Оскільки заздалегідь не відомо які саме дані будуть передаватися у мережі, тому для порівняння необхідно використовувати різні види пакетів даних.

Через невизначеність даних, що передаватимуться у реальних мережах необхідно використовувати пакети даних, що теоретично можуть передаватися у мережі, та на основі їх проводити аналіз об'ємів трафіку для передачі у мережі.

Для аналізу було сформовано три пакети даних. Перший пакет даних використовує для передачі лише числові дані. Другий пакет містить текстові дані. Третій пакет використовує змішаний варіант даних з використанням числових та текстових даних.

У таблицях 3.1 – 3.3 відображено сформовані пакети даних.

Таблиця 3.1 – Пакет даних для першого пакету

Тип даних	Ідентифікатор	Значення
Ціле число	number1	60
Ціле число	number2	2000000000
Ціле число	number3	384

Таблиця 3.2 – Пакет даних для другого пакету

Тип даних	Ідентифікатор	Значення
Текст	line1	action start
Текст	line2	Kherson National Technical University
Текст	line3	command end transition

Таблиця 3.3 – Пакет даних для третього пакету

Тип даних	Ідентифікатор	Значення
Ціле число	number1	384
Текст	line1	Kherson National Technical University
Текст	line2	status info
Ціле число	number2	2000000000

3.3 Формування тіла повідомлення

Кожен з протоколів структури даних має або точний опис формату за яким необхідно розміщувати дані, або рекомендації по формуванню пакету, якщо протокол контролює передачу бітових даних.

Спершу за кожним протоколом формується тіло повідомлення. Тіло повідомлення – це корисні дані, що передаватимуться за обраним протоколом та мають чітко визначену структуру, за якою можна знайти дані маючі ідентифікатор цих даних.

За протоколами HTTP та STOMP тіло повідомлення першого пакету даних має наступний формат:

number1:15

number2:11

number3:2004

Для другого пакету:

line1:action start

line2:Kherson National Technical University

line3:command end transition

Для третього пакету:

number1:384

line1:Kherson National Technical University

line2:status info

number2:2000000000

Перехід на наступний рядок визначає, що між даними та наступним ідентифікатором у протоколі SOAP використовується символ переходу на новий рядок «\n». Протокол HTTP для розділення даних та наступного ідентифікатору використовує комбінацію символів «\r\n».

Для протоколів XMPP та SOAP тіло першого пакету має вигляд:

<number1>60</int1>

<number2>2000000000</number2>

<number3>384</number3>

Для другого пакету тіло матиме вигляд:

<line1>action start</line1>

<line2>Kherson National Technical University</line2>

<line3>command end transition</line3>

Тіло третього пакету має вигляд:

```
<number1>384</number1>
<line1>Kherson National Technical University</line1>
<line2>status info</line3>
<number2>2000000000</number2>
```

У протоколах CoAP, MQTT та MQTT-SN можливий обмін бінарними даними. Для опису тіла використовуватимуться шістнадцятирична система числення.

У протоколі CoAP для першого пакету даних тіло матиме вигляд:

```
00 00 00 3C 01 77 35 94 00 02 00 00 07 D4
```

Другий пакет даних:

```
line1:action start
line2:Kherson National Technical University
line3:command end transition
```

Третій пакет даних:

```
number1: 00 00 3C 01
line1:Kherson National Technical University
line2:status info
number2: 77 35 94 00
```

Для першого пакету ідентифікатори даних були замінені на інші. Це зменшує об'єм даних при передачі лише числових даних.

У таблиці 3.4 зазначені нові ідентифікатори для даних у першому пакеті даних протоколу CoAP. Нові ідентифікатор використовують числову форму запису.

Таблиця 3.4 – Нові ідентифікатори для бінарної форми запису

Ідентифікатор	Новий ідентифікатор
number1	00
number2	01

Ідентифікатор	Новий ідентифікатор
number3	02

Між даними та ідентифікатором використовується символ переходу на нову строку «\n» для розділення корисних даних та наступним ідентифікатором.

Протоколи MQTT та MQTT-SN використовують бінарне кодування для даних. Через це формат пакету визначається під час розробки системи. Усі дані будуть закодовані у шістнадцятиричну форму.

Тіло першого пакету має вигляд:

00 00 00 3C 01 77 35 94 00 02 00 00 01 80

Тіло другого пакету має вигляд:

6C 69 6E 65 31 3A 61 63 74 69 6F 6E 20 73 74 61 72 74 0D 6C 69 6E 65 32 3A 4B 68 65 72 73 6F 6E 20 4E 61 74 69 6F 6E 61 6C 20 54 65 63 68 6E 69 63 61 6C 20 55 6E 69 76 65 72 73 69 74 79 0D 6C 69 6E 65 33 3A 63 6F 6D 6D 61 6E 64 20 65 6E 64 20 74 72 61 6E 73 69 74 69 6F 6E

Тіло третього пакету:

6E 75 6D 62 65 72 31 3A 01 80 0D 6C 69 6E 65 31 4B 68 65 72 73 6F 6E 20 4E 61 74 69 6F 6E 61 6C 20 54 65 63 68 6E 69 63 61 6C 20 55 6E 69 76 65 72 73 69 74 79 6C 69 6E 65 32 3A 73 74 61 74 75 73 20 69 6E 66 6F 6E 75 6D 62 65 72 31 3B 77 35 94 00

Перший пакет також використовує нові ідентифікатор згідно таблиці 3.4. Це зручніша форма запису у двійковій формі. Другий та третій пакет використовують форму даних як CoAP з кодуванням у шістнадцятиричну форму.

3.4 Розрахунок об'ємів даних у тілі повідомлення

Об'єм даних у тілі повідомлення – це кількість байт, що займають ідентифікатори та дані згідно формату у протоколі структури даних.

Для проведення розрахунку прийнято, що для кодування текстових даних використовується кодування UTF-8, за яким кожен символ кодується одним байтом.

Протокол HTTP відділяє дані від наступного ідентифікатору комбінацією символів «\r\n». Це додає по два символи до кожного рядку. Після підрахунку кількості символів отримані наступні значення:

- перший пакет має обсяг 38 байт;
- другий пакет 95 байт;
- третій пакет 97 байт.

Інші пакети, що мають текстову форму опису даних для відділення даних від наступного ідентифікатору використовують символ переходу на наступний рядок «\n». Це додає один байт до кожного рядка.

Враховуючи вищесказаний фактор отримані об'єми даних для протоколу STOMP:

- обсяг першого пакету складає 41 байт;
- обсяг другого пакету 91 байт;
- обсяг третього пакету 93 байт.

Протокол CoAP може використовувати бінарну форму запису даних або текстову. Через це перший пакет використовує бінарну форму для запису, другий пакет текстову форму, а третій змішану: числові дані у бінарній формі, а інші у текстовій. На основі цього отримані об'єми тіла повідомлення:

- обсяг першого пакету складає 14 байт;
- другий пакет складає 96 байт;
- третій пакет складає 93 байт.

Протоколи SOAP та XMPP мають однакове тіло повідомлення. Через це обсяги даних на тіло повідомлення однакові та складають:

- перший пакет 71 байт;
- другий пакет 118 байт;
- третій пакет 129 байт.

Протоколи MQTT та MQTT-SN використовують лише бінарне кодування для даних. Через це форма запису використовує шістнадцятирічні коди. При цьому дані у тілі повідомлення мають схожий запис з протоколом CoAP тільки у бінарній формі. На основі цього розраховані об'єми даних у тілі пакетів:

- тіло першого пакету має обсяг 14 байт;
- тіло другого пакет 91 байт;
- тіло третього пакету 82 байт.

На основі вище розрахованих даних отримана таблиця 3.5 у якій зазначені об'єми тіла повідомлення за кожним з протоколів структури даних.

Таблиця 3.5 – Об'єми тіла протоколів структури даних

	Пакет 1	Пакет 2	Пакет 3
HTTP	38	95	97
SOAP	71	118	129
XMPP	71	118	129
STOMP	41	91	93
CoAP	14	96	93
MQTT	14	91	82

3.5 Розрахунок об'ємів даних у тілі повідомлення

Для отримання повного пакету структури даних необхідно додати до тіла пакету технічні дані. Серед технічних даних є ідентифікатор ресурсу, ідентифікатор серверу та ідентифікатор клієнту.

У таблиці 3.6 зазначені технічні дані для пакетів структури даних.

Таблиця 3.6 – Технічні дані для пакетів структури даних

Дані	Значення
Ідентифікатор ресурсу	/send/info

Дані	Значення
Ідентифікатор серверу	iot.server
Ідентифікатор клієнту	iot.client

На основі таблиці 3.6 формуються повні пакети даних для протоколів структури даних. Також додаються технічні дані, що регламентуються самим протоколом.

Для протоколу HTTP повний пакет даних має вигляд:

PUT /send/info HTTP/1.1

Host: iot.server

Content-Length: 41

number1:15

number2:11

number3:2004

Другий пакет даних має вигляд:

PUT /send/info HTTP/1.1

Host: iot.server

Content-Length: 91

line1:action start

line2:Kherson National Technical University

line3:command end transition

Третій пакет:

PUT /send/info HTTP/1.1

Host: iot.server

Content-Length: 93

number1:384

line1:Kherson National Technical University

line2:status info

number2:2000000000

Перший пакет протоколу SOAP має форму:

```
<soap:Envelope>
```

```
<soap:Body>
```

```
<put resource="/send/info">
```

```
<number1>60</int1>
```

```
<number2>2000000000</number2>
```

```
<number3>384</number3>
```

```
</put>
```

```
</soap:Body>
```

```
</soap:Envelope>
```

Другий пакет має форму:

```
<soap:Envelope>
```

```
<soap:Body>
```

```
<put resource="/send/info">
```

```
<line1>action start</line1>
```

```
<line2>Kherson National Technical University</line2>
```

```
<line3>command end transition</line3>
```

```
</put>
```

```
</soap:Body>
```

```
</soap:Envelope>
```

Третій пакет має форму:

```
<soap:Envelope>
```

```
<soap:Body>
```

```
<put resource="/send/info">
```

```
<number1>384</number1>
```

```
<line1>Kherson National Technical University</line1>
```

```
<line2>status info</line3>
```

```
<number2>2000000000</number2>
</put>
</soap:Body>
</soap:Envelope>
```

У протоколі ХМР перший пакет має форму:

```
<message from="iot.client@domain" to="iot.server@domain/send/info">
<body>
<number1>60</int1>
<number2>2000000000</number2>
<number3>384</number3>
</body>
</message>
```

Другий пакет:

```
<message from="iot.client@domain" to="iot.server@domain/send/info">
<body>
<line1>action start</line1>
<line2>Kherson National Technical University</line2>
<line3>command end transition</line3>
</body>
</message>
```

Третій пакет:

```
<message from="iot.client@domain" to="iot.server@domain/send/info">
<body>
<number1>384</number1>
<line1>Kherson National Technical University</line1>
<line2>status info</line3>
<number2>2000000000</number2>
</body>
</message>
```

Протокол STOMP після додавання технічних даних формує наступну форму першого пакету:

SEND

destination:/send/info

content-type:text/plain

content-length:41

number1:60

number2:2000000000

number3:384

Другий пакет:

SEND

destination:/send/info

content-type:text/plain

content-length:91

line1:action start

line2:Kherson National Technical University

line3:command end transition

Третій пакет:

SEND

destination:/send/info

content-type:text/plain

content-length:93

number1:384

line1:Kherson National Technical University

line2:status info

number2:2000000000

Повний перший пакет для протоколу CoAP:

48 03 7B 7E 00 00 00 00

Uri-Path="/send/info"

FF

00 00 00 3C 01 77 35 94 00 02 00 00 07 D4

Другий пакет:

48 03 7B 7E 00 00 00 00

Uri-Path="/send/info"

FF

line1:action start

line2:Kherson National Technical University

line3:command end transition

Третій пакет:

48 03 7B 7E 00 00 00 00

Uri-Path="/send/info"

FF

number1: 00 00 3C 01

line1:Kherson National Technical University

line2:status info

number2: 77 35 94 00

Для протоколів MQTT та MQTT-SN перший пакет має формат:

12 18

2F 73 65 6E 64 2F 69 6E 66 6F 00 00 00 00 3C 01 77 35 94 00 02 00 00 01 80

Другий пакет має вигляд:

12 65

2F 73 65 6E 64 2F 69 6E 66 6F 00 6C 69 6E 65 31 3A 61 63 74 69 6F 6E 20 73 74

61 72 74 0D 6C 69 6E 65 32 3A 4B 68 65 72 73 6F 6E 20 4E 61 74 69 6F 6E 61 6C 20

54 65 63 68 6E 69 63 61 6C 20 55 6E 69 76 65 72 73 69 74 79 0D 6C 69 6E 65 33 3A
63 6F 6D 6D 61 6E 64 20 65 6E 64 20 74 72 61 6E 73 69 74 69 6F 6E

Третій пакет:

12 5D

2F 73 65 6E 64 2F 69 6E 66 6F 00 6E 75 6D 62 65 72 31 3A 01 80 0D 6C 69 6E
65 31 4B 68 65 72 73 6F 6E 20 4E 61 74 69 6F 6E 61 6C 20 54 65 63 68 6E 69 63 61 6C
20 55 6E 69 76 65 72 73 69 74 79 6C 69 6E 65 32 3A 73 74 61 74 75 73 20 69 6E 66 6F
6E 75 6D 62 65 72 31 3B 77 35 94 00

Після формування пакетів структури даних розраховані об'єми пакетів. У таблиці 3.7 відображені отримані результати.

Таблиця 3.7 – Об'єми даних у пакетів

	Пакет 1	Пакет 2	Пакет 3
HTTP	104	161	163
SOAP	154	201	216
XMPP	153	200	215
STOMP	111	161	162
CoAP	45	127	124
MQTT	27	104	95

3.6 Аналіз пакетів у протоколах структури даних

На основі даних таблиці 3.7 були побудовані гістограми обсягів даних для пакетів даних для усіх протоколів даних (рис. 3.1).

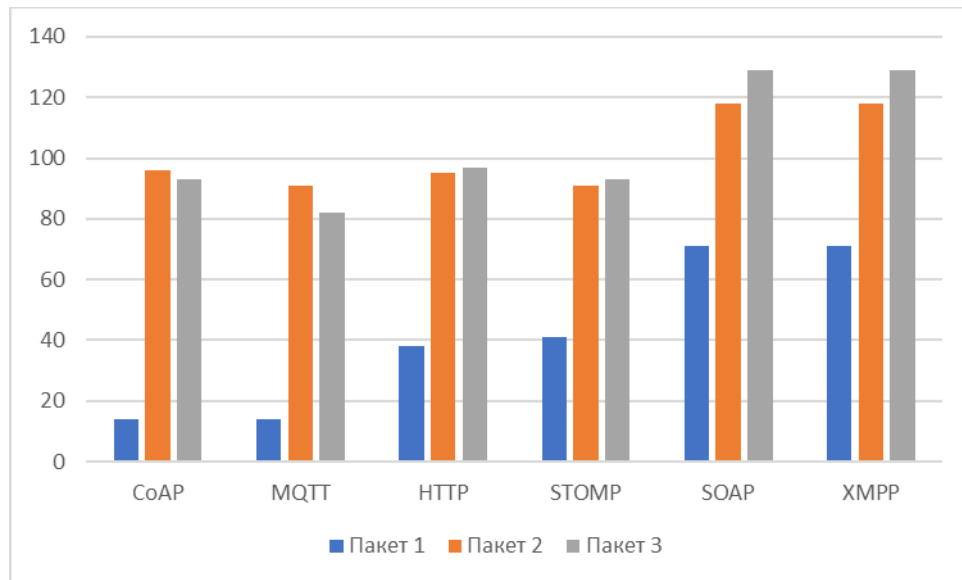


Рисунок 3.1 – Гістограма обсягів даних

Згідно гістограми видно, що протоколи SOAP та XMPP в середньому мають обсяг даних більше на 20% ніж інші протоколи на тестових даних, при текстових та змішаних типах даних. Особливо виділяються числові дані. Для них збільшення обсягу склало близько 40%, відносно протоколів з текстовим кодуванням чисел. Бінарне кодування чисел дає вигоду близько 300% в обсязі даних [73].

Через таку різницю в обсягах даних використання протоколів SOAP та XMPP не є доцільним в рамках оптимізації енергоспоживання.

Також гістограма відображає, що при передачі лише числових даних у системі використання протоколів CoAP та MQTT або MQTT-SN є оптимальним варіантом.

При передачі текстових або змішаних даних усі протоколи крім SOAP та XMPP демонструють приблизно однакові результати, що дозволяє обрати будь-який з протоколів за бажанням розробника системи.

Випадки коли можна ігнорувати недоліки протоколів:

- Протоколи SOAP та XMPP можна використовувати у системах для яких можливий варіант підключення до інших систем сторонніми розробниками. Це дозволить швидше розібратися у структурі даних та легше розробити обробник для них.

- Протокол HTTP дозволяє одразу передавати дані через мережу Інтернет без додаткової обробки, що зменшує навантаження на центри обробки даних у системі.

3.7 Розрахунок об'ємів технічних даних у повному пакеті даних

Технічні дані протоколів передачі сигналів містять інформацію про протокол, версію протоколу, об'єми тіла повідомлення. Ці дані являють собою заголовок пакету.

Технічні дані визначають максимальний обсяг тіла повідомлення на один пакет. Через це виникає необхідність розділення тіла на декілька пакетів, для кожного з яких будуть свої технічні дані, що збільшить обсяг даних на передачу.

Використовуючи інформацію про структуру пакетів протоколів передачі сигналів визначено обсяг технічних даних необхідних для кожного пакету даних. Отримані результати відображені у таблицях 3.8 - 3.10.

Таблиця 3.8 – Обсяги технічних даних для першого пакету

	Wi-Fi	Bluetooth	NB-IoT	ZigBee	6LoWPAN	WirelessHART
HTTP	50	14	14	108	66	80
SOAP	50	14	21	162	66	80
XMPP	50	14	21	162	66	80
STOMP	50	14	14	108	66	80
CoAP	50	14	7	54	33	40
MQTT	50	14	7	54	33	40

Таблиця 3.9 – Обсяги технічних даних другого пакету

	Wi-Fi	Bluetooth	NB-IoT	ZigBee	6LoWPAN	WirelessHART
HTTP	50	14	21	162	99	120
SOAP	50	14	28	162	99	120
XMPP	50	14	28	162	99	120

	Wi-Fi	Bluetooth	NB-IoT	ZigBee	6LoWPAN	WirelessHART
STOMP	50	14	21	162	99	120
CoAP	50	14	14	108	66	80
MQTT	50	14	14	108	66	80

Таблиця 3.10 – Обсяги технічних даних третього пакету

	Wi-Fi	Bluetooth	NB-IoT	ZigBee	6LoWPAN	WirelessHART
HTTP	50	14	21	162	99	120
SOAP	50	14	28	216	99	120
XMPP	50	14	28	216	99	120
STOMP	50	14	21	162	99	120
CoAP	50	14	14	108	66	80
MQTT	50	14	14	108	66	80

3.8 Розрахунок обсягів повного пакету даних

Згідно формули (3.1) обсяг пакету даних визначається сумою даних тіла протоколу структури даних та технічних даних протоколу передачі сигналів. Для розрахунку використана формула (3.1), у яку підставлено значення з таблиці 3.5 та значення з таблиць 3.8 - 3.10. Отримані результати відображені у таблицях 3.11 - 3.13.

Таблиця 3.11 – Загальний обсяг даних у першому пакеті

	Wi-Fi	Bluetooth	NB-IoT	ZigBee	6LoWPAN	WirelessHART
HTTP	154	118	118	212	170	184
SOAP	204	168	175	316	220	234
XMPP	203	167	174	315	219	233
STOMP	161	125	125	219	177	191
CoAP	95	59	52	99	78	85

	Wi-Fi	Bluetooth	NB-IoT	ZigBee	6LoWPAN	WirelessHART
MQTT	77	41	34	81	60	67

Таблиця 3.12 – Загальний обсяг даних у другому пакеті

	Wi-Fi	Bluetooth	NB-IoT	ZigBee	6LoWPAN	WirelessHART
HTTP	211	175	182	323	260	281
SOAP	251	215	229	363	300	321
XMPP	250	214	228	362	299	320
STOMP	211	175	182	323	260	281
CoAP	177	141	141	235	193	207
MQTT	154	118	118	212	170	184

Таблиця 3.13 – Загальний обсяг даних у третьому пакеті

	Wi-Fi	Bluetooth	NB-IoT	ZigBee	6LoWPAN	WirelessHART
HTTP	213	177	184	325	262	283
SOAP	266	230	244	432	315	336
XMPP	265	229	243	431	314	335
STOMP	212	176	183	324	261	282
CoAP	174	138	138	232	190	204
MQTT	145	109	109	203	161	175

3.9 Аналіз обсягів повних пакетів даних

Базуючись на інформації з таблиць 3.11-3.13 було побудовано порівняльні гістограми розмірів пакетів даних, що відправляються за протоколами передачі сигналів та мають в собі дані закодовані протоколами структури даних (рис. 3.2-3.4).

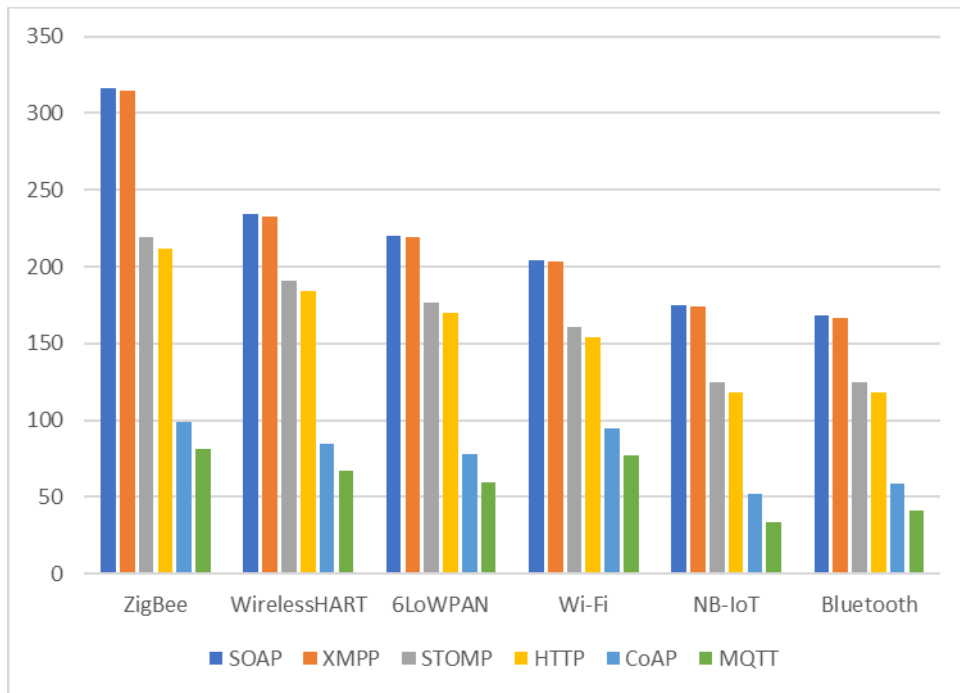


Рисунок 3.2 – Гістограми для першого пакету даних

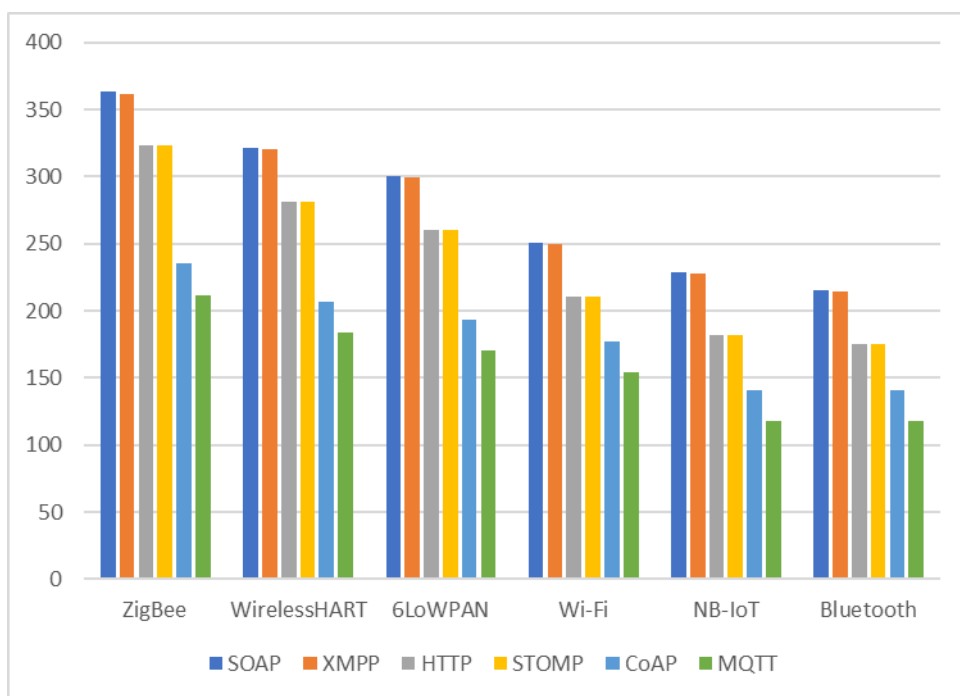


Рисунок 3.2 – Гістограма для другого пакету даних

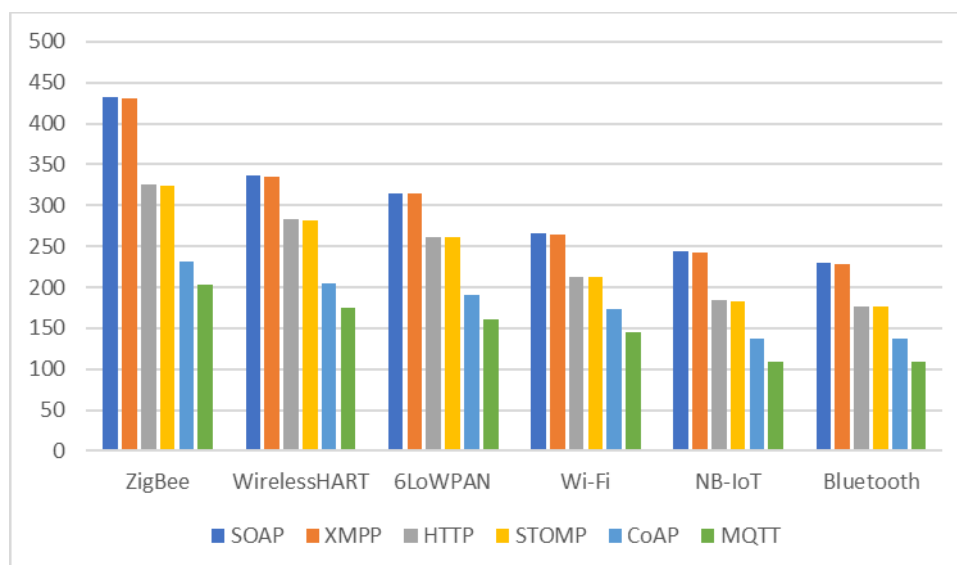


Рисунок 3.3 – Гістограма для третього пакету даних

Згідно з результатами найгірші результати має протокол ZigBee у комбінації з SOAP або XMPP через те що у протоколах SOAP та XMPP є надмірність через подвійний опис змінних, через що дані не вміщуються в одному пакеті ZigBee і необхідно застосовувати декілька пакетів між якими розділяться усі данні. Також технічні дані протоколу ZigBee займають великі обсяги, що вимагає більше передач при декількох пакетах.

Краще себе показують інші протоколи стандарту IEEE 802.15.4, але вони програють за показниками Wi-Fi, NB-IoT та Bluetooth.

Використання SOAP та XMPP найкраще в комбінації з Bluetooth. Інші протоколи структури краще взаємодіють з NB-IoT. Найкращі результати виявлені у комбінації NB-IoT з MQTT.

Оскільки протокол NB-IoT може використовуватися для обміну даними на відстань до 50 км, то використання цього протоколу у локальних системах не рекомендується.

Друге обмеження накладається протоколом MQTT: протокол вимагає від пристроїв у мережі постійної активності для очікування підключення. Це дозволяє виконати передачу команд пристрою гарантуючи їх негайне виконання, але також потребує більшої кількості енергії, що зменшить час використання джерела живлення.

Якщо використовувати цю комбінацію лише для передачі інформації від пристрою на віддалений сервер на великій відстані, то необхідності у підтримці постійної активності пристрою відпадає. Вона необхідна лише, якщо пристроєм можливо віддалено керувати.

В усіх пакетах протокол MQTT може бути замінено на протокол CoAP. Цей протокол також не потребує постійної активності пристрою. Хоч кількість даних для передачі більша ніж у MQTT, але це нівелюється простотою формування пакету даних та обробки.

Хоч і Bluetooth має гірші показники за NB-IoT, але в локальних мережах він більш підходящий, оскільки не вимагає потужної центральної станції. Відстані передачі до 50м вистачить для роботи більшості локальних систем.

Розповсюдження Wi-Fi дозволяє використовувати його для систем інтернету речей. Із недоліків можна виділити, що для своєї роботи вимагає постійного обміну даними між кінцевим пристроєм та базовою станцією для підтримки зв'язку. Такі постійні передачі виснажують джерело живлення. Bluetooth також має підтримувати зв'язок, але кількість передач менше.

Перевагою для Wi-Fi є швидкість передачі, що дозволяє передавати великі обсяги данні які притаманні для аудіо або відео передач.

На відміну від Wi-Fi та Bluetooth у інших протоколах немає необхідності підтримувати зв'язок. Кінцеві пристрою прокидаються лише під час своєї передачі даних.

Протокол HTTP рекомендовано використовувати лише для текстових або змішаних даних, де різниця між об'ємами відносно протоколів CoAP та MQTT не настільки велика. В поєднанні з протоколом 6LoWPAN або Wi-Fi дозволить більш швидко інтегрувати систему у мережу Інтернет.

3.10 Висновки до розділу

В третьому розділі проведено аналіз протоколів структури даних. Для аналізу впливу їх на загальний трафік у системі були сформовані три пакети даних з різним наповненням: числові дані, текстові дані, змішані дані.

З даних були сформовані пакети за кожним протоколом та проаналізовано їх розміри. Найкращі показники має протокол MQTT, але оскільки він вимагає постійної активності пристрою для підтримки каналу зв'язку, то оптимальніше використати протокол CoAP, що не має такої вимоги та має найближчі показники до MQTT.

Використання SOAP та XMPP найкраще в комбінації з Bluetooth. Інші протоколи структури краще взаємодіють з NB-IoT. Найкращі результати виявлені у комбінації NB-IoT з MQTT.

Оскільки протокол NB-IoT може використовуватися для обміну даними на відстань до 50 км, то використання цього протоколу у локальних системах не рекомендується та рекомендовано використати протоколи Bluetooth або Wi-Fi.

РОЗДІЛ 4

ДОСЛІДЖЕННЯ ВПЛИВУ ШИФРУВАННЯ НА РОЗМІРИ ПАКЕТІВ

4.1 Аналіз шифрування у протоколах передачі сигналів

Одним із аспектів, що може впливати на енергоспоживання пристроїв Інтернету речей, є шифрування, яке може збільшувати обсяги трафіку. Для перевірки впливу шифрування було перевірено пакети передачі сигналів.

У протоколі Wi-Fi основна увага приділяється частинам пакету CCMP Header, Payload та MIC, що беруть участь у шифруванні даних. Для шифрування корисних даних використовується алгоритм AES у парі з протоколом CCMP [74, 75, 76]. Частина MIC – це код, що розраховується як контрольна сума корисних даних (Payload) у пакеті та дозволяє виконати перевірку правильності передачі даних. Частини Payload та MIC шифруються як одне ціле. Алгоритм AES має декілька варіантів шифрування. Для шифрування даних у Wi-Fi використовується версія AES-CTR з ключем на 128 біт (16 байт). Згідно з ним на вхід алгоритму подається деякий лічильник, що збільшувався з моменту старту. Для кожного блоку корисних даних використовуються унікальні значення лічильника, що зменшує ризик розшифровки сторонніми особами. На початку процесу шифрування корисні дані мають бути доповнені до розміру, який буде кратний розміру ключа, що відповідає 128 бітам. Доповнення початкових даних може сягати до 127 біт, що має додатковий негативний вплив на обсяги даних для передачі. Обсяги даних, отриманих після доповнення, не змінюються під час шифрування. Другим доповненням до обсягів пакету даних є заголовок CCMP. У ньому зберігається номер пакету даних, пустий байт, зарезервований на майбутнє, та байт параметрів шифрування. У цьому байті 5 біт завжди встановлений у значення 1, щоб вказати, що є шифрування за допомогою AES. Біти 6-7 зберігають ідентифікатор ключа, за яким виконувалось шифрування, якщо ключі були заздалегідь обумовлені. Біти 0-4 є зарезервованими на майбутнє. З цього випливає, що один байт додатково додано для виконання шифрування, і загальний обсяг

пакету даних збільшується від 8 до $8+127 = 135$ бітів в залежності від обсягів доповнення початкових даних. При малих обсягах розмір корисних даних може бути меншим, ніж обсяг додаткової інформації, необхідної для шифрування.

В стандарті Bluetooth є дві основні частини, що задіяні у шифруванні – Payload та MIC. Bluetooth має схожий з Wi-Fi алгоритм шифрування. Він використовує метод шифрування AES у версії AES-CTR з ключем на 128 біт [77]. Частина Payload відповідає за корисні дані, а MIC - за контрольну суму для перевірки цілісності корисних даних. Під час шифрування Payload та MIC об'єднуються та шифруються як одне ціле. Оскільки алгоритм AES вимагає, щоб довжина даних була кратна ключу шифрування (128 біт), то об'єднані дані доповнюються до кратності. На відміну від Wi-Fi, у стандарті Bluetooth немає додаткового байту з параметрами шифрування, тому максимальне збільшення розмірів пакету може складати до 127 біт.

У протоколі NB-IoT під час передачі даних шифрується частина Payload. Для шифрування використовується протокол ESP на основі алгоритму 128-EEA2 [78, 79]. Цей алгоритм передбачає доповнення корисних даних до розміру ключа, що складає 128 біт. Через це обсяг зашифрованих даних може збільшуватися від 1 до 127 біт.

У пакеті протоколу ZigBee шифруються частини APS Header, ZCL Header та Payload. Частина Payload зберігає корисні дані, що передаються у пакеті. APS Header відповідає за контроль зв'язку з кластером, до якого відноситься пристрій, що передає дані. ZCL Header визначає напрям передачі пакету між клієнтом та сервером, та за тип команди, що передається.

Для шифрування також використовується алгоритм AES у версії AES-CTR з ключем шифрування 128 біт [80]. Оскільки необхідно забезпечити кратність ключу, потрібно виконати доповнення пакету даних обсягом до 127 біт, що впливає на обсяги трафіку.

Окремо шифрується частина MIC за алгоритмом AES у версії AES-CBC. Після шифрування залишаються лише частина із старших 4 байт. Оскільки розмір частини MIC є фіксованим, то він не має впливу на розмір пакету даних.

У пакеті даних протоколу 6LoWPAN шифруються частини IPv6+UDP Header та Payload. Частина Payload відповідає за корисні дані. У IPv6+UDP Header зберігаються дані адресації за протоколом IPv6 та заголовки передачі даних за протоколом UDP.⁷

Для шифрування використовуються алгоритм AES у версії AES-CTR з розміром ключа 128 біт [81, 82]. Під час шифрування ці частини об'єднуються в єдиний текст та шифруються разом. Для шифрування необхідне доповнення від 1 до 127 біт, що впливає на обсяги даних у пакеті.

Для протоколу WirelessHART шифруються частини Security Header та Payload у пакеті даних. Для шифрування використовуються алгоритм AES у версії AES-CBC [83]. Алгоритм вимагає доповнення даних до розміру ключа, що складає 128 біт. Через це обсяги доповнення можуть становити від 1 до 127 бітів, що збільшують обсяги пакету.

4.2 Аналіз впливу шифрування на пакети даних

У кожному із представлених протоколів існує проблема збільшення пакету, що передається, внаслідок шифрування даних, що вимагає доповнення корисних даних додатковими бітами до розміру ключа. Оскільки очевидний негативний вплив доповнення, то необхідно перевірити співвідношення обсягів доповнення до корисних даних, щоб мати підстави для вибору оптимального протоколу. Для цього були побудовані графіки обсягів доповнень у відсотках відносно обсягів корисних даних для кожного протоколу (рис. 4.1-4.6) [84].

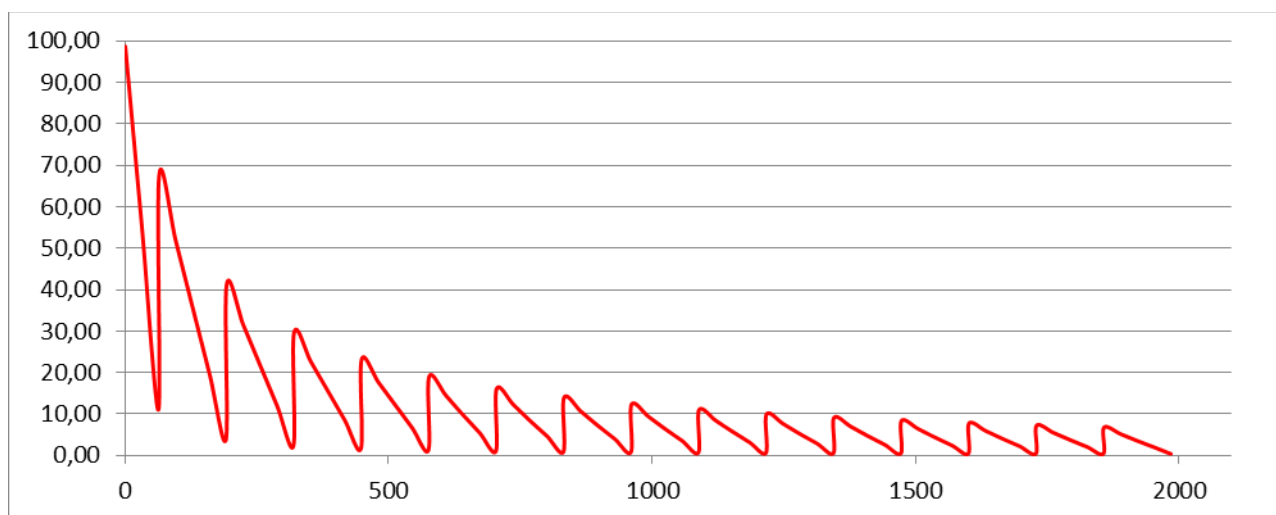


Рисунок 4.1 – Обсяги доповнень у стандарті Wi-Fi

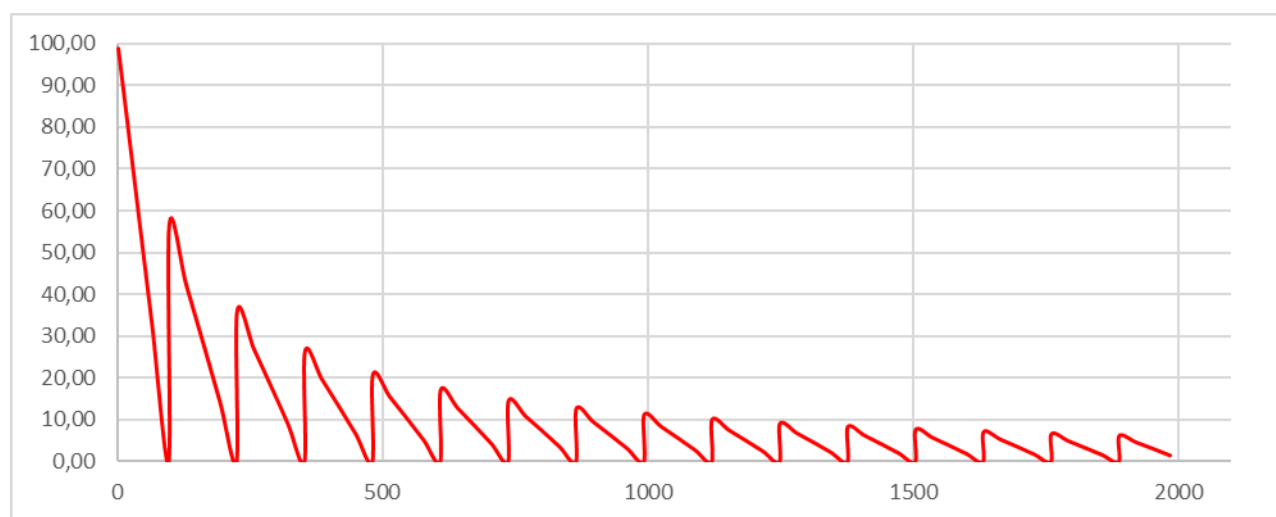


Рисунок 4.2 – Обсяги доповнень у стандарті Bluetooth

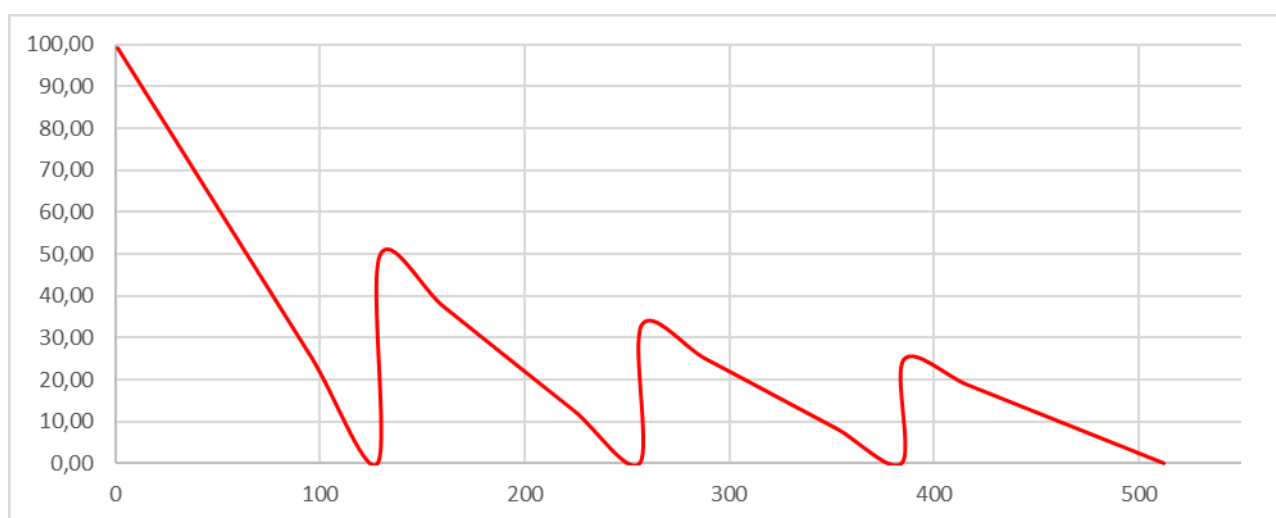


Рисунок 4.3 – Обсяги доповнень у протоколі NB-IoT

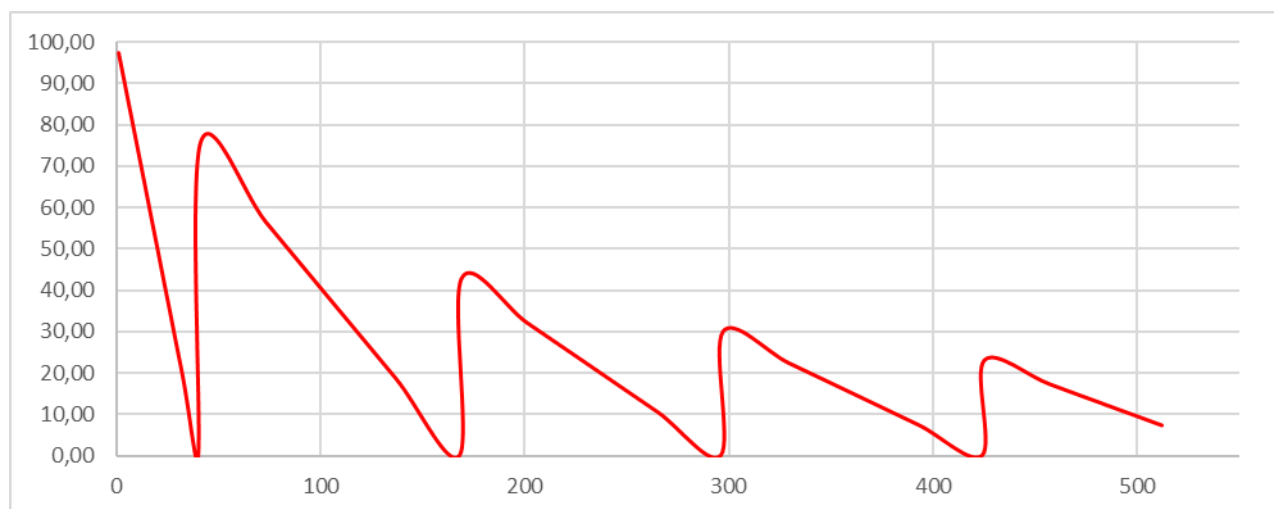


Рисунок 4.4 – Обсяги доповнень у протоколі ZigBee

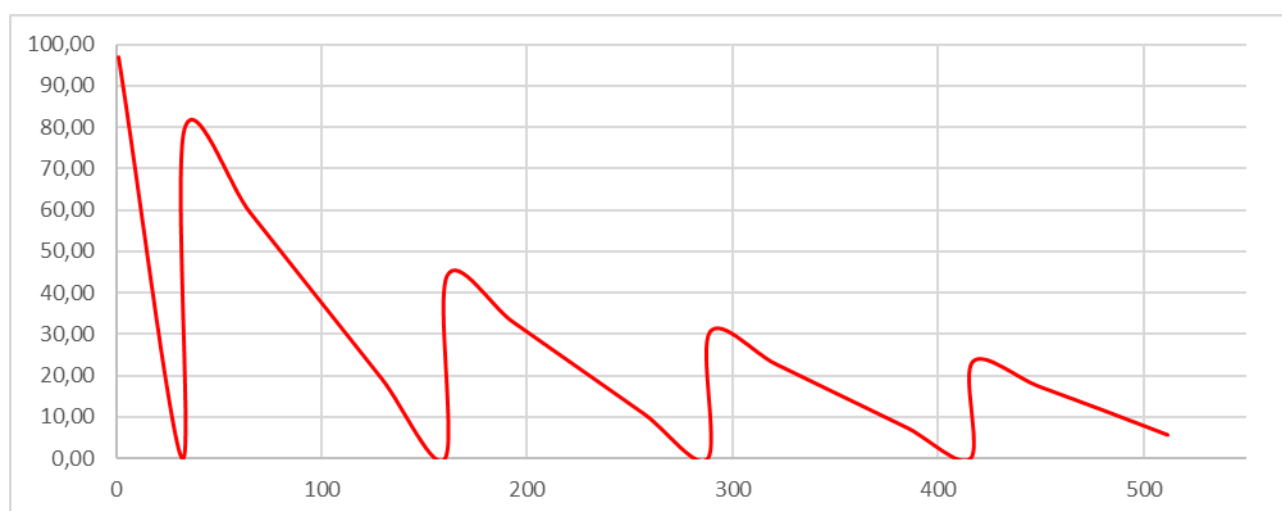


Рисунок 4.5 – Обсяги доповнень у стандарті 6LoWPAN

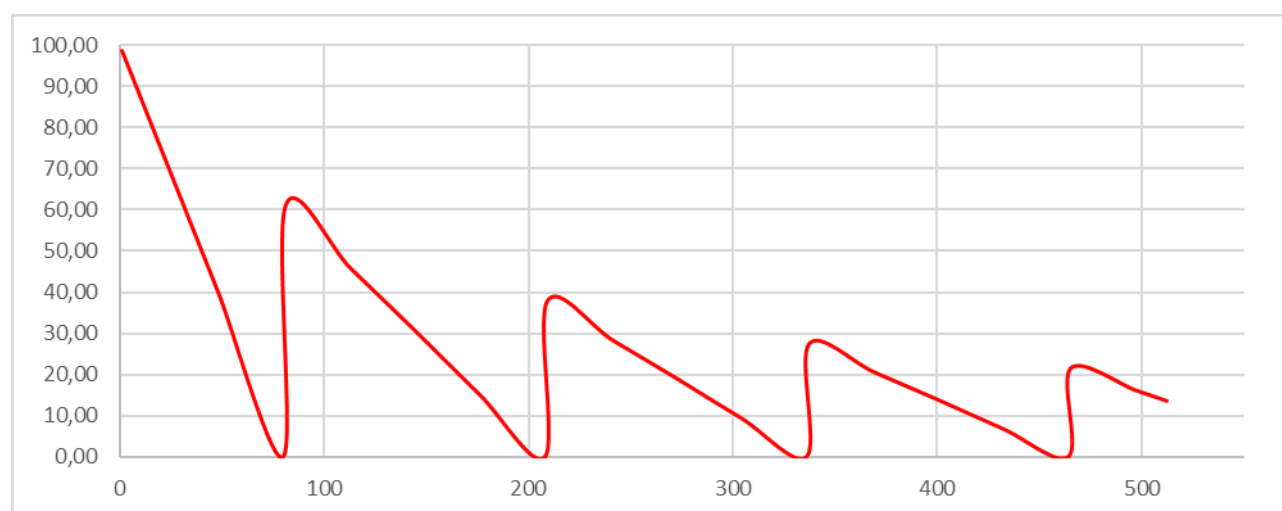


Рисунок 4.6 – Обсяги доповнень у протоколі WirelessHART

З графіків видно, що при збільшенні обсягів корисних даних зменшується вплив доповнення на загальний обсяг пакету даних. Окремо необхідно виділити піки на графіку, що виникають, коли необхідно виконати доповнення на велику кількість бітів, для досягнення кратності ключа шифрування.

Також було проведено аналіз середнього значення відсотка доповнення відносно корисних даних для кожного протоколу та стандарту. Результат представлено у таблиці 4.1.

Таблиця 4.1 – Середній відсоток доповнень [84]

Протокол або стандарт	Середній відсоток доповнень відносно корисних даних	Максимальний обсяг корисних даних
Wi-Fi	2.13%	2304 байт
Bluetooth	10.64%	251 байт
6LoWPAN	23.22%	80 байт
WirelessHART	23.08%	80 байт
ZigBee	24.81%	68 байт
NB-IoT	25.96%	65 байт

З таблиці 4.1 видно, що чим більший обсяг даних здатний передавати протокол або стандарт, тим менший середній відсоток доповнень буде містити пакет даних. Але сучасні пристрої інтернету речей є енергоефективними та передають малі обсяги даних, тому обчислення долі доповнень відносно корисних даних недоцільно робити на основі їх максимального обсягу.

Якщо обмежити максимальний обсяг корисних даних до 64 байт, то можна розрахувати середній відсоток для більшості енергоефективних систем, що мають малі обсяги трафіку (табл. 4.2) [84].

Таблиця 4.2 – Середній відсоток доповнень відносно 64 байтів корисних даних

Протокол або стандарт	Середній відсоток доповнень при обсязі корисних даних до 64 байт
Wi-Fi	9.95%
Bluetooth	16.31%
6LoWPAN	11.55%
WirelessHART	10.12%
ZigBee	8.00%
NB-IoT	20.21%

Обсяги доповнення найбільший вплив мають на протокол NB-IoT. Далі за ним йде стандарт Bluetooth. Найкраще себе показує протокол ZigBee, що може свідчити про те, що він добре підходить для систем з обмеженими за обсягом корисними даними [84].

4.3 Висновки до розділу

У розділі отримані наступні результати:

- Представлено нову кількісну оцінку впливу шифрування на обсяги трафіку в протоколах Інтернету речей (IoT).
- Проведено емпіричну оцінку з використанням реальних сценаріїв передавання даних для вимірювання розміру додаткових даних, спричинених шифруванням.
- Запропоновано новий критерій вибору для удосконалення процесу прийняття рішень під час вибору протоколів для IoT-систем.

Сучасні стандарти та протоколи Інтернету речей включають обов'язкове шифрування даних. Оскільки алгоритми шифрування вимагають доповнення корисних даних до розміру ключа, який зазвичай становить 128 біт, виникає необхідність передачі «зайвої» інформації, що в середньому складає близько 26%

від загального обсягу корисних даних у пакеті. Загальний вплив на обсяги трафіку в більшості протоколів становить приблизно 10%, за винятком Bluetooth (16%) та NB-IoT (20%).

Для оптимізації обсягів трафіку слід враховувати розмір пакета даних, зважаючи на те, що шифруються не тільки корисні дані, але й інші частини пакету. Аналіз популярних протоколів щодо частки "додаткової" інформації, пов'язаної з шифруванням, показав, що протокол ZigBee є найбільш рекомендованим для використання в системах з обмеженим обсягом корисних даних.

Отримані результати дозволять більш точно обирати протокол або стандарт під час проєктування систем інтернету речей, оскільки є можливість провести моделювання передачі даних у системі та визначити чи буде передаватися надлишкова інформація.

РОЗДІЛ 5

АВТОМАТИЗАЦІЯ ПРОЦЕСУ ВИБОРУ ПРОТОКОЛІВ ІНТЕРНЕТУ РЕЧЕЙ

5.1 Математична основа вибору протоколів Інтернету речей

У деяких випадках під час розробки системи може з'явитися необхідність спеціалізованого опису даних під час передачі. Це може зменшити час передачі та покращити енергоефективність.

Деякі з протоколів вимагають постійної підтримки з'єднання IoT пристрою з системою. Якщо система не повинна негайно реагувати на зміни стану, то протоколи з постійним з'єднанням лише виснажуватимуть джерело живлення.

Можливі ситуації, коли декілька протоколів будуть давати однакові результати. У такому випадку вже розробники системи мають вирішувати, який з альтернативних протоколів використовувати.

У таблиці 1 приведена інформація про протоколи передачі сигналів щодо відстані передачі, часу виснаження джерела живлення, швидкості передачі та об'єму пакету даних.

Таблиця 5.1 – Характеристики IoT-протоколів [85, 86, 87, 88]

Протокол	Відстань передачі	Час виснаження джерела живлення	Швидкість передачі	Об'єм пакету даних
Wi-Fi	< 100 м.	декілька місяців	2 Гбіт/сек.	50 байта
Bluetooth	< 50 м.	декілька місяців	24 Мбіт/сек.	14 байт
NB-IoT	< 50 км.	1-2 роки	250 Кбіт/сек.	7 байт
ZigBee	< 100 м.	5-10 років	250 Кбіт/сек.	54 байт

Протокол	Відстань передачі	Час виснаження джерела живлення	Швидкість передачі	Об'єм пакету даних
6LoWPAN	< 800 м.	декілька місяців	250 Кбіт/сек.	33 байт
WirelessHART	< 255 м.	4-10 років	250 Кбіт/сек.	40 байт

Представлені дані спираються лише на теоретичний розрахунок часу роботи під час проектування протоколів.

Для спрощення вибору протоколів було сформовано список правил, за якими можна обрати оптимальну комбінацію протоколів [89, 90]:

- Передача відео/аудіо даних у мережі потребує великих швидкостей, через це для таких мереж рекомендовано протокол передачі сигналів Wi-Fi.
- При великій відстані передачі оптимальним протоколом передачі сигналів є NB-IoT.
- Для спрощеної інтеграції системи у мережу Інтернет рекомендується протокол передачі сигналів 6LoWPAN та протокол структури даних HTTP.
- Якщо необхідна розподілена мережа обробки даних, то рекомендується протокол структури даних XMPP, що має можливість міжсерверної передачі без додаткових модулів.
- При необхідності віддаленого керування пристроями рекомендується протокол MQTT, що дозволяє підтримувати активність пристроїв в режимі очікування пакету даних з командою на виконання.
- При необхідності спрощеної інтеграції у систему для сторонніх розробників рекомендується протокол структури даних SOAP, оскільки він використовує мову розмітки XML, що полегшує формування та аналіз пакету даних.
- Пристрої у персональних мережах здатні виконати передачу даних на відстань до 10 метрів, у локальних мережах на відстань до 200 м. У системах з

великомасштабним розгортанням передача може виконуватися на відстань до 50 км.

Для формалізації процесу вибору протоколів на основі наведених вище правил було вирішено використовувати декілька показників.

У якості критеріїв для вибору було обрано наступні:

- Відстань передачі: в залежності від відстані передачі визначається один із протоколів передачі сигналів.
- Розподілена мережа серверів обробки: при необхідності розподілених обчислень буде обраний один з протоколів структури даних, що здатний на це.
- Спрощена інтеграція для сторонніх розробників: якщо після розробки системи з нею працюватимуть сторонні розробники, то буде обрано один з протоколів структури даних, що використовує простий метод опису інформації.
- Відео/аудіо дані: при передачі такого виду даних необхідні протоколи, що мають велику швидкість передачі та здатні на просту передачу бінарних даних.
- Швидке реагування пристроїв у системі на команди від серверу: якщо є необхідність керувати пристроями Інтернету речей у системі віддалено, то можлива ситуація, коли пристрої мають отримати команду якомога швидше, що потребуватиме вибору протоколу, який може одразу відправити команду пристрою.

Процес вибору можна звести до відповіді «так» або «ні» на шість запитань:

- Відстань передачі у протоколі відповідає вимогам системи?
- Потрібна розподілена мережа серверів обробки?
- Потрібна спрощена інтеграція для сторонніх розробників?
- Наявна передача відео/аудіо даних?
- Необхідна швидка реакція пристроїв на команди від серверу?
- Проста інтеграція у мережу інтернет?

Наявність лише двох варіантів відповіді на запитання дозволяє створити булеві функції від n змінних.

Було прийняте рішення розділити запитання на дві групи відповідно до типу протоколів.

Протоколи передачі сигналів:

- Відстань передачі у протоколі відповідає вимогам системи?
- Наявна передача відео/аудіо даних?
- Проста інтеграція у мережу інтернет?

Протоколи структури даних:

- Потрібна розподілена мережа серверів обробки?
- Потрібна спрощена інтеграція для сторонніх розробників?
- Необхідна швидка реакція пристроїв на команди від серверу?
- Проста інтеграція у мережу інтернет?

На основі вхідних питань побудовані таблиці булевих функцій для кожного з протоколів (табл. 2 та табл. 3). Для протоколів передачі сигналів зміни мають наступні значення: а – відповідь на питання «відстань передачі у протоколі відповідає вимогам системи?», b – «наявна передача відео/аудіо даних?», c – «проста інтеграція у мережу інтернет?». Булеві значення представлені у таблиці 5.2.

Таблиця 5.2 – Таблиця значень булевих функцій протоколів передачі сигналів

a	b	c	$F_{Wi-Fi}(abc)$	$F_{Bluetooth}(abc)$	$F_{NB-IoT}(abc)$	$F_{ZigBee}(abc)$	$F_{MeshLogic}(abc)$	$F_{6LoWPAN}(abc)$	$F_{WirelessHART}(abc)$
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1

a	b	c	$F_{Wi-Fi}(abc)$	$F_{Bluetooth}(abc)$	$F_{NB-IoT}(abc)$	$F_{ZigBee}(abc)$	$F_{MeshLogic}(abc)$	$F_{6LoWPAN}(abc)$	$F_{WirelessHART}(abc)$
1	0	1	0	0	0	0	0	1	0
1	1	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0

Для протоколів структури даних (табл. 5.3) змінні відповідають наступним питанням: a – «потрібна розподілена мережа серверів обробки?», b – «потрібна спрощена інтеграція для сторонніх розробників?», c – «необхідне швидка реакція пристроїв на команди від серверу?», d – «проста інтеграція у мережу інтернет?».

Таблиця 5.3 – Таблиця значень булевих функцій протоколів структури даних

a	b	c	d	$F_{HTTP}(abcd)$	$F_{SOAP}(abcd)$	$F_{XMPP}(abcd)$	$F_{STOMP}(abcd)$	$F_{CoAP}(abcd)$	$F_{MQTT}(abcd)$
0	0	0	0	0	0	0	1	1	0
0	0	0	1	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	1
0	0	1	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0
1	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0

a	b	c	d	$F_{HTTP}(abcd)$	$F_{SOAP}(abcd)$	$F_{XMPP}(abcd)$	$F_{STOMP}(abcd)$	$F_{CoAP}(abcd)$	$F_{MQTT}(abcd)$
1	0	1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0

Результатом є значення булевих функцій $F(abc)$ та $F(abcd)$ для кожного з наявного протоколів. Якщо результат функції для протоколу має значення «1», то його можна використовувати у даній системі.

Таблиця 5.2 дозволяє прийти до функцій від 3 змінних для протоколів передачі сигналів:

$$F_{Wi-Fi}(abc) = a\bar{b}\bar{c} + ab\bar{c} + abc = a\bar{c} + ab$$

$$F_{Bluetooth}(abc) = a\bar{b}\bar{c}$$

$$F_{NB-IoT}(abc) = a\bar{b}\bar{c}$$

$$F_{ZigBee}(abc) = a\bar{b}\bar{c}$$

$$F_{MeshLogic}(abc) = a\bar{b}\bar{c}$$

$$F_{6LoWPAN}(abc) = a\bar{b}\bar{c} + a\bar{b}c = a\bar{c}$$

$$F_{WirelessHART}(abc) = a\bar{b}\bar{c}$$

Таблиця 5.3 приводиться до функцій від 4 змінних для протоколів структури даних:

$$F_{HTTP}(abcd) = a\bar{b}\bar{c}d$$

$$F_{SOAP}(abcd) = a\bar{b}\bar{c}\bar{d}$$

$$F_{XMPP}(abcd) = a\bar{b}\bar{c}\bar{d}$$

$$F_{STOMP}(abcd) = a\bar{b}\bar{c}\bar{d}$$

$$F_{CoAP}(abcd) = a\bar{b}\bar{c}\bar{d}$$

$$F_{MQTT}(abcd) = \bar{a}\bar{b}cd$$

Отримані булеві функції дозволяють перевірити протокол до можливості застосування його у конкретній системі.

5.2 Автоматизація процесу вибору протоколів Інтернету речей

Оскільки значення булевих функцій можна представити у програмному коді за допомогою базових операторів «І» та «АБО», то за отриманими функціями було розроблене програмне забезпечення вибору протоколів «LProtocols Selector», що дозволяє під час підготовки нової системи одразу сформувати список протоколів, що відповідатимуть вимогам.

Програма запитує критерії системи, що відповідають значенням змінних у булевих функціях. На основі вхідних значень змінних за допомогою умовних операторів перевіряється кожний з протоколів на значення його булевої функції. При значенні «1» протокол вважається підходящим для системи.

Програму написано мовою C#. Для використання програми необхідний комп'ютер з операційною системою сімейства Microsoft Windows, починаючи з ОС Microsoft Windows XP. На рисунку 5.1 відображено головне вікно програми «LProtocols Selector» [91].

Рисунок 5.1 – Головне вікно програми LProtocols Selector

Вікно програми розділено на 2 частини. Зліва розміщені критерії, яким має відповідати мережа Інтернету речей. У правій частині відображатиметься інформація про обрані протоколи після аналізу критеріїв на основі правил, зазначених вище.

Користувач вводить дані та обирає критерії, і після натискання кнопки «Обрати» програма на основі правил та критеріїв вибере комбінацію, що складається з протоколу передачі сигналів та протоколу структури даних. Отримані результати будуть відображені у правій частині вікна. На рисунку 5.2 показано результат роботи програми.

Рисунок 5.2 – Результат роботи програми LProtocols Selector

Програма відображає такі додаткові дані:

- Максимальна відстань, на яку можливо виконати передачу за допомогою обраного протоколу передачі сигналів.
- Максимальна швидкість, з якою можлива передача даних обраним протоколом передачі сигналів.
- Коментар до протоколу передачі сигналів, де наявна інформація про протокол та причину його вибору.

– Коментар до протоколу структури даних відображає інформацію про протокол, а також причину, через яку програма його обрала.

Протоколи, що обрані у результаті роботи програми, не є обов'язковими для використання. В деяких випадках обрані протоколи можуть не підходити через територіальне розміщення або через необхідність використання технологій, що не мають сумісності з протоколами.

Прикладом проблеми територіального розміщення є проблема розміщення проміжних маршрутизаторів для протоколів Wi-Fi, або Bluetooth. Через це необхідно змінити протокол передачі сигналів на такий, що матиме комірчасту топологію, у якій маршрутизаторами виступатимуть самі пристрої.

5.3 Висновки до розділу

В п'ятому розділі розглянуто математичний підхід до вибору протоколів при побудові систем Інтернету речей. Отримані булеві функції за яким визначається можливість застосування кожного з протоколів.

На основі математичного рішення розроблене програмне забезпечення, що дозволяє обрати протоколи вказавши характеристики системи, що необхідно побудувати. Результати роботи програми є рекомендаційними і не обов'язковими для застосування, якщо є невраховані розбіжності між можливостями протоколів та системою Інтернету речей.

ВИСНОВКИ

У роботі досліджено протоколи Інтернету речей для оптимізації енергоспоживання кінцевих пристроїв. Вони були розділені на дві категорії згідно їх призначення: протоколи структури даних та протоколи передачі сигналів.

До протоколів передачі сигналів відносяться: Wi-Fi, Bluetooth, NB-IoT, ZigBee, 6LoWPAN, WirelessHART та Matter. Через те, що Matter не має власних особливостей для передачі сигналів, а використовує мережі Wi-Fi та ZigBee, то він виключений з цього дослідження, оскільки результати його ефективності можна отримати з результатів протоколів Wi-Fi та ZigBee.

Протоколами структури даних є: HTTP, SOAP, XMPP, STOMP, CoAP, MQTT.

Були проаналізовані протоколи передачі сигналів. Для кожного з них визначено типовий вигляд пакету даних для проведення аналізу обсягів трафіку як одного з чинників енерговитрати пристроїв.

Також було проведено аналіз наявних моделей та методів оптимізації енерговитрат для протоколів передачі сигналів. Наявні методи були розбиті за категоріями їх призначення: керування часом активності пристрою; керування частотними характеристиками передачі; керування потужністю передавача; керування пакетами даних; керування маршрутизацією.

Протоколи ZigBee та WirelessHART мають методи оптимізації за всіма наявними категоріями. Протоколи Wi-Fi, 6LoWPAN мають по 4 категорії методів оптимізації. Bluetooth та NB-IoT мають по 3 категорії оптимізації. Найкращі показники у методів оптимізації, що відносяться до критерію «Керування часом активності пристрою», що зумовлено повним відключенням пристрою на певні проміжки часу, поки не настає момент збору даних та відправки їх до центру обробки.

Було проведено аналіз протоколів структури даних. Для аналізу впливу їх на загальний трафік у системі були сформовані три пакети даних з різним наповненням: числові дані, текстові дані, змішані дані.

З даних були сформовані пакети за кожним протоколом та проаналізовано їх розміри. Найкращі показники має протокол MQTT, але оскільки він вимагає постійної активності пристрою для підтримки каналу зв'язку, то оптимальніше використати протокол CoAP, що не має такої вимоги та має найближчі показники до MQTT.

Використання SOAP та XMPP найкраще в комбінації з Bluetooth. Інші протоколи структури краще взаємодіють з NB-IoT. Найкращі результати виявлені у комбінації NB-IoT з MQTT.

Оскільки протокол NB-IoT може використовуватися для обміну даними на відстань до 50 км, то використання цього протоколу у локальних системах не рекомендується та рекомендовано використати протоколи Bluetooth або Wi-Fi.

Сучасні стандарти та протоколи Інтернету речей включають обов'язкове шифрування даних. Оскільки алгоритми шифрування вимагають доповнення корисних даних до розміру ключа, який зазвичай становить 128 біт, виникає необхідність передачі «зайвої» інформації, що в середньому складає близько 26% від загального обсягу корисних даних у пакеті. Загальний вплив на обсяги трафіку в більшості протоколів становить приблизно 10%, за винятком Bluetooth (16%) та NB-IoT (20%).

Для оптимізації обсягів трафіку слід враховувати розмір пакета даних, зважаючи на те, що шифруються не тільки корисні дані, але й інші частини пакету. Аналіз популярних протоколів щодо частки "додаткової" інформації, пов'язаної з шифруванням, показав, що протокол ZigBee є найбільш рекомендованим для використання в системах з обмеженим обсягом корисних даних.

Отримані результати дозволять більш точно обирати протокол або стандарт під час проєктування систем інтернету речей, оскільки є можливість провести моделювання передачі даних у системі та визначити чи буде передаватися надлишкова інформація.

Було розглянуто математичний підхід до вибору протоколів, при побудові систем Інтернету речей. У результаті були отримані булеві функції за яким визначається можливість застосування кожного з протоколів.

На основі математичного рішення розроблене програмне забезпечення, що дозволяє обрати протоколи вказавши характеристики системи, що необхідно побудувати. Результати роботи програми є рекомендаційними і не обов'язковими для застосування, якщо є невраховані розбіжність між можливостями протоколів та системою Інтернету речей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Shendge, Antima. INTERNET OF THINGS (IOT): AN OVERVIEW ON RESEARCH CHALLENGES AND FUTURE APPLICATIONS. International Journal of Engineering Applied Sciences and Technology. 2021 Vol. 6. P. 66-71. DOI: 10.33564/IJEAST.2021.v06i08.011.
2. Nimodiya, Aditi & Ajankar, Shruti. A Review on Internet of Things. International Journal of Advanced Research in Science, Communication and Technology. 2022. P. 135-144. DOI: 10.48175/IJAR SCT-2251.
3. OSI Model. Imperva. URL: <https://www.imperva.com/learn/application-security/osi-model/> (дата звернення 17.11.2020).
4. «What Is Wi-Fi?». Cisco. URL: <https://www.cisco.com/c/en/us/products/wireless/what-is-wifi.html> (дата звернення 17.11.2020).
5. Wi-Fi - Access Protocols. Tutorials Point. URL: https://www.tutorialspoint.com/wi-fi/wifi_access_protocols.htm (дата звернення 17.11.2020).
6. Wireless Security Protocols: WEP, WPA, WPA2, and WPA3. URL: <https://www.netspotapp.com/wifi-encryption-and-security.html> (дата звернення 17.11.2020).
7. Bluetooth. Techopedia. URL: <https://www.techopedia.com/definition/26198/bluetooth> (дата звернення 17.11.2020).
8. Bluetooth. Versions. URL: <https://sites.google.com/site/project3bluetooth/versions> (дата звернення 17.11.2020).
9. Bluetooth. Univers System integration. URL: <http://www.univers-spb.ru/technologys/bluetooth.php> (дата звернення 17.11.2020).
10. Bluetooth Adaptive Frequency Hopping on a R&S CMW. Rohde & Schwarz. 2016. URL: https://www.rohde-schwarz.com/ua/applications/-bluetooth-r-s-cmw-application-note_56280-364992.html (дата звернення 17.11.2020).

11. NB-IoT overview. Haltian. 2019. URL: <https://haltian.com/resource/nb-iot-overview/> (дата звернення 17.11.2020).
12. Weidong Fang, Wuxiong Zhang, Lianhai Shan, Quianqian, Wei Chen. A Converged Network Architecture Oriented Towards NB-IoT. International Conference on 5G for Future Wireless Networks. 2019. pp 31-39.
13. Afzal Junaid. NB-IoT - Data Rates and Latency. NetManias. 2017. URL: <https://www.netmanias.com/en/post/blog/12609/iot-nb-iot/nb-iot-data-rates-and-latency> (дата звернення 17.11.2020).
14. 3GPP Specification Release Numbers. URL: <https://www.electronics-notes.com/articles/connectivity/3gpp/standards-releases.php> (дата звернення 17.11.2020).
15. James Agajo, Jonathan Gana Kolo, Mutiu A. Adegboye, Bello Kontagora Nuhu. Experimental Performance Evaluation and Feasiblity Study of 6LoWPAN Based Internet of Things. Acta Electrotechnica et Informatica. 2017. Vol. 17(2). DOI: 10.15546/aeei-2017-0011
16. P. Arun Mozhi Devan, Fawnizu Azmadi Hussin, Rosdiazli Ibrahim, Kishore Bingi, Farooq Ahmad Khanday. A Survey on the Application of WirelessHART for Industrial Process Monitoring and Control. Sensors. 2021. Vol. 21(15). DOI: doi.org/10.3390/s21154951
17. M. Kostadinovic, M.K. Stojčev, Zlatko Bundalo, Dusanka Bundalo. Design, implementation and simulation of wirelesshart network. Telecommunication in Modern Satellite, Cable, and Broadcasting Services. 2009. DOI:10.1109/TELSKS.2009.5339447
18. Song Han, Xiuming Zhu, Aloysius Mok, Mark Nixon. Control over WirelessHART network. 2010. DOI: 10.1109/IECON.2010.5675278
19. Pavlo Matiieshyn. Matter Smart Home Standard: What's It All About? 2023. URL: <https://lebergssolutions.com/blog/matter-protocol-smart-home-industry> (дата звернення: 27.11.2023 р.)
20. David Pierce. How Matter became the future of the smart home. 2021. URL: <https://www.protocol.com/matter-smart-home> (дата звернення: 27.11.2023 р.)

21. Sam Barros. HTTP: An overview of web development. 2019. URL: <https://medium.com/swlh/http-an-overview-of-web-development-4f6033efb9da> (дата звернення 17.11.2020).
22. What is SOAP: Formats, Protocols, Message Structure, and How SOAP is Different from REST. 2019. URL: <https://www.altexsoft.com/blog/engineering/what-is-soap-formats-protocols-message-structure-and-how-soap-is-different-from-rest/> (дата звернення 17.11.2020).
23. An Overview of XMPP. URL: <https://xmpp.org/about/technology-overview/> (дата звернення 17.11.2020).
24. The Simple Text Oriented Messaging Protocol. URL: <https://stomp.github.io/> (дата звернення 17.11.2020).
25. Z. Shelby, K. Hartke, C. Borman. The Constrained Application Protocol (CoAP). 2014. URL: <https://tools.ietf.org/html/rfc7252> (дата звернення 17.11.2020).
26. MQTT: The Standard for IoT Messaging. URL: <https://mqtt.org/> (дата звернення 17.11.2020).
27. MQTT return codes. URL: <https://notes.networklessons.com/mqtt-return-codes> (дата звернення 17.11.2020).
28. 802.11 Frame Types and Formats. How I Wi-Fi. 2020. URL: <https://howiwifi.com/2020/07/13/802-11-frame-types-and-formats/> (дата звернення 29.11.2024).
29. Rob Brownstein. What are the benefits and drawbacks of WiFi power saving modes (PSM and U-APSD)? 2023. URL: <https://www.linkedin.com/advice/1/what-benefits-drawbacks-wifi-power-saving-modes-psm-u-apsd> (дата звернення: 07.10.2024).
30. Michael De Nil. Wi-Fi HaLow: What is it and why you might need it. 2024. URL: <https://www.iotinsider.com/iot-insights/technical-insights/wi-fi-halow-what-is-it-and-why-you-might-need-it/> (дата звернення: 07.10.2024).
31. Joel Hruska. How to Boost Your Wi-Fi Speed by Choosing the Right Channel. 2021. URL: <https://www.extremetech.com/internet/179344-how-to-boost-your-wifi-speed-by-choosing-the-right-channel> (дата звернення: 07.10.2024).

32. Transmit Power Control. URL: <https://www.winncom.com/en/glossary/162/transmit-power-control> (дата звернення: 07.10.2024).

33. IEEE 802.11 Frame Aggregation. URL: <https://inet.omnetpp.org/docs/showcases/wireless/aggregation/doc/index.html> (дата звернення: 07.10.2024).

34. Bluetooth Adaptive Frequency Hopping on a R&S CMW. Rohde & Schwarz. 2016. URL: https://www.rohde-schwarz.com/ua/applications/-bluetooth-r-s-cmw-application-note_56280-364992.html (дата звернення 17.11.2020).

35. Jinxiao Zhang. The application of bluetooth technology in the internet of things. Applied and Computational Engineering. 2023. Vol. 12(1). Pp. 177-183. DOI: 10.54254/2755-2721/12/20230334

36. Optimizing Current Consumption in Bluetooth Low Energy Devices. URL: <https://docs.silabs.com/bluetooth/6.1.0/bluetooth-fundamentals-system-performance/current-consumption> (дата звернення: 07.10.2024).

37. Malvinder Singh Bali, Kamali Gupta, Kanwalpreet Kour Bali, Pramod K. Singh. Towards energy efficient NB-IoT: A survey on evaluating its suitability for smart applications. Materials Today: Proceedings. 2022. Volume 49, Part 8. pp. 3227-3234. DOI: doi.org/10.1016/j.matpr.2020.11.1027.

38. eDRX. URL: <https://www.narrowband.com/nbiot-glossary/edrx> (дата звернення: 07.10.2024).

39. Mwakwata CB, Malik H, Mahtab Alam M, Le Moullec Y, Parand S, Mumtaz S. Narrowband Internet of Things (NB-IoT): From Physical (PHY) and Media Access Control (MAC) Layers Perspectives. Sensors. 2019. Vol. 19(11):2613. DOI: 10.3390/s19112613

40. Bilal Doori, Ahmed Zurfi. Decreasing the RA Collision Impact for Massive NB-IoT in 5G Wireless Networks. Jordanian Journal of Computers and Information Technology. 2021. Vol. 07(03):1. DOI: 10.5455/jjcit.71-1620292048

41. Internet of Things (IoT). URL: <https://www.cablefree.net/wirelesstechnology/internet-of-things-iot/> (дата звернення: 07.10.2024).

42. Lozano Jesús, Suárez José Ignacio, Arroyo Patricia, Ordiales José, Álvarez, Fernando. Wireless Sensor Network For Indoor Air Quality Monitoring. Chemical Engineering Transactions. 2012. Vol. 30. P. 319-324. DOI: 10.3303/CET1230054.

43. Sleep Options | ZigBee Networking Guide. 2020. URL: <https://development.libelium.com/zigbee-networking-guide/sleep-options> (дата звернення: 07.10.2024).

44. Zigbee Frequency Bands. URL: <https://www.everythingrf.com/community/zigbee-frequency-bands> (дата звернення: 07.10.2024).

45. Khalid A. Fakieh. A Survey on 6LowPAN & its Future Research Challenges. International Journal of Computer Science and Mobile Computing. 2014. Vol. 3. Pp. 558-570.

46. CHENG Hong Ubin, WANG Xiao Una, LIANG Wei, NIE Pan Uhong, SUN Xia. Study on Energy Saving for 6LoWPAN under Enabled Node Sleep State. Computer Engineering & Science. 2012. Vol. 34.

47. Geetha V. Archana Bhat, S. Thanmayee. New Bit Pattern Based IPv6 Address Compression Techniques for 6LoWPAN Header Compression. IEEE Access. 2022. Vol. 10. Pp. 80055-80070. DOI: 10.1109/ACCESS.2022.3193235

48. Ruchi Garg. Comparative Study on Techniques of IPv6 Header Compression in 6LoWPAN. Fourth International Conference on Advances in Information Processing and Communication Technology - IPCT 2016. 2016. DOI: 10.15224/978-1-63248-099-6-33

49. Hu J, Luo J, Zhang Y, Wang P, Liu Y. Location-Based Data Aggregation in 6LoWPAN. International Journal of Distributed Sensor Networks. 2015. Vol. 11(10). DOI:10.1155/2015/912926

50. Gee Keng Ee, Chee Kyun Ng, Nor Kamariah Noordin, Borhanuddin Mohd. Ali. A Review of 6LoWPAN Routing Protocols. Proceedings of the Asia-Pacific Advanced Network. 2010. Vol. 30. DOI: 10.7125/APAN.30.11

51. WirelessHART® networks: 7 myths that cloud their consideration for process control. URL: <https://new.abb.com/products/measurement-products/wireless-products-and-solutions/wirelesshart-networks-seven-myths-that-cloud-their-consideration-for-process-control> (дата звернення: 07.10.2024).

52. Tran Duc Chung, Rosdiazli Ibrahim, Vijanth Sagayan Asirvadam, Nordin Saad, Sabo Miya Hassan. Energy Consumption Analysis of WirelessHART Adaptor for Industrial Wireless Sensor Actuator Network. Procedia Computer Science. 2017. Volume 105. Pp. 227-234. DOI: 10.1016/j.procs.2017.01.215.

53. WirelessHART Radio Communication Standard. URL: <https://control.com/textbook/wireless-instrumentation/wirelesshart/> (дата звернення: 07.10.2024).

54. Jianping Song, Song Han, Aloysius Mok, Deji Chen. WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control. Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS '08. IEEE. 2008. DOI:10.1109/RTAS.2008.15

55. Feng Li, Lei Ju, Zhiping Jia. Data aggregation framework for energy-efficient WirelessHART networks. Journal of Systems Architecture. 2016. Volume 63. Pp 70-79. DOI: 10.1016/j.sysarc.2016.01.009.

56. G. Anastasi, M. Conti, E. Gregori, A. Passarella. Saving Energy in Wi-Fi Hotspots through 802.11 PSM: an Analytical Model. 2011.

57. Serena Santi, Le Tian, Jeroen Famaey. Accurate Energy Modeling and Characterization of IEEE 802.11ah RAW and TWT. Sensors. 2019. № 11. 2614 p. DOI: 10.3390/s19112614.

58. Raksha Upadhyay, Arpita Tiwari, Uma Rathore Bhatt. Energy Efficient Rate Adaptation Algorithm for FiWi Access Network. Journal of Microwaves Optoelectronics and Electromagnetic Applications. 2017. Vol. 16(4). Pp. 908-921. DOI: 10.1590/2179-10742017v16i41013

59. What is Transmit Power & Transmit Power Control in Wi-Fi? 2022. URL: <https://thenetworkguys.wordpress.com/2022/11/10/what-is-transmit-power-transmit-power-control-in-wi-fi/> (дата звернення: 07.10.2024).

60. Mohammad Afaneh. Bluetooth Low Energy Power Consumption – How to Achieve Maximum Battery Life. 2023. URL: <https://novelbits.io/ble-power-consumption-optimization/> (дата звернення: 07.10.2024).

61. Ashish Kumar Sultania, Pouria Zand, Chris Blondia, Jeroen Famaey. Energy Modeling and Evaluation of NB-IoT with PSM and eDRX. IEEE Global Communications Conference. 2018. DOI:10.1109/GLOCOMW.2018.8644074

62. Nassim Labdaoui, Fabienne Nouvel, Stéphane Dutertre. NB-IoT Power Consumption: A Comparison of SFR and Objenious Network Operators. 17ème Colloque du GDR SoC2. 2023.

63. Zheng Jiang, Bin Han, Peng Chen, Fengyi Yang. On Novel Access and Scheduling Schemes for IoT Communications. Mobile Information Systems 2016. 2016. Vol. 1-9. DOI:10.1155/2016/3973287

64. Rachaen M. Huq, Kevin P. Moreno, Hui Zhu, Jue Zhang. On the Benefits of Clustered Capillary Networks for Congestion Control in Machine Type Communications over LTE. Computer Communication and Networks (ICCCN), 2015 24th International Conference. 2015. DOI:10.1109/ICCCN.2015.7288439

65. Chris Baumann. The Importance of sleep mode power consumption in ZigBee/802.15.4 applications. 2006. URL: <https://www.eetimes.com/the-importance-of-sleep-mode-power-consumption-in-zigbee-802-15-4-applications/> (дата звернення: 07.10.2024).

66. M. C. Ekström, M. Bergblomma, M. Lindén, M. Björkman, M. Ekström. Comparison study of ZigBee and Bluetooth with regards to power consumption, packet-error-rate and distance. Computer Science, Engineering. 2012.

67. Shi Longlong, Qiu Chunling, Gao Peng, Jia Zhengsen. The Research and Simulation of CSMA/CA Mechanism of ZigBee Protocol. Procedia Engineering. 2012. Volume 29. Pp 3466-3471. DOI: 10.1016/j.proeng.2012.01.513.

68. Emre Ertekin, Christos Christou, Booz Allen Hamilton. IPv6 Header Compression. North American IPv6 Summit, June 2004. 2004.
69. Simon Vedel Johansen. Node energy consumption variations with WirelessHART and RPL in a Wireless Sensor Network. 2018.
70. Stig Petersen, Simon Carlsen. Performance Evaluation of WirelessHART for Factory Automation. 2009 IEEE Conference on Emerging Technologies & Factory Automation. 2009. Pp. 1-9. DOI: 10.1109/ETFA.2009.5346996
70. Kozel, V., Ivanchuk, O., Drozdova, I., Prykhodko, O. Automation of the Protocol Selection Process for IoT Systems. International Journal of Computing. 2022. 21(2). Pp. 251-257. DOI: 10.47839/ijc.21.2.2594
71. Іванчук О.В. Дослідження методів оптимізації енерговитрат у протоколах Інтернету речей. Вісник Херсонського національного технічного університету. 2024. № 4(91). С. 273-279. DOI: 10.35546/kntu2078-4481.2024.4.35
72. Ivanchuk O., Kozel V., Drozdova Ie., Prykhodko O. Research of the Methods for optimizing energy consumption in IEEE 802.15.4 protocols. Вчені записки Таврійського національного університету імені В.І. Вернадського. 2024. Том 35 (74) №6. с. 112-117. DOI: 10.32782/2663-5941/2024.6.1/19
73. Іванчук О.В., Завгородній В.В., Козел В.М., Дроздова Є.А. Аналіз протоколів обміну даними для керування системами Інтернету речей. Вчені записки Таврійського національного університету. Серія: Технічні науки. 2020. №31(70). С. 99-104. DOI: 10.32838/2663-5941/2020.2-1/15
74. Ali M. Alsahlany, Zainalabdin H. Alfatlawy, Alhassan R. Almusawy. Experimental Evaluation of Different Penetration Security Levels in Wireless Local Area Network. Journal of Communications. 2018. № 13 (12). P. 723-729. DOI: 10.12720/jcm.13.12.723-729
75. Rasika Nayanajith. CWSP – CCMP Encryption Method. 2014. URL: <https://mrncciew.com/2014/08/19/cwsp-ccmp-encryption-method/> (дата звернення 14.10.2024).
76. Iman Saberi, Bahareh Shojaie, Mazleena Salleh, Mahan Niknafsgermani. Enhanced AES-CCMP key structure in IEEE 802.11i. Proceedings of 2011 International

Conference on Computer Science and Network Technology. 2011. DOI: 10.1109/ICCSNT.2011.6182011

77. Nthatisi Hlapisi, Understanding Security Keys in Bluetooth Low Energy. 2023. URL: <https://www.allaboutcircuits.com/technical-articles/understanding-security-keys-in-bluetooth-low-energy/> (дата звернення: 27.11.2024).

78. Guosheng Zhao, Huan Chen, Jian Wang, A Lightweight Block Encryption Algorithm for Narrowband Internet of Thing, 2022. DOI: 10.21203/rs.3.rs-2033728

79. Ashutosh Dhar Dwivedi, Gautam Srivastava, Differential Cryptanalysis in ARX Ciphers with specific applications to LEA. 2018.

80. ZigBee Specification. Davis, CA, 2015. URL: <https://lucidar.me/en/zigbee/files/docs-05-3474-21-0csg-zigbee-specification.pdf>

81. Muhammad Tanveer, Ghulam Abbas, Ziaul Haq Abbas, Muhammad Waqas, Fazal Muhammad, Sunghwan Kim, S6AE: Securing 6LoWPAN Using Authenticated Encryption Scheme, Sensors, 2020. doi: 10.3390/s20092707

82. Packet format for 6LoWPAN/AD6LoWPAN? URL: <https://ez.analog.com/wireless-sensor-networks-reference-library/ad6lowpan/w/documents/15056/packet-format-for-6lowpan-ad6lowpan>

83. Shahid Raza, Adriaan Slabbert, Thiemo Voigt, Krister Landernäs, Security considerations for the WirelessHART protocol, IEEE Conference on Emerging Technologies & Factory Automation, 2009. DOI: 10.1109/ETFA.2009.5347043

84. Іванчук О.В., Козел В.М. Дослідження впливу захисту інформації на обсяги пакетів даних протоколів Інтернету речей. Науковий журнал "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" Луцьк, 2024. Випуск № 57. с. 43-50. DOI: 10.36910/6775-2524-0560-2024-57-06

85. S. Sen, J. Koo, S. Bagchi. TRIFECTA: Security, Energy Efficiency, and Communication Capacity Comparison for Wireless IoT Devices. IEEE Internet Computing. 2018. vol. 22. pp. 74–81. DOI: 10.1109/MIC.2018.011581520

86. Kabalci Y., Kabalci E., Padmanaban S., Holm-Nielsen J.B., Blaabjerg F. Internet of Things Applications as Energy Internet in Smart Grids and Smart Environments. Electronics 2019. P. 8, 972/ DOI: 10.3390/electronics8090972

87. Lee J.S., Dong M.F., Sun Y.H. A preliminary study of low power wireless technologies: ZigBee and Bluetooth low energy. In Proceedings of the 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), 15–17 June 2015, Auckland. 2015. p. 135-139. DOI: 10.1109/ICIEA.2015.7334098

88. Tran Duc Chung, Rosdiazli Ibrahim, Vijanth Sagayan Asirvadam, Nordin Saad, Sabo Miya Hassan. Energy Consumption Analysis of WirelessHART Adaptor for Industrial Wireless Sensor Actuator Network. 2016 IEEE International Symposium on Robotics and Intelligent Sensors, IRIS 2016, 17-20 December 2016, Tokyo, Japan. Tokyo. 2016. p. 8. DOI: 10.1016/j.procs.2017.01.215

89. U. Tandale, B. Momin and D. P. Seetharam. An empirical study of application layer protocols for IoT. 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS). 2017. pp. 2447-2451, DOI: 10.1109/ICECDS.2017.8389890 .

90. C. Sharma and N. K. Gondhi. Communication Protocol Stack for Constrained IoT Systems. 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU). 2018. pp. 1-6, DOI: 10.1109/IoT-SIU.2018.8519904 .

91. Козел В.М., Іванчук О.В., Дроздова Є.А. Розробка системи збору інформації від IoT пристроїв. Вісник Херсонського національного технічного університету. 2019. № 3(70). С. 126-132. DOI: 10.35546/kntu2078-4481.2019.3.14

ДОДАТОК А

Список опублікованих праць за темою дисертації

Публікації у міжнародних фахових журналах Scopus або Web of Science:

1. Kozel, V., Ivanchuk, O., Drozdova, I., Prykhodko, O. Automation of the Protocol Selection Process for IoT Systems. International Journal of Computing. 2022. 21(2). Pp. 251-257. DOI: 10.47839/ijc.21.2.2594

Публікації в наукових фахових виданнях України:

1. Козел В.М., Іванчук О.В., Дроздова Є.А. Розробка системи збору інформації від IoT пристроїв. Вісник Херсонського національного технічного університету. 2019. № 3(70). С. 126-132. DOI: 10.35546/kntu2078-4481.2019.3.14

2. Козел В.М., Іванчук О.В., Дроздова Є.А., Завгородній В.В. Аналіз протоколів обміну даними для керування системами Інтернету речей. Вчені записки Таврійського національного університету імені В.І. Вернадського. 2020. № 31(70). С. 99-104. DOI: 10.32838/2663-5941/2020.2-1/15

3. Іванчук О.В., Козел В.М. Дослідження впливу захисту інформації на обсяги пакетів даних протоколів Інтернету речей. Науковий журнал "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" Луцьк, 2024. Випуск № 57. с. 43-50. DOI: 10.36910/6775-2524-0560-2024-57-06

4. Ivanchuk O., Kozel V., Drozdova Ie., Prykhodko O. Research of the Methods for optimizing energy consumption in IEEE 802.15.4 protocols. Вчені записки Таврійського національного університету імені В.І. Вернадського. 2024. Том 35 (74) №6. С. 112-117. DOI: 10.32782/2663-5941/2024.6.1/19

5. Іванчук О.В. Дослідження методів оптимізації енерговитрат у протоколах Інтернету речей. Вісник Херсонського національного технічного університету. 2024. № 4(91). С. 273-279. DOI: 10.35546/kntu2078-4481.2024.4.35

Матеріали і тези Всеукраїнських наукових та науково-практичних конференцій:

1. Іванчук О.В., Козел В.М., Дроздова Є.А. Аналіз методів шифрування протоколів інтернету речей . Сучасні комп'ютерні системи та мережі в управлінні:

матеріали VI Всеукраїнської наук.-практ. Інтернет-конф. здобувачів вищої освіти та молодих вчених (30 листопада 2023 р., м. Хмельницький, м. Херсон) / за ред. А.А. Григорової. – Херсон: Книжкове видавництво ФОП Вишемирський В. С., 2023. –с 245-247

2. Іванчук О.В., Козел В.М., Зоря Р.І. Дослідження протоколів та стандартів IoT. Сучасні комп'ютерні системи та мережі в управлінні: матеріали VI Всеукраїнської наук.-практ. Інтернет-конф. здобувачів вищої освіти та молодих вчених (30 листопада 2023 р., м. Хмельницький, м. Херсон) / за ред. А.А. Григорової. – Херсон: Книжкове видавництво ФОП Вишемирський В. С., 2023. –с 173-174

3. Іванчук О.В., Козел В.М., Дроздова Є.А. Методи оптимізації енерговитрат у протоколах Інтернету речей. Сучасні комп'ютерні системи та мережі в управлінні: матеріали VII Всеукраїнської наук.-практ. Інтернет-конф. здобувачів вищої освіти та молодих вчених (29 листопада 2024 р., м. Хмельницький, м. Херсон) / за ред. А.А. Григорової. – Херсон: Книжкове видавництво ФОП Вишемирський В. С., 2024.

Матеріали і тези Міжнародних наукових та науково-практичних конференцій:

1. Іванчук О.В., Козел В.М. Проблеми використання IoT-пристроїв. Молодь у світі сучасних технологій за тематикою: Використання інформаційних технологій в системах управління : матеріали міжнар. наук.- практ. конф. м. Херсон, 6-7 червня 2019 р.) / за заг. ред. Г.О. Райко – Херсон : Видавництво ФОП Вишемирський В. С., 2019. – 216 с.

2. Іванчук О.В., Козел В.М., Дроздова Є.А. Інтернет речей у післявоєнному відновленні Херсонщини. Синергія науки і бізнесу у повоєнному відновленні Херсонщини : матеріали Міжнародної науково-практичної конференції (ХНТУ, 26–28 квітня 2023 р.) у 2-х т. ; Т. 1 / за ред. О. В. Чепелюк. – Одеса : Олді+, 2023. – 456 с.