

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій**

До захисту допущено:

Завідувач кафедри

_____ Сергій КРАВЧУК

«__» _____ 2023 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія та програмування
інфокомунікацій»**

спеціальності 172 «Телекомунікації та радіотехніка»

**на тему: «Застосування літальних сенсорних мереж для побудови
Інтернету речей»**

Виконав:

студент ІV курсу, групи ТЗ-91

Третяк Артем Вікторович _____

Керівник:

Доцент кафедри ТК, к.т.н.

Валуйський Станіслав Вікторович _____

Рецензент:

Старший викладач кафедри ІКТС НН ІТС, к.т.н.

Новіков Валерій Іванович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій КРАВЧУК

«__» _____ 2023 р.

ЗАВДАННЯ

на дипломну роботу студенту

Третяку Артему Вікторовичу

1. Тема роботи «Застосування літальних сенсорних мереж для побудови Інтернету речей», керівник роботи Валуйський Станіслав Вікторович, к.т.н., затверджені наказом по університету від «22» травня 2023 р. № 1884-с.
2. Термін подання студентом роботи 18 червня 2023 р.
3. Вихідні дані до роботи: технологія LoRa. Мережевий сервер TTN.
4. Зміст роботи:
 - проаналізувати принципи функціонування технологій збору та передачі даних з сенсорних мереж;
 - сформулювати елементи методології реалізації ЛСМ Інтернету речей;
 - навести характеристику експлуатації розробленої системи;
 - розглянути питання програмної реалізації;
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо):
 - 1) Тема роботи
 - 2) Актуальність роботи, мета

- 4) Що таке LoRa
- 5) Сценарій застосування LoRa, дронів, сенсорів
- 6) Сценарій застосування шлюзу у поєднанні з дроном
- 7) Простий кейс: шлюз, мобільний вузол, вузол
- 8) Надсилання даних по LoRa Mesh
- 9) Надсилання даних по LoRaWAN
- 10) Загальні висновки по роботі

6. Дата видачі завдання 30 жовтня 2022р

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Дослідження можливостей використання безпілотних літальних апаратів для збору і передачі даних з сенсорних мереж, що використовуються в Інтернеті речей.	05.11.2022-01.01.2023	Виконано
2	Ознайомлення з сучасними технологіями, які застосовуються для створення сенсорних мереж, а також особливості їх функціонування.	01.01.2023-15.03.2023	Виконано
3	Ознайомлення з технічною стороною процесу збору та передачі даних з дронів до серверів Інтернету речей	15.03.2023-15.04.2023	Виконано
4	Ознайомлення з перевагами та недоліками літальних сенсорних мереж для побудови Інтернету речей порівняно з іншими технологіями збору та передачі даних	15.04.2023-05.05.2023	Виконано
5	Розгляд питання програмної реалізації	05.05.2023-25.05.2023	Виконано

Студент

Артем ТРЕТЯК

Керівник

Станіслав ВАЛУЙСЬКИЙ

РЕФЕРАТ

Дипломна робота містить 63 сторінки, 18 рисунків. Було використано 32 використаних джерел.

КВАДРОКОПТЕР, МОНІТОРИНГ НАВКОЛИШНЬОГО СЕРЕДОВИЩА, ІНТЕРНЕТ РЕЧЕЙ, ЛІТАЛЬНА СЕНСОРНА МЕРЕЖА, SJRC F7 4K PRO, LOPY4, LORA, LORA MESH, LORAWAN.

Актуальність роботи. Актуальність застосування літальних сенсорних мереж для побудови Інтернету речей можна пояснити кількома ключовими факторами, такими як, охоплення територій, передача даних в реальному часі, гнучкість та масштабованість та інноваційні можливості. Літальні сенсорні мережі можуть покривати значно більші території та здатні швидко переміщатись і працювати в різних локаціях, що дозволяє забезпечувати підключення до Інтернету речей навіть в віддалених або важкодоступних місцях. Вони можуть бути підключені до хмарних обчислювальних систем, що дозволяє передавати зібрані дані в реальному часі. Це має важливе значення для реалізації швидко реагуючих систем. Загалом, застосування літальних сенсорних мереж для побудови Інтернету речей має великий потенціал у різних галузях, сприяючи розвитку технологій та покращенню якості життя людей.

Мета. Застосування літальної сенсорної мережі для побудови Інтернету речей. В якості літальної мережі буде використан квадрокоптер із прикріпленим сенсорним датчиком на базі LoRa, завдяки якому, за протоколами LoRa Mesh та LoRaWAN буде збиратись дані з датчиків за для передачі їх на сервер TTN для подальшої обробки.

Об'єкт дослідження – процес функціонування безпроводових сенсорних мереж із застосуванням безпілотних літальних апаратів (БПЛА).

Предмет дослідження – методи побудови безпроводових сенсорних мереж із застосування БПЛА.

Мета дослідження – розробити методику побудови безпроводових сенсорних мереж із застосування (БПЛА) на основі технології LoRa.

Поставлена мета передбачає розв'язання таких завдань:

1. Проаналізувати основні характеристики сенсорних мереж у контексті використання дрона у якості мобільної літаючої платформи.
2. Визначити функціональні особливості сенсорних мереж з подальшим вибором типу дрона.
3. Проаналізувати можливості технологій LoRa та TTN.
4. Навести характеристику розроблюваної системи.
5. Розглянути питання програмної реалізації сенсорної мережі.

Методи дослідження. Аналіз технологій IoT, синтез типології мережі із застосуванням БПЛА, експеримент побудови каналу LoRa

Отримані результати. Методологія побудови літальної сенсорної мережі для Інтернету речей. Досліджувана реалізація системи характеризується відносно низькою вартістю вузлів та значним охопленням території. Аналіз технології LoRa дає можливість розгорнути БСМ на великі відстані. Використовуючи маленьку схему таку як вузол - мобільний вузол - шлюз можна досягнути охоплення території протяжністю 80 км. А за допомогою веб-додатку можна відслідковувати збирані дані у реальному часі.

ABSTRACT

The thesis contains 63 pages, 18 figures. 32 used sources were used.

QUADROCOPTER, ENVIRONMENTAL MONITORING, INTERNET OF THINGS, FLIGHT SENSOR NETWORK, SJRC F7 4K PRO, LOPY4, LORA, LORA MESH, LORAWAN.

Relevance of work. The relevance of using aerial sensor networks to build the Internet of Things can be explained by several key factors, such as coverage of territories, real-time data transmission, flexibility and scalability, and innovative capabilities. Airborne sensor networks can cover much larger areas and are able to move quickly and work in different locations, which allows you to provide connectivity to the Internet of Things even in remote or hard-to-reach places. They can be connected to cloud computing systems, which allows for the transfer of collected data in real time. This is important for the implementation of fast-reacting systems. In general, the application of aerial sensor networks to build the Internet of Things has great potential in various fields, contributing to the development of technology and improving the quality of people's lives.

Goal. Application of an aerial sensor network for building the Internet of Things. A quadcopter with an attached sensor based on LoRa will be used as an aerial network, thanks to which, according to the LoRa Mesh and LoRaWAN protocols, data from the sensors will be collected and transmitted to the TTN server for further processing.

The object of research is the process of functioning of wireless sensor networks using unmanned aerial vehicles (UAVs).

The subject of research is methods of building wireless sensor networks using UAVs.

The purpose of the research is to develop a methodology for building wireless sensor networks with applications (UAVs) based on LoRa technology.

The set goal involves solving the following tasks:

1. To analyze the main characteristics of sensor networks in the context of using a drone as a mobile flying platform.

2. Determine the functional features of sensor networks with subsequent selection of the type of drone.

3. To analyze the possibilities of LoRa and TTN technologies.

4. Describe the characteristics of the developed system.

5. Consider the issue of software implementation of a sensor network.

Research methods. Analysis of IoT technologies, synthesis of network typology using UAVs, experiment of building a LoRa channel.

The results obtained. Methodology for building an aerial sensor network for the Internet of Things. The studied implementation of the system is characterized by a relatively low cost of nodes and a significant coverage of the territory. Analysis of LoRa technology makes it possible to deploy BSM over long distances. Using a small scheme such as a node - mobile node - gateway, it is possible to reach an area coverage of 80 km. And with the help of a web application, you can monitor the collected data in real time.

ЗМІСТ

ВСТУП.....	12
РОЗДІЛ 1	13
АНАЛІЗ ПОТОЧНОЇ СИТУАЦІЇ ЗА НАПРЯМКОМ ДОСЛІДЖЕНЬ.....	13
1.1 Основні принципи та характеристики сенсорних мереж.....	13
1.2 Технології збору та передачі даних з сенсорних мереж	15
1.3 Технічні можливості дронів для збору та передачі даних	18
Висновок.....	22
РОЗДІЛ 2	24
МЕТОДИКА ПОБУДОВИ ЛІТАЛЬНОЇ СЕНСОРНОЇ МЕРЕЖІ ІОТ	24
2.1 Вибір типу дрона для збору та передачі даних	24
2.2 Технологія LoRa контексті систем ІоТ	27
2.2.1 Особливості передачі даних	27
2.2.2 Протокол LoRaWAN	32
2.2.3 Вузол LoRa	38
2.3 Мережа TTN	42
2.4 Розгляд сценарію застосування технології LoRa та дронів	45
2.5 Характеристика вибраних технологій.....	47
Висновок.....	49
РОЗДІЛ 3	50
АНАЛІЗ ТА ОБРОБКА ДАНИХ ЗІБРАНИХ З ДРОНІВ	50
3.1 Характеристика експлуатації розроблюваної системи	50
3.2 Питання програмної реалізації	52
3.3 Особливості системи.....	58
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
ДОДАТОК 1	65

ПЕРЕЛІК СКОРОЧЕНЬ

БПЛА	Безпілотний літальний апарат.
БСМ	Бездротова сенсорна мережа.
FANET	Flying Ad-Hoc Network
VANET	Vehicular Ad hoc Network
MANET	Mobile Ad hoc Network
РПС	Рівень потужності сигналу
IoT	Internet of Things
SF	Spreading Factor
БАП	Бездротова активація пристрою
TTN	The Things Network
TTS	The Things Stack
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
ЛСМ	Літальна сенсорна мережа
АПП	Активація пристрою за допомогою персоналізації

ВСТУП

Безпілотні літальні апарати (БПЛА), такі як дрони, до категорії яких включено квадрокоптери, створені завдяки значного прогресу у сфері обчислювальної техніки, комунікацій та датчиків, а також мініатюризації пристроїв. Вони отримують значну увагу в дослідницькому співтоваристві. Завдяки простоті розгортання, низьким витратам на технічне обслуговування та високій маневреності БПЛА стають невід'ємним компонентом у таких критично важливих програмах, як спостереження за кордоном, моніторинг стихійних лих, моніторинг руху та дистанційне зондування. Такі літальні платформи також використовуються у військових, сільськогосподарських та промислових програмах. Один або декілька БПЛА зазвичай використовуються як ретранслятори зв'язку або базові станції для забезпечення функціональності відповідної мережі [1]. Зовсім недавно почали з'являтися нові можливості щодо комерційного застосування та суспільних послуг для БПЛА, які можуть суттєво змінити повсякденне життя, як-от служба авіа доставки.

Дійсно, кілька проектів у Європі досліджують використання повітряних базових станцій для встановлення мережевих зв'язків і спеціального радіопокриття під час несподіваних і тимчасових подій. Крім того, включення БПЛА в наземні мережі привертає значну увагу дослідницького співтовариства в даний час. Мережі з підтримкою БПЛА, класифіковані як мережі FANET [2], є особливою формою мобільних спеціальних мереж - MANET і транспортних спеціальних мереж - VANET. FANET мають відмінні характеристики від інших форм мереж, такі як мобільність вузла, зміна частоти щільності вузла та топології мережі тощо. Мобільність є головною проблемою в мережі з підтримкою БПЛА. Такі відмінні характеристики вимагають розробки та тестування нових протоколів зв'язку в багаторівневому підході, придатному для FANET.

РОЗДІЛ 1

АНАЛІЗ ПОТОЧНОЇ СИТУАЦІЇ ЗА НАПРЯМКОМ ДОСЛІДЖЕНЬ

1.1 Основні принципи та характеристики сенсорних мереж

Бездротова сенсорна мережа (БСМ) - це мережа, що складається з великої кількості малопотужних пристроїв, які сприймають навколишнє середовище та надсилають свої показання одному чи кільком приймачам [3]. Пристрої, що входять до складу БСМ, називаються сенсорними вузлами, і вони мають такі характеристики:

- компактні розміри;
- мають обмежену пам'ять, обчислювальну потужність та енергоспоживання (більшість із них живляться від батарейок);
- складаються з компонентів зондування, обробки даних і зв'язку;
- мають обмежену дальність зв'язку та швидкість передачі даних.

Як правило, мережі БСМ щільно розгорнуті через наведені вище характеристики характеристики.

Категорії датчиків

Робота з широким діапазоном типів програм навряд чи буде можливою з будь-яким одним типом сенсорного вузла. Тим не менш, деякі спільні риси проявляються, особливо щодо характеристик і необхідних механізмів таких мереж. Дійсно, у більшості застосувань датчики вимагають готовності до розгортання на місцях з точки зору економічної та інженерної ефективності. Масштабованість датчика також важлива в завданнях розподіленого моніторингу навколишнього середовища, які вимагають, щоб датчики були малими та недорогими, щоб масштабувати їх до багатьох розподілених систем. Виробництво таких датчиків розгорнуто мільйонами. Таким чином, очікується, що вартість знизиться, але датчики поточного покоління все ще дорогі, щоб забезпечити їх широке застосування [4].

Датчики можна класифікувати за тим, де вони розгортаються або використовуються [4]. Наприклад, для моніторингу якості води фізичні датчики, як правило, більш готові до роботи в полі та масштабовані, ніж хімічні датчики, які, у свою чергу, значно більш готові до роботи в полі та масштабовані, ніж біологічні датчики.

Вузли БСМ

Сенсорні вузли можуть бути неоднорідними за конструкцією, тобто деякі вузли мають більші батареї, комунікаційні пристрої з більшою віддаленістю або більшу обчислювальну потужність. Вони також можуть бути гетерогенними внаслідок еволюції, тобто всі вузли почали з однакового стану, але оскільки деякі вузли повинні були виконувати більше завдань під час роботи мережі, вони вичерпали свої енергетичні ресурси або інші вузли мали кращі можливості для збирання інформації від навколишнього середовища.

Незалежно від того, чи то конструкція, чи то еволюція, неоднорідність мережі є одночасно обмеженням і можливістю. БСМ складається з сенсорних вузлів з різними можливостями, такими як різні типи датчиків і діапазон зв'язку, що забезпечує більшу гнучкість у розгортанні. Наприклад, можна побудувати БСМ, у якій вузли оснащені різними типами датчиків для надання різноманітних служб зондування [5].

У сенсорних бездротових мережах ще потрібно вирішити кілька проблем, таких як визначення теоретичної потужності БСМ, інтерпретації різних технологій, мобільності, якості обслуговування тощо. Є кілька переваг БСМ на відміну від традиційної однорідної бездротової мережі, включаючи підвищену надійність, покращену ефективність використання спектру та збільшене покриття. Надійність покращується, оскільки коли одна конкретна технологія доступу в межах БСМ дає збій, можливо, все ще можна підтримувати з'єднання, повернувшись до іншої технології доступу.

1.2 Технології збору та передачі даних з сенсорних мереж

БСМ широко використовуються в кількох сферах застосування, починаючи від військових, сільського господарства та закінчуючи моніторингом якості показників навколишнього середовища. Збір даних є одним із найважливіших питань у БСМ, і ця проблема є рушієм значної кількості досліджень протягом десятиліть. Традиційно схеми збору даних базуються на статичній топології, де вузли статично розгорнуті, але пізніше на мобільних вузлах, які є більш енергоефективними.

Основна ідея збору даних полягає в тому, щоб сприймати дані та передавати їх збирачам (наприклад, центральний сервер) для подальшої обробки. Традиційно передача даних здійснювалася кількома етапами. Вузли ретрансляції допомагають простим вузлам пересилати свої дані. Таким чином, вони швидко втрачають енергію, що зрештою призводить до втрати покриття. Потім були представлені мобільні вузли в такому контексті. Мобільні вузли переміщуються по мережі заздалегідь визначеним або випадковим шляхом для збору та пересилання даних. Збір даних за допомогою мобільних вузлів складається з трьох етапів [6], що наведені нижче.

1) *Введення*. На цьому етапі мобільні колектори транслюють повідомлення присутності на свою зону покриття, щоб повідомити вузли в межах свого діапазону про прибуття колектора. Датчики, які отримали повідомлення, можуть визначити наявність мобільного колектора.

2) *Передача даних*. Тут вузли починають надсилати дані після того, як вони ідентифікують присутність збирача. Метою етапу передачі даних є досягнення максимальної пропускну здатності даних протягом обмеженого часу сеансу між збирачем і простими вузлами. Загалом, обидва: вузли та збирачі є мобільними, збір даних у такому динамічному контексті включає багато важливих параметрів, серед яких тривалість сеансу комунікації та швидкість передачі даних між збирачами та простими вузлами, мають величезний вплив на збір даних.

3) *Маршрутизація*. На цьому етапі мобільні збирачі пересилають зібрані дані до приймача або базової станції (рис. 1.1).

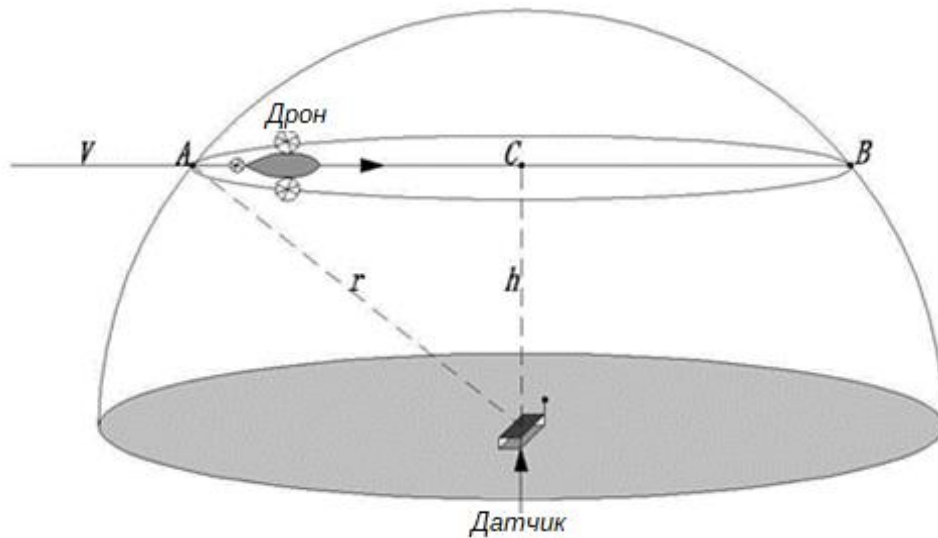


Рис. 1.1 Простий сценарій збору та передачі даних з одним дроном і одним датчиком [5]

За рис. 1.1, дрон рухається певним маршрутом, щоб забезпечити безперервне підключення та допомогти збалансувати навантаження на вузли. Датчик має можливість зв'язуватися з дроном, коли останній знаходиться в зоні його зв'язку. Дуже важливо знати переривчастість зв'язку в такому контексті.

Як показано на рис. 1.1, коли дрон летить над датчиком, пропускна спроможність передачі даних може досягти свого максимального значення. Чим довший час контакту між дроном та датчиком, тим більше повідомлень було надіслано між ними.

Механізм множинної швидкості

На продуктивність зв'язку впливають втрати на шляху, перешкоди, затінення тощо. Швидкість передачі даних між дроном та вузлами залежить від відносної відстані між ними, і відносна відстань змінюється з часом. Таким чином, швидкість передачі даних також змінюється залежно від руху мережі. Отже, більш розумно використовувати механізм множинної швидкості передачі даних між різними вузлами протягом різного часу.

Парне налаштування параметрів зв'язку [7] використовується в багатьох програмах. Чим вище швидкість передачі даних між дроном та вузлом, тим більше даних передано між ними.

Відповідно до розгорнутого середовища, де використовується програма, схеми збору даних можна приблизно розділити на:

1) *Схеми збору даних на основі статичної топології.* У цій категорії як вузли датчика, так і колектор розгорнуті статично.

2) *Схеми збору даних на основі мобільної топології.* У мобільній топології конструкція вузлів і колекторів залежить від додатків; деякі програми потребують лише рухомих вузлів або лише рухомих колекторів, тоді як інші програми повинні розглядати обидва рухомі елементи.

Класифікація базових протоколів збору даних

Протоколи зв'язку з одним ступенем

У цьому випадку вузли джерела, які знаходяться в межах діапазону вузла призначення, можуть безпосередньо комунікувати з колекторами. Передача в цьому контексті вимагає відповідної відстані між вузлами-джерелами та вузлами-одержувачами, і між ними немає перешкод, які б заважали їм комунікувати один з одним. Якщо в діапазоні цільового вузла є декілька вузлів одночасно, середовище можна спільно використовувати відповідно до різних протоколів, які можна далі класифікувати як засновані на: конфлікті протоколи, безконфліктні і гібридні протоколи. Було запропоновано багато протоколів щодо цих трьох класифікацій, таких як В-МАС [8] (на конфліктній основі), FLAMA [9] (безконфліктна основа), Z-МАС [10] (гібридна основа) тощо.

Багатоступеневі протоколи зв'язку

Для вихідних вузлів, які знаходяться поза зоною дії вузла призначення або мають погані умови передачі (наприклад, високе енергоспоживання, великі втрати пакетів тощо), краще надсилати пакети за допомогою проміжних вузлів, які знаходяться в зоні дії вузла. Вузол призначення та вузол джерела одночасно або мають кращі ситуації зв'язку. Групу проміжних вузлів зазвичай називають потенційними ретрансляторами. Використання механізмів в основному

залежить від мережевої інформації, що надається кожним вузлом. У цьому сенсі протоколи з кількома ступенями можуть бути додатково класифіковані на протоколах маршрутизації.

Протоколи маршрутизації

Протоколи маршрутизації в сенсорних мережах є дуже складними через кілька характеристик:

1) Як правило, сенсорні вузли живляться від батареї, вони суворо обмежені щодо потужності передачі, бортової енергії, обчислювальних можливостей і зберігання, і тому вимагають ретельного управління ресурсами.

2) У великомасштабних сенсорних мережах важко розробити глобальну схему адресації для розгортання сенсора.

3) У більшості додатків БСМ зазвичай вимагається передача отриманих даних від кількох вузлів джерела до одного вузла призначення.

4) Трафік генерації даних має певну надлишковість у багатьох програмах, оскільки кілька датчиків можуть генерувати ті самі дані. Така надлишковість повинна використовуватися протоколами маршрутизації для покращення продуктивності мережі.

1.3 Технічні можливості дронів для збору та передачі даних

Безпілотні літальні апарати (БПЛА), а у контексті поточної роботи - дрони, набули великої популярності в різних сферах застосування, які не потребують взаємодії з людиною або є небезпечними (наприклад, під час урагану дрони використовувалися для пошуку та порятунку тих, хто вижив [11]) для людей-операторів. Дрони широко застосовуються у різних сферах життя, від ранніх військових, екологічних і міських додатків до сучасних додатків Facebook, Google і Amazon. Facebook успішно випробував свої безпілотники, що транслюють інтернет і Amazon забезпечує спеціальну доставку продукту протягом 30 хвилин [2].

Завдяки прогресу в обчислювальній техніці, комунікації та датчиках, а також мініатюризації пристроїв, дрони, такі як повітряні кулі, квадрокоптери та планери, приділяють значну увагу дослідницькому співтовариству. Вони стають невід'ємним компонентом у кількох критично важливих програмах, таких як нагляд за кордоном, військові операції [12], моніторинг стихійних лих, моніторинг руху, дистанційне зондування, а також транспортування вантажів, медикаментів та надання першої допомоги.

Серед багатьох технічних проблем, які супроводжують вищезгадані програми, використання дрона для забезпечення широкопasmового з'єднання відіграє центральну роль у системах зв'язку наступного покоління [13]. Так, Facebook і Google оголосили, що використовуватимуть мережу дронів, які кружляють у стратосфері над певними населеними пунктами, щоб забезпечити широкопasmовий зв'язок. Дрони також були запропоновані як ефективне рішення для передачі широкопasmових даних в надзвичайних ситуаціях через низьковисотні платформи. Вони можуть служити тимчасовою, динамічною та гнучкою інфраструктурою для забезпечення широкопasmового зв'язку та швидкої локалізації постраждалих у разі надзвичайних ситуацій.

Категорії дронів

Дрони можуть бути дистанційно керованими літальними апаратами (наприклад, керованими людиною на наземній станції управління) або можуть літати автономно на основі попередньо запрограмованих планів польоту або більш складних динамічних систем автоматизації. Дрони можуть бути квадрокоптерами, планерами тощо, а також можуть бути їх вдосконаленими моделями, що переносять корисне навантаження (наприклад, мікродрони, перенесення камер).

Існує більше ніж одна категорія, коли йдеться про класифікацію «дрон». Як правило, дрони класифікуються за розміром, радіусом дії та витривалістю тощо. Витривалість - це час польоту дрона, а дальність - робочий радіус дрона.

Мережа Flying Ad-Hoc (FANET)

Спеціальні мережі - це бездротові мережі, які можуть організовуватися без попередньо визначеної інфраструктури. Кожен вузол безпосередньо комунікує зі своїми сусідами. Щоб комунікувати з іншими вузлами, необхідно передати свої дані іншим, які відповідатимуть за їх пересилання. Для цього перш за все важливо, щоб вузли були розташовані відносно один одного і могли будувати маршрути між собою: це роль протоколу маршрутизації. Таким чином, робота мережі «ad-hoc» суттєво відрізняє її від таких мереж, як стільникова мережа або мережа Wi-Fi з точками доступу, де одна або кілька базових станцій потрібні для більшості зв'язків між різними вузлами мережі, спеціальні мережі організовуються самі, і кожен вузол може виконувати різні функції.

Подібним чином спеціальна мережа Flying (FANET) - це просто спеціальна мережа між кількома дронами (рис. 1.2). Вона є частиною транспортних спеціальних мереж VANET (Vehicular Ad hoc Network), які самі є частиною мобільних спеціальних мереж MANET (Mobile Ad hoc Network).

FANET - це комунікаційна мережа з декількома дронами, де ступінь мобільності вузлів набагато вищий, ніж ступінь мобільності вузлів MANET або VANET. У той час як типові вузли MANET і VANET - це люди у русі та автомобілі відповідно, вузли FANET переміщуються у повітряному просторі (здійснюють польоти) [4].



Рис. 1.2 Комунікація між FANET, VANET і MANET [4]

Використання дронів значно розширює можливості розгортання, і їх передбачувані застосування будуть набагато різноманітнішими в повсякденному житті. Програми для БСМ з підтримкою дронів можуть бути або традиційними, як домашня автоматизація; моніторинг навколишнього середовища та середовища існування; промисловий моніторинг і контроль; військовий; безпека; і охорона здоров'я або абсолютно нові типи програм. Дійсно, дрони забезпечать швидке та просте розгортання, особливо в складних та небезпечних умовах. Вони можуть запропонувати сенсорній мережі швидкий доступ до більш важливої технології з точки зору наскрізної затримки та пропускну здатності, що дозволяє передавати різні типи даних, включаючи місцезнаходження систем глобального позиціонування (англ. GPS), потокове відео/голос, зображення тощо. Це набагато важливіше в нових напрямках, таких як рятувальні операції.

Далі підсумовуються деякі вже розгорнуті програми:

1) Військове застосування, включаючи військовий бій, операції на полі бою та військовий зв'язок [14].

2) Застосування в навколишньому середовищі. Homeland Surveillance & Electronics (HSE) пропонує 11 моделей сільськогосподарських дронів, обприскувачів, 6 гелікоптерів і 5 мультикоптерів для моніторингу вологи, паразитів і росту культур тощо [15].

3) Міські програми, такі як моніторинг дорожнього руху, міське спостереження та цивільна безпека [16].

4) Промислове застосування, включаючи моніторинг якості продукції та вимірювання Smart-Grid. Крім того, програми також можна класифікувати за зоною, де розгортаються датчики. У першому класі вузли розміщені на землі для моніторингу посівів або під землею, наприклад для безпеки та моніторингу трубопроводів [17].

Категорія промислового застосування є найбільш широко використовуваною і зазвичай не може використовуватися під водою. Другий клас відповідає підводному розгортанню, де датчики використовують

акустичний зв'язок. Цей клас застосування зазвичай коштує дорого через захист датчиків від води, коли вони були розгорнуті протягом тривалого часу. Третій клас - гібридний. Він поєднує в собі обидва попередні режими розгортання, що не тільки розширює сферу застосування, але й знижує вартість. В [1] наприклад, сенсорні вузли плавають у межах обмеженої зони на поверхні моря для моніторингу морських катастроф.

Незалежно від сфери застосування, дрони або БСМ мають різні функції. Дрон в основному бере участь у перевірці пакетів, зборі даних, підтримці з'єднання БСМ і локалізації. В інших місцях БСМ відповідає за збір даних, моніторинг, відстеження об'єктів, виявлення та обробку подій.

Після цього режим збору та вилучення даних є ключовою характеристикою в цих програмах і має важливий вплив на продуктивність усієї системи. Крім того, він визначає траєкторію польоту дрона (запланована чи незапланована траєкторія, нерухомий чи рухомий).

Режими збору та пошуку даних можна розділити на [18]:

- 1) На основі запитів, де вузли надсилають дані лише за потреби.
- 2) Керований подією, вузли передають дані на дрон лише тоді, коли відбувається подія.
- 3) У часі, вузли датчиків періодично передають свої показання дронам. Застосування, кероване часом, є найпоширенішим і широко використовується у військових і екологічних програмах.

Висновок

З цього розділу впливає наступне:

- 1) Літальні сенсорні мережі є перспективною технологією для побудови Інтернету речей. Вони використовують безпілотні апарати (наприклад, дрони), обладнані датчиками, для збору даних і передачі їх у реальному часі.
- 2) Застосування ЛСМ для побудови IoT відкриває нові можливості в таких галузях, як моніторинг навколишнього середовища, агрокультура, логістика та безпека.

3) ЛСМ можуть здійснювати нагляд за важкодоступними або віддаленими областями, збирати дані про навколишнє середовище, а також виявляти небезпеки та здійснювати візуальний контроль.

Розділ дозволяє проаналізувати напрямок дослідження та зробити обґрунтовані рішення під час проектування ЛСМ для Інтернету речей.

РОЗДІЛ 2

МЕТОДИКА ПОБУДОВИ ЛІТАЛЬНОЇ СЕНСОРНОЇ МЕРЕЖІ ІОТ

2.1 Вибір типу дрона для збору та передачі даних

Однією з основних цілей додатків БСМ є обробка значущої інформації з даних, отриманих сенсорними вузлами, розгорнутими у полі. Таким чином, отримані дані повинні бути передані в приймач для обробки та отримання важливих висновків.

Традиційно збір даних оброблялося в кілька етапів. Вузли, розташовані ближче до колектора, стають ретрансляційними вузлами для даних інших вузлів, які знаходяться далі від колектора. У результаті ці вузли втрачають енергію набагато швидше порівняно з тими, яким не доводиться часто працювати як вузли ретрансляції. Отже, більш швидка деактивація цих вузлів призводить до отримання відключеної мережі, що, у свою чергу, призводить до втрати покриття. Для вирішення цієї проблеми були введені мобільні вузли (наприклад, на базі дронів).

Традиційні мобільні вузли зазвичай переміщуються з обмеженими швидкостями та умовами руху. Застосовувати традиційних мобільних вузлів в складних умовах (таких як снігові гори, дикий ліс тощо) є складним. Дрони стають кращим вибором для таких застосувань через широкі та гнучкі можливості застосування. Як комунікаційний вузол у БСМ, основні функціональні можливості дрона, включають збір даних, підтримку зв'язку та локалізацію тощо.

Локалізація

Послуги на основі визначення місця розташування відіграють усе більш важливу роль у повсякденному житті людей. Інформація про місцезнаходження має велике значення для розуміння подій, виявлених у полі зондування. Локалізація датчиків важлива для нормальної роботи БСМ. Розташування датчиків є фундаментальною та важливою проблемою бездротової сенсорної мережі.

Щоб гарантувати зону охоплення та хорошу точність локалізації, БСМ зазвичай складається з великої кількості сенсорних вузлів із високою щільністю розгортання в полі. Як правило, датчики розгортаються без попередньої інформації про їх місцезнаходження. Простий спосіб отримати інформацію про місцезнаходження - розмістити їх у певних місцях вручну. Однак, коли кількість датчиків велика, це стає складним процесом.

GPS [19] є ще одним популярним способом, який може запропонувати хорошу точність локалізації. Однак проблематично оснастити кожен сенсорний вузол модулем GPS з урахуванням факторів енергоспоживання, обсягу та вартості.

Схеми, засновані на локалізації транспортних засобів, поділяються на локалізацію статичних транспортних засобів і локалізацію мобільних транспортних засобів. У статичному випадку, точність визначення місцезнаходження сильно залежить від кількості та розташування транспортних засобів. Функція рівномірного розподілу та щільного розміщення транспортних засобів призведе до кращої точності локалізації, але також до додаткових витрат. Для подолання цієї проблеми пропонуються схеми з мобільними транспортними засобами. Мобільний транспортний засіб може отримати інформацію про власне місцезнаходження за допомогою GPS або інших технологій локалізації, і він рухається по зоні зондування, транслюючи свої поточні координати у формі повідомлення-маяка. Датчик, який отримав сигнал маяка, може визначити, що він знаходиться в радіусі зв'язку мобільного засобу. Коли отримано достатню кількість маяків, датчик може визначити своє місцезнаходження за допомогою кількох запропонованих методів.

У контексті БСМ з підтримкою мобільного засобу, автори у роботі [20] розглядають проблему тривимірної (3D) локалізації в БСМ за допомогою дрона. Дрон оснащений GPS і літає над зоною спостереження, транслюючи своє географічне положення. Таким чином, сенсорні вузли здатні оцінювати своє географічне положення без оснащення GPS-приймачем. GPS є ефективним методом для оцінки положення у відкритому просторі. GPS краще

використовувати на достатній відстані від будівель або перешкод, інакше сигнали GPS стануть ненадійними. У БСМ з підтримкою дрона, останні потребують додаткової оцінки, щоб продовжувати працювати після збою GPS.

Таким чином, на основі аналізу умов експлуатації дрона у контексті мобільної платформи та у складі сенсорної мережі, визначено, що доцільним варіантом є квадрокоптер SJRC F7 4K Pro (рис. 2.1).



Рис. 2.1 Квадрокоптер SJRC F7 4K Pro

Зазначений дрон має наступну технічну характеристику:

- 1) Вартість: 8000-9000 грн (станом на 2023 рік).
- 2) Рекордна дальність польоту для свого класу: 3000 м (25 хв.).
- 3) Тип керування: пульт дистанційного керування та смартфон.
- 4) Камера з якістю зображення 4K та відповідний підвіс зі стабілізацією у трьох осях (EIS).
- 5) Вбудовані модулі: GPS та гіроскоп.
- 6) Режими інтелектуального польоту (автопілот): «Політ за точками», «Політ по колу», «Слідування за об'єктом», «Повернення додому».
- 7) Ємність вбудованого акумулятора: 2500 мА·г.

Ще однією важливою особливістю даного дрона, є надійна конструкція вбудованих опорних стійок. Так, можна налаштовувати функціонал літальної сенсорної мережі, комбінуючи необхідні модулі (наприклад, можна

демонтувавши відеокамеру та встановити додаткову акумуляторну батарею), які будуть надійно захищені при встановленні на штатне місце для навісного обладнання.

2.2 Технологія LoRa контексті систем IoT

LoRa - бездротова технологія передачі даних за допомогою радіохвиль, запатентована компанією Semtech, де LoRa означає «Long Range», і типовий пристрій, що використовує цю технологію, має наступні характеристики [21]:

1) *Велике охоплення.* Пристрій LoRa може приймати і відправляти сигнали в міській місцевості на відстані 5 кілометрів, в сільській місцевості радіус дії становить до 15 кілометрів, а якщо передавач і приймач знаходяться в прямій видимості, радіус дії сигналу становить десятки-сотні кілометрів.

2) *Низька швидкість передачі даних.* Швидкість передачі даних коливається від 0,3 до 37,5 кілобіт на секунду в залежності від налаштувань пристрою. Технологія LoRa використовується для надсилання невеликих даних розміром близько 12 байт, які не сильно змінюються з часом і надсилаються з інтервалом у хвилини або години.

3) *Низьке енергоспоживання.* Оскільки дані, що передаються пристроєм LoRa, невеликі, а часовий інтервал між повідомленнями великий, споживання такого пристрою дуже низьке і вимірюється в одиницях міліват. Тому передавач LoRa може житися від батареї та служити десятиліттями.

2.2.1 Особливості передачі даних

Рівень потужності отриманого сигналу (РПС) або (англ. RSSI), і чутливість приймача – Rx визначають, чи зможе приймач обробити повідомлення від передавача. І РПС, і чутливість приймача вказуються в дБм (англ. dBm). Значення РПС має бути більше, ніж значення чутливості приймача для обробки сигналу.

При передачі сигнал може бути посилений або ослаблений. Можна посилити сигнал за допомогою радіомодуля, антен і приймача. Сигнал може

бути послаблений в кабелях і на шляху передачі (відстань, перешкоди, погодні умови). Значення РПС визначається сумою цих позитивних факторів і втрат. На рис. 2.2 показано залежність потужності сигналу від відстані передавача до приймача.

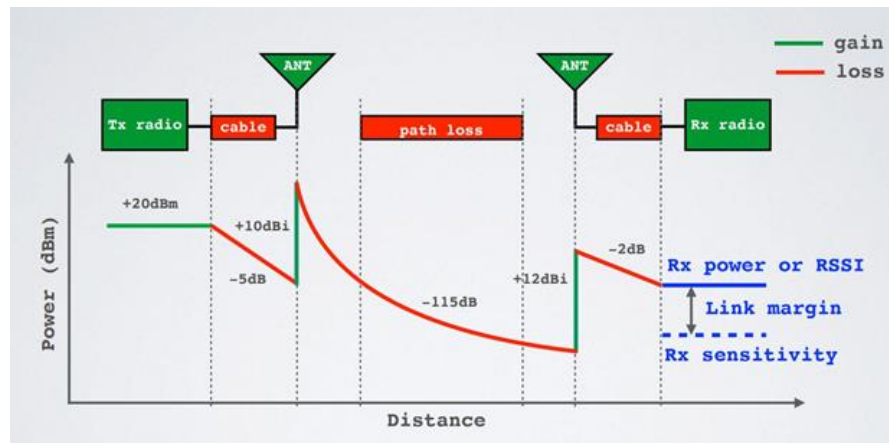


Рис. 2.2 Характеристика РПС [22]

Коефіцієнт посилення передавачів LoRa обмежений значеннями, близькими 20 дБм, а чутливість приймачів LoRa може досягати -148 дБм. Так, максимальна потужність сигналу зв'язку - 168 дБм. Для порівняння, сигнали Wi-Fi мають середній рівень 100 дБм. Тут важливо зазначити, що залежність між потужністю сигналу та відстанню не є лінійною. Додавши 6 дБм до загального значення, можна подвоїти діапазон сигналу. З'єднання LoRa, яке має на 68 дБм більшу потужність з'єднання, ніж з'єднання Wi-Fi, теоретично має діапазон у 4000 разів більший.

LoRa має високі значення посилення через те, як тут модулюються дані. Вона використовує технологію Chirp Spread Spectrum для модуляції даних. Це метод кодування інформації в так звані тріскотіння, які є повторюваною лінійною зміною частоти, яка збільшується або зменшується. Якщо частота лінійно збільшується з часом до свого найвищого значення, це буде висхідний тріскіт (рис. 2.3а,б), а якщо частота лінійно зменшується з часом до найнижчого значення, це низхідний тріскіт (рис. 2.3в). Інформація інтегрується в ці сигнали за допомогою стрибків між частотами (рис. 2.4).

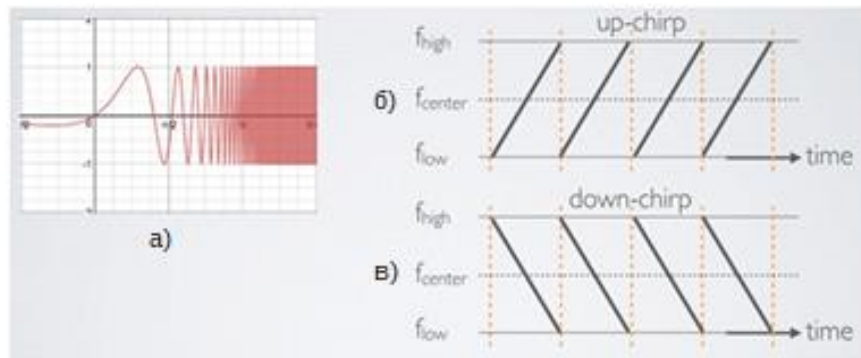


Рис. 2.3 Тріскіт а), б) висхідний; в) низхідний [23]

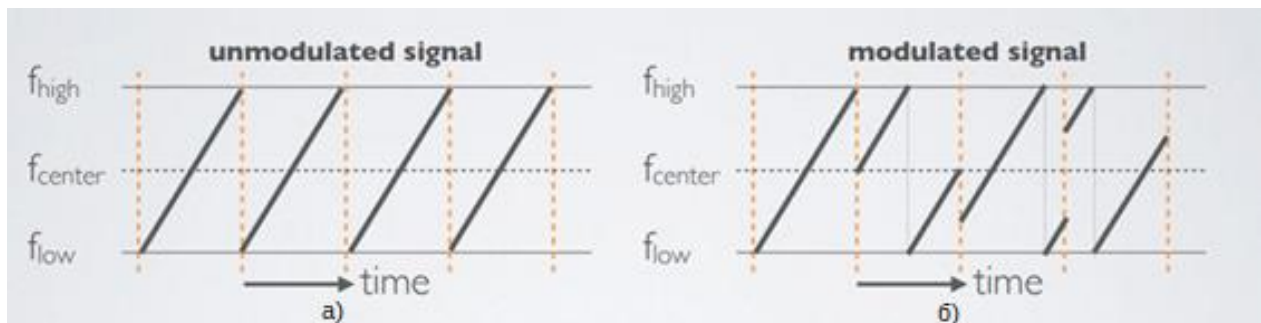


Рис. 2.4 Модуляція сигналу LoRa а) немодульований сигнал; б) модульований сигнал [23]

Кількість бітів, які можна вставити в один тріскіт, визначається так званим параметром коефіцієнта поширення (англ. SF). В одному тріскоті може бути значення $2SF$. Більшість пристроїв LoRa пропонують коефіцієнт поширення від SF7 до SF12.

Різниця між верхньою та нижньою межами, в яких рухається частота, називається смугою пропускання (англ. BW - BandWidth). Одиницею пропускну здатності є Герц.

Швидкість передачі даних (Data Rate, (DR)) залежить від пропускну здатності, коефіцієнта поширення та швидкості кодування (англ. CR), яка має значення від 1 до 4.

У більшості випадків значення BW є постійним і змінюється лише SF. Так, збільшення SF зменшить швидкість передачі даних і збільшуючи SF, можна збільшити час передачі, отже, дальність сигналу (рис. 2.5).

Існують певні правила оператора мережі, які обмежують час передачі пристроїв LoRa, тому, збільшуючи SF, також обмежується частота надсилання повідомлень.

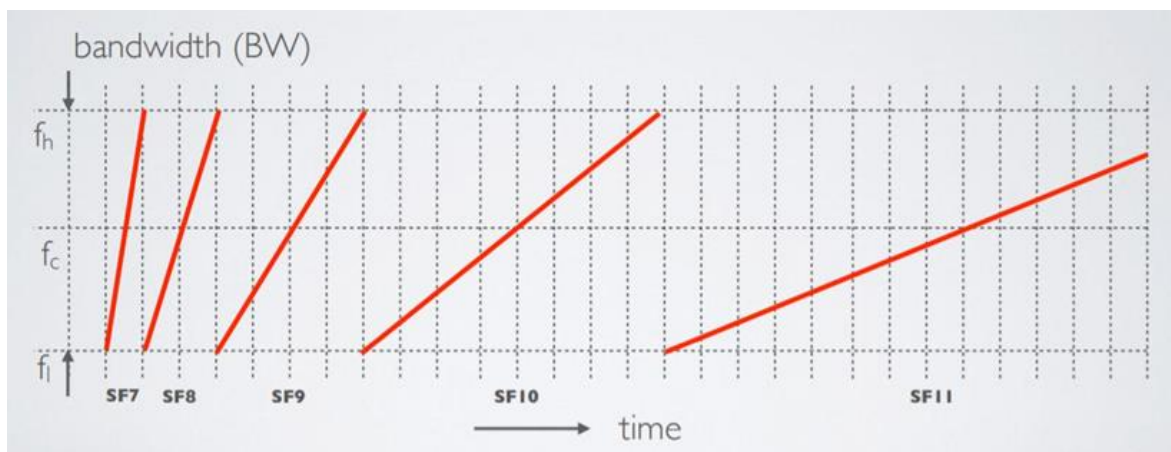


Рис. 2.5 Тріскіт на різних SF [23]

Технологія LoRa та IoT

Зв'язок LoRa може бути: «один-до-одного» (рис. 2.6а), тобто один пристрій є передавачем, а інший - приймачем; LoRa Mesh (рис. 2.6б), яка являє собою мережу пристроїв LoRa, які кумунікують один з одним; LoRaWAN (рис. 2.6в), який є протоколом зв'язку, де пристрій, так званий шлюз LoRaWAN, отримує повідомлення від кількох пристроїв LoRa, які він надсилає на мережевий сервер - це топологія «Зірка».

Пристрої LoRa, що працюють за протоколом LoRaWAN, відносяться до категорії Інтернет речей (IoT), яка являє собою мережу пристроїв, «речей», оснащених датчиками, програмним забезпеченням та іншими технологіями. Ці пристрої повинні мати підключення до мережі, що дозволяє їм кумунікувати один з одним і обмінюватися даними через Інтернет [24].

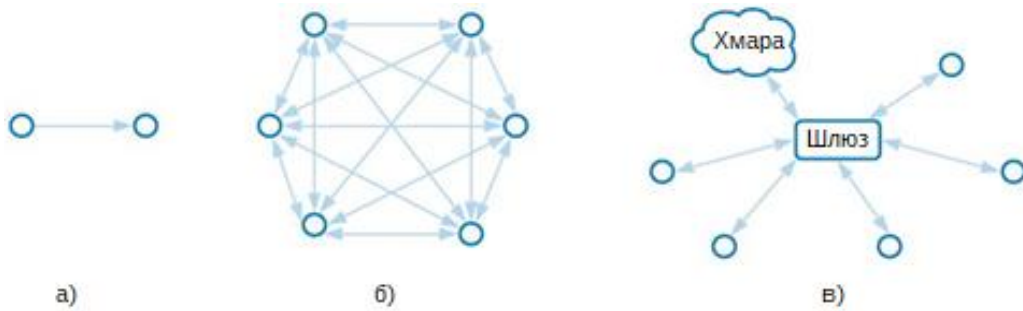


Рис. 2.6 Типи комунікацій LoRa: а) «один-до-одного»; б) LoRa Mesh; в) LoRaWAN [24]

Технології, які підпадають під IoT (рис. 2.7) (табл. 2.1) можуть бути бездротовими технологіями малого радіусу дії (Wi-Fi, Bluetooth), бездротовими технологіями середнього радіусу дії (3G, 4G, 5G) і бездротовими технологіями дальнього радіусу дії, до яких входить LoRaWAN, точніше до глобальної мережі малої потужності (LPWAN).

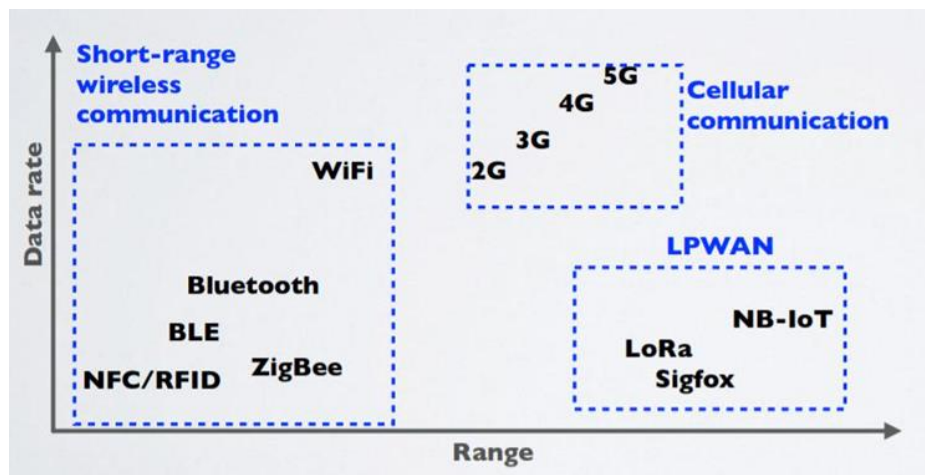


Рис. 2.7 Розподіл технологій IoT за охопленням [26]

Таблиця 2.1

Порівняння різних бездротових технологій

Бездротові технології	Діапазон (м)	Енергопоживання (мВт)
Bluetooth	~50	~2,5
Wi-Fi	~50	~80
3G/4G	~5000	~500
LoRa	2000-5000 (міська місцевість) 5000-15000 (сільська місцевість) >15000 (пряма видимість)	~20

2.2.2 Протокол LoRaWAN

LoRaWAN - це протокол зв'язку для пристроїв LoRa, створений LoRa Alliance. З одного боку у тут є апаратне забезпечення, яке надсилає дані за допомогою технології LoRa, а з іншого боку - програма IoT, розгорнута на сервері. LoRaWAN служить для двостороннього зв'язку, тому пристрій може надсилати дані на сервер, і в той же час можна надсилати дані з сервера на апаратне забезпечення.

Перспективним є використання архітектури LoRaWAN у розумному місті (контроль освітлення, паркування, моніторинг забруднення), промисловості (відстеження вантажів), сільському господарстві (моніторинг температури, вологості, рівня води), розумному домі (автоматизація побутової техніки) чи навіть у сфері охорони здоров'я (моніторинг здоров'я за допомогою носимих технологій).

Топологія

На рис. 2.8 наведено топологію LoRaWAN.

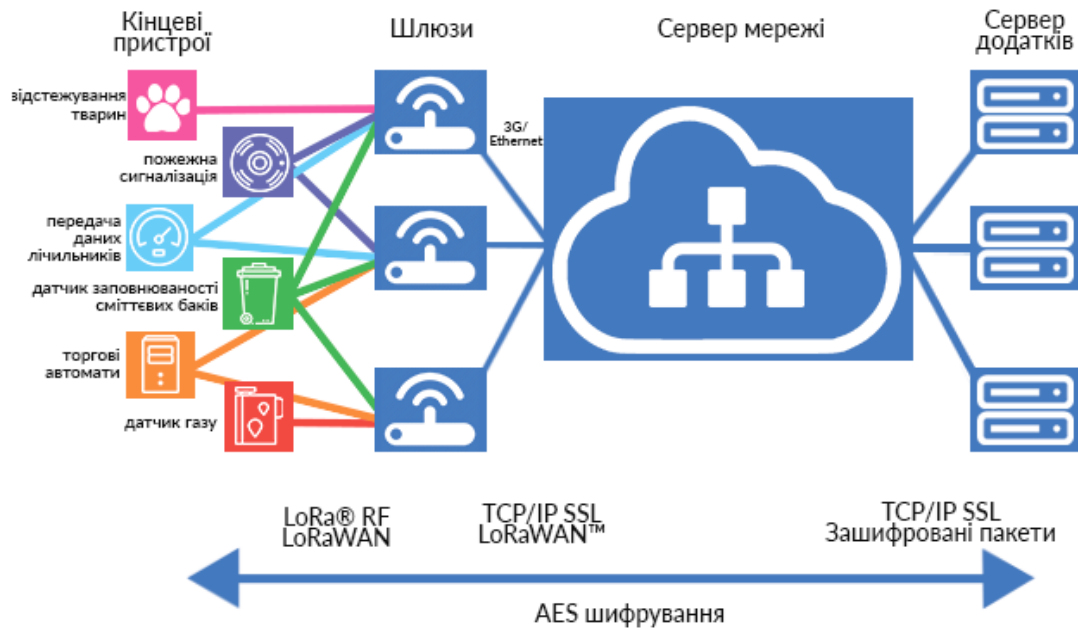


Рис. 2.8 Топологія протоколу LoRaWAN [26]

На рис. 2.8 можна побачити архітектуру LoRaWAN, яка складається з:

1) *Кінцевий вузол*. Кінцевий вузол або кінцевий пристрій - це пристрій, оснащений датчиками та радіомодулем LoRa, який може надсилати та отримувати дані одночасно. Повідомлення, яке надсилає кінцевий вузол, називається висхідним каналом (вони далі поділяються на підтверджені та непідтверджені висхідні канали), а повідомлення, яке отримує кінцевий вузол, називається низхідним каналом зв'язку. Кінцевий вузол служить, в деяких застосуваннях, як виконавчий механізм (керування світлом, замком, клапанами і т.д.).

2) *Шлюз*. Пристрій, підключений до Інтернету, який може декодувати радіосигнал від кінцевого вузла LoRa та надсилати його на мережевий сервер. Він служить точкою доступу між пристроями LoRa та Інтернетом, на зразок маршрутизатора Wi-Fi. Шлюз LoRaWAN отримує дані від усіх найближчих кінцевих вузлів, а кінцевий вузол надсилає дані на всі можливі шлюзи.

3) *Мережевий сервер*. Обробкою даних займається мережевий сервер. Оскільки кінцевий пристрій надсилає повідомлення на всі можливі шлюзи, генеруються дублікати повідомлень, які мережевий сервер фільтрує, розшифровує вхідні та шифрує вихідні повідомлення. Він вибирає найбільш

підходящий шлюз для надсилання низхідного каналу, перенаправляє висхідні канали на сервер додатків.

4) *Сервер додатків.* Сервер, на якому знаходиться сам додаток IoT, який працює з даними з кінцевого пристрою. Цей сервер може бути у приватній або загальнодоступній хмарі. Ця програма відповідає за створення низхідного каналу, який пересилає через мережевий сервер до кінцевого вузла.

Перевагою протоколу LoRaWAN є його масштабованість. Найдешевший вид шлюзу може обробляти сотні тисяч повідомлень на день. Якби один кінцевий вузол LoRa надсилав 10 повідомлень на день, вищезгаданий шлюз міг би отримувати повідомлення від 10 000 кінцевих вузлів. Якщо потрібно обробити більше повідомлень, необхідно додати інший шлюз.

Ще однією перевагою є вартість покриття. Оскільки це структура за архітектурою зірки, де один шлюз відповідає за кілька кінцевих вузлів, а ці пристрої мають радіус дії в кілометрах, відносно небагато шлюзів буде достатньо для створення широкого покриття для пристроїв LoRa. Ціна одного шлюзу становить приблизно 2000 грн (станом на 2023 рік).

Регулювання

LoRa Alliance розробив документ регіональних параметрів LoRaWAN [27], де вказано, в якому діапазоні частот може працювати кінцевий вузол LoRa (табл. 2.2). Це залежить від регіону, де знаходиться пристрій. Для кінцевого вузла, розташованого в Європі, він може передавати на частотах від 863 МГц до 870 МГц. Пристрій також має обмеження робочого циклу 1% або 0,1% залежно від використовуваного каналу. Пристрої з робочим циклом 1% можуть передавати 36 секунд на годину.

Крім того, для кінцевих вузлів LoRaWAN у Європі існують обмеження на висхідну та низхідну лінії зв'язку. Потужність передачі не повинна перевищувати 25 мВт для висхідних каналів і 0,5 Вт для низхідних каналів.

На додаток до цих правил можуть також існувати обмеження, введені окремими мережевими серверами.

Частотний план для різних регіонів [28]

Регіон	Частота (МГц)
Австралія	915-928
США	902-928
Європа, Індія, частини Африки	863-870
Канада	779-787
Китай	779-787, 470-510
Країни Азії	433

Безпека

Перш ніж кінцевий пристрій зможе кумунікувати за допомогою мережі LoRaWAN, пристрій LoRa має бути активовано.

Активований пристрій має такі значення:

1) *Адреса пристрою (DevAddr)*. DevAddr - це 32-розрядне число, яке ідентифікує кінцевий пристрій із мережевим сервером, який використовується. Це значення, яке призначає пристрою мережевий сервер.

2) *Ключ мережевого сеансу (NwkSKey)*. Це ключ, який використовують кінцеві пристрої і мережеві сервери під час обміну даними один з одним. Він використовується для кодування вихідних повідомлень і декодування вхідних повідомлень.

3) *Ключ сеансу програми (AppSKey)*. Цей ключ використовується в наскрізному зв'язку, тобто в зв'язку між кінцевим пристроєм і сервером додатків. Знову ж таки, він використовується для кодування та декодування вхідних і вихідних повідомлень. Повідомлення між сервером додатків і кінцевим пристроєм повинні проходити через мережевий сервер, який може впливати на повідомлення, що передаються, але не може бачити вміст. Мережеві сервери вважаються надійними.

Так, можна активувати кінцевий пристрій двома способами: бездротову активацію (БАП) (англ. OTC) і активацію пристрою за допомогою персоналізації (АПП) (англ. ABR). БАП є більш безпечною та кращою формою активації, коли АПП в основному використовується в розробці програм.

Бездротова активація кінцевого пристрою

Перш ніж спробувати активувати пристрій, у ньому повинні бути збережені три значення:

1) DevEUI, який є 64-розрядним унікальним ідентифікатором пристрою, більшість пристроїв уже мають вбудований DevEUI.

2) AppEUI, який є ідентифікатором сервера додатків, до якого кінцевий пристрій має надсилати повідомлення.

3) AppKey (ключ), який має бути однаковим як для пристрою, так і для мережевого сервера, використовується для перевірки цілісності повідомлень.

Кінцевий пристрій створює запит на приєднання, який містить DevEUI, AppEUI і два нових значення.

DevNonce - це випадково згенероване число, яке зберігається як пристроєм, так і мережевим сервером. Він використовується під час повторного підключення, щоб пристрій, який було виведено зі сну, не проходив повну активацію знову. Іншим значенням є код цілісності повідомлення (англ. MIC), який відповідає за цілісність повідомлення та генерується за допомогою AppKey.

Коли мережевий сервер отримує запит на приєднання, він перевіряє, чи використовувався раніше DevNonce, якщо так, це буде повторне приєднання. Якщо це новий DevNonce, мережевий сервер перевірить значення MIC за допомогою свого AppKey.

Якщо Join-Request успішно проходить перевірку, мережевий сервер створює Join-Accept. Join-Accept містить DevAddr, який відображає DevEUI пристрою на коротшу (32-розрядну) адресу пам'яті мережевого сервера. Також AppNonce або JoinNonce, які є випадково згенерованим числом. Потім у повідомленні Join-Accept вишукується ідентифікатор мережі та налаштування. Join-Accept шифрується за допомогою AppKey.

Після того, як пристрій успішно отримує повідомлення Join-Accept, пристрій LoRa та мережевий сервер спільно використовують AppNonce та DevNonce, які використовуватимуться для генерації NwkSKey та AppSKey,

мережевий сервер передає AppSKey та DevAddr на сервер додатків, що дозволяє зв'язок між кінцевим пристроєм і мережевим сервером, а також між кінцевим пристроєм і сервером додатків. Процес активації пристрою за допомогою БАП можна побачити на рис. 2.9.

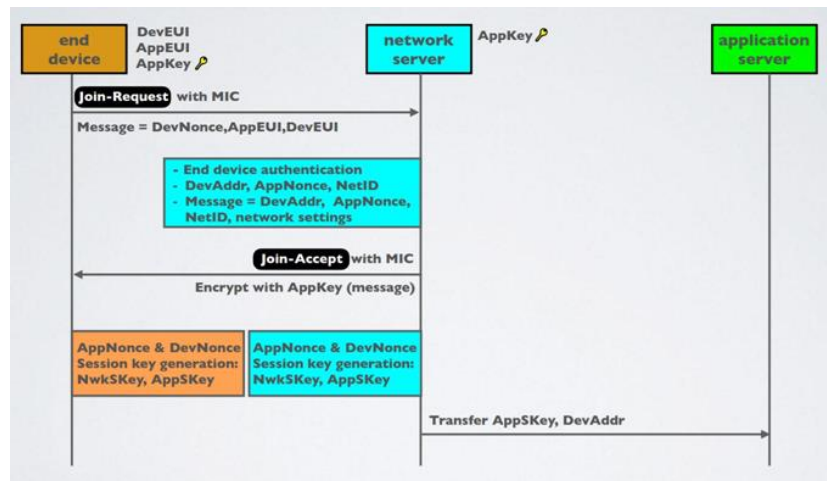


Рис. 2.9 Активація за допомогою БАП [25]

Активація пристрою за допомогою персоналізації

Коли АПП активовано, процеси Join-Request і Join-Асерт пропускаються. Замість цього значення DevAddr, AppSKey і NwkSKey вставляються безпосередньо в кінцевий вузол. Значення DevAddr і NwkSKey мережевого сервера попередньо встановлені, а також значення DevAddr і AppSKey сервера додатків.

АПП прискорює активацію пристрою, не чекаючи приєднання-прийняття від мережевого сервера. Це також може збільшити радіус дії пристрою. Якщо шлюз, який надсилає Join-Асерт, знаходився далеко від пристрою під час активації БАП, можливо, Join-Асерт не надійде в той час, коли пристрій очікує на вхідні повідомлення.

Проблема з АПП полягає в тому, що він знижує безпеку всієї програми. Під час активації БАП для кожного пристрою генеруються унікальні DevAddr, AppSKey і NwkSKey. Коли АПП активовано, ці значення вставляються безпосередньо в пристрій, які кумунікують із сервером додатків і мають

однакові значення. Таким чином, компрометація одного пристрою може скомпрометувати всі інші, які кумунікують з тим самим сервером.

Іншим недоліком АПП є фіксоване значення DevAddr, яке дозволяє кінцевому вузлу кумунікувати лише з визначеним сервером. Якщо змінити налаштування сервера додатків, доведеться перепрограмувати всі пристрої, які кумунікують з цим сервером. Крім того, мережевий сервер і пристрій мають вбудований лічильник кадрів, який підраховує кількість надісланих повідомлень. Значення цих компонентів повинні бути синхронізовані, щоб висхідна лінія була прийнята сервером. Якщо щось трапиться з пристроєм і його лічильник кадрів буде перезапущено, це призведе до відхилення висхідного зв'язку. Активація БАП синхронізує лічильники кадрів під час процесу запиту на приєднання, на відміну від АПП.

2.2.3 Вузол LoRa

Обладнання

Усі кінцеві вузли LoRa повинні мати радіомодуль LoRa, мікроконтролер та антену. Іншим поширеним обладнанням є датчики та виконавчі механізми.

Радіомодуль LoRa або чіп LoRa - це апаратне забезпечення, яке відповідає за надсилання та отримання даних. Вибраний чіп визначає максимальну пропускну здатність зв'язку, посилення передавача, чутливість приймача, максимально можливу швидкість передачі, можливий клас трафіку та необхідну бібліотеку для програмування чіпа.

Мікроконтролер відповідає за обробку вихідних даних, які він найчастіше отримує від якогось датчика, і вхідних даних. Вибір мікроконтролера має найбільший вплив на енергоспоживання всього кінцевого вузла LoRa та визначає мову програмування. Антена додає підсилення і таким чином збільшує потужність каналу зв'язку.

Плати розширення LoRa

Є багато плат розширення LoRa, які вже мають більшість інтегрованих апаратних компонентів.

Для практичної частини було розглянуто три плати розширення:

1) RN2483 LoRa Technology Mote (рис. 2.10а) фірми Microchip, яка має вбудований радіомодуль LoRa, мікроконтролер, роз'єм для зовнішньої антени, датчики, USB-порт для програмування плати, роз'єм для джерела живлення. На платі відсутній програматор мікроконтролера PICkit.

2) Dragino LoRa shield (рис. 2.10б) має радіомодуль LoRa, роз'єм для зовнішньої антени та датчики. На цій платі відсутній мікроконтролер, порт USB для програмування плати та блок живлення. Плата сумісна з мікроконтролером Arduino UNO, в якому є всі відсутні компоненти.

3) LoPy4 (рис. 2.10в) від Русом. Ця плата має вбудований радіомодуль LoRa, мікроконтролер і роз'єм для зовнішньої антени. Для цієї плати необхідно було б придбати плату PySense, на якій присутні компоненти, датчики, USB-порт для програмування плати та роз'єм для живлення.

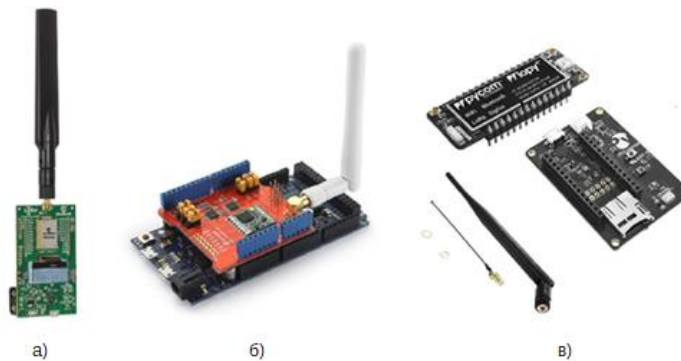


Рис. 2.10 Плати розширення LoRa: а) RN2483 LoRa Mote; б) Dragino LoRa shield і клон ArduinoUNO; в) LoPy4 і PySense

Порівняння плат розширення

У таблиці 2.3 наведено характеристику окремих плат розширення.

Таблиця 2.3

Порівняння плат розширення LoRa

Назва плати	RN2483Mot	LoPy4	Dragino shield
Додаткове обладнання	PICkit	PySense	ArduinoUNO
Модуль LoRa RF	RN2483	SX1276	RFM95W
Макс. потужність каналу(дБ)	-	168	164
Посилення (дБм)	14	14	14
Чутливість приймача(дБм)	-146	-148	-144
Макс. швидкість передачі(кбіт/с)	300	300	300
Мікроконтролер	PIC18LF45K50	ESP32	ArduinoUNO
Мова програмування	C	MicroPython	C, C++
Роз'єм антени	SMA	IPEX MHF	SMA
Можливі джерела живлення	USB, елементи живлення AAA	USB, батареї LiPo	USB
Датчики	Температури, світла	Температури, вологості, світла, тиску, прискорення	Температури
Споживання в режим сну(мкА)	Всього - 16	LoPy4 - 1, PySense - 9, Всього - 10	Dragino shield - 1, ArduinoUNO - 31, Всього - 32
Інші функції	LED-дісплей, кнопки	WiFi, Bluetooth, SigFox, слод для карти Micro SD	Клемні колодки входів/виходів

У табл. 2.3 можна побачити властивості плат, отримані з таблиці даних. Для практичної частини була обрана плата LoPy4 з PySense, тому опис цього ширший, порівняно з іншими варіантами.

RN2483 LoRa Mot

LoRa Mote від Microchip має мікросхему LoRa RN2483, також розроблену Micro-chip. Але не знайдено великої підтримки відповідної спільноти для цього продукту в Інтернеті. З іншого боку, у Microchip усі апаратні компоненти дуже добре задокументовані. Плата може працювати від елемента живлення типу AAA. Мікроконтролер запрограмовано на C, що дає розробнику великий контроль над продуктом.

Dragino LoRa shield

Перевага Dragino LoRa shield від HOPERF із мікросхемою RFM95 полягає в тому, що вона використовується з Arduino UNO або його клоном, який є одним із найпопулярніших мікроконтролерів, і можна знайти велику підтримку в Інтернеті як від спільноти, так і від розробників цих продуктів. Ще однією перевагою є вже підключені контакти роз'єму, тому розробник має доступ до багатьох вхідних і вихідних контактів, які можна знайти на Arduino, і це дозволить використовувати цей продукт як привод, який також може працювати з механічними компонентами. Недоліком є високе енергоспоживання, викликане мікроконтролером і його несумісність з низьковольтними батареями, що є однією з типових характеристик пристроїв LoRa.

LoPy4

Останній варіант, поєднання LoPy4 і PySense від Русом, використовує мікросхему LoRa SX1276 від Semtech, яка володіє технологією LoRa. SX1276 має найкращу пропускну здатність зв'язку, діапазон, чутливість приймача з усіх згаданих модулів і є єдиним модулем, який пропонує роботу класу С.

Плата PySense [32] пропонує широкий спектр датчиків, які можуть живитися від акумулятора LiPo. LoPy4 і PySense мають найменше енергоспоживання в режимі сну. LoPy4 [31] також підтримує бездротовий зв'язок за допомогою Wi-Fi, Bluetooth і SigFox, що розширює використання IoT і пропонує велику гнучкість роботи з даними. Плата програмується на мові MicroPython.

MicroPython - це інтерпретована мова, а не скомпільована мова, як C і C++, тому під час розробки можна використовувати командний рядок REPL для виконання коду, що дає розробнику можливість швидко протестувати код і допомагає з налагодженням. Недоліком інтерпретованих мов є те, що вони більш вимогливі до пам'яті пристрою, яка обмежена при роботі з мікроконтролерами. PySense пропонує можливість вставити SD-карту для розширення пам'яті.

LoPy4 не пропонує роз'єм SMA для антени, як решта плат. Антену не можна підключати безпосередньо до плати, замість цього потрібно використовувати коаксіальний кабель.

2.3 Мережа TTN

The Things Network (TTN) - це інфраструктура для IoT. TTN пропонує мережевий сервер The Things Stack (TTS) LoRaWAN, який є відкритим, децентралізованим і безкоштовним для використання. Усі шлюзи, зареєстровані на TTS, вільно доступні для всіх. TTN також пропонує інструменти для розробників додатків LoRaWAN, такі як шлюзи, кінцеві вузли LoRa та сервери додатків.

У практичній частині буде використовуватися The Things Stack V3 (TTS V3), який є сервером, який підтримує всі класи трафіку (A, B і C) і всі регіональні параметри. Він забезпечує глибоку безпеку як для мережі, так і для серверів додатків. Крім того, можлива інтеграція між власною програмою та сервером програм, що працює на TTS V3. Так, можна підключити власні шлюзи до мережевого сервера TTS V3. Є можливість контролювати через консоль TTN повідомлення, які проходять через наш шлюз, з'єднання кінцевого вузла, вхідні висхідні канали зв'язку з пристроїв LoRa та підготовлені низхідні канали зв'язку, які надсилатимуться на відповідні пристрої.

Послуги TTN

Після входу в консоль «The Things Stack V3» можна створити сервер додатків, для якого потрібно вибрати ідентифікатор додатка.

Можливі функції програми:

- 1) «End devices»: знаходження потрібного пристрою або додавання нового.
- 2) «Live data» - відстежування прогресу даних програми.
- 3) «Payload formatters»: дозволяють формувати як вхідні, так і вихідні дані (наприклад, із Hex у Json).

4) «Integrations»: використовується для інтеграції з власною програмою за допомогою MQTT, Webhooks, Storage Integration, AWS IoT або LoRa Cloud.

5) «Collaborators»: сюди можна додати співавторів, які братимуть участь у розробці програми, і призначити їм різні права.

6) «API keys»: тут створюється ключ, за допомогою якого можна отримати доступ до сервера додатків через зовнішні додатки.

Реєстрація кінцевого пристрою LoRa

Під час реєстрації потрібного пристрою LoRa є два варіанти реєстрації пристрою: або використовувати наявний пристрій із бібліотеки пристроїв LoRaWAN, або реєструвати пристрій вручну.

У даній роботі буде описана процедура ручної реєстрації, з наступними особливостями:

- можна вибрати, як потрібно активувати пристрій (БАП, АПП, multicast (варіація АПП) або не налаштовувати);

- сама форма встановлює мережу, програму та сервер приєднання, але користувач також має можливість використовувати інші параметри;

- користувач вибирає унікальний ідентифікатор «End device ID» для потрібного пристрою (частотний план встановлюється відповідно до розташування кінцевого пристрою);

- за специфікацією пристрою вибирається «DevEUI», підтримувана версія LoRaWAN, версія регіонального параметра та клас роботи;

Подальші етапи реєстрації відрізняються залежно від способу активації. Якщо вибрано БАП, можна вказати значення «AppEUI» та «AppKey» або дозволити їх генерувати. Якщо вибрано АПП, можна вказати значення «DevAddr», «NwkSKey» і «AppSKey» або дозволити їх генерувати.

Відображення даних на TTS

Консоль TTS V3 в основному використовується для моніторингу даних, що проходять через мережевий сервер. Так, можна побачити, коли повідомлення надійшло на сервер, який це тип повідомлення та здійснити попередній перегляд вмісту. Можна спостерігати обробку повідомлень запиту

на приєднання, переадресацію приєднання-прийняття, обробку висхідної лінії зв'язку та підготовку низхідної лінії зв'язку.

Обмеження

На додаток до правил, визначених протоколом LoRaWAN, існують правила та обмеження мережевого сервера, і це також стосується використання мережевого сервера TTN, політика добросовісного використання [29].

TTN визначає обмеження розміру для надісланих висхідних і низхідних каналів та це обмеження залежить головним чином від обраного коефіцієнта поширення (SF):

- для SF7 і SF8 при BW 125 кГц, верхня межа становить 222 байти;
- для SF9 при BW 125 кГц, верхня межа становить 115 байт;
- для SF10, SF11 і SF12 при BW 125 кГц, верхня межа становить 51 байт.

Важливо відзначити, що протокол LoRaWAN додає принаймні 13 байт до кожного повідомлення (DevAddr, MIC, мережевий порт і інформація про налаштування мережі). Крім того, TTN дозволяє лише 10 низхідних каналів на 24 години, включаючи підтвердження висхідних каналів. Один пристрій обмежений 30 секундами трансляції на 24 години. Час передачі залежить від розміру повідомлення та обраної SF при постійній BW. Так, потрібно запрограмувати пристрій LoRa таким чином, щоб часові інтервали між передачами були достатньо великими і загальний час передачі за 24 години не перевищував 30 секунд.

В мережі Інтернет доступним є спеціальний калькулятор [13], до якого вносять розмір повідомлення, і він обчислює мінімальний інтервал часу між передачами для різних SF. За рис. 2.11 розмір повідомлення встановлюється як 16 байт, у другому рядку можна побачити час відправки одного повідомлення, а в третьому мінімальний інтервал між повідомленнями. Це стосується протоколу LoRaWAN.

Щодо «політики справедливого доступу», яка застосовується до TTN і визначає максимальний час передачі на день, то згідно з онлайн-калькулятором, якщо система працюватиме впродовж усього дня і надсилатиме 16 байт за

допомогою SF7, можна надіслати 448 повідомлень на день, а часовий інтервал має становити принаймні 192,4 секунди.

Airtime calculator for LoRaWAN

AS923 AU915 AU915 DL CN470 **EU868** IN865 KR920 US915 US915 DL

EU863-870 uplink and downlink

overhead size^⓪ payload size^⓪ share^⓪

- 13 + - 16 + [share icon] [refresh icon]

	DR6 ^⓪	DR5	DR4	DR3	DR2	DR1 ^⓪	DR0 ^⓪
<i>data rate</i>	SF7 ^{BW 250}	SF7 ^{BW 125}	SF8 ^{BW 125}	SF9 ^{BW 125}	SF10 ^{BW 125}	SF11 ^{BW 125}	SF12 ^{BW 125}
<i>airtime</i>	33.4 _{ms}	66.8 _{ms}	123.4 _{ms}	226.3 _{ms}	411.6 _{ms}	905.2 _{ms}	1,646.6 _{ms}
<i>1% max duty cycle</i>	3.3 _{sec} 1,077 _{msg/hour}	6.7 _{sec} 538 _{msg/hour}	12.3 _{sec} 291 _{msg/hour}	22.6 _{sec} 159 _{msg/hour}	41.2 _{sec} 87 _{msg/hour}	90.5 _{sec} 39 _{msg/hour}	164.7 _{sec} 21 _{msg/hour}
<i>fair access policy</i>	96.2 _{sec (avg)} 37.4 _{avg/hour} 897 _{msg/24h}	192.4 _{sec (avg)} 18.7 _{avg/hour} 448 _{msg/24h}	355.4 _{sec (avg)} 10.1 _{avg/hour} 243 _{msg/24h}	651.8 _{sec (avg)} 5.5 _{avg/hour} 132 _{msg/24h}	1,185.5 _{sec (avg)} 3.0 _{avg/hour} 72 _{msg/24h}	2,607.0 _{sec (avg)} 1.4 _{avg/hour} 33 _{msg/24h}	4,742.2 _{sec (avg)} 0.8 _{avg/hour} 18 _{msg/24h}

Рис. 2.11 Онлайн-калькулятор для розрахунку часу трансляції [30]

2.4 Розгляд сценарію застосування технології LoRa та дронів

Огляд сценарію

На малюнку 2.12 наведено приклад застосування технології LoRa із використанням дронів та сенсорів

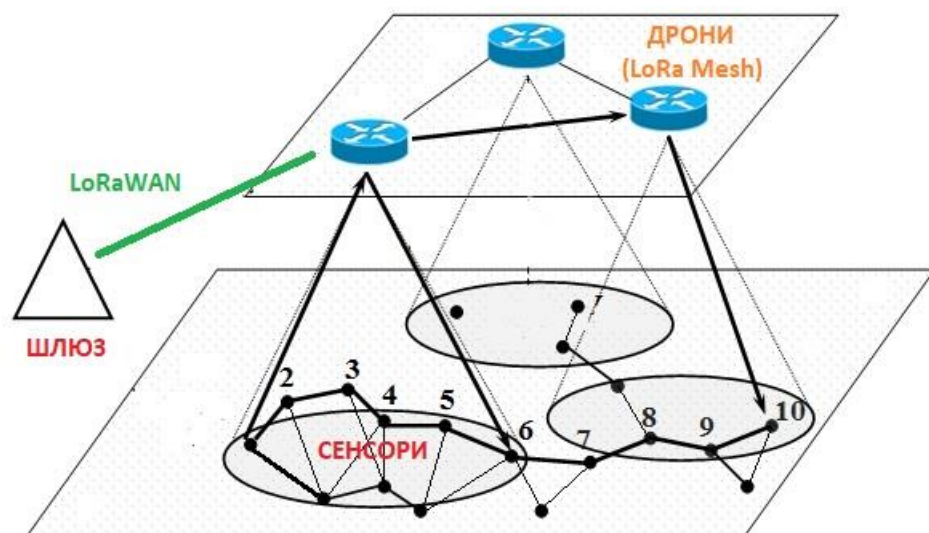


Рис. 2.12 Сценарій застосування LoRa, дронів, датчиків

Цей сценарій передбачає те що дрони, які начинені пристроями LoRa перебувають в Mesh зв'язку, як і між собою так і між датчиками. Вони збирають інформацію, отриману з сенсорів та пересилають до дрона який має зв'язок зі шлюзом по LoRaWAN. Інформація передається шлюзові за для подальшої обробки.

Огляд сценарію якщо сталася якась трагедія зі шлюзом

Але не завжди все може працювати і наприклад ми можемо отримати аварію: будинок в якому перебував шлюз затопило, або він був зруйнованим, або ще щось, і тут як раз таки можна трошки ширше розглянути застосування нашої схеми. На рисунку 2.13 наведений приклад застосування дрону в якості шлюза.

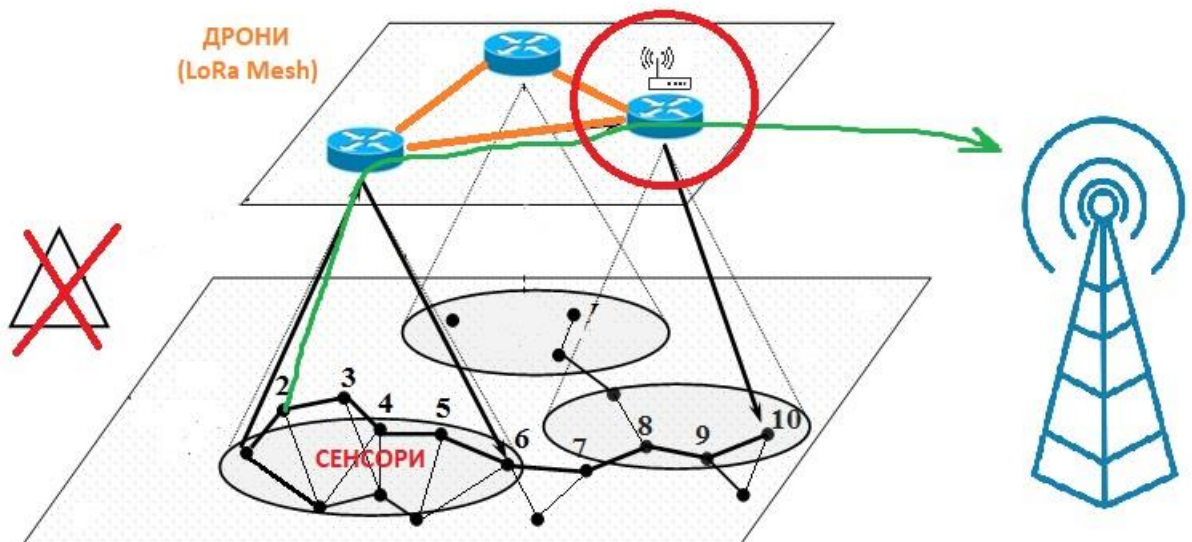


Рис. 2.13 Сценарій застосування шлюзу у поєднанні з дроном

Цей сценарій передбачає те що в нас може і не залишитись ніякої можливості розгорнути шлюз на землі. І тому ми можемо прикріпити його до дрона. Питання з'єднання шлюза із інтернет зв'язком може бути вирішеним під'єднавши до нього, наприклад, USB-модем, таким чином забезпечуємо можливість шлюзу комунікувати з серверами.

2.5 Характеристика вибраних технологій

Програми IoT

TTN дозволяє отримувати дані з кінцевого вузла LoRa. Однак, якщо потрібно продовжувати працювати з інформацією, яку надсилає пристрій LoRa, тут потрібен сам додаток IoT, який отримуватиме дані від сервера додатків TTN і продовжуватиме роботу з ним. Найпопулярнішими формами додатків IoT є настільні, мобільні та веб-програми.

Веб-додаток

Єдине, що потрібно користувачеві для запуску веб-додатку, це веб-браузер і доступ до Інтернету. Веб-додаток складається з двох частин: бекенда та інтерфейсу.

Бекенд - це частина, яка відповідає за сервер додатків, саму функціональність додатка, роботу з даними, базу даних тощо. Бекенд зв'язується з TTN і обробляє дані. Типовими інструментами розробки бекенда є мови програмування наприклад PHP, Ruby, Java, Python, Golang тощо. Крім того, такі технології баз даних, як SQL, NoSQL, MongoDB, Oracle та інші.

Інтерфейс - це частина веб-програми, з якою взаємодіє користувач. Інтерфейс відповідає за текст, зображення, меню, кнопки тощо. Робота інтерфейсу представляє отримані дані користувачеві та пересилає запити бекенда. Для розробки інтерфейсу потрібно використовувати HTML, CSS і Javascript, іншими інструментами програмування для інтерфейсу можуть бути Boot-strap, jQuery, Ajax тощо.

Оскільки LoRaWAN пропонує двосторонній зв'язок, можна реалізувати у веб-додатку як відображення отриманих даних, так і зв'язок із кінцевим вузлом LoRa, коли користувач зможе надіслати команду. Також можна запропонувати користувачеві кумунікувати з базою даних, щоб отримати старіші дані, які не потрібно відображати на інтерфейсі.

Як згадувалося раніше, бекенд відповідає роботу сервера додатків. Мови програмування для бекенд-розробки пропонують багато варіантів запуску локального сервера, на якому буде працювати програма. Однак якщо потрібно,

щоб веб-додаток був доступний не лише на локальному пристрої, додаток має бути розгорнуто в хмарі. Одним із можливих рішень є модель хмарних обчислень під назвою «Platform as a Service» (PaaS). Вся інфраструктура веб-додатків вводиться в модель, а потім PaaS створює публічний сервер на основі цієї моделі.

Вузол LoRa

В якості пристроїв LoRa були обрані LoPy4 і PySense від Русом. Причиною тому став широкий спектр датчиків для плати PySense, великий бюджетний радіомодуль LoRa SX1276, низьке енергоспоживання обох пристроїв, можливість живлення від акумулятора, а також мова програмування MicroPython.

Вузол LoRa працює як клас «А», частота встановлюється 868 МГц. Для активації використовується метод БАП, який має більш високий рівень безпеки, ніж АПП. Він надсилатиме дані датчиків, а саме температуру навколишнього середовища, барометричний тиск і напругу батареї живлення, як непідтверджену висхідну лінію. Відповідно до встановленого значення SF також встановлюється час режиму сну, щоб кінцевий вузол не порушував обмеження, встановлене сервером мережі.

Русом пропонує антену для LoPy4, яку було обрано, оскільки вона забезпечує найвищий коефіцієнт посилення з усіх доступних антен.

Мережевий сервер

В якості мережевого сервера обрано The Things Stack V3 від The Things Network. Це безкоштовна послуга, де всі користувачі можуть використовувати шлюзи LoRaWAN від усіх користувачів, створюючи всесвітнє покриття LoRaWAN. Крім того, TTN пропонує створення сервера додатків, у який можна інтегрувати зв'язок із додатками IoT.

Програми для веб-додатку

Сервер додатку може реалізовано на Python з використанням Flask, який є фреймворком Python для створення веб-додатків. Flask пропонує можливість

швидкого розгортання, що є перевагою при створенні меншої програми, у якій бере участь невелика кількість розробників.

MongoDB обрано як базу даних. Інтерфейс реалізовано на HTML, CSS і JavaScript, оскільки вони є найпопулярніші методами для створення інтерфейсу з широким набором інструкцій в Інтернеті.

Для хмарного розгортання веб-додатку придатним є Heroku - це хмарний PaaS, який пропонує безкоштовний сервіс і підтримує мову програмування Python.

Висновок

Підсумовуючи все вище сказане, можна виділити головне:

1. За основу дрона було взято квадрокоптер із міцними стійками та розумними режимами, наприклад, «політ по точкам» для побудови ЛСМ. В практичній частині буде розглянуто простий сценарій в якому він буде виступати як ретранслятор сигналу.

2. Для передачі даних використовується технологія LoRa (Long Range). Вона забезпечує високу дальність передачі даних та низьку споживання енергії. Ця технологія особливо ефективна для бездротової комунікації в широкому радіусі дії.

3. Для організації мережі IoT в нашій роботі буде використовується протокол LoRaWAN, який забезпечує надійну передачу даних між сенсорними вузлами та шлюзами.

4. Вузли LoRa можуть бути розташовані на дронах, на землі або в інших місцях, де потрібно здійснювати збір та передачу даних. В якості вузла була обрана конструкція LoPy4 та PySense.

5. Розглядання можливостей TTN дозволяє ширше подивитись на майбутню структуру з використанням ЛСМ. TTN дозволяє отримувати дані з кінцевого вузла LoRa. Однак, щоб працювати з інформацією, яку надсилає пристрій LoRa, можна зробити додаток IoT, який отримуватиме дані від сервера додатків TTN і продовжуватиме роботу з ним.

РОЗДІЛ 3

АНАЛІЗ ТА ОБРОБКА ДАНИХ ЗІБРАНИХ З ДРОНІВ

3.1 Характеристика експлуатації розроблюваної системи

Архітектура системи

В практичній ж частині ми будемо розглядати простий кейс (рис. 3.2): поєднання вузла (сенсор розгорнутий на землі), мобільного вузла (сенсор закріплен на дроніві) та шлюза.

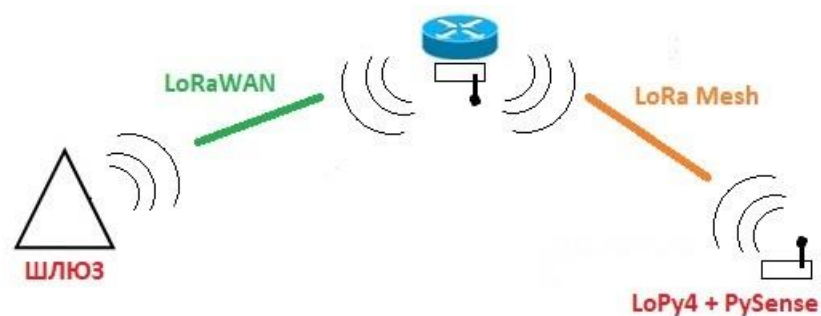


Рис. 3.1 Ілюстрація простого кейсу

За для того щоб розкрити потенціал TTN, ми можемо перенаправити надходжені на нього дані на інші ресурси. На рисунку 3.2 наведено архітектуру розроблюваної системи.

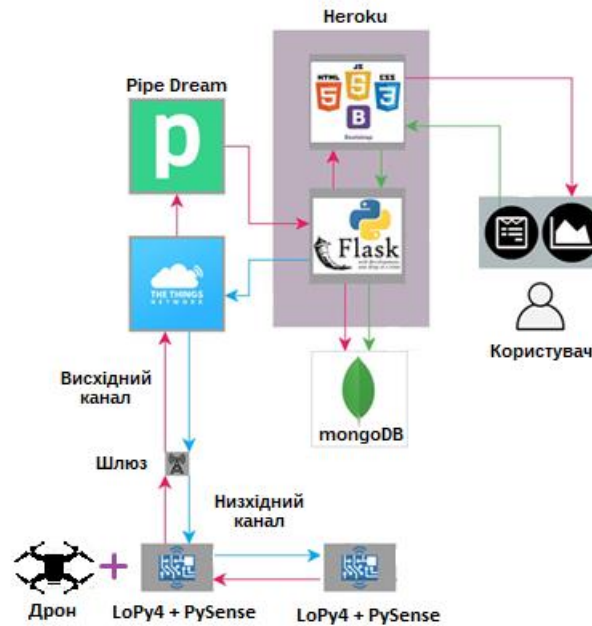


Рис. 3.2 Архітектура системи

За рис. 3.2 червона лінія вказує на потік даних від LoPy4 до користувача (відображення даних у веб-додатку); синя лінія вказує на підтвердження надходження сигналу, надіслану від Flask; зелена лінія вказує на ідентифікацію користувача у веб-додатку. Шлях червоної лінії починається від датчика PySense і продовжується до LoPy4. LoPy4 створює висхідний канал, який надсилає до LoPy4 (що закріплений на дроні) по LoRa mesh, від нього вже йде до TTN через шлюз LoRaWAN. Висхідна лінія не йде безпосередньо від TTN до веб-додатку, а проходить через блок PipeDream.

Одним із основних способів комунікації програм в Інтернеті між собою є Rest API, де одна сторона надсилає HTTP-запит, а якщо інша сторона його приймає, вона надсилає назад необхідну відповідь. Що і роблять PipeDream та додаток створений на Flask.

PipeDream пропонує службу RequestBin, яка є короткостроковим сховищем для HTTP-запитів. Сервер додатків на TTN може бути налаштований на створення запиту HTTP Post, коли надходить нове висхідне з'єднання, і пересилання висхідного з'єднання на PipeDream. Серверна частина може взяти на себе висхідну лінію за допомогою PipeDream API.

Інтерфейс може відображати отримані дані у вигляді графіка та інформаційних блоків.

Апаратне підключення

LoPy4 та PySense з'єднуються між собою. Плата може працювати від LiPo акумулятора з роз'ємом JST PHR-2 або від micro USB. Порт USB також використовується для програмування плати.

3.2 Питання програмної реалізації

Підготовка середовища розширення

Для програмування LoPy4 потрібен PyMake, плагін для редакторів коду, такий як VS code, який забезпечує зв'язок із платою розробки Русом.

Використовуючи інструмент пошуку пакетів в VS code, можна встановити необхідну версію PyMake.

Після встановлення PyMake консоль REPL додається до VS code. PyMake допомагає підключити пристрій Русом до VS code. Тут потрібно запустити код (на мові MicroPython) у мікроконтролері, увівши його або завантаживши з пристрою. Також є можливість отримати інформацію про підключену плату, наприклад тип розробки, версію мікропрограми, використану пам'ять тощо.

Структура проекту

Проект, який вставляється в LoPy4, має бути структурований так, щоб файли «boot.py» і «main.py» містилися в папці проекту. Файл «boot.py»- це сценарій, який запускається першим, найчастіше він містить код, який ініціалізує налаштування. Сценарій «main.py» запускається після «boot.py» або, якщо «boot.py» не існує. Так, «main.py» записується в проект. Каталог «lib»- це каталог для всіх бібліотек, які імпортуються в проект, як показано нижче.

```
|--Project folder
  |--lib
    |--library1.py
    |--library2.py
  |--boot.py
  |--main.py
```

Конфігурація

LoPy4 має пам'ять NVRAM, яка зберігає інформацію навіть після відключення живлення або після виходу з режиму сну. Бажано записати в нього значення «AppEui» і «AppKey», а також значення конфігурації, це може бути: час сну, налаштування LoRaWAN. Доступ до пам'яті NVRAM забезпечує бібліотека «русом».

Підключення до сервера мережі TTN

Нижче показано приклад для активації пристрою.

```

1  # ініціалізує LoRa в режимі LORAWAN і намагається відновити
   з'єднання
2  lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU)
3  lora.nvram_restore()
4  if not lora.has_joined():
5      # створює параметри автентифікації rssi
6      app_eui = ubinascii.unhexlify(русом.nvs_get("app_eui"))
7      app_key = ubinascii.unhexlify(русом.nvs_get("app_key"))
8      # приєднатися до мережі за допомогою rssi (повітряна
   активація)
9      lora.join(activation=LoRa.rssi, auth=(app_eui, app_key),
   timeout=0)
10 else:
11     print("Connection restored")

```

Бібліотека LoRa буде використовуватися для налаштування пристрою відповідно до протоколу LoRaWAN і встановлення частоти передачі відповідно до заданого регіону (рядок 2). Він спробує відновити з'єднання з пам'яті NVRAM (рядок 3). Якщо повторне підключення не вдалося, пристрій активується за допомогою БАП (рядки 6-9). Після успішної активації встановлюється значення SF і встановлюється з'єднання між пристроєм і сервером за допомогою бібліотеки сокетів.

Налагодження вузлів у lora mesh.

Нижче наведений приклад налагодження lora mesh та додавання до неї вузла.

```

1  lora.mesh(true) # активація режиму мережі
2  mesh = lora.mesh()
3  sync_word = 0x34 # кодове слово синхронізації
4  mesh.set_sync_word(sync_word) # використання слова
5  # Налаштування параметрів вузла
6  mesh.adaptive_sf(True) # автоматичний режим SF
7  mesh.frequency(868000000) # Частота
8  mesh.adaptive_sensitivity(True) # режим чутливості
9  mesh.tx_power(14) # потужність передавача (dBm)
10 DEVICE_ADDR = b'\x10\x01\x02\x03\x04\x05' # адреса вузла
11 DEVICE_ID = b'S1' # ідентифікатор вузла
12 mesh.add_peer(DEVICE_ADDR, DEVICE_ID) # додавання вузла в меш
13 mesh.start() # запуск mesh

```

Надсилання даних в LoRa Mesh

У додатку 1 (див. Додаток 1) показано, як lora4, використовуючи бібліотеку Pysense, ініціалізує зв'язок між датчиками. З датчиків отримується температура навколишнього середовища, тиск та вологість. Та відправляє отримані дані всім у мережі.

Прийняття даних в LoRa Mesh

Прийняття даних у меш має схожу структуру коду (див. Додаток 1) але з додаванням рядка про очікування даних. Приклад нижче

```

1  data = mesh.recv()

```

Надсилання зібраних даних до шлюза

Нижче наведено приклад

```

1  # Перевірка статусу приєднання до мережі LoRaWAN
2  while not lora.has_joined():
3      pass
4  # Відправка даних до шлюзу через LoRaWAN
5  s.send(struct.pack(">1", dev_addr) + data.encode('utf-8'))

```

Рядок 2 перевіряє чи приєднався вузол до мережі. Рядок 3 призупиняє виконання коду, щоб програма не перейшла до наступних дій, поки не відбудеться з'єднання.

Збереження налаштувань і переведення пристрою в режим сну

Параметри з'єднання LoRaWAN, LoRa Mesh зберігаються в пам'яті NVRAM. Глибокий сон переводить майже всі функції в сплячий режим, тому пробудження схоже на перезавантаження всього пристрою. Усі оголошені змінні будуть забуті, тому нам доведеться записати значення конфігурації в пам'ять NVRAM.

Структура веб-додатку

```
|--Project folder
  |--templates
    |--index.html
  |--app.py
  |--config.py
  |--mongo_db_communication.py
  |--ttn_communication.py
```

Реалізація

Структура проекту наведена вище, де можна знайти більшість файлів, які забезпечують роботу веб-додатку. Так, «index.html» використовується для візуалізації інтерфейсу. У «ttn_communication.py» знаходяться функції, які відповідають зв'язок із TTS V3. У «mongo_db_communication.py» функції, які кумунікують з базою даних. Файл «app.py» містить серверну частину веб-програми, а «config.py» - значення її конфігурації.

У «app.py» створюється зразок класу «Flask», який відповідає за ініціалізацію веб-сервера, на якому працює програма. За допомогою «Flask» реалізуються маршрути, які запускають різні функції програми: рендеринг інтерфейсу, завантаження висхідних посилань, створення низхідних посилань.

Візуалізація інтерфейсу

Коли відвідується домен веб-програми, серверна частина викликає файл «index.html», який відображає необхідні компоненти. Такими компонентами є: заголовок, інформаційні вікна модуля, інтерактивні графіки та веб-форми.

Для того щоб створити інтерактивні графіки для відображення залежності температури та тиску від часу, використовується бібліотека «highcharts».

Графік буде оновлюватися в реальному часі, бо інтерфейс буде постійно запитувати останні дані на сервері, щоб оновити графік. Приклад графіка контрольованих показників наведено на рис.3.3.

Веб-форма, що буде використовуватися за для запити логіна та пароля для відображення даних користувачеві, який перейде по лінці.

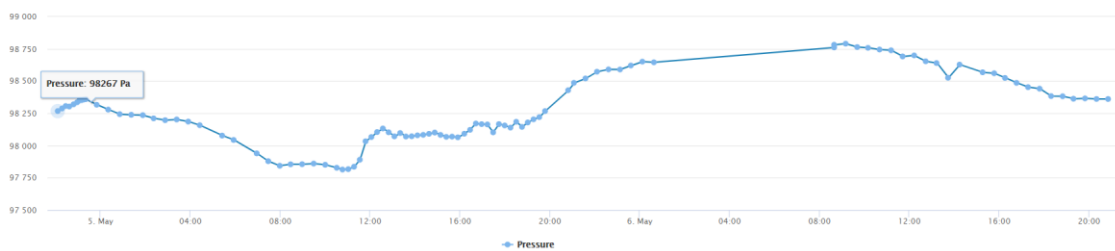


Рис. 3.3 Графік контрольованих показників

Завантаження висхідних каналів

Сервер отримує висхідні канали від TTN, надсилаючи запит «HTTP GET» на сервер PipeDream. Відповідь сервера - поле «json», в якому можна знайти вказану кількість останніх завантажень. Дані, надіслані з LoPy4, можна знайти в «frm_payload». TTN кодує висхідну лінію за допомогою кодування Base64, яке серверна частина декодує перед надсиланням на зовнішню частину. Зразок відповіді від PipeDream наведено далі.

```
"uplink_message": {
  "frm_payload": "TV9+4ADQYE0ArsBH6voEQpovmUA=",
  "rx_metadata": {
    "gateway_ids": "eui-b827ebfffec6e5d0",
    "rssi": -116,
    "snr": -11,
  },
  "settings": {
    "bandwidth": 125000,
    "spreading_factor": 10
  },
  "frequency": "868000000"
}
```

Створення низхідного каналу

Сервер надсилає «HTTP Post Request» на сервер TTN для створення низхідного каналу. Потрібно помістити повідомлення для пристрою LoRa, порт і тип повідомлення в тіло запиту. Повідомлення повинне мати кодування Base64. Порт для низхідних каналів має значення «1».

Запит на публікацію наведено нижче.

```
{"downlinks":[{"frm_payload":downlink, "f_port":1, "confirmed": True}]}
```

Отримання інформації з бази даних

Зв'язок із базою даних MongoDB підтримується бібліотекою «rumongo». Дані в MongoDB представлені в «json». Для пошуку та отримання даних із бази використовується відповідний запит, приклад якого наведено нижче.

```
1 # знаходить дані з коефіцієнтом поширення SF12
2 results = collection.find({'sf': 12})
3 # знаходить дані між вказаними часовими мітками та сортує їх
  від найвищого до найнижчого
4 results = collection.find(
    {'time': {'$gte': from_timestamp, '$lte':
    to_timestamp}}).sort([('time', -1)])
```

Зберігання інформації у базі даних

Функція, яка відповідає за збереження інформації в базі даних, переглядає ідентифікатор нового висхідного каналу та порівнює його з існуючими ідентифікаторами в базі. Якщо це унікальний ідентифікатор, то зберігає висхідний канал до бази даних.

Хмарне розгортання

Щоб завантажити веб-додаток у хмару за допомогою Heroku, потрібно створити файл «Procfile». Він повідомляє Heroku, який сервер хостингу використовувати та який сценарій запускати. Для створення хмарного сервера

використовується бібліотека «unicorn». Скриптом для запуску буде «app.py», у якому створюється серверна частина веб-програми. Приклад наведений нижче.

```
web: unicorn app:app
```

3.3 Особливості системи

LoRa характеризується значною дальністю дії, яка вимірюється кілометрами/десятками кілометрів. Цей показник стосується параметрів, що впливають на успішність з'єднання, таких як коефіцієнт поширення та розташування пристрою.

При відправці висхідного каналу на TTN можна дізнатися, які шлюзи можуть отримати повідомлення. Нижче наведено спрощений приклад повідомлення на сервері TTN, коли надходить висхідний канал. Так, можна побачити шлюз, який є досяжними для пристрою, і які повідомлення РПС досягли шлюзу. Також можна побачити ідентифікатор шлюзу.

```
{
  "payload_raw": "UvHrRQBF",
  "gateways": [
    {
      "gtw_id": "eui-313532352a004e00",
      "rssi": -115,
    }
  ]
}
```

Залежність дальності від коефіцієнта поширення

У LoPy4 реалізовано сценарій, який поступово збільшуватиме значення SF і надсилатиме чотири висхідні канали для кожного значення коефіцієнта поширення. Збільшуючи коефіцієнт поширення, час передачі збільшується, це збільшує ймовірність декодування сигналу шлюзом і стійкість сигналу до втрат на шляху. Так, за допомогою SF7 можна транслювати на шлюз на відстані приблизно 3 км, а з SF12, зменшивши кількість перешкод, наприклад, піднявши

дрон, можна суттєво збільшити дальність сигналу і є шанс охопити шлюзи на відстані 40 км.

Таким чином, розташування наших конструкцій вузол-мобільний вузол-шлюз може досягати протяжності у 80 км. Взявши випадок, якщо наші вузли будуть розташовані в місті із великою кількістю перешкод, то можна використати функцію у дрона як «політ за точками». Політ за точками передбачає, що дрон буде пролітати у місця що визначені на контролері польотів (гео позиції розташування наших сенсорів) тим самим збирати дані у важкодоступних місцях і передавати їх на шлюз.

Енергоефективність системи

Проблема яка розповсюджена серед БПЛА це енергія. Літальна платформа - квадрокоптер SJRC F7 4K Pro є досить енергоефективним апаратом. Так, час безперервного польоту становить 25 хвилин з акумуляторною батареєю 2500 мА·г. У той же час, пристрій LoRa, що працює за класом енергоспоживання «А», надсилає дані щохвилини або щогодини, тому більшу частину часу пристрій перебуває в режимі сну. Так, залежно від потреб експлуатації літальної мережі, квадрокоптер можна налаштувати на синхронне перебування у режимі сну разом із пристроєм LoRa, урахувавши час підйому дрона на необхідну висоту, наприклад, 100 м, для отримання найкращої ефективності передачі сигналу. У такому режимі, з урахування характеристик досліджуваного обладнання, система зможе працювати автономно, до однієї доби, за умови здійснення сеансу зв'язку один раз на годину.

Висновок

В розділі 3 ми розглянули простий кейс реалізації LoRa із застосуванням дрона як ретранслятора сигналу. Ми розглянули процес формування, обробки та передачі сигналу між пристроями в мережах LoRa Mesh, LoRaWAN. Розглянули питання розповсюдження сигналу конструкціями. Та затронули енергоефективність цієї системи.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

В цілому були розглянуті всі завдання що впливають з мети дослідження. Зробили аналіз обраних технологій та обладнання для створення літальної сенсорної мережі для Інтернету речей. Відповідно до задач роботи, обраним обладнанням для створення мережі є мобільна літальна платформа – квадрокоптер sjrc f7 4k pro, плата розробки LoPy4 з модулем SX1276 LoRa, де до плати підключається модуль PySense, який оснащено датчиками вимірювання температури та тиску навколишнього середовища. LoPy4 оснащений мікроконтролером ESP32, який програмується за допомогою мови MicroPython.

Для передачі даних за технологією LoRa обрано протоколи LoRa Mesh, LoRaWAN і мережевий сервер TTS. Сервер веб-додатку, який отримує і зберігає дані з мережевого сервера TTS, може бути реалізовано за допомогою фреймворку Flask. Інтерфейс додатку може відображати дані через бібліотеку Highcharts. Базою даних для зберігання даних може виступати MongoDB.

Цю структуру можна покращити: замінивши наш квадрокоптер на дрон з підтримкою LoRa або ж дрон з підтримкою під'єднання до контролеру польоту модульного пристрою LoRa, що дозволить нам синхронізувати та автоматизувати процес накладання маршруту у дронів; додавши більше: дронів, датчиків та шлюзів. Але ж якщо ми додамо більше вузлів в системі, то ми можемо забути про їх розташування, тому можна покращити нашу схему, якщо замість PySense використаємо плату PyTrack. PyTrack має вбудований GPS і якщо ми використаємо декілька таких датчиків, то сенсори, без вбудованих GPS, зможуть визначити своє місце розташування завдяки розповсюдженню геолокації інших.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Фесенко, Г. (2020). Особливості визначення оптимального плану використання флотів БПЛА заданого радіусу дії для моніторингу об'єктів на території атомних станцій. Науковий журнал «Інженерія природокористування». 62-67. 10.37700/enm.2019.4(14).62-67.
2. Zhai, Di-Hua & Yu, Sheng & Xia, Yuanqing. (2023). FANet: Fast and Accurate Robotic Grasp Detection Based on Keypoints. IEEE Transactions on Automation Science and Engineering. PP. 1-13. 10.1109/TASE.2023.3272664.
3. Durairaj, Balakumaran. (2023). DISASTER MANAGEMENT USING WIRELESS SENSOR NETWORKS. International Scientific Journal of Engineering and Management. 02. 10.55041/ISJEM00410.
4. Al-Saedi, Ahmed & Boeva, Veselka & Casalicchio, Emiliano & Exner, Peter. (2022). Context-Aware Edge-Based AI Models for Wireless Sensor Networks—An Overview. Sensors. 22. 5544. 10.3390/s22155544.
5. Kumaresan, Praghash & Ramaraj, R. Ravi. (2022). Energy Consumption Architecture for Wireless Sensor Networks With Different Clusters.
6. Ullah, Ihsan & Kim, Chan-Myung & Heo, Joo-Seong & Han, Youn-Hee. (2022). An Energy-efficient Data Collection Scheme by Mobile Element based on Markov Decision Process for Wireless Sensor Networks. Wireless Personal Communications. 123. 1-17. 10.1007/s11277-021-09241-1.
7. X. Ma, R. Kacimi, and R. Dhaou. Fairness-aware uav-assisted data collection in mobile wireless sensor networks. In 2016 International Wireless Communications and Mobile Computing Conference (IWCMC), pages 995–1001, Sep. 2016.
8. Liu, Weixue. (2022). An Embedded Microcontroller-Based Access Authentication System of Wireless Sensor Network. Journal of Cyber Security Technology. 7. 1-20. 10.1080/23742917.2022.2132345.
9. Khan, Koffka. (2023). Routing Protocols in Sensor Networks.

10. Tao, Hai & Zhou, Jincheng & Padmavathy, T. & Md, Abdul & Jawawi, Dayang & Aksoy, Muammer. (2022). Design and Validation of Lifetime Extension Low Latency MAC Protocol (LELLMAC) for Wireless Sensor Networks Using a Hybrid Algorithm. *Sustainability*. 14. 15547. 10.3390/su142315547.
11. Лещенко, Геннадій & Мандрик, Яна & Стратонов, Вадим & Давидов, Сергій. (2021). СПОСОБИ ЗАСТОСУВАННЯ БПЛА ПІД ЧАС АВІАЦІЙНОГО ПОШУКУ І РЯТУВАННЯ. *Science-based technologies*. 51. 271-280. 10.18372/2310-5461.51.15998.
12. W. A and K. M. Unmanned aerial vehicles, volume 134 of *In :Combat modeling, international series in operations research & management science*. US : Springer, 2009.
13. I. Guvenc, W. Saad, M. Bennis, C. Wietfeld, M. Ding, and L. Pike. Wireless communications, networking, and positioning with unmanned aerial vehicles [guest editorial]. *IEEE Communications Magazine*, 54(5) :24–25, May 2016.
14. Печурін, М.К & Боярінова, Ю.С & Кондратова, Л.П & Воронін, М.Г & Сіренко, М.А.. (2022). Моделі топологій слабовипромінюючої телекомунікаційної системи взаємодіючих БПЛА. *Problems of Informatization and Management*. 4. 48-54. 10.18372/2073-4751.72.17461.
15. Agriculture application. Available : <http://www.hse-uav.com>.
16. Alawad, Wedad & Ben Halima, Nadhir & Aziz, Layla. (2023). An Unmanned Aerial Vehicle (UAV) System for Disaster and Crisis Management in Smart Cities. *Electronics*. 12. 1051. 10.3390/electronics12041051.
17. Pipeline safety and monitoring. Available : <http://napipelines.com/unmanned-aircrafts-pipeline-safetymonitoring/>.
18. Sarmadi, Hassan & Entezami, Alireza & Magalhães, Filipe. (2023). Unsupervised data normalization for continuous dynamic monitoring by an innovative hybrid feature weighting-selection algorithm and natural nearest neighbor searching. *Structural Health Monitoring*. 10.1177/14759217231166116.
19. ХАЗАНОВИЧ, Юрій & МОСПАН, Антон & КИРИЧУК, Юрій & НАЗАРЕНКО, Наталя. (2023). ОГЛЯД СИСТЕМ НАВІГАЦІЇ РУХОМИХ

ОБ'ЄКТІВ. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES. 58-65. 10.31891/2219-9365-2023-73-1-9.

20. Tran, Thanh-Nam & Nguyen, Thanh-Long & Truong Hoang, Vinh & Vozňák, Miroslav. (2023). Sensor Clustering Using a K-Means Algorithm in Combination with Optimized Unmanned Aerial Vehicle Trajectory in Wireless Sensor Networks. *Sensors*. 23. 10.3390/s23042345.

21. Xuewen, He & Kaihua, Cao. (2023). LoRa network communication protocol based on location and time planning. *Peer-to-Peer Networking and Applications*. 1-13. 10.1007/s12083-023-01473-1.

22. Shin, Won-Jae & Lee, Yejin & Cho, Junghyun & Jang, Jiyeon & Seo, Yejune & Kahng, Sungtek. (2023). RSSI Improved for LoRa Wireless Communication, Field-Tested in the Wide-Open Area. *IEEE Access*. PP. 1-1. 10.1109/ACCESS.2023.3241485.

23. Jia, Zhenyu & Zheng, Wenjun & Yuan, Fei. (2022). A Two-Dimensional Chirp-MFCSK Modulation Method for Underwater LoRa System. *IEEE Internet of Things Journal*. 9. 24388-24397. 10.1109/JIOT.2022.3188755.

24. Francis, Imelda & Shah, Shaharil. (2022). Cost-Effective Arduino-Based RFID Automated Cage Door and Pet Tagging with GPS Tracker using Peer-to-Peer LoRa WAN. *Journal of Electronic Voltage and Application*. 3. 10.30880/jeva.2022.03.02.005.

25. Tayebi, Amin & Veltri, Luca & Zanichelli, Francesco & Caselli, Stefano. (2022). Interworking between LoRaWAN and non-LoRa IoT Systems. 415-420. 10.1109/PerComWorkshops53856.2022.9767494.

26. YEGIN a SELLER. RP002-1.0.2 LoRaWAN® Regional 40 Parameters [online]. 2020 [cit. 2021-5-14]. Dostupné z: https://lora-alliance.org/wpcontent/uploads/2020/11/RP_2-1.0.2.pdf

27. Herard, Joffrey. (2022). Structuration autonome des réseaux IoT de type LoRa / LoRaWAN.

28. Sarkar, Nurul & Kavitha, Asish & Ali, Md. (2022). A Secure Long-Range Transceiver for Monitoring and Storing IoT Data in the Cloud: Design and Performance Study. *Sensors*. 22. 8380. 10.3390/s22218380.

29. Gupta, Anubhav & Campos, Daniel & Ganeriwala, Parth & Bhattacharyya, Siddhartha & Oconnor, Terrence & Dcosta, Adolf. (2023). Modeling Internet-of-Things (IoT) Behavior for Enforcing Security and Privacy Policies.

30. Avbentem/airtime-calculator Github [online].
<https://avbentem.github.io/airtime-calculator/ttn/eu868/16>

31. Placencia, Sebastian & Astudillo, Darwin & Vázquez-Rodas, Andrés & Minchala, Ismael & Guaman, Jhonny. (2019). Rainfall Intensity Datalogger System. LoPy4-Based Design and Implementation. 45-50. 10.1145/3345860.3361513.

32. Voisin-Grall, Andrea & Malaolu, Oba & Zhu, Yingbo & Ahmed, Tanveer & Al-Ahmed, Shahriar & Zeeshan, Muhammad. (2019). Remote Condition Monitoring: A Prototype Based on Pycom Development Board FiPy and Pysense. 1-6. 10.1109/UCET.2019.8881870.

ДОДАТОК 1

```
1  # Імпорт необхідних бібліотек
2  from network import LoRa
3  import socket
4  import time
5  from machine import I2C
6  from pysense import Pysense
7  from SI7006A20 import SI7006A20
8  from MPL3115A2 import MPL3115A2

9  # Ініціалізація LoRa
10 lora = LoRa(mode=LoRa.LORA, region=LoRa.EU868)
11 lora.mesh(True) # Активація режиму мережі

12 # Ініціалізація PySense та датчиків
13 i2c = I2C(0, pins=('P22', 'P21'))
14 py = Pysense(i2c=i2c)
15 temp_hum_sensor = SI7006A20(py)
16 pressure_sensor = MPL3115A2(py)

17 # Функція для зчитування даних з датчиків
18 def read_sensor_data():
19     temperature = temp_hum_sensor.temperature()
20     humidity = temp_hum_sensor.humidity()
21     pressure = pressure_sensor.pressure()
22     return temperature, humidity, pressure

23 # Основний цикл
24 while True:
25     # Зчитування даних з датчиків
26     temperature, humidity, pressure = read_sensor_data()

27     # Форматування повідомлення для надсилання
28     data = "Temperature: {}C, Humidity: {}%, Pressure:
29     {}Pa".format(temperature, humidity, pressure)
30     data_bytes = data.encode('utf-8')

31 # Відправка даних до всіх вузлів в мережі
32 for peer in lora.meshed_devices():
33     s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
34     s.setsockopt(socket.SOL_LORA, socket.SO_CONFIRMED, True)
35     s.sendto(data_bytes, peer)
```