

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки  
(повна назва)

## АТЕСТАЦІЙНА РОБОТА

### Пояснювальна записка

рівень вищої освіти другий (магістерський)  
(рівень вищої освіти)

Комп'ютерне діагностування станів розподілених об'єктів з випадковим вибором вузлів сенсорної мережі  
\_\_\_\_\_  
\_\_\_\_\_  
(тема)

Виконав: студент 2 курсу, групи СКСм-18-1  
Степанова К.О.  
(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія  
(код і повна назва спеціальності)

Тип програми Освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані комп'ютерні системи  
(повна назва освітньої програми)

Керівник д.т.н., проф. Кривуля Г.Ф.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_

(підпис)

Чумаченко С.В.

(прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
Кафедра Автоматизації проектування обчислювальної техніки  
Рівень вищої освіти другий (магістерський)  
Спеціальність 123 – Комп'ютерна інженерія Тип  
програми Освітньо-професійна Освітня  
програма Спеціалізовані комп'ютерні системи

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

## ЗАВДАННЯ НА АТЕСТАЦІЙНУ РОБОТУ

студентові Степановій Крістині Олександрівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютерне діагностування станів розподілених об'єктів з випадковим вибором вузлів сенсорної мережі

затверджена наказом по університету від 04 \_\_\_\_\_ 11 \_\_\_\_\_ 2019 р. № 1624ст \_\_\_\_\_ 2.

Термін подання студентом роботи до екзаменаційної комісії 19 \_\_\_\_\_ 12 \_\_\_\_\_ 2019 р.

3. Вихідні дані до роботи збір розподілених даних в бездротових сенсорних мережах, випадковий вибір вузлів сенсорної мережі, кількість датчиків в сенсорній мережі

4.

Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

Аналіз предметної області \_\_\_\_\_

Актуальні проблеми розгортання бездротової сенсорної мережі \_\_\_\_\_

Постановка задачі \_\_\_\_\_

Розгляд реалізації бездротової сенсорної мережі \_\_\_\_\_

Алгоритм розташування датчиків бездротової сенсорної мережі \_\_\_\_\_

Реалізація системи підтримки прийняття рішень для розміщення сенсорів \_\_\_\_\_

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

13 слайдів

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	02.09.2019- 07.09.2019	
2	Аналіз предметної області	08.09.2019-24.09.2019	
3	Аналіз джерел з проблемної галузі	25.09.2019-05.10.2019	

4	Вивчення актуального стану та проблем предметної області	06.10.2019-19.10.2019	
5	Розробка алгоритму розташування датчиків БСМ	20.10.2019-02.11.2019	
6	Розміщення вузлів БСМ на основі вибраного алгоритму	03.11.2019-16.11.2019	
7	Аналіз отриманих результатів	17.11.2019-20.11.2019	
8	Оформлення пояснювальної записки	20.11.2019-16.12.2019	
9	Перевірка виконаного проекту керівником	17.12.2019-28.12.2019	
10	Захист роботи	20.12.2018	

Дата видачі завдання 02.09.2019 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ д.т.н., проф. Кривуля Г.Ф.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка містить 85 сторінок, 44 рисунків, 4 таблиці, 20 джерел за переліком посилань.

ZIGBEE, MESHLOGIC, ДІАГРАМА ВОРОНОГО, АЛГОРИТМ, СЕНСОР, ДАТЧИК, МЕРЕЖА, БСМ, MESH-ТОПОЛОГІЯ, МАРШРУТИЗАТОР, КЛАСТЕРИ, ДІАГРАМА ВОРОНОГО, НЕЧІТКА ЛОГІКА

У даній роботі представлені сучасні технології з моніторингу та управління оточуючими приладами. Електронне управління, що забезпечує економію

електроенергії та часу людей, надаючи величезний комфорт в управлінні. Зокрема описані технології ZigBee і MeshLogic, одну з яких буде використано в цій роботі.

Продемонстровані технології енергоефективні, споживання електроенергії у них знаходиться на низькому рівні, в порівнянні з можливими аналогами.

Продемонстровано можливість побудови діаграми Вороного для розподілу сенсорів на території на основі єдиного підходу: формулювання безперервної завдання оптимального розбиття множини з критерієм якості розбиття, що забезпечує відповідний вид діаграми Вороного, і застосування математичного та алгоритмічного апарату рішення таких задач. Результатом такого підходу є можливість будувати не тільки вже відомі діаграми Вороного, а й конструювати нові.

## ABSTRACT

Explanatory note for this work contains 85 pages, 44 figures, 4 tables, 20 sources.

ZIGBEE, MESHLOGIC, VORONOI DIAGRAM, ALGORITHM, SENSOR, DETECTOR, NETWORK, WSN, MESH-TOPOLOGY, ROUTER, CLUSTERS, VORONOI DIAGRAM, FUZZY LOGIC

This paper presents the most advanced technologies for monitoring and management of the surrounding devices. Electronic control provides power savings, giving people time to manage a huge comfort. Specifically described MeshLogic ZigBee and technology, one of which we will use in our work.

Demonstrated technologies a energy-efficient, the consumption of electricity from them at a low level in comparison with possible analogs.

The possibility of constructing the Voronoi diagram for the distribution of sensors in the territory based on a unified approach is demonstrated: formulation of a continuous

problem of optimal partition of a set with the criterion of the quality of the partition, which provides the appropriate type of Voronoi diagram, and application of mathematical and algorithmic apparatus for solving such problems. The result of this approach is the ability to construct not only already known Voronoi diagrams, but also to construct new ones.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І .....	8
ТЕРМІНІВ .....	8
ВСТУП .....	9
1 АКТУАЛЬНІСТЬ ТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ .....	11
1.1 Актуальність теми .....	11
1.2 Постановка задачі .....	12
2 ВАРІАНТИ РЕАЛІЗАЦІЇ БЕЗДРОТОВИХ СЕНСОРНИХ МЕРЕЖ: ZIGBEE І MESHLOGIC .....	13
2.1 Порівняння MeshLogic та ZigBee .....	16
2.1.1 Фізичний рівень .....	16
2.1.2 Стійкість до співканальної інтерференції .....	17
2.1.3 Профілі ZigBee .....	20
2.1.4 Мережеві протоколи .....	22
2.1.5 Типи сенсорних мереж .....	25
2.1.6 Маршрутизація в мережах ZigBee .....	27
2.1.7 Бібліотека кластерів ZigBee .....	34
2.1.8 Безпека ZigBee .....	35
2.1.9 Енергоспоживання БСС .....	39
2.1.10 Вимоги до апаратних ресурсів .....	40
3 ДІАГРАМА ВОРОНОГО ДЛЯ РОЗМІЩЕННЯ СЕНСОРІВ .....	40
3.1 Діаграма Вороного .....	41

3.2 Алгоритми побудови діаграми .....	43
3.3 Алгоритм побудови діаграми Вороного шляхом перетину напівплощин .....	43
3.4 Можливості застосування діаграм Вороного .....	49
4 РЕАЛІЗАЦІЯ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ АНАЛІЗУ .....	50
СЕНСОРІВ .....	50
4.1 Основні теоретичні відомості .....	51
4.1.1 Визначення лінгвістичної змінної .....	51
4.1.3 Логічні операції з нечіткими множинами .....	55
4.2 Системи нечіткої логіки .....	56
4.2.1 Прості системи нечіткої логіки .....	56
4.2.2 Системи нечіткої логіки з фаззифікатором і дефаззифікатором .....	59
4.2.3 Базис нечітких правил .....	60
4.2.4 Приклад системи нечіткої логіки .....	61
4.3 Реалізація системи підтримки прийняття рішень .....	65
4.4 Створення нейро-нечіткої мережі в середовищі Matlab .....	70
ВИСНОВКИ .....	77
ПЕРЕЛІК ПОСИЛАНЬ .....	77
ДОДАТОК А .....	<b>Ошибка! Закладка не определена.</b>
ДОДАТОК Б .....	<b>Ошибка! Закладка не определена.</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

OEM – original equipment manufacturer – оригінальний виробник обладнання;

АСУ – автоматична система управління;

LAN – local area network – локальна мережа;

WAN – world area network – світова мережа;

ОС – операційна система;

IEEE – Institute of Electrical and Electronics Engineers – інститут інженерів електротехніки та електроніки;

BCM – бездротова сенсорна мережа;

WEP – wired equivalent privacy – протокол безпеки бездротових локальних мереж;

WPA – Wi-Fi protected access – протокол безпеки для захисту бездротових мереж;

API – application programming interface – програмний інтерфейс приложения;

MAC – media access control – управління доступом до середовища;

PHY – physical layer – управління фізичним рівнем;

DHCP – dynamic host configuration protocol – протокол динамічного налаштування вузла;

DNS – domain name system – система доменних імен;

ZDO – ZigBee device object – об'єкт пристрою ZigBee;

ZDP – ZigBee device profile – профіль пристрою ZigBee;

NWK – network layer – мережевий рівень;

- ВПС – Військові Повітряні Сили;
- UART – universal asynchronous receiver-transmitter – універсальний асинхронний приймач;
- SPI – stateful packet inspection – інспекція пакетів зі зберіганням стану;
- GPIO – general-purpose input/output – інтерфейс введення/виводу загального призначення;
- АЦП – аналого-цифровий перетворювач;
- SoC – system-on-a-chip – система на кристалі;
- PAN – personal area network – персональна мережа;
- ЛВ - логічна відстань;
- ZCL – ZigBee cluster – бібліотека кластерів ZigBee;
- NWK – network layer – мережевий рівень;
- APS – application support sublayer – підрівень підтримки програм;
- CSMA/CD – carrier sense multiple access with collision detection – множинний доступ з прослуховуванням несучої і виявленням колізій;
- ОЗП – оперативний запом'ятовуючий пристрій;
- MIMO – multiple input multiple output – кілька входів, кілька виходів;
- MISO – master in slave out – вхід керуючого, вихід керованого.

## ВСТУП

Минуло не більше десятиліття, коли почалися інтенсивні наукові і технологічні дослідження можливостей використання сенсорних датчиків спільно з бездротовою мережею. Їх результатом стало створення нового виду телекомунікаційних мереж, що отримав назву БСМ.

Великий інтерес до вивчення таких систем обумовлений широкими можливостями застосування сенсорних мереж. Бездротові сенсорні мережі, зокрема, можуть використовуватися для передбачення відмови обладнання в

аерокосмічних системах і автоматизації будівель. Через свою здатність до самоорганізації, автономності та високою відмовостійкості такі мережі активно застосовуються в системах безпеки і військових додатках. Успішне застосування бездротових сенсорних мереж в медицині для моніторингу здоров'я пов'язано з розробкою біологічних сенсорів сумісних інтегральними схемами сенсорних вузлів. Але найбільшого поширення бездротові сенсорні мережі отримали в області моніторингу навколишнього середовища і живих істот.

На основі успішної реалізації ряду дослідницьких проектів на початку нашого століття були створені комерційні компанії, що забезпечують просування технологій БСМ на телекомунікаційні ринки світу. Інтенсивний розвиток мікроелектроніки дозволив вирішити завдання створення дешевих, низькоспоживаючих, багатofункціональних пристроїв, що мають малі габарити, і здатних передавати телеметричну інформацію по радіоканалу на короткі відстані. Вони стали базовими елементами в структурі бездротових сенсорних мереж - нового класу бездротових систем, які являють собою розподілену, стійку до відмови окремих елементів та здатну до самоорганізації мережу мініатюрних електронних пристроїв (сенсорів) з автономними джерелами живлення. Вузли такої мережі здатні ретранслювати інформацію, використовуючи безліч малопотужних передавачів і забезпечуючи значну площу покриття бездротової системою. Відмінною особливістю сенсорів є мініатюрні розміри і низька собівартість, що дозволяє використовувати їх у великій кількості для створення мережі збору і бездротової передачі телеметричних даних. При цьому передача даних відбувається поетапно, від одного пристрою на інший, а маршрути передачі формуються автоматично таким чином, щоб за кінцеве число пересилань по мережі між найближчими вузлами пакет з даними був переданий на шлюз, що має з'єднання з сервером. У разі виходу з ладу одного або декількох вузлів, структура мережі

автоматично змінюється, забезпечуючи можливість доставки інформації з усіх працюючих сенсорів до шлюзу.

У числі основних функціональних і експлуатаційних переваг БСМ слід виділити:

- можливість самонастроювання і самовідновлення мережі;
- масштабованість мереж з щільним розміщенням вузлів у просторі (від десятків до тисяч пристроїв);
- високі показники надійності і відмовостійкості за рахунок комунікаційної надмірності і наявності безлічі альтернативних маршрутів доставки даних;
- низька вартість і малі масогабаритні показники вузлів;
- висока енергетична ефективність (термін експлуатації може досягати декількох років при автономному електроживленні вузлів);
- стійкість до модифікації топології мережі і змін характеристик середовища поширення радіохвиль;
- здатність вузлів спільно обробляти отримані дані і приймати рішення на базі розподілених алгоритмів;
- можливість швидкого і при необхідності прихованого розгортання мережі.

## 1 АКТУАЛЬНІСТЬ ТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Актуальність теми

БСМ – це нова перспективна технологія, на основі якої інтенсивно ведуться прикладні розробки і виконуються масштабні проекти для різних галузей промисловості і систем військового призначення. Так, наприклад, на основі розробленого вченими університетів США прикладного програмного

забезпечення на базі ВПС у Флориді розгорнута БСМ, що забезпечує виявлення проникнення і переміщення людей і транспортних засобів в контрольованій зоні. Мережа охоплює зону площею 5 кв. км. і обладнана сенсорами з магнітоелектричними і температурними датчиками для розпізнавання і відстеження різних металевих об'єктів, в тому числі рухомих. У числі відомих прикладів прикладного використання БСМ - розгортання мережі, побудованої на основі обладнання Intel на борту нафтового танкера для моніторингу і профілактичного обслуговування судна. Робота БСМ проходила в умовах екстремальних температур, високої вібрації та значний рівень радіочастотних перешкод від суднових систем. При цьому розгорнута БСМ стійко функціонувала, автоматично здійснюючи реконфігурацію і відновлення працездатності при впливі екстремальних факторів. За останній період на розгортання БСМ для моніторингу інфраструктури світової нафтогазодобувної галузі компаній витрачали щорічно більше 200 млн. \$.

## 1.2 Постановка задачі

Метою роботи є дослідження технологій комп'ютерного діагностування станів розподілених об'єктів з випадковим вибором вузлів сенсорної мережі для зменшення енергоспоживання та підвищення безпеки в цілому.

На меті маєтись розглянути варіанти реалізації бездротових сенсорних мереж: MeshLogic та ZigBee та провести їх порівняння. Також важливою складовою атестаційної роботи є дослідження з метою визначення, і подальшого застосування, ефективного алгоритму Вороного для покриття контрольованої території датчиками для скорочення вартості розгортання мережі, а також забезпечення достатнього ступеня покриття території, що відслідковується, при збереженні максимально здійсненою тривалості функціонування і підтримки масштабованості мережі.

## 2 ВАРІАНТИ РЕАЛІЗАЦІЇ БЕЗДРОТОВИХ СЕНСОРНИХ МЕРЕЖ: ZIGBEE I MESHLOGIC

При виникненні завдання, що вимагає застосування технологій БСМ, розробники мають кілька варіантів вирішення проблеми. Перший варіант - побудова власної системи на основі стандарту ZigBee. Сьогодні є досить велика кількість різних ZigBee-продуктів, починаючи від мікросхем

приймачів стандарту IEEE 802.15.4 та закінчуючи готовими OEM-модулями з вбудованим програмним забезпеченням мережевого стека ZigBee. Переваги такого підходу – відносна простота та короткі терміни отримання працездатного рішення. Недолік – необхідність у витраті часу та ресурсів як на детальне вивчення самого стандарту ZigBee, так і на освоєння реалізації цієї технології в обраному продукті, тому що потрібне глибоке розуміння закладених в ZigBee механізмів для побудови ефективних ZigBee-мереж в реальних умовах експлуатації. Прискорити процес освоєння можна за допомогою засобів налагодження, які є практично у всіх виробників ZigBee-продуктів, однак вартість таких наборів розробника є досить високою. Крім того, при вирішенні поставленого завдання розробник обмежений рамками самого стандарту, тому не завжди можливо отримати «елегантне» рішення, а будь-який відступ від специфікації автоматично позбавить систему сумісності - основної переваги ZigBee [1].

Іншим варіантом є застосування однієї з доступних пропрієтарних платформ (наприклад, від Sensicast, Millennial Net або Dust Networks). Деякі з них використовують власні приймачі, а інші засновані на стандарті IEEE 802.15.4. Мережевий стек пропрієтарних платформ реалізований на базі патентованих алгоритмів і протоколів, які забезпечують ряд переваг в порівнянні з ZigBee, але ціною несумісності з рішеннями від інших виробників.

Як правило, подібна продукція являє собою набір апаратних модулів з різною функціональністю, вбудованого ПЗ вузлів і ПЗ персонального комп'ютера для налагодження та управління мережею. Недоліком цього підходу є закритість пропрієтарної платформи та відсутність можливості внесення змін до логіки мережевої взаємодії для більшої відповідності вимогам конкретного додатку.

Отже, подібні платформи підходять для інтеграції в більшу систему в якості єдиного цілого без будь-якої модифікації. Можна також спробувати самостійно розробити набір мережевих протоколів і реалізувати їх на власній апаратній базі. Таким чином, може бути отримане рішення, яке найбільш точно відповідає прикладній задачі, але часові та фінансові витрати на розробку можуть бути занадто великі, тому що для цього потрібні серйозні знання в області бездротових сенсорних мереж, що мають масу особливостей і відмінностей від традиційних систем передачі даних. Слід також зазначити, що налагодження мережі всього з декількох бездротових вузлів вже є досить нетривіальним завданням і вимагає високої кваліфікації розробника.

Таким чином, на даний момент часу існуючі комерційні рішення в області БСМ не відрізняються гнучкістю, яка необхідна для отримання ефективної мережі в конкретному додатку, а також є складними в освоєнні нефахівцями в даній області. В результаті підвищуються вартість і терміни розробки системи.

Саме тому було прийнято рішення про створення апаратно-програмної платформи MeshLogic для реалізації бездротових сенсорних мереж в різних областях застосування. Платформа не є готовим продуктом (виробом), який надається кінцевому споживачу, тому що передбачається її використання в якості бази для створення різних бездротових систем під конкретні вимоги замовника та з урахуванням специфіки прикладної задачі. Крім того, платформа може використовуватися як інструмент для проведення досліджень з розробки нових алгоритмів і протоколів в області БСМ.

При розробці платформи MeshLogic було сформовано завдання використовувати найбільш сучасні досягнення в області БСМ, а також враховувати переваги і недоліки існуючих комерційних рішень, в тому числі стандарту ZigBee[2].

Звісно, виникає резонне питання про відмінності платформи MeshLogic від інших існуючих рішень, тому нижче наведено короткий порівняльний аналіз платформи MeshLogic зі стандартом ZigBee. Порівняння з іншими комерційними платформами викликає труднощі через відсутність їх докладного опису.

## 2.1 Порівняння MeshLogic та ZigBee

Розглянемо основні характеристики стандарту ZigBee і платформи MeshLogic в різних аспектах, за допомогою опису відмін цих технологій.

### 2.1.1 Фізичний рівень

У ZigBee фізичний рівень і рівень доступу до середовища відповідають стандарту IEEE 802.15.4, в якому визначені наступні неліцензовані частотні діапазони для використання: 868 ~ 868,6 МГц, 902 ~ 928 МГц і 2405 ~ 2483,5 МГц.

Як і слід було очікувати, найбільша частина вироблених пристроїв працює в діапазоні 2,4 ГГц, тому будемо розглядати тільки цю область частот.

В IEEE 802.15.4 діапазон 2,4 ~ 2,4835 ГГц розділений на 16 каналів шириною 2 МГц з кроком 5 МГц.

Використовуються квадратурна фазова маніпуляція із зсувом і розширення спектра методом прямої послідовності 16 фіксованими квазіортогональними 32-розрядними кодами. В результаті технічна швидкість передачі даних в ZigBee становить 250 Кбіт / с при коефіцієнті розширення спектра (processing gain), що дорівнює 9 дБ.

Поточна реалізація бездротового вузла MeshLogic побудована на основі приймачів Cypress, які також працюють в діапазоні 2,4 ~ 2,4835 ГГц з розподіленням на 78 каналів (з урахуванням необхідного рознесення 39

каналів) шириною 1 МГц. У WirelessUSB застосовуються гауссова частотна маніпуляція і розширення спектра ортогональними послідовностями Голда довжиною 32 або 64 елементарних символів. Приймачі WirelessUSB дозволяють змінювати коди, що розширюються, тому також можлива організація множинного доступу з кодовим розділенням. Залежно від обраного режиму роботи швидкість передачі даних може бути 15625 біт / с, 31250 біт / с або 62500 біт / с, при цьому коефіцієнт розширення спектра становить 18, 15 і 12 дБ відповідно.

Таким чином, ZigBee і MeshLogic функціонують в загальнодоступному неліцензованому діапазоні частот 2,4 ~ 2,4835 ГГц, при цьому в ZigBee швидкість передачі даних вище, але фізичний рівень платформи MeshLogic володіє більшою стійкістю до перешкод. Крім того, для типових додатків БСМ швидкість передачі даних не є критичним параметром, в той час як втрата пакетів безпосередньо впливає на якість обслуговування мережі та енергоспоживання вузлів.

Слід зазначити, що принципово платформа MeshLogic дозволяє використовувати на фізичному рівні і інші приймачі, аналогічні Cypress WirelessUSB, в тому числі відповідні стандарту IEEE 802.15.4 [3].

### 2.1.2 Стійкість до співканальної інтерференції

З огляду на зростання кількості бездротових систем в неліцензованому діапазоні частот 2,4 ~ 2,4835 ГГц, БСМ повинна мати механізми підвищення стійкості до співканальної інтерференції.

Наприклад, в офісній будівлі велика ймовірність появи однієї або декількох мереж стандарту IEEE 802.11b / g, які можуть бути серйозною перешкодою для мережі ZigBee або MeshLogic. Як правило, в ZigBee і MeshLogic потужність корисного сигналу становить приблизно 1 мВт при ширині смуги 2 МГц і 1 МГц відповідно, в той час як в IEEE 802.11b / g

потужність може бути до 100 мВт з шириною смуги 22 МГц за рівнем ослаблення 30 дБ .

При створенні ZigBee-мережі необхідно вибрати один з 16 каналів у вільному від інших систем частотному діапазоні. Однак в процесі експлуатації може з'явитися джерело внутрішньосмугових перешкод, через яке частина мережі вийде з ладу, тому буде потрібно знову «вручну» задати новий активний канал, причому для всієї мережі. Отже, ускладнюється обслуговування системи [3].

Розглянемо для прикладу таку ситуацію. У великому будинку існує кілька Wi-Fi-мереж, налаштованих на частотні канали так, що весь діапазон 2,4 ~ 2,4835 ГГц ними перекритий (рисунок 2.1). Отже, всі 16 каналів стандарту IEEE 802.15.4 будуть потрапляти в область однієї або декількох Wi-Fi-мереж, тому в даних умовах практично неможлива нормальна робота розподіленої по всій будівлі ZigBee-мережі, тому що частина пристроїв не буде функціонувати через вплив розташованої поблизу мережі IEEE 802.11b/g.

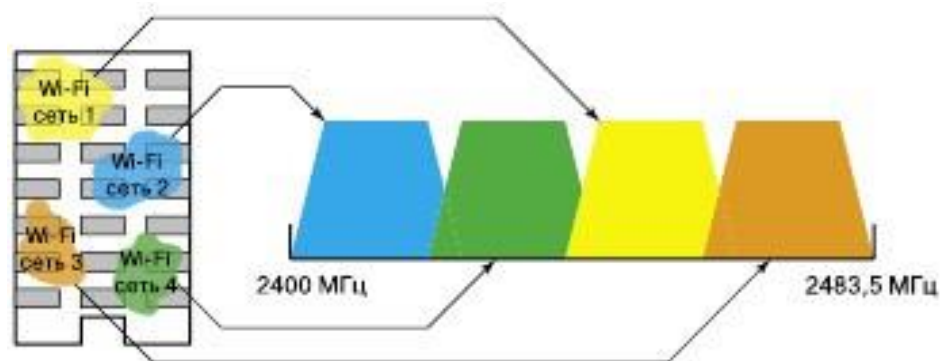


Рисунок 2.1 – Поширення Wi-Fi мереж.

Для вирішення подібних проблем в платформі MeshLogic реалізований механізм динамічного виділення каналів: в залежності від локальної ситуації з

перешкодами вибирається вільний частотний діапазон, при цьому різні ділянки мережі можуть працювати на різних каналах.

Спочатку у всій мережі MeshLogic використовується канал, обраний на етапі інсталяції системи. При досить великій і розподіленій у просторі бездротовій мережі, область впливу покриє тільки частину вузлів, через перешкоди, які з'явилися в певний момент часу на даній частоті. Після падіння якості зв'язку нижче порогового рівня, вузли перейдуть на нову частоту відповідно до алгоритму динамічного виділення каналів.

Відзначимо, що зміну каналу виконують тільки ті вузли, для яких це дійсно необхідно, інша ж частина мережі продовжує функціонувати на колишньому каналі. Звісно, зв'язність всієї мережі зберігається.

Очевидно, даний механізм дозволяє експлуатувати мережу MeshLogic в описаній вище ситуації, тому що різні ділянки мережі будуть використовувати ті чи інші частотні канали, обрані в залежності від сусідньої Wi-Fi-мережі (рисунок 2.2). Всі описані дії виконуються автоматично і не потребують втручання обслуговуючого персоналу [4].



Рисунок 2.2 – вибір каналів мережі MeshLogic

### 2.1.3 Профілі ZigBee

Структуру стека ZigBee / 802.15.4 складають програмна і апаратна частини. У першій реалізований код, написаний за специфікацією ZigBee, в другій - механізми MAC і РНУ рівнів, що містяться в стандарті 802.15.4. Код в скомпільованому вигляді, з доданими функціями управління інтерфейсами прийнято називати - профіль.

Як правило, виробники компонентів мережі ZigBee постачають кілька готових профілів. Профіль HomeLightControll управляє режимами освітлення в будинку і передає сигнал вмикання / вимикання від вимикачів до лампочок. Однак для створення мереж зі складною топологією і великою кількістю периферійних пристроїв такі стандартні профілі не завжди є універсальними. Для вирішення подібних завдань розробникам надається можливість створення власних профілів. Ця процедура виконується за допомогою програмного продукту ProfileBuilder, з використанням API постачальника стека ZigBee. Далі, створений профіль завантажується в ZigBee пристрій, який, в залежності від типу профілю, стає координатором, роутером або кінцевим пристроєм. При цьому профіль додатково володіє тим набором функцій, який було визначено користувачем.

Крім широких можливостей з реалізації профілів для побудови бездротової мережі телеметрії, виробники ZigBee пристроїв наділяють свою продукцію досить великою кількістю апаратних інтерфейсів. На сьогоднішній день доступні ZigBee пристрої, в яких реалізовані такі інтерфейси як UART, SPI, GPIO і ADC / DAC [4].

Одним із прикладів таких пристроїв є модулі виробництва компанії Cirronet, у яких є такі апаратні інтерфейси 1 SPI, 1 UART, 6 GPIO і 3 АЦП. Ці модулі виконані на базі трансиверів від Chipcon CC2420 і контролера ATmega128 компанії Atmel. Компанія MaxStream випускає аналогічні продукти, на яких доступні 1 UART, 6 GPIO і 6 АЦП. Ці модулі складаються з

RF-частини і контролера компанії Freescale, об'єднаних на одній платі розміром 30 \* 20 мм. Від більшості подібних продуктів інших виробників ці модулі відрізняє можливість використання AT-команд, завдяки яким суттєво спрощується процедура впровадження модуля в готовий виріб.

Крім модульних платформ, деякі виробники представляють інтегральні рішення SoC. SoC вдає із себе збірку, в якій rf-частина і контролер об'єднані на одному чіпі.

Стандарт ZigBee описує профіль стека, який визначає мережу, служби програм і параметри безпеки для всієї мережі. Профіль стека для конкретної ZigBee мережі вибирається мережевим адміністратором (проектувальником), виходячи з області застосування (автоматизація будівлі, управління техпроцесом, позиціонування). Кожен такий профіль стека потім буде визначати параметри мережі, такі як:

- загальні розміри мережі;
- підтримувані алгоритми маршрутизації;
- розміри таблиць маршрутизації і прив'язки програм; – характер служби безпеки для цієї мережі.

Особливий інтерес з точки зору топології мережі являє специфікація трьох параметрів: максимальна глибина мережі (`nwkMaxDepth`), максимальна кількість дочірніх зв'язків маршрутизатора (`nwkMaxChildren`), і максимальна кількість зв'язків з дочірніми маршрутизаторами (`nwkMaxRouters`). Разом вони визначають загальну структуру мережі [5].

Відповідно до стандарту IEEE 802.15.4 кожен мережевий пристрій має 64-розрядний (довгий) IEEE адресу, яка унікально ідентифікує цей пристрій. Щоб продовжити життя акумуляторних батарей, для скорочення розмірів пакетів, і відповідно часу активної взаємодії пристроїв використовуються більш короткі адреси. Стандарт ZigBee вимагає, щоб обмін повідомленнями після приєднання до мережі виконувався з 16-бітними (короткими)

мережевими адресами. 16-бітна мережева адреса присвоюється знов приєднаним пристроям під час обміну повідомленнями в процесі приєднання.

Координатор ZigBee завжди має адресу 0000. Координатор за встановленим стандартом ZigBee алгоритмом визначає, виходячи з параметрів профілю стека, діапазони адрес, які присвоюються приєднаним пристроям відповідно до їх розташування в мережевій ієрархії. Хоча ці діапазони адрес розрізняються, залежно від параметрів профілю стека, діапазони адрес завжди однакові для конкретного набору параметрів профілю стека.

#### 2.1.4 Мережеві протоколи

Стандарти IEEE 802.15.4 і ZigBee 2007 Specification визначають стандартизовані протоколи, які забезпечують мережеву інфраструктуру, необхідну для бездротових мережевих додатків з великим числом датчиків і виконавчих механізмів. IEEE 802.15.4 визначає фізичний і MAC рівні, а ZigBee визначає мережевий рівень і рівень програм.

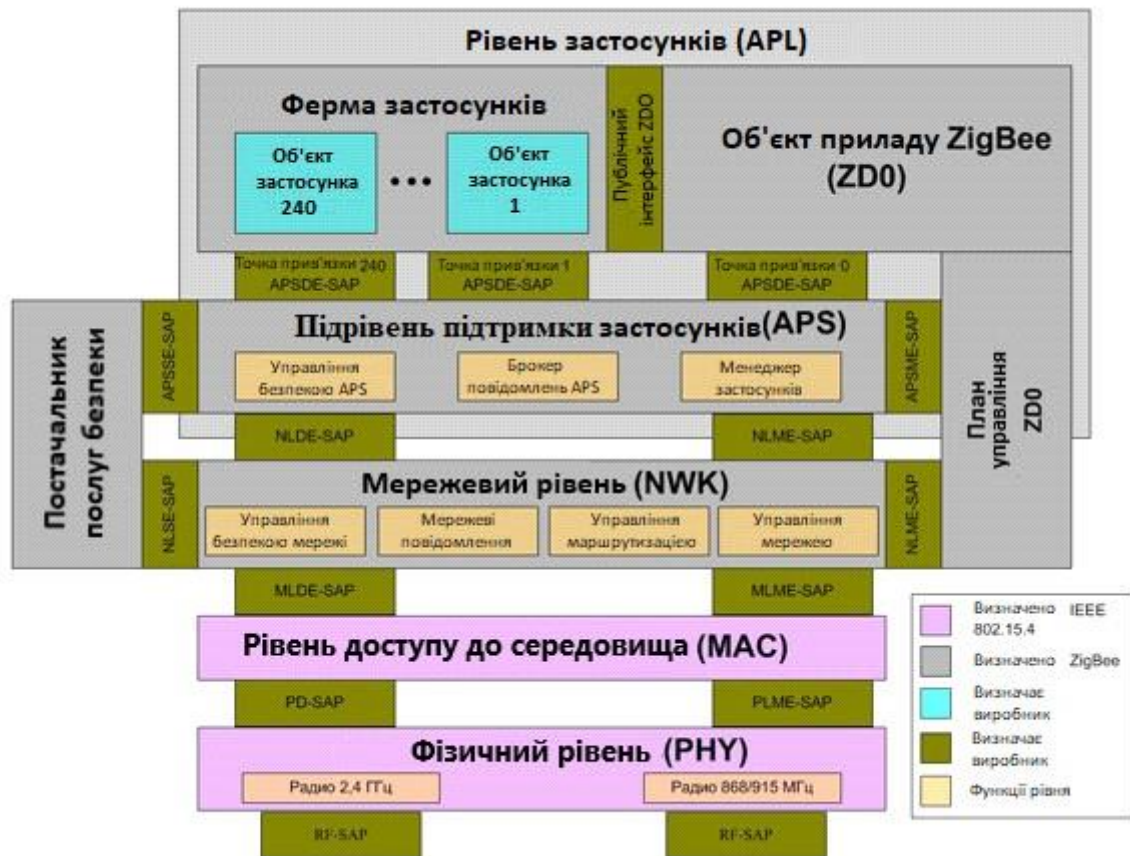


Рисунок 2.3 – Рівні мережевих протоколів ZigBee

Специфікація ZigBee регламентує стек протоколів взаємодії вузлів мережі, в якому протоколи верхніх рівнів використовують сервіси, що надаються протоколами нижніх рівнів.

Рівень додатків складається з ферми програм, об'єкта пристрою ZigBee і підрівня підтримки програм.

Ферма програм містить опис порядку створення профілів в стеці ZigBee, що гарантує їх сумісність, а також визначає:

- стандартні типи даних для профілів;
- дескриптори, які допомагають у виявленні служб;
- формати кадрів для транспортування даних; – значення пар ключів.

Це дозволяє швидко розробляти прості профілі на основі атрибутів.

Об'єкти програм – програмні модулі, що управляють пристроями ZigBee в кінцевих точках. Одна базова станція ZigBee може підтримувати до 240 об'єктів програм. Об'єкти програм підтримують кінцеві точки, пронумеровані від 1 до 240. Кінцева точка 0 зарезервована для об'єкта пристрою ZigBee.

Об'єкт пристрої ZigBee:

- визначає роль пристрою ZigBee: координатор, маршрутизатор або кінцевий пристрій;
- ініціює запити пошуку і приєднання і / або відповідає на такі запити; – встановлює безпечний зв'язок між пристроями ZigBee.

Він також надає великий набір команд управління, які визначаються в профілі пристрою ZigBee (використовуються при введенні в експлуатацію).

План управління підтримує зв'язок об'єкту пристрою з підрівнем підтримки програм та мережевим рівнем. Дозволяє об'єкту пристрою обробляти запити програм на доступ до мережі і забезпечення безпеки, використовуючи повідомлення профілю пристрою ZigBee [3].

Підрівень підтримки програм відповідає за надання даних програм і профілів пристрою ZigBee. Підрівень також керує приєднаннями в мережі ZigBee і зберігає дані про такі приєднання в своїй таблиці.

Постачальник послуг безпеки забезпечує механізми безпеки для рівнів, що використовують шифрування – мережевого і підтримки застосунка. Ініціюється і конфігурується об'єктом пристрої [6].

Мережевий рівень обробляє мережеві адреси і маршрутизацію за викликами MAC рівня. У його завдання також входять:

- запуск мережі (якщо пристрій є координатором);
- присвоєння мережевих адрес;
- додавання і видалення мережевих пристроїв;
- маршрутизація повідомлень; – застосування політики безпеки;

– здійснення пошуку маршрутів.

### 2.1.5 Типи сенсорних мереж

Однією з важливих особливостей мереж ZigBee є можливість вибору топології мережі, яка в найбільшій мірі відповідає конкретному застосуванню. Точна структура мережі ZigBee в більшості випадків не може бути заздалегідь визначена, так як залежить від взаємного розташування пристроїв і проходження радіохвиль під час формування мережі. Однак три згаданих параметра профілю стека будуть диктувати загальну структуру (топологію) мережі.

Наприклад, якщо програма має пристрої, фізично розташовані в лінію (наприклад, уздовж конвеєра), мережа має лінійну структуру, як показано на рисунку 2.4.

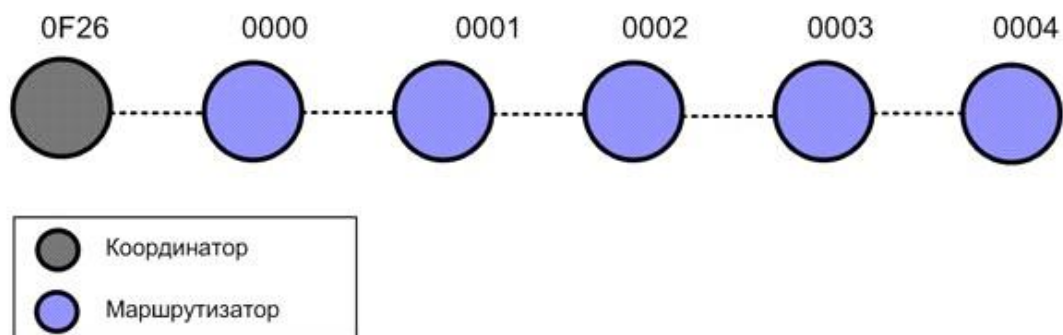


Рисунок 2.4 – Лінійна топологія мережі

Гнучкість мережевих структур ZigBee, що настроюються відповідно до галузі застосування, найкращим чином проявляється в ситуаціях, коли більшість інших мереж виявляються непридатні.

Найбільш перспективною в мережі ZigBee є Mesh-мережі (meshтопологія).

Сітчаста мережа - це мережа взаємопов'язаних маршрутизаторів і кінцевих пристроїв, в якій кожен маршрутизатор має, принаймні, два зв'язки і

може транслювати повідомлення своїх сусідів. Як показано на рисунку 2.5, сітчаста мережа складається з одного координатора і декількох маршрутизаторів і кінцевих пристроїв [7].

У такій мережі кожен пристрій може зв'язуватися з будь-яким іншим пристроєм як безпосередньо, так і через проміжні вузли мережі. Сітчаста топологія підтримує «багатокроковий» (multi-hop) зв'язок, при якому дані проходять кроками від одного пристрою до іншого, використовуючи найбільш надійні лінії зв'язку і найбільш ефективні маршрути, поки не досягнуть мети.

Здатність до багатокрокової передачі допомагає забезпечити живучість мережі (здатність до самовідновлення). Якщо один з пристроїв відмовляє або виявляється під впливом перешкод, мережа здатна перемаршрутизовуватись, використовуючи пристрої, які залишилися.

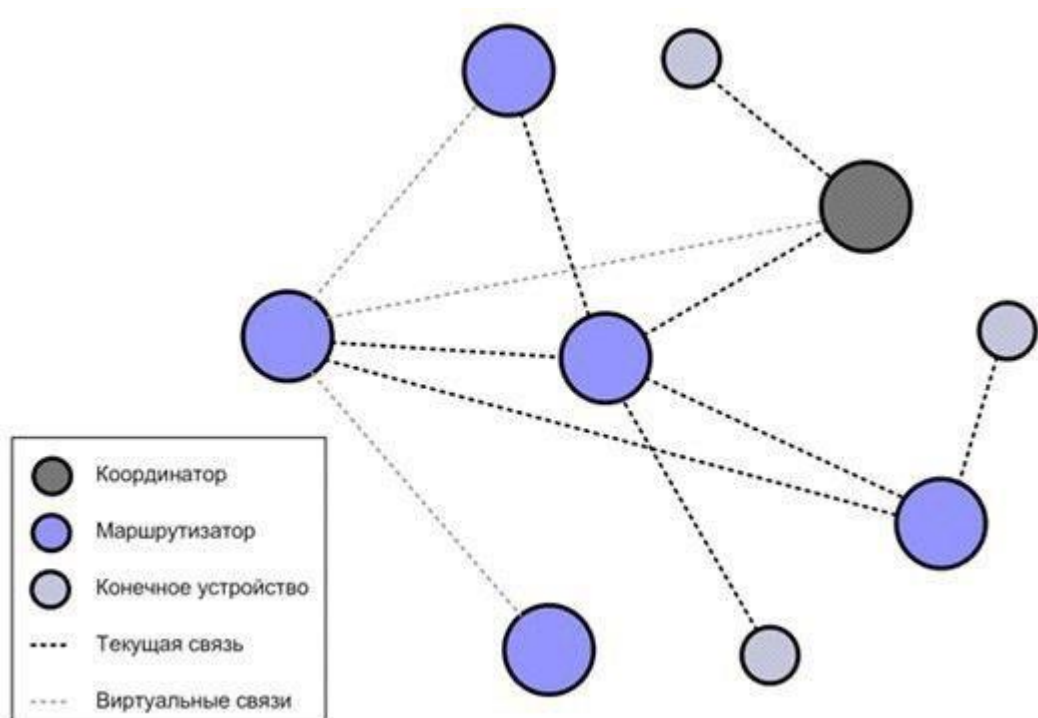


Рисунок 2.5 – Сітчаста мережа ZigBee

Принципова відмінність мереж ZigBee від інших бездротових мереж, таких як IEEE 802.11 / WiFi, в тому, що замість централізованої зіркоподібної

структури мережі ZigBee допускають ієрархію приєднання до мережі. В результаті може бути досягнуто кілька рівнів приєднання. Така ієрархія пристроїв служить для створення дійсно бездротових мереж.

Для порівняння, IEEE 802.11 / WiFi пропонує технологію точки бездротового доступу: бездротове підключення проводиться між, припустимо, ноутбуком або телевізором і точкою доступу, яка не є бездротовим маршрутизатором. Тому далі потік даних направляється на дроти (наприклад, по провідній мережі до інших провідних пристроїв або до іншої точки бездротового доступу). Мережа ZigBee здатна доставляти пакети повністю по бездротовій мережі за допомогою декількох проміжних бездротових пристроїв.

Переваги мережі ZigBee:

- сітчаста топологія має високу живучість і надійність. Якщо який небудь маршрутизатор стає недоступним, можуть бути знайдені і використані альтернативні маршрути;
- використання проміжних пристроїв при передачі даних робить сітчасту мережу дуже масштабованою;
- слабкі сигнали і мертві зони можуть бути легко виключені простим додаванням додаткових маршрутів.

#### 2.1.6 Маршрутизація в мережах ZigBee

Основний алгоритм маршрутизації в мережах ZigBee - «Ad hoc On Demand Distance Vector» заснований на понятті «вектор відстані» маршруту, коли кожен маршрутизатор, який бере участь в трансляції запиту маршруту від конкретного джерела до певного пункту призначення створює свій запис до

маршрутної таблиці. Цей запис як мінімум містить «логічну відстань» від джерела запиту і адресу попереднього маршрутизатора.

Алгоритм працює наступним чином:

- коли пристрою «джерело» (I) необхідно відправити пакет на пристрій «адресат» (A), джерело ініціює процес виявлення маршруту шляхом розсилки широкомовного пакета «запит маршруту», запитуючи маршрут до пристрою A (рисунок 2.6);

- кожен пристрій, який отримав цей запит маршруту, вносить свій запис до таблиці маршрутизації і може транслювати цей запит, включивши в нього таблицю зі своїм записом (рисунок 2.7);

- такого типу пристосуваннями, які отримали ретрансльований запит, транслюють його далі, і з кожним кроком «логічна відстань», пройдена запитом, збільшується на величину, що залежить від якості зв'язку на черговій ділянці;

- процес триває поки запит не досягає пристрою A, причому зазвичай пристрою A досягають декілька ретрансльованих запитів (від різних маршрутизаторів) і «логічні відстані» маршрутів цих пакетів різні;

- пристрій A відправляє відповідь за маршрутом, що має мінімальну «логічну відстань»;

- відповідь буде повертатися у відповідності з таблицею маршруту за шляхом, який пройшов попередній запит, поки не прибуде на пристрій I (рисунок 2.8).

Ця серія переданих в зворотньому напрямку відповідей формує прямий маршрут для майбутньої передачі пакетів від I до A (рисунок 2.9). Механізм «логічної відстані» дозволяє джерелу і вузлам, розташованим на шляху запиту, вибрати мінімальну «логічну відстань» маршруту від джерела до місця призначення .

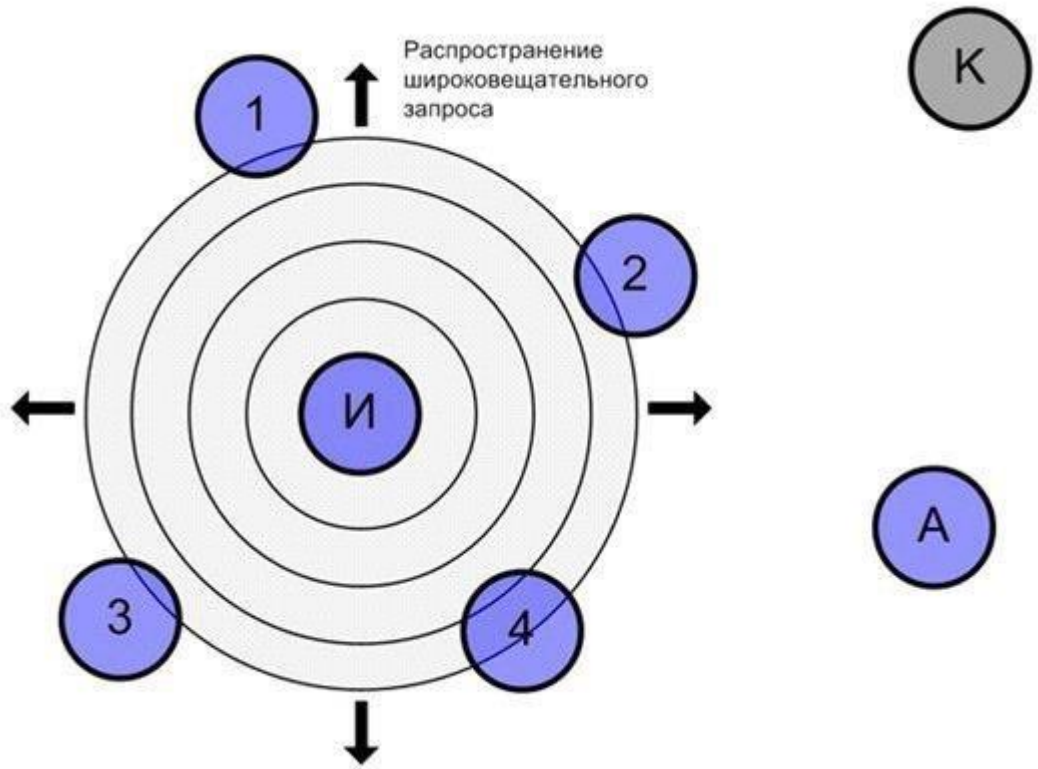


Рисунок 2.6 – Пошук маршруту. Крок 1

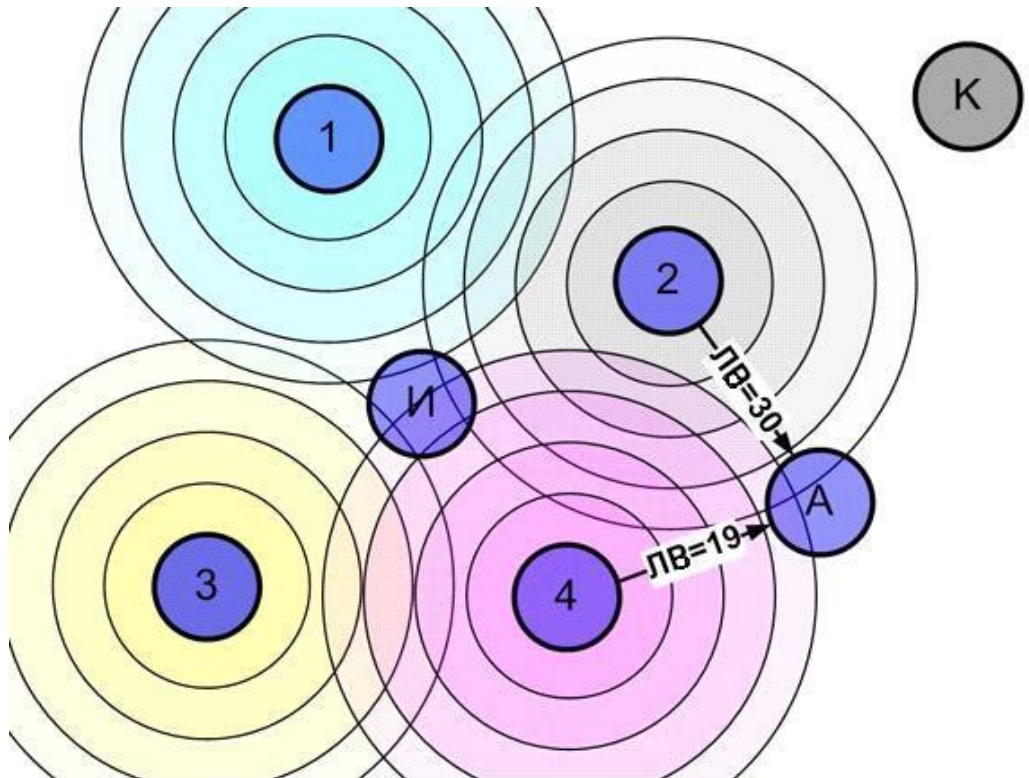


Рисунок 2.7 – Пошук маршруту. Крок 2

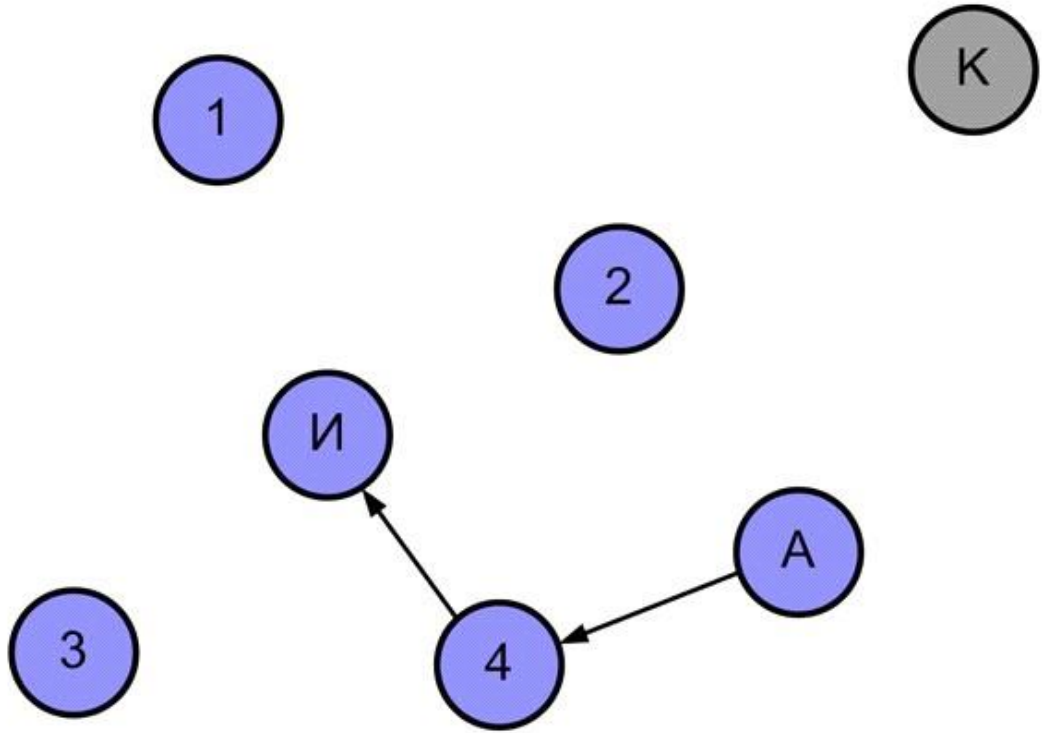


Рисунок 2.8 – Пошук маршруту. Крок 3

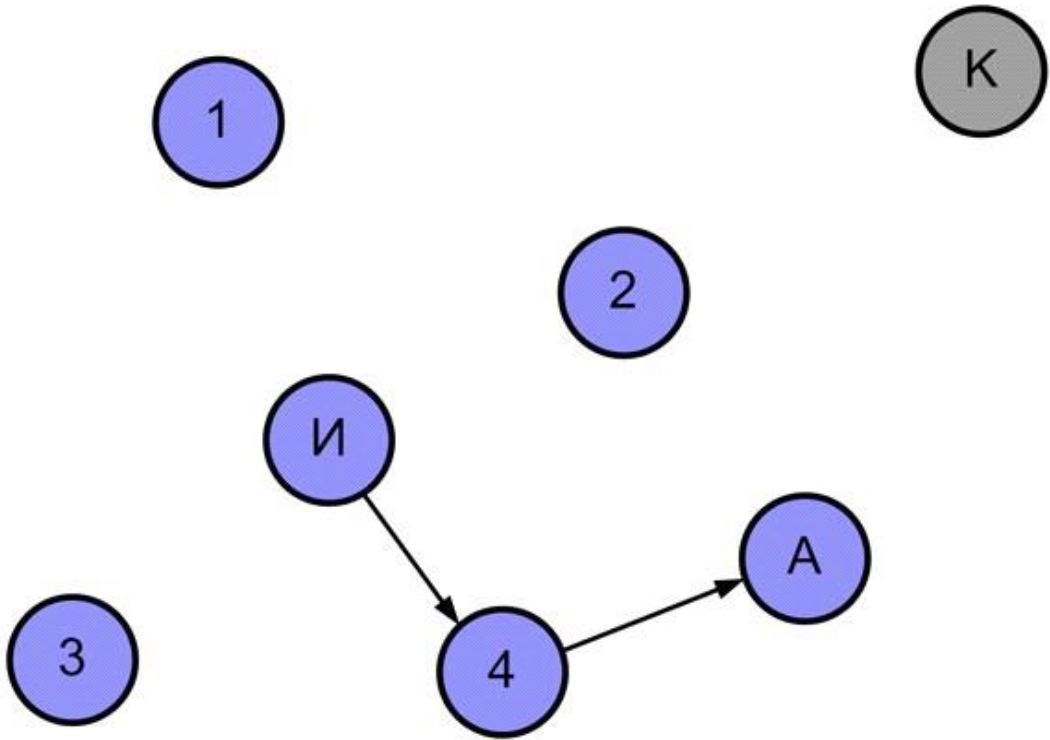


Рисунок 2.9 – Пошук маршруту. Крок 4

Описаний базовий алгоритм ефективний і універсальний, однак вимагає значного обсягу пам'яті для зберігання маршрутів, що ускладнює і здорожує пристрої.

Тому в мережах ZigBee реалізований і інший алгоритм, який дозволяє знизити вимоги до обсягу пам'яті, а іноді і зменшити мережевий трафік, необхідний для пошуку маршрутів. Дія алгоритму заснована на тому, що адреси в ZigBee мережі розподіляються ієрархічно, починаючи з координатора. Пристрій, що не має можливостей маршрутизації або пристрій, можливості маршрутизації якого вичерпані, може скористатися ієрархічної маршрутизацією - менш ефективною, але, тим не менш, цілком практичною.

У процесі формування ZigBee мережі алгоритм розподілу адрес задає діапазони адрес мережевих пристроїв в ієрархічному порядку, починаючи з координатора. В результаті будь-який пристрій в мережі, який знає свою адресу і адресу одержувача пакета, може визначити, чи належить конкретна мережева адреса до «спадної» гілки (і до якої саме), або знаходиться в іншому місці в ієрархії пристроїв. Отже, будь-який пристрій може прийняти просте рішення маршрутизації: передавати пакет «вгору» - в напрямку координатора або «вниз» - до дочірнього пристрою.

Приклад ієрархічної маршрутизації представлений на рисунку 2.10.

Як у випадку на рисунку 2.9, пакет, відправлений пристроєм І, призначений для пристрою А. Однак пристрій 4 вичерпав свої можливості маршрутизації, тому він не може транслювати пакет безпосередньо на пристрій А, а замість цього, використовуючи ієрархічну маршрутизацію, направляє цей пакет «вгору» по ієрархічній драбині – на пристрій 2. Далі пакет транслюється на координатор К, який передає його на шукану адресу А.

Переваги ієрархічної маршрутизації полягають в її простоті і меншому використанні ресурсів, що дозволяє створювати дуже недорогі пристрої без можливостей маршрутизації, які, тим не менш, можуть брати участь в будь-якій

ZigBee сумісній мережі. Недолік алгоритму в тому, що пакети будуть рухатися вгору і вниз до координатора і назад навіть в тому випадку, коли між джерелом і приймачем можливий прямий зв'язок [8].

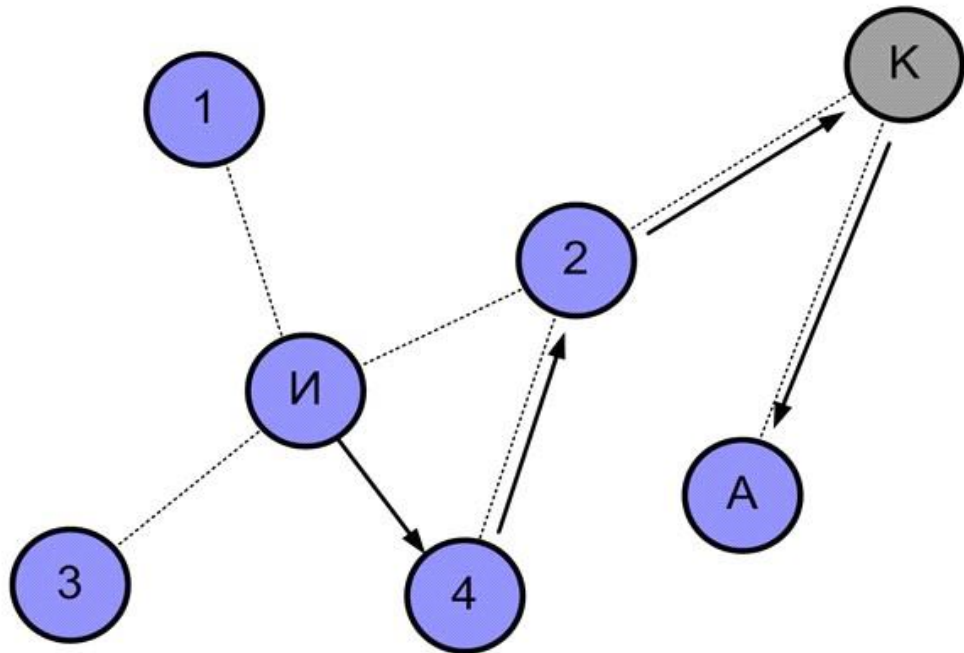


Рисунок 2.10 – Ієрархічна маршрутизація

У багатьох, якщо не в більшості програм, що використовують бездротові мережі, є спеціальний електронний пристрій, часто званий агрегатором, до якого інші мережеві пристрої повинні регулярно направляти дані. З метою економії мережевого трафіку в ZigBee PRO передбачений спеціальний випадок пошуку маршрутів, при якому єдиний широкомовний запит маршруту від агрегатора дозволяє встановити в роутінгових таблицях всіх маршрутизаторів мережі записи де агрегатор є одержувачем [9].

Типова інфраструктура системи складається з декількох зон ZigBee мереж, пов'язаних з сервером через сегменти локальної IP мережі. Кожна з зон є самостійною ZigBee мережею, що складається, як правило, не менше ніж з чотирьох анкерів, щодо яких позиціонуються мітки, які знаходяться в зоні ZigBee мережі. Один з анкерів в кожній зоні завжди є ще й координатором мережі. Два анкера в кожній ZigBee зоні системи забезпечені IP інтерфейсами

і виконують функції шлюзів, що зв'язують зону з сервером через незалежні IP канали, що забезпечує резервування і відповідно надійність інфраструктури в цілому. Приклад інфраструктури, яка обслуговує багатоповерховий будинок і прилеглу територію, показаний на рисунку 2.11.

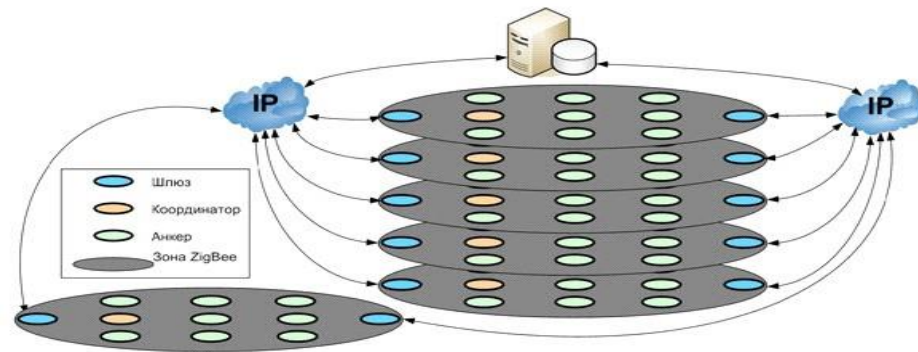


Рисунок 2.11 – Інфраструктура системи позиціонування у реальному часі

Профіль програми описує набір пристроїв, що використовуються для певної програми, і непрямо схеми обміну повідомленнями між цими пристроями. Наприклад, є профілі програм для «розумного будинку», автоматизації будівлі, системи безпеки і так далі. Для унікальної ідентифікації програм кожній з них виділяється ідентифікатор профілю.

Існує два типи профілів програм:

– загальні профілі програм – сумісне прикладне програмне

забезпечення, розроблене союзом ZigBee, яке виконує конкретне завдання;

– профілі виробників обладнання – приватні профілі програм, розроблені компаніями для роботи пристроїв ZigBee.

Пристрої в межах профілю програми спілкуються один з одним за допомогою кластерів, які можуть бути входами або виходами пристрою. Наприклад, в профілі «розумного будинку» є кластер, призначений для

підсистем контролю освітлення. Ідентифікатор кластера унікально ідентифікує кластер в рамках конкретного профілю.

Кінцева точка визначає об'єкт зв'язку всередині пристрою, через який реалізується певний додаток. Наприклад, пульт дистанційного керування може виділити кінцеву точку 6 для контролю освітлення в спальні, кінцеву точку 8 для управління системою обігріву та кондиціонування повітря і кінцеву точку 12 для управління системою безпеки. Це дозволяє пульту дистанційного керування незалежно зв'язуватися з цими пристроями і визначати, які пакети призначені для кожної програми і пристроя.

У будь-якому пристрої ZigBee можна використовувати до 240 кінцевих точок (1 - 240). Кінцева точка нуль виділена для об'єкта пристрою ZigBee, який забезпечує команди контролю і управління самим пристроєм ZigBee [3].

### 2.1.7 Бібліотека кластерів ZigBee

ZCL - це бібліотека кластерів, які можуть використовуватися будь-яким додатком. Це дозволяє використовувати кластери в різних сферах. Зокрема, кластери освітлення можуть використовуватися в будь-якому додатку, яке потребує управління освітленням, наприклад, в «розумному будинку» або в системі автоматизації комерційної будівлі.

Кожен кластер має два «кінця» - клієнт і сервер (рисунок 2.12).



Рисунок 2.12 – Зв'язка «клієнт-сервер»

Кластери в бібліотеці кластерів організовані в ряд функціональних доменів, включаючи:

- освітлення;
- клімат-контроль;
- вимірювання;
- охорона і безпека;
- протоколи;
- загального призначення і так далі.

Зв'язування є сполуками між двома кінцевими точками, причому кожне зв'язування підтримує конкретний профіль програми, а кожен тип повідомлення представлений кластером в цьому профілі.

Зв'язування можуть бути створені між окремими кінцевими точками або їх групами, такими, як світильники і вимикачі, які мають збіжні кластери введення і виведення (тобто, один і той же ідентифікатор кластера). Пристрій ZigBee може мати до 240 кінцевих точок, тому кожен пристрій може підтримувати безліч зв'язувань [5].

Зв'язування можуть зберігатися всередині вихідного пристрою. Наприклад, в пульті дистанційного керування можуть зберігатися адреси і ідентифікатори кінцевих точок усіх програм, з якими він взаємодіє. Це називається прямим зв'язуванням або зв'язуванням джерела.

Або відомості про зв'язування можуть зберігатися в проміжному пристрої, що підтримує таблицю підстановки, яка співвідносить всі кінцеві точки джерел і одержувачів. Це називається непрямим зв'язуванням.

### 2.1.8 Безпека ZigBee

Система безпеки ZigBee, яка заснована на 128-бітному AES-алгоритмі, доповнює модель безпеки, що надається IEEE 802.15.4. Служби безпеки

ZigBee включають методи створення ключів і транспорту, управління пристроями і захисту пакетів даних.

Специфікація ZigBee визначає безпеку для MAC, NWK і APS рівнів.

Безпека додатків зазвичай забезпечується через профілі додатків.

Складові системи безпеки ZigBee:

Центр управління безпекою.

Центр управління безпекою визначає, чи слід дозволити або заборонити приєднання нових пристроїв до своєї мережі.

Центр управління безпекою може періодично оновлювати ключ мережі і переходити на новий ключ. Спочатку він транслює новий ключ, зашифрований за допомогою старого ключа мережі. Потім він повідомляє всім пристроям про перехід на новий ключ.

Центром управління безпекою зазвичай є координатор мережі, але це може бути і виділений пристрій.

Центр управління відіграє наступні ролі в забезпеченні безпеки:

- довірений менеджер по перевірці достовірності пристроїв, що бажають приєднатися до мережі;
- менеджер мережі, що підтримує і поширює мережеві ключі;
- диспетчер конфігурації, що забезпечує безпеку взаємодії пристроїв.

ZigBee використовує три типи ключів для управління безпекою: головний ключ, мережевий ключ і ключ каналу зв'язку.

Головні ключі не використовуються для шифрування пакетів. Вони використовуються як початковий секретний код, що поділяється двома пристроями, коли пристрої виконують процедуру генерації ключа каналу зв'язку.

Головні ключі, які створюються центром управління безпекою, називаються основними джерелами центру безпеки, в той час як всі інші ключі називаються основними ключами рівня додатків.

Мережеві ключі забезпечують безпеку мережевого рівня в мережі ZigBee. Мережеві ключі мають всі пристрої в мережі ZigBee.

Мережеві ключі високої безпеки повинні відправлятися по бездротових каналах тільки в зашифрованому вигляді, в той час як стандартні мережеві ключі безпеки можуть відправлятися як зашифрованими, так і не зашифрованими.

Ключі каналів зв'язку забезпечують безпеку одноадресної передачі повідомлень між двома пристроями на рівні додатків.

У режимі стандартної безпеки перелік пристроїв, головних ключів, ключів каналів зв'язку і мережевих ключів можна зберігати як в центрі управління безпекою, так і в самих пристроях. Центр управління безпекою, проте, відповідає за підтримання стандартного мережевого ключа і контролює політику прийому в мережу. В цьому режимі вимоги до ресурсів пам'яті центру управління безпекою набагато нижче, ніж для режиму підвищеної безпеки.

У режимі підвищеної безпеки центр управління безпекою веде список пристроїв, головних ключів, ключів каналів зв'язку і мережевих ключів, необхідних для контролю і застосування політики поновлення мережевих ключів і доступу в мережу. В цьому режимі по мірі зростання кількості пристроїв в мережі швидко зростає необхідний центру управління безпекою обсяг пам'яті.

Додаткові можливості безпеки, які забезпечує ZigBee PRO, надзвичайно важливі, тому що мережі ZigBee використовуються в відповідальних додатках.

Під введенням в експлуатацію розуміється фізичне розгортання конкретного рішення і логічна прив'язка вузлів для формування

функціональної мережі. У широкому сенсі цього слова введення в експлуатацію охоплює широке коло завдань, включаючи:

- вимір радіо і фізичного середовища;
- розміщення пристроїв;
- конфігурацію їх параметрів;
- прив'язку додатків;
- оптимізацію мережі та параметрів пристроїв;
- тестування і підтвердження правильності роботи.

Крім того, при введенні в експлуатацію найчастіше доводиться вирішувати суміжні питання, такі як наявність знаючого і досвідченого персоналу, наявність необхідного обладнання, функціональна сумісність використовуваних пристроїв, а також їх сумісність з іншими бездротовими або дротовими системами.

В ідеалі пристрої повинні запускатися самі. Наладчик тільки подає на них напругу, включає і стоїть осторонь, спостерігаючи, як пристрої працюють поза мережею, до якої вони повинні приєднатися, як вони безпечно включаються в цю мережу, до якого пристрою (або пристроїв) вони приєднуються і з якими пристроями спілкуються.

Але щоб досягти цього ідеалу наладчик спершу повинен виконати всю свою роботу.

А якщо подивитися на кількість завдань, пов'язаних з введенням в експлуатацію, і врахувати, що чимало наладчиків, швидше за все, мають обмежене уявлення про застосовувану технологію, легко зрозуміти цінність налагоджувальних інструментів.

Інструменти наладчика зазвичай працюють на ноутбучі або КПК і розроблені, щоб спростити наладку і введення в експлуатацію. Ці інструменти мають інтуїтивно зрозумілий інтерфейс, що приховує складність базової

технології. А головне, забезпечують гнучку візуалізацію мережі і пристроїв, а також надають кошти для настройки, запуску та управління системою.

Налагоджувальні інструменти, як правило, не призначені для роботи в складі мережі при поточній експлуатації, а просто, не будучи частиною основного додатка, полегшують введення в експлуатацію та періодичне обслуговування або управління.

### 2.1.9 Енергоспоживання БСС

Одна з ключових особливостей БСС полягає в енергетичній ефективності. У більшості випадків мінімальне енергоспоживання є більш пріоритетним критерієм, ніж пропускна здатність, латентність, рівномірність надання доступу до середовища. У цьому полягає основна специфіка мережевих протоколів для БСС.

У ZigBee найменше енергоспоживання досягається при синхронізованому (beacon mode) доступі до середовища, який дозволяє переводити в «сплячий» режим як кінцеві пристрої, так і маршрутизатори.

Однак на практиці досить важко налаштувати розклад доступу до середовища при складній топології мережі або при нерівномірній швидкості генерації трафіку. Більш простим є чисельний доступ до середовища за алгоритмом CSMA / CA, але при цьому координатори повинні постійно перебувати в режимі прослуховування каналу, тому маршрутизаторам потрібна стаціонарна мережа живлення.

Отже, можна очікувати, що в багатьох практичних завданнях тільки кінцеві пристрої будуть працювати на автономному джерелі живлення (батарея), а маршрутизатори і PAN-координатор - від мережі живлення.

У платформі MeshLogic усі вузли здатні працювати від автономного джерела живлення, при цьому не потрібне будь-яке налаштування режимів доступу до середовища. Розроблений MAC-протокол динамічно вибирає

оптимальний з точки зору енергоспоживання режим роботи кожного вузла в залежності від його поточного мережевого навантаження. Система самостійно адаптується під мінливі вимоги прикладного рівня, що необхідно, наприклад, в мережах зі змінним періодом формування пакетів даних.

#### 2.1.10 Вимоги до апаратних ресурсів

Прагнення мінімізувати енергоспоживання і габарити пристроїв накладає обмеження на апаратне забезпечення бездротового вузла.

Як правило, в подібних пристроях застосовуються 8- або 16-розрядні низькоспоживаючі мікроконтролери, які не відрізняються обчислювальними можливостями і великими обсягами пам'яті. Отже, мережеві протоколи повинні висувати мінімальні вимоги до апаратних ресурсів вузла.

Вважається, що реалізація ZigBee-стека може займати від 4 до 32 Кбайт флеш-пам'яті залежно від функціонального типу пристрою.

Зрозуміло, що розмір стека впливає на величину доступних прикладному рівню ресурсів мікроконтролера, тому багато сучасних ZigBeeплатформ припускають двопроцесорну реалізацію вузла, в якій один мікроконтролер займається виконанням мережевих функцій, а другий - рішенням прикладних завдань. У наслідок збільшуються енергоспоживання, вартість і габарити.

У платформі MeshLogic вимоги до апаратних ресурсів усіх вузлів однакові і для поточної реалізації становлять близько 12 Кбайт пам'яті програм і 2 Кбайт ОЗП.

Отже, в більшості випадків буде достатньо одного мікроконтролера в складі вузла. [5]

### 3 ДІАГРАМА ВОРОНОГО ДЛЯ РОЗМІЩЕННЯ СЕНСОРІВ

Для розподілу сенсорів на території використовується діаграма Вороного.

### 3.1 Діаграма Вороного

Перше використання цієї діаграми зустрічається у праці Рене Декарта (1596-1650) «Засади філософії» (1644). Декарт запропонував поділ Всесвіту на зони гравітаційного впливу зірок. Тільки через два століття, відомий німецький математик Йоганн Петер Густав Лежен-Діріхле (1805 - 1859) ввів діаграми для двох-і тривимірних випадків. Тому їх іноді називають діаграмами Діріхле. Ну а вже в 1908 році російський математик Георгій Феодосійович Вороний (16 (28) квітня 1868 - 7 (20) листопада 1908) описав цю діаграму для просторів великих розмірностей, з того часу діаграма носить його прізвище [10].

Діаграма складається з так званих локусів - областей, в яких присутні всі точки, які знаходяться ближче до цієї точки, ніж до всіх інших. У діаграмі Вороного локуси є опуклими багатокутниками.

Локус будується наступним чином: нехай дана множина з  $n$  точок, для якої буде будуватися діаграма (рисунок 3.1). Береться конкретна точка  $p$ , для якої будується локус, і ще одна точка з даної множини -  $q$  (не дорівнює  $p$ ). Проводиться відрізок, що з'єднує ці дві точки, і проводиться пряма, яка буде серединним перпендикуляром даного відрізка. Ця пряма ділить площину на дві півплощини - в одній лежить точка  $p$ , в іншій лежить точка  $q$ . В даному випадку локусами цих двох точок є отримані півплощини. Тобто для того, щоб побудувати локус точки  $p$ , потрібно отримати перетин всіх таких напівплощин (тобто на місці  $q$  побувають всі точки даної множини, крім  $p$ ).

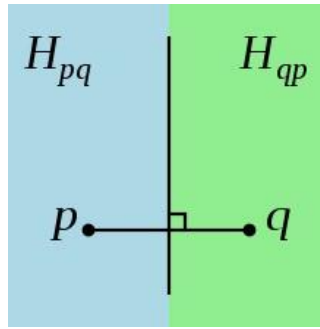


Рисунок 3.1 – Побудова локусу

Точку, для якої будується локус, називають сайтом (site). На наступному рисунку 3.2 локуси позначені різними кольорами [11].

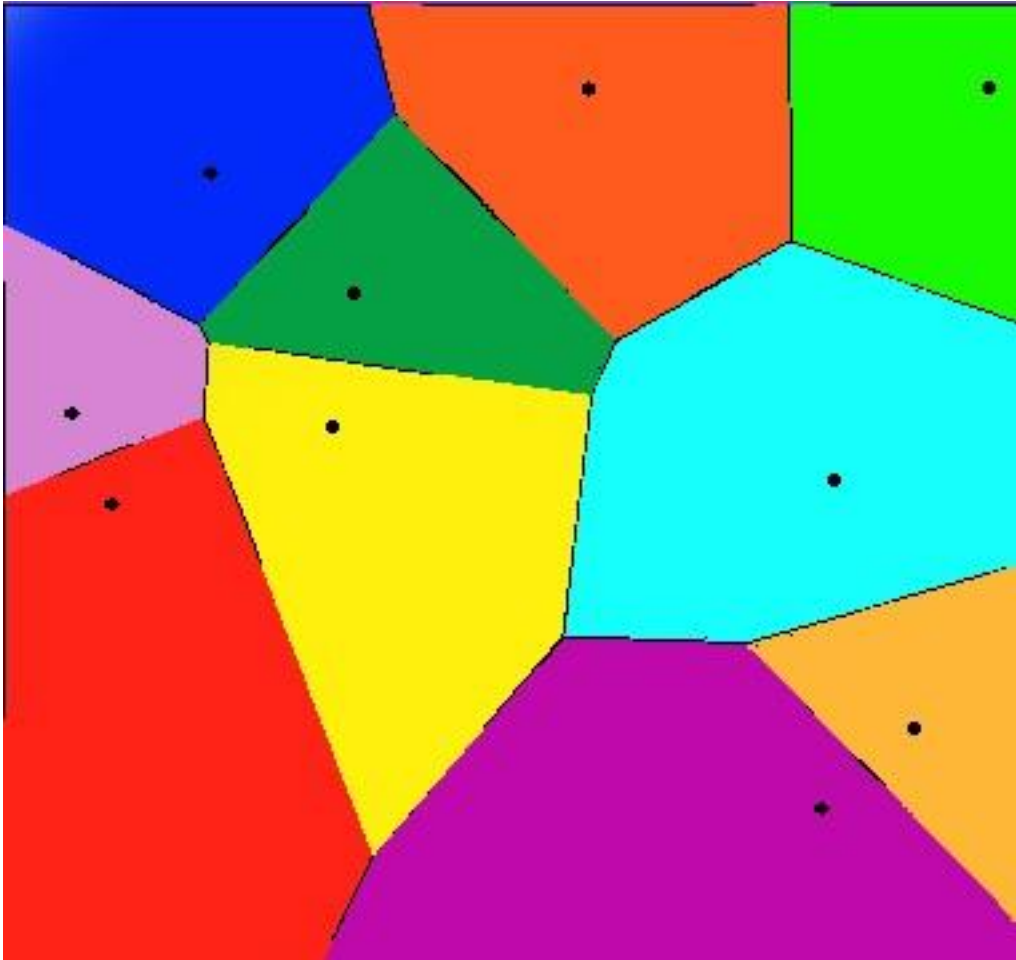


Рисунок 3.2 – Приклад з багатьма локусами

Алгоритми побудови діаграми як раз і є не що інше, як алгоритми побудови цих самих локусів для всіх точок з заданого набору. Локуси в даній задачі також називають багатокутниками Вороного або комірками Вороного.

### 3.2 Алгоритми побудови діаграми

Існує 4 алгоритма побудови діаграми Вороного:

- алгоритм побудови діаграми Вороного «в лоб». Складність:  $O(n^4)$ ;
- алгоритм побудови діаграми Вороного шляхом перетину

напівплощин складність:  $O(n^2 \log(n))$ ;

- алгоритм Форчун побудови діаграми Вороного на площині.

Складність:  $O(n \log(n))$ ;

- рекурсивний алгоритм побудови діаграми Вороного.

Складність:  $O(n \log(n))$ .

### 3.3 Алгоритм побудови діаграми Вороного шляхом перетину напівплощин

Нижче наведений алгоритм “поділяй і пануй” побудови діаграми Вороного при складності  $O(n^2 \log(n))$ .

Початкова множина точок сортується по одній з координат (припустимо,  $x$ ) і ділиться на дві, приблизно рівні множини. Кожна з отриманих множин знову ділиться на два і так відбувається до тих пір, поки в кожній з множин залишиться не більше двох точок. Легко побачити, що таких розбиття буде не більше ніж  $\log(n)$  [11].

Далі, для кожної одержаної множини будуються діаграми Вороного, після чого, в зворотньому порядку до ділення, ці діаграми об'єднуються в одну. Отримана діаграма Вороного і буде кінцевим результатом.

Щоб описаний алгоритм мав складність порядку  $O(n \cdot \log(n))$ , необхідно виконувати процес об'єднання двох діаграм Вороного за  $O(N)$ . Для множини з двох точок, діаграмою Вороного буде серединний перпендикуляр відрізка утвореного цими двома точками.

Припустимо, що всі точки відсортовані по X координаті, а вихідна множина розділена на два і для кожної побудована діаграма Вороного.

Виконаємо об'єднання даних множин та їх діаграм наступним чином:

1. Для кожного з підмножин знайдемо опуклу оболонку (побудова опуклої оболонки для кожної з множин можна виконувати всі тим же 'поділяй і пануй': тобто, на кожному кроці об'єднання діаграм Вороного ми об'єднуємо і опуклі оболонки даних множин за  $O(N)$ ).

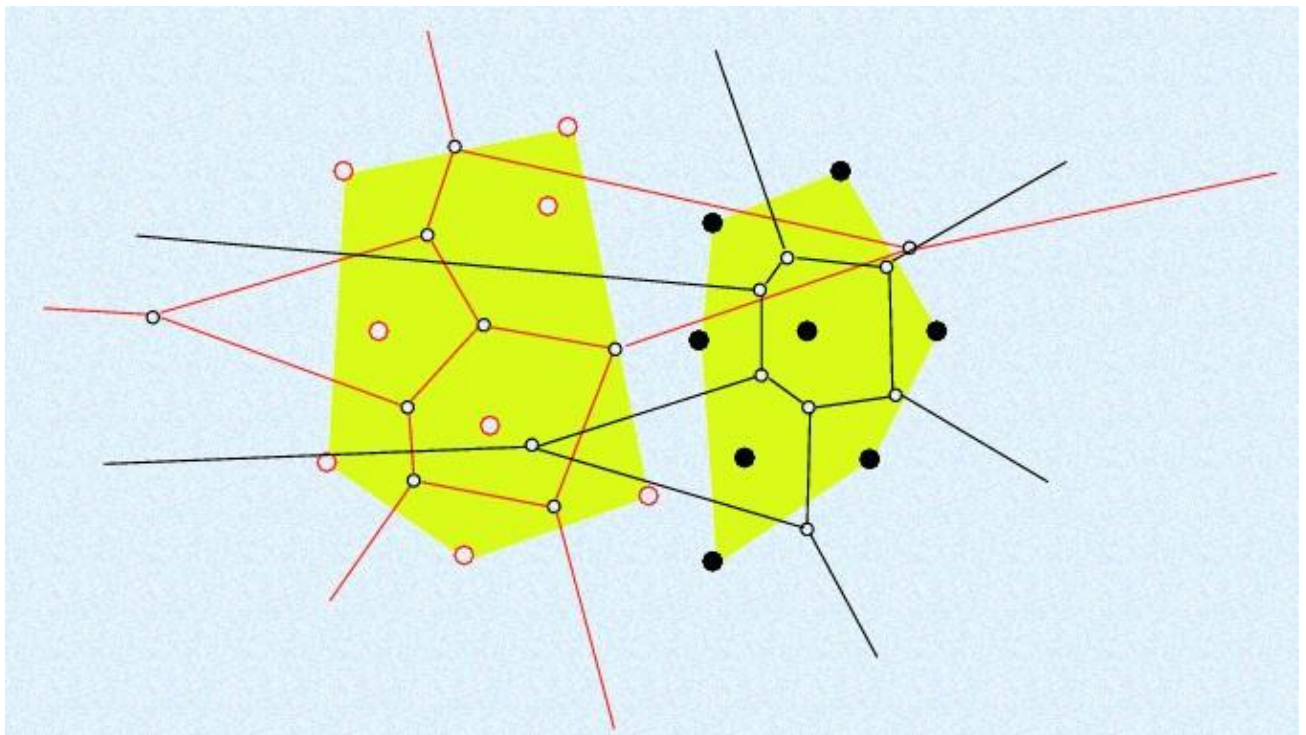


Рисунок 3.3 – Побудова діаграми Вороного (крок 1)

На рисунку показано початкову множину точок, де червоні точки - це точки першої множини, а чорні - другої. Для кожної множини побудована опукла оболонка, яка зафарбована зеленим кольором. Для першої множини діаграма Вороного червона, для другої – чорна [11].

2. Тепер, коли є дві опуклі оболонки вихідних множин, знайдемо верхню і нижню кордону даних множин: тобто, ми повинні знайти два таких відрізка, які об'єднують дві дані опуклі оболонки в одну (звісно, опуклу). Таким чином ми виконаємо умови кроку 1, а також отримаємо значення, що ініціалізує, для кроку 3. Даний крок можна виконати за  $O(N)$ .

3. З отриманих відрізків на кроці 2, виберемо будь-який і позначимо за  $L$  (останній, що залишився, позначимо за  $Q$ ), і через його середину, перпендикулярно пускаємо безкінечний промінь. Уявімо, що даний промінь тільки входить у початкову множину, і знайдемо його перетин з комірками діаграм Вороного початкових множин (вважається, що промінь тягнеться вперед, тобто у нього є напрям). Ми перетинаємо промінь тільки з тими комірками Вороного, центрами яких є кінці відрізка, перпендикулярно яким ми пускаємо промінь. Нам потрібно знайти точки перетину даного променя з відповідними осередками Вороного і вибрати серед них ту, що перетинається раніше. Позначимо цю точку за  $M$ , а комірку, яку ми перетнули, запам'ятаємо і позначимо  $V$ . Кінець відрізка  $L$ , що є центром для комірки Вороного, яка не перетиналася, залишаємо в спокої, а ось той, що був центром комірки, яку перетнули - оновлюємо: перетнули одну зі сторін комірки Вороного, тоді новим кінцем відрізка  $L$  стане центр комірки Вороного, суміжної по цій стороні з коміркою, яка була перетнута. У спеціальну множину  $S$  (в ній зберігається межа двох Діаграм Вороного) треба додати ту частину променя, яка простягається до перетину зі стороною комірки. Повторюємо крок 3 до тих пір, поки значення кінців відрізка  $L$  не стануть дорівнювати значенням кінців

відрізка  $Q$ . В результаті, в множині  $S$  виявиться безперервний ламаний промінь.

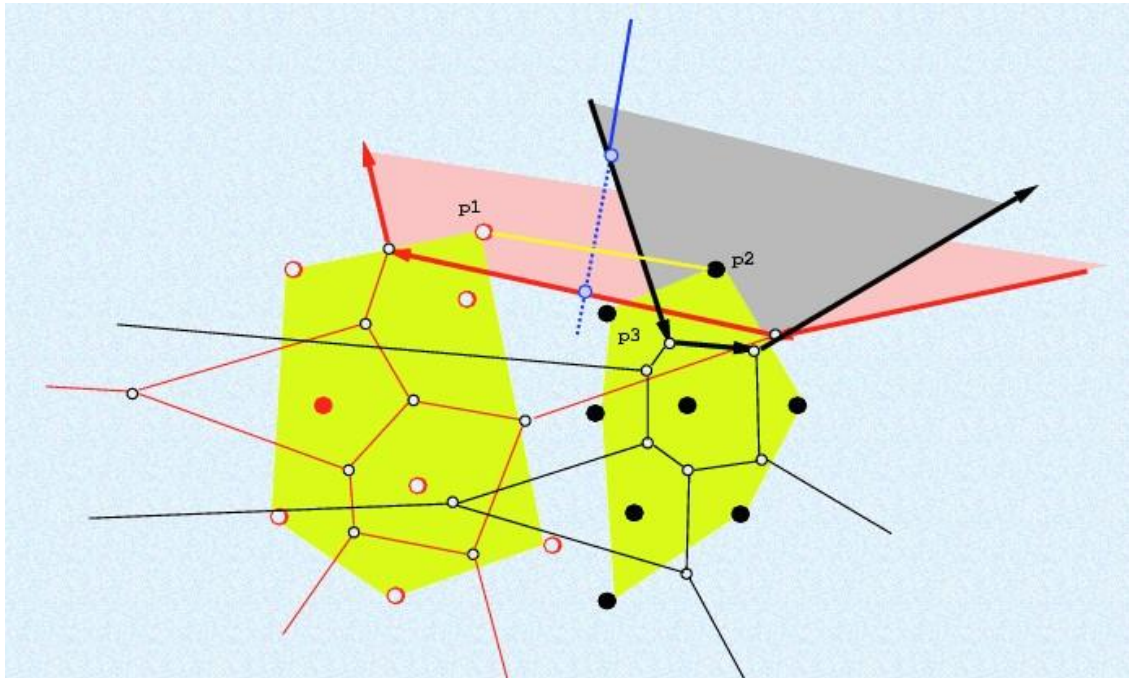
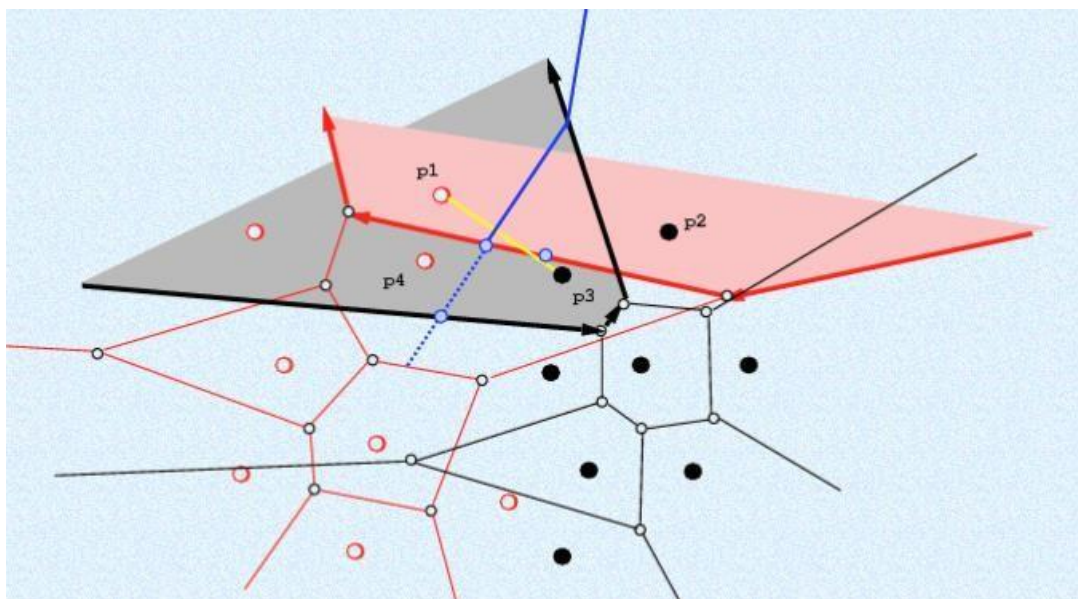


Рисунок 3.4 – Побудова діаграми Вороного (крок 2)

Знайшли верхню межу  $[p_1p_2]$ . Синім кольором позначено промінь, що проходить через середину  $[p_1p_2]$  під прямим кутом. Знайшли перетини з комірками Вороного лівої та правої множини. Синій суцільний промінь – те, що розміщуємо в  $S$ .





зрозуміло, що спочатку промінь буде в неї входити, а потім, в якийсь певний момент – вийде. Потрібно відловити ці події, і в залежності від того, з діаграмою якої множини працюємо (лівою або правою), видалити лівий чи правий ланцюг ребер комірки Вороного, яку була перетнута променем, і додати туди потрібну частину з множини  $S$ .

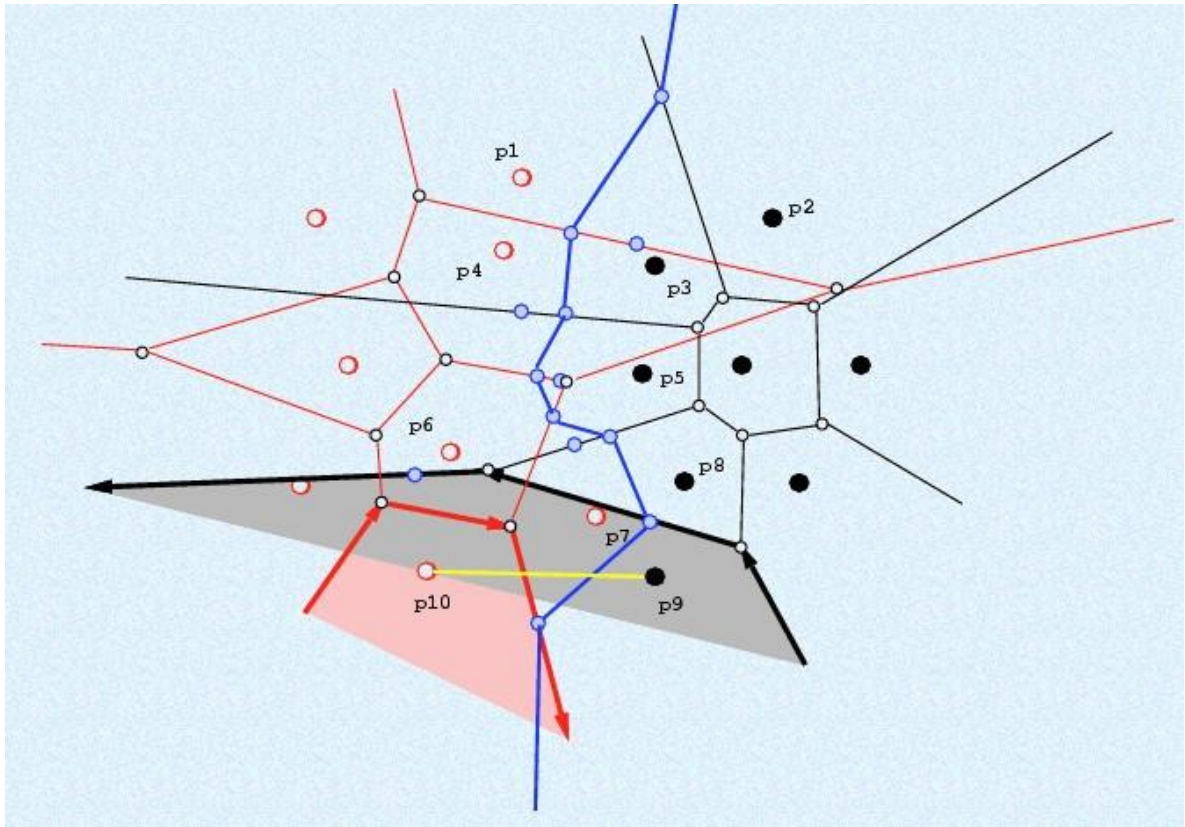


Рисунок 3.7 – Побудова діаграми Вороного (крок 5)

На рисунку 3.7 показаний фінальний результат. Кінці відрізка співпадають з нижньою межею, що значить треба завершити побудову межі між двома діаграми Вороного [12].

Важливо відзначити, що якщо точки мають однакову координату  $X$ , то варто їх сортувати по координаті  $Y$ , таким чином, щоб рівномірно і послідовно їх розділити [5].

### 3.4 Можливості застосування діаграм Вороного

В обчислювальній геометрії діаграма Вороного потрібна перш за все для вирішення завдання близькості точок, а точніше, особливий вигравш діаграма дає в рішенні завдання «усі найближчі сусіди», тому як аналогічні їй засоби не такі прості.

Широке застосування діаграми Вороного знайшли у розробці дизайну ігор, де система навігації в ігровому двигуні заснована на діаграмі.

Також різний геолокаційні софт використовує діаграми Вороного. Геолокаційні рекомендаційні системи можуть використовувати діаграму для визначення, наприклад, найближчого до вас продуктового магазину, для різного пошуку та аналізу місця розташування (рисунок 3.8) [13].



Рисунок 3.8 – Приклад використання діаграми Вороного у геолокації

Тут же можна згадати і застосування діаграми в картографії - для окреслення меж регіонів і подальшого аналізу на їх основі (рисунок 3.9). Та й взагалі, будь-які географічні діаграми, що показують розподіл чого-небудь,

можна наочно проілюструвати за допомогою розфарбованих діаграм Вороного, і там буде видно перехід потрібного нам показника (наприклад, температури).



Рисунок 3.9 – Приклад використання діаграми Вороного у картографії

В археології багатокутники Вороного використовуються для нанесення на карту ареалу застосування знаряддя праці в давніх культурах і для вивчення впливу конкуруючих центрів торгівлі.

В екології – можливості організму на виживання залежать від числа сусідів, з якими він повинен боротися за їжу і світло.

3D-сканування різних об'єктів теж може використовувати діаграму Вороного і триангуляцію Делоне, також це тісно пов'язано з робототехнікою – рух робота з урахуванням перешкод на шляху.

#### 4 РЕАЛІЗАЦІЯ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ АНАЛІЗУ СЕНСОРІВ

Розглянемо реалізацію систем підтримки прийняття рішень на базі нечіткої логіки з використанням системи MATLAB.

#### 4.1 Основні теоретичні відомості

Основою систем нечіткої логіки є математична теорія нечітких множин, яка бере свій початок зі статті американського вченого Лотфі Заде, опублікованої під назвою "Fuzzy Sets" (нечіткі множини) в 1965 році в журналі *Information and Control*. На даний час ця теорія отримала досить глибокий розвиток, а системи на базі нечіткої логіки знайшли широке застосування в промисловості від виробництва фотоапаратів, пральних машин, мікрохвильових печей до управління великими промисловими виробництвами і систем підтримки прийняття рішення.

Ідея, що лежить в основі теорії нечітких множин, полягає в тому, що людина в своєму повсякденному житті мислить і приймає рішення на основі нечітких понять. Створення теорії нечітких множин - це спроба формалізувати людський спосіб міркувань. Розвиток обчислювальної техніки дозволяє на даний час створювати на базі теорії нечітких множин системи нечіткої логіки, які копіюють спосіб міркувань людини.

У більшості вирішуваних людиною завдань не потрібна висока точність. Навпаки, часто доводиться знаходити розумний компроміс між поняттями "точність" і "важливість" при спілкуванні з реальним світом. Так, для прийняття рішення про перехід вулиці людина не оцінює швидкість автомобіля, що наближається з точністю до десятих часток метрів в секунду. Він визначає для себе швидкість автомобіля як "дуже швидко", "швидко", "повільну" і т.д., тобто використовує для позначення швидкості лінгвістичні змінні [15].

##### 4.1.1 Визначення лінгвістичної змінної

Лінгвістичною змінною називається п'ятірка  $(x, T(x), X, G, M)$ , де  $x$  - ім'я змінної;  $T(x)$  - множина імен лінгвістичних значень змінної  $x$ , кожне з яких є

нечіткою множиною на множині  $X$ ;  $G$  є синтаксичне правило для утворення імен значень  $x$ ;  $M$  є семантичне правило для асоціювання кожної величини значення з її поняттям.

Це визначення може викликати відчуття, що лінгвістична змінна - дуже складне поняття, але насправді це не так. Мета концепції лінгвістичної змінної полягає в тому, щоб формальним чином сказати, що змінна може приймати в якості значень слова з природної мови. Наприклад, якщо ми говоримо "швидка швидкість", то змінну "швидкість" слід розуміти як лінгвістичну змінну, але це не означає, що змінна "швидкість" не може приймати реальні значення [16].

#### 4.1.2 Визначення лінгвістичної змінної (інтуїтивне)

Якщо змінна може приймати значення слів в природній мові (наприклад, "маленький", "швидкий" і т.п.), то ця змінна визначається як лінгвістична змінна. Слова, значення яких бере лінгвістична змінна, зазвичай позначають собою нечіткі множини. Лінгвістична змінна може приймати своїми значеннями або слова, або числа.

Нечітка множина - це така множина, яка утворюється шляхом введення узагальненого поняття належності, тобто розширення двоелементною множиною значень функції належності  $\{0,1\}$  до відрізка  $[0,1]$ . Це означає, що перехід від повної належності об'єкта множини до його повної неналежності відбувається не стрибком, як в звичайних "чітких" множинах, а плавно, поступово, причому ступінь належності елемента множини виражається числом з інтервалу  $[0,1]$ .

Таким чином, нечітка множина  $A = \{(x, A(x))\}$  визначається математично як сукупність упорядкованих пар, складених з елементів  $x$  множини  $X$  і відповідних їм ступенів належності  $A(x)$  або безпосередньо у вигляді функції  $A: X \rightarrow [0, 1]$ . Розглянемо приклад нечіткої множини [7].

Розглянемо приклад з множиною високих людей.

Нехай  $x$  є лінгвістична змінна, що позначає "зріст людини", а її функція належності до множини високих людей  $\mu_A: X \rightarrow \{0, 1\}$ , де  $X$  - множина, що включає в себе всі можливі значення зросту людини, задана в такий спосіб:

$$\mu_A(x) = \begin{cases} 1, \text{ якщо } x \geq 180, \\ 0, \text{ якщо } x < 180. \end{cases}$$

$$A = \{x \mid \mu_{A(x)} = 1\}, x \in X$$

Множина "високих людей" задається виразом

Графічно це представлено на рисунку 4.1 (суцільна лінія).

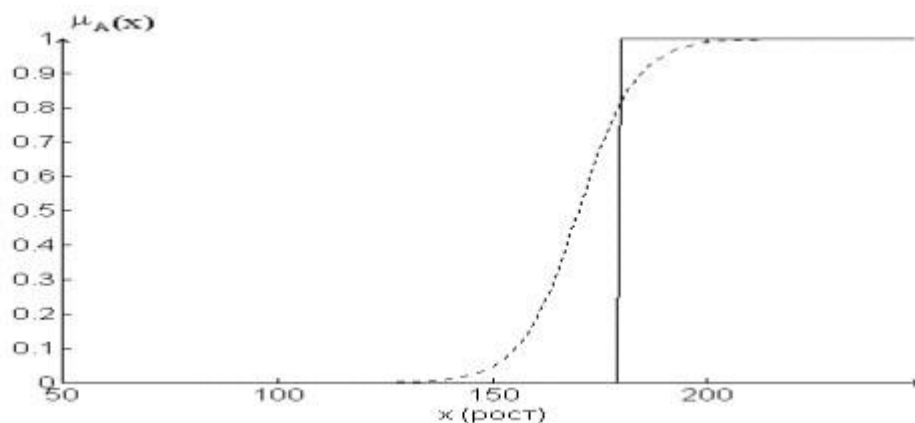


Рисунок 4.1 – Графічне відображення множини "високих людей"

Однак у реальному житті таке поняття, як "зріст найвищої людини", є суб'єктивним, тобто залежить від індивідуума, що робить оцінку. Одні вважають, що високий чоловік повинен бути ростом понад 170 см, інші - більш 180 см, треті - більше 190 см. Нечіткі множини дозволяють врахувати

таку розмитість оцінок. Так, нехай функція належності  $\mu_A: X \rightarrow \{0, 1\}$  має

вигляд, представлений на рисунку 1 пунктирною лінією. Тоді множина  $A$

$$A = \{x, \mu_A(x)\}, x \in X$$

("високих людей") задається множиною пар  $(x, \mu_A(x))$  і є нечіткою множиною.

Таким чином, людина ростом 145 см буде належати множині  $A$  зі ступенем

$$\mu_A(145) = 0 \text{ належності, } \mu_A(165) = 0,3$$

людина

$$\mu_A(185) = 0,9 \text{ зростом } 165 \text{ см - , } \mu_A(205) = 1.$$

см - ,

зростом 185 см -

, а зростом 205 см -

Можна сказати, що в нечіткій логіці правдивість кожного твердження розглядається в рамках ступеню правдивості, що може також розглядатися як ступінь впевненості експерта, що робить оцінку, в тому, що елемент  $x$  належить множині  $A$ .

З вищесказаного можна зробити наступні висновки:

- нечіткі множини описують невизначені поняття (швидкий бігун, гаряча вода, спекотна погода);
- нечіткі множини допускають можливість часткової належності до них (п'ятниця – частково вихідний день (укорочений), погода швидше спекотна);
- ступінь належності об'єкта до нечіткої множини визначається відповідним значенням функції належності на інтервалі  $[0,1]$  (п'ятниця належить до вихідних днів зі ступенем належності 0,3);
- функція належності ставить у відповідність об'єкту (або логічній змінній) значення ступеню його належності до нечіткої множини.

### 4.1.3 Логічні операції з нечіткими множинами

Однією з важливих речей для розуміння механізму нечіткої логіки є те, що вона може розглядатися як розширення стандартної, добре відомої булевої логіки (подібно до того, як нечіткі множини є розширенням звичайних множин). Іншими словами, якщо розглядати нечіткі значення в їх екстремумах, тобто в 1 (абсолютна правда) і в 0 (абсолютна неправда), то можна застосовувати стандартні логічні операції: І, АБО, НЕ. У нечіткій логіці істинність кожного твердження розглядається в певній мірі. У нечіткої логіки за аналогією з операціями І та АБО вводяться операції  $\min(A, B)$  і  $\max(A, B)$ , а операції НЕ відповідає операція  $1-A$  [17]. Графічно це представлено на рисунках 4.2 – 4.4.

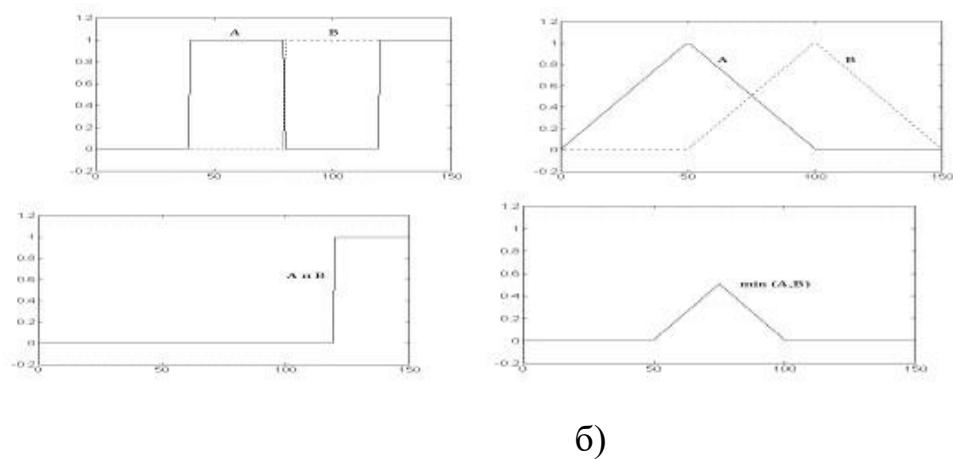


Рисунок 4.2 – Операція  $\min(A, B)$ : а) – "чітка" логіка; б) – "нечітка" логіка

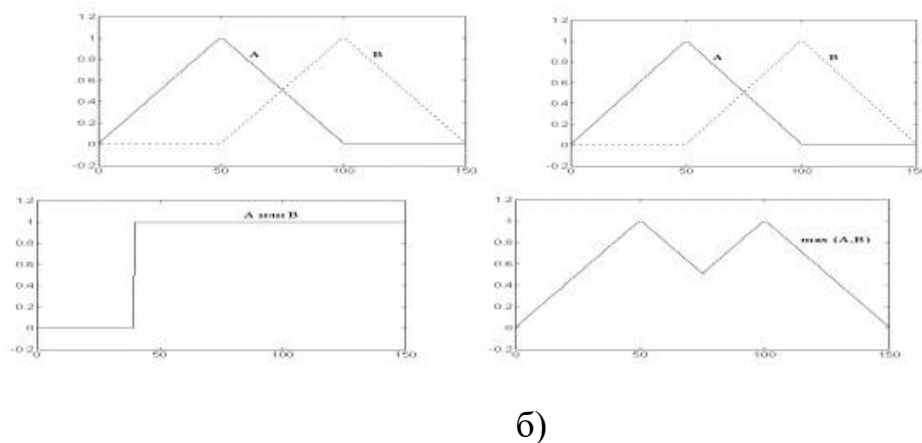


Рисунок 4.3 - Операція  $\max(A,B)$ : а) – "чітка" логіка; б) – "нечітка" логіка

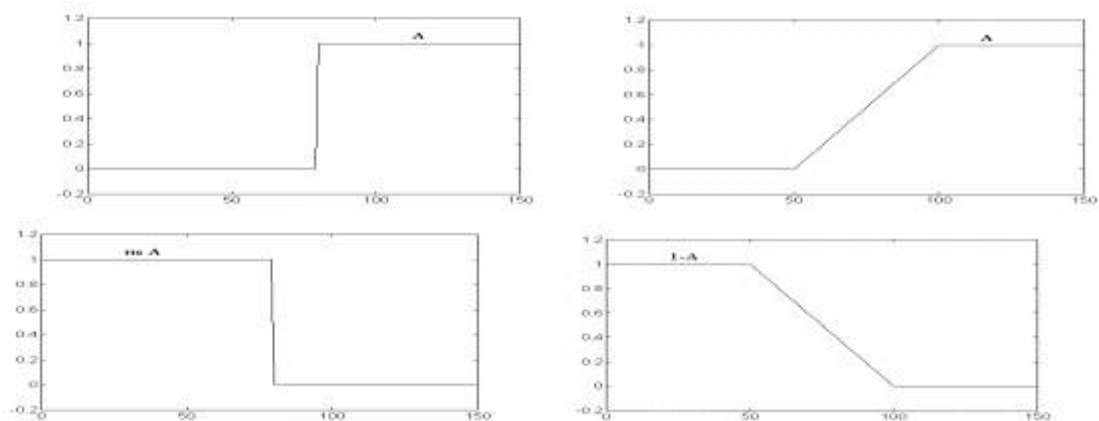


Рисунок 4.4 – Операція  $1-A$ : а) – "чітка" логіка; б) – "нечітка" логіка

## 4.2 Системи нечіткої логіки

Системами нечіткої логіки називаються системи, які оперують нечіткими поняттями, такими як нечіткі множини, лінгвістичні змінні і т.п., і використовують при цьому нечітку логіку. Системи нечіткої логіки можуть бути класифіковані за трьома основними типами:

- прості системи нечіткої логіки (pure Fuzzy Logic Systems);
- нечіткі системи Такагі та Сугено (Takagi and Sugeno);
- системи нечіткої логіки з фаззифікатором і дефаззифікатором.

### 4.2.1 Прості системи нечіткої логіки

Базова конфігурація простий системи нечіткої логіки представлена на рисунку 4.5.

Базис нечітких правил містить набір нечітких IF-THEN (ЯКЩО-ТО) правил, а механізм нечіткого виводу на основі принципів нечіткої логіки використовує ці IF-THEN-правила для відображення нечітких множин з

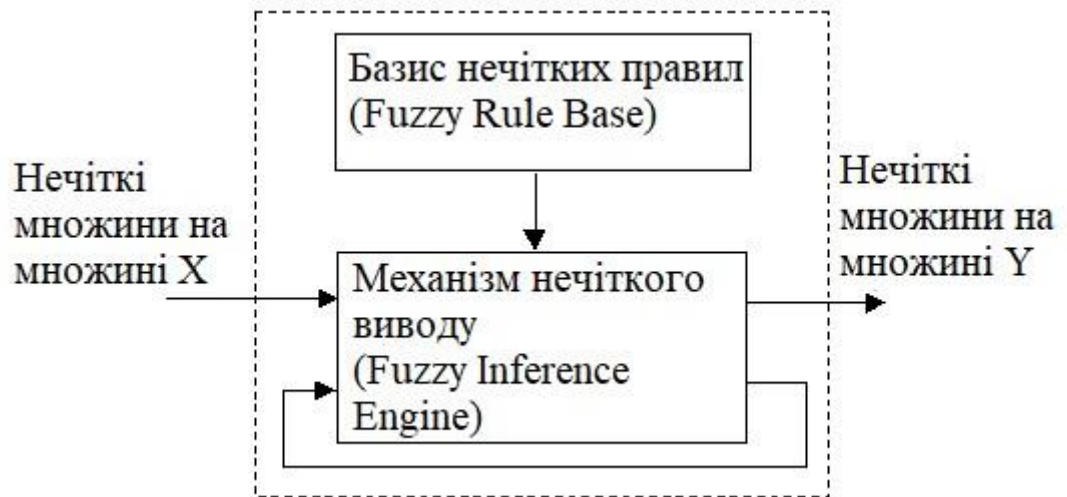
вхідної множини висловлювань  $X$  в нечіткі множини з множини висловлювань  $Y$  на виході системи.

Нечіткі IF-THEN-правила виглядають наступним чином:

$$R^{(p)}: IF x_1 \in F_{1^p} \text{ і } \dots \text{ і } x_n \in F_{n^p} THEN y \in G^p \quad (1)$$

$F_{1^p}$  та  $G^p$  – нечіткі множини;  $\bar{x} = (x_1, \dots, x_n)^T \in X$  та  $x \in Y$

вхідною і вихідною лінгвістичними змінними відповідно;  $p = \overline{1, m}$ .



де

є

## Рисунок 4.5 – Проста система нечіткої логіки

Практика показує, що ці нечіткі IF-THEN-правила забезпечують зручний механізм для представлення знань людини-експерта. Кожне нечітке

$$F_1^p \times \dots \times F_n^p \rightarrow G_p.$$

IF-THEN-правило визначає нечітку множину

Найбільш загальним принципом нечіткої логіки, що використовується

$A'$

в механізмі нечіткого виводу, є композиція. Нехай є довільна нечітка

$A'$

множина з  $X$  і нехай є входом простої системи нечіткої логіки,

представленої на рисунку 4.5. Тоді вихід, який визначається кожним

$$A' \circ R^{(p)}$$

нечітким IF-THEN-правилом, є нечітка множина в  $Y$  з функцією належності

$$\mu_{A' \circ R^{(p)}}(y) = \sup_{\bar{x} \in X} (\mu_{A'}(\bar{x}) \otimes \mu_{F_1^p \times \dots \times F_n^p \rightarrow G^p}(\bar{x}, y)) \quad \otimes$$

, де оператор « $\otimes$ » є

одна з T-норм.

Остаточний вихід простої системи нечіткої логіки є нечітка множина

$$A' \circ (R^{(p)}, \dots, R^{(M)})$$

в  $Y$ , яке є комбінацією  $M$  нечітких множин, так що

$$A' \circ (R^{(1)}, \dots, R^{(M)})(y) = \mu_{A' \circ (R^{(1)})}(y) \otimes \dots \otimes \mu_{A' \circ (R^{(M)})}(y)$$

$\otimes$

, де оператор « $\otimes$ » є одна із T-

конорм [6].

Якщо присутній зворотний зв'язок, як показано на рисунку 4.5, то ми маємо так звану нечітку динамічну систему, тобто просту систему нечіткої логіки, у якій входи залежать від її виходів.

Структура простої системи нечіткої логіки є суттєвою частиною систем нечіткої логіки. У загальному випадку, коли лінгвістична інформація від

експерта перекладається в кількісні показники, принципи нечіткої логіки використовуються для систематизації інформації. Головним недоліком простих систем нечіткої логіки є та обставина, що її входи і виходи - нечіткі множини, тоді як в більшості технічних систем входи і виходи є змінними, які приймають реальні значення [18].

#### 4.2.2 Системи нечіткої логіки з фаззифікатором і дефаззифікатором

Для того, щоб використовувати розглянуті вище прості системи нечіткої логіки в технічних додатках, коли входи і виходи систем приймають реальні значення, найбільш простим шляхом є додавання фаззифікатора до входу і дефаззифікатора до виходу простої системи нечіткої логіки.

Базова конфігурація системи нечіткої логіки з фаззифікатором і дефаззифікатором представлена на рисунку 4.6.

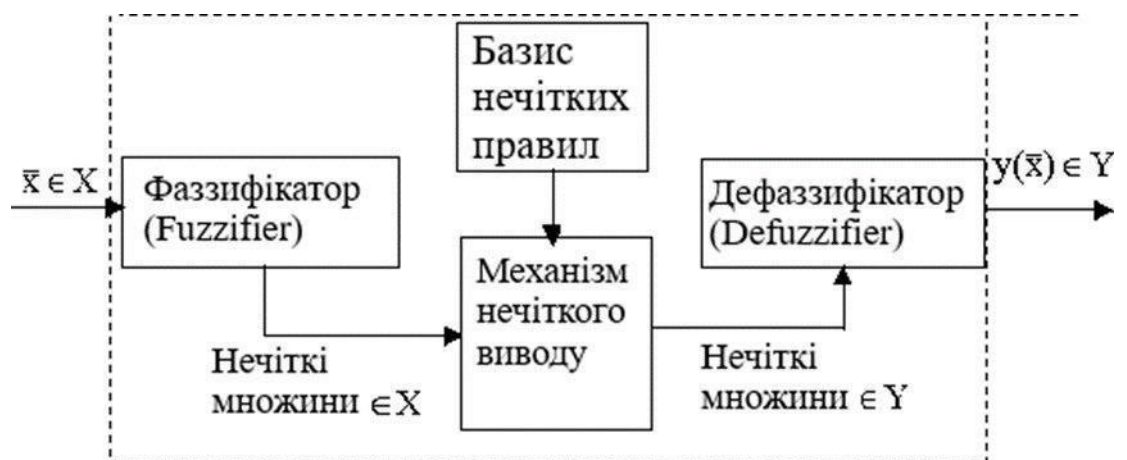


Рисунок 4.6 – Система нечіткої логіки з фаззифікатором і дефаззифікатором

Фаззифікатор відображає "чітку" точку (або значення змінної) з  $X$  в нечіткі множини з  $X$ . Дефаззифікатор відображає нечіткі множини з  $Y$  в чітке значення виходу. Базис нечітких правил і механізм нечіткого виведення залишаються такими ж, як у простих системах нечіткої логіки.

Системи нечіткої логіки з фаззифікатором і дефаззифікатором мають кілька привабливих рис:

- по-перше, вони придатні для використання в технічних системах, так як їх вхідні і вихідні змінні приймають реальні значення;
- по-друге, вони надають можливість природного переходу від висновків експерта до нечітким IF-THEN-правилам;
- по-третє, вони надають велику свободу у виборі фаззифікатора, механізму нечіткого виведення і дефаззифікатора, тобто можна підібрати систему нечіткої логіки, найбільш підходящу для вирішення конкретного завдання;
- по-четверте, можуть бути розроблені різні алгоритми настройки таких систем нечіткої логіки, що дозволяє ефективно поєднувати чисельну і лінгвістичну інформацію.

#### 4.2.3 Базис нечітких правил

Базис нечітких правил складається з набору нечітких IF-THEN-правил виду. Без втрати спільності можна розглядати багатовходову систему нечіткої логіки з одним виходом (MISO-систему), тому що система з багатьма виходами (MIMO - система) може бути представлена як композиція групи систем, що мають один вихід.

Базис нечітких правил є основною частиною системи нечіткої логіки в тому сенсі, що інші її компоненти використовуються для інтерпретації цих правил і їх застосування при вирішенні конкретних завдань. Практика показує, що нечіткі IF-THEN-правила надають зручний механізм людині-експерту для вираження своїх знань [19].

#### 4.2.4 Приклад системи нечіткої логіки

Розглянемо використання систем нечіткої логіки для реалізації функцій прийняття рішення на міжцеховому рівні машинобудівного підприємства на прикладі функції "прийняття оптимального плану".

Завдання прийняття оптимального плану полягає у виборі номенклатури виробів таким чином, щоб цільова функція, де -  $i$ -й виріб; - планований прибуток від реалізації  $i$ -го виробу, брала максимальне значення. На цю формулу накладаються обмеження, пов'язані з виробничими потужностями підприємства.

Дана формула описує ідеальну ситуацію, коли всі замовники тієї чи іншої продукції мають стовідсоткової платоспроможністю. Реальне життя вносить свої корективи, і формула для обчислення цільової функції набуває такого вигляду: , де - коефіцієнт платоспроможності замовника. Зрозуміло, що платоспроможність є цілком нечітким поняттям і визначення коефіцієнта платоспроможності підприємства є складним завданням прийняття рішення. У сучасних умовах економіки платоспроможність підприємства визначається не тільки наявністю грошових коштів на банківському рахунку або можливістю їх своєчасного залучення з інших джерел, а й наявністю виробів або матеріалів, здатних зацікавити підприємство-постачальника як так званого "бартеру".

У зв'язку з цим функція прийняття рішення за завданням коефіцієнта платоспроможності була реалізована представленої на рисунку 4.7 системою нечіткої логіки з двома входами, одним виходом і трьома правилами. Входами системи є експертні бальні оцінки наявності грошей на рахунку замовника та наявності цікавить підприємство бартеру. Мінімальна кількість балів (0) відповідає абсолютної впевненості дає оцінку експерта в тому, що замовник не має на рахунку грошових коштів і не зможе їх вчасно залучити для оплати поставляються йому виробів, а наявний в його розпорядженні бартер абсолютно не потрібен підприємству. Максимальна кількість балів (10)

відповідає зворотному стану речей. Оцінки про наявність грошових коштів і корисності бартеру виносяться різними фахівцями. Оцінка наявності грошових коштів на рахунку замовника проводиться фахівцями відділу власної безпеки підприємства і здійснюється за непрямими ознаками. Корисність бартеру оцінюють фахівці з відділу постачання.

Виходом системи нечіткої логіки є значення коефіцієнта платоспроможності, який приймає значення від 0 до 1.



Рисунок 4.7 – Приклад системи нечіткої логіки

Система працює наступним чином. Інформаційний потік рухається зліва направо, від двох входів до одного виходу. Паралельна природа виконання правил є одним з важливих аспектів систем нечіткої логіки. Замість різкого перемикавання між моделями (правилами) в точках переходу виходить гладкий перехід від області, в якій поведінка системи описується одним правилом, до області, де воно описується іншим правилом [7].

Для входів системи визначені відповідні їм нечіткі множини і задані визначають ці множини функції належності. В даному випадку входи задаються на множині чисел (балів) від 0 до 10. Експертом, що брав участь в побудові системи нечіткої логіки, були задані три правила, які використовують нечіткі множини, і визначено функції їх належності. Для наявності грошей на

рахунку замовника (змінна "гроші") - це нечіткі множини "маленькі", "середні" і "великі", для змінної "бартер" - "поганий" і "хороший", для змінної "платоспроможність" - "низька", "середня" і "висока".

Перш ніж застосувати той чи інший метод імплікації, необхідно враховувати вагу правила. Кожне правило може мати свою вагу (число від 0 до 1) в залежності від того, наскільки це правило є важливим. Вага правила накладається на вихід ІF-частини. В даному випадку все правила прийняті рівноцінними, і вага кожного правила дорівнює 1.

Метод імплікації визначається як формування виведення з правила на основі заданих умов. Входом процесу імплікації є число, отримане з умов (ступінь істинності умови), а виходом - нечітка множина. Для даної системи нечіткої логіки був обраний найпростіший і добре себе зарекомендував метод імплікації по мінімуму.

У разі, коли кілька правил працюють паралельно, необхідно об'єднати (агрегатувати) нечіткі множини, що представляють виходи окремих правил, в одну нечітку множину для підготовки до заключного кроку - дефазифікації. Входом процесу агрегування є набір нечітких множин, отриманих з кожного правила шляхом імплікації. Його виходом є одна нечітка множина для кожної вихідної змінної. В даному випадку для агрегування нечітких множин використовується операція об'єднання [19].

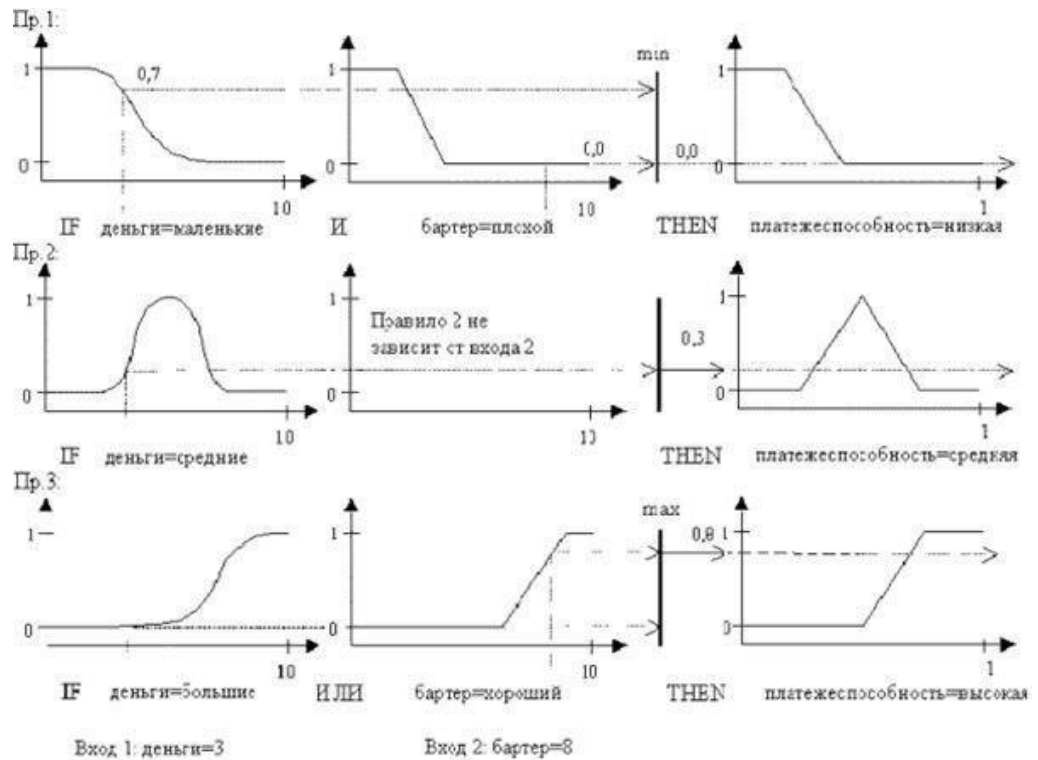


Рисунок 4.8 – Загальна схема пропонованої системи нечіткої логіки

Зауважимо, що так як методи агрегування є комутативними (що мають виконуватися завжди), то порядок, в якому стоять правила, не має значення. Для ілюстрації роботи системи на рисунку 4.8 задані входи "гроші = 3" і "бартер = 8". Виходом операції агрегування для заданих входів є нечітка множина з функцією належності, представленої на рисунку 4.9.

Дефазифікація вихідної нечіткої множини методом центру ваги дає для наведеного прикладу вихідне значення коефіцієнта платоспроможності 0,8. Зауважимо, що всі нечіткі множини задані на відповідних множинах. Для грошей і бартеру – на множині балів від 1 до 10, для коефіцієнта платоспроможності – на безперервній множині від 0 до 1.

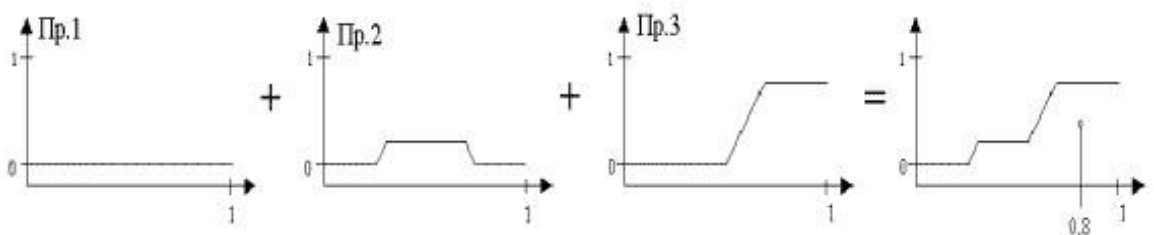


Рисунок 4.9 – Дефазифікація вихідної нечіткої множини методом центру ваги

### 4.3 Реалізація системи підтримки прийняття рішень

Задамо зв'язки між змінними та параметри системи у вікні редактора FIS Editor, що графічно представлено на рисунку 4.10.

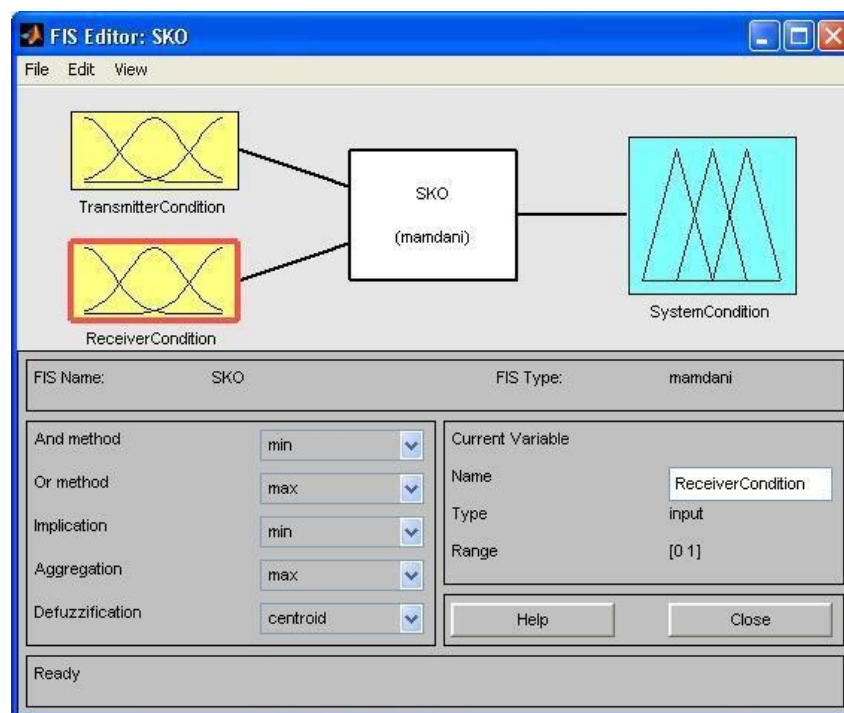


Рисунок 4.10 – Параметри системи у вікні редактора FIS Editor

Задамо функцію належності змінної TransmitterCondition. Для лінгвістичної оцінки цієї змінної використано 5 термів з трикутними функціями належності. В результаті був отриманий графічний результат, зображений на рисунку 4.11.

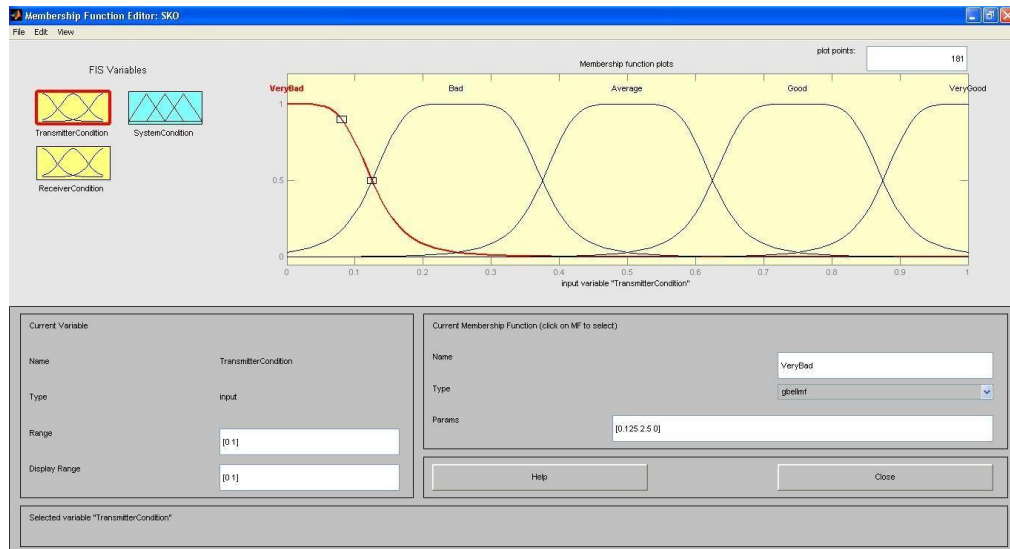


Рисунок 4.11 - Функції належності змінної TransmitterCondition

Задамо функцію належності змінної ReceiverCondition. Для лінгвістичної оцінки цієї змінної використано 5 термів з дзвоноподібними функціями належності. В результаті був отриманий графічний результат, зображений на рисунку 4.12.

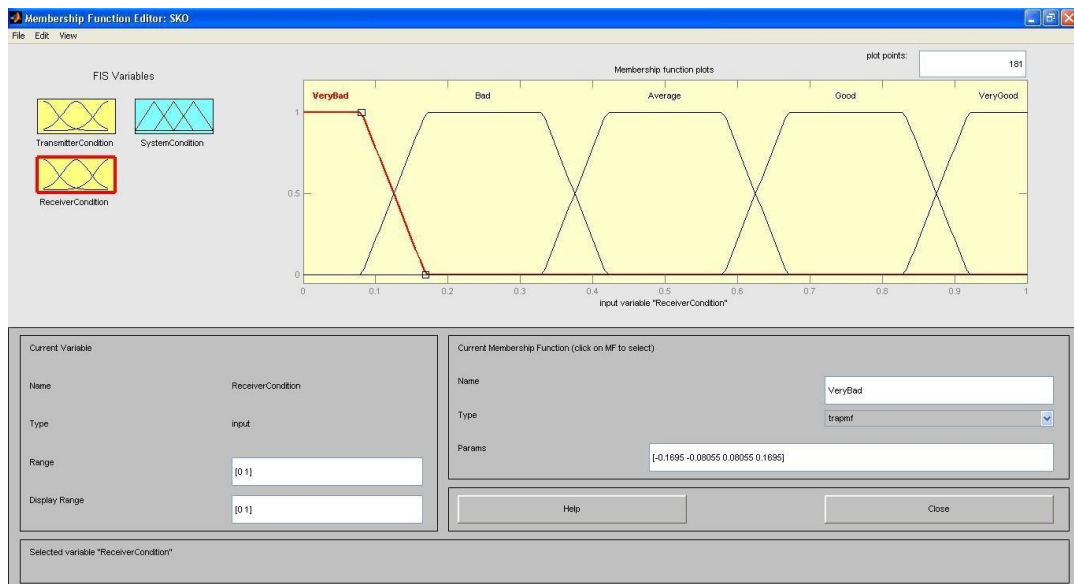


Рисунок 4.12 – Функції належності змінної ReceiverCondition

Була задана функція належності змінної SystemCondition. Для

лінгвістичної оцінки цієї змінної використано 4 терма з трапецієподібними функціями належності. В результаті було отримано графічне вікно, зображене на рисунку 4.13.

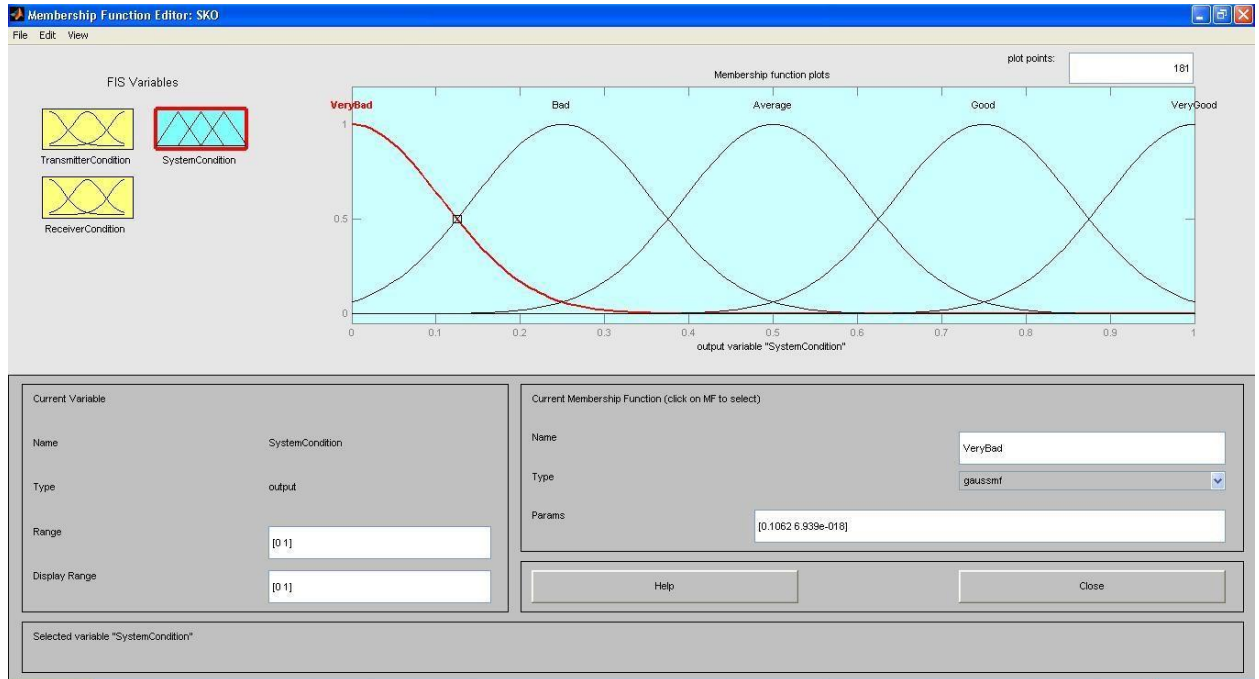


Рисунок 4.13 – Функції належності змінної SystemCondition

Введемо в систему правила логічного висновку. Правила логічного виводу представлені в таблиці 4.1.

Таблиця 4.1 - Таблиця правил логічного виводу

TransmitterCondition						
		Very Bad	Bad	Average	Good	Very Good
ReceiverCondition	Very Bad	Very Bad	Very Bad	Bad	Bad	Average
	Bad	Very Bad	Bad	Bad	Average	Average
	Average	Bad	Bad	Average	Average	Good
	Good	Bad	Average	Average	Good	Good
	Very Good	Average	Average	Good	Good	Very Good

На рисунку 4.14 зображено вікно редактора бази знань після введення всіх 25 правил.

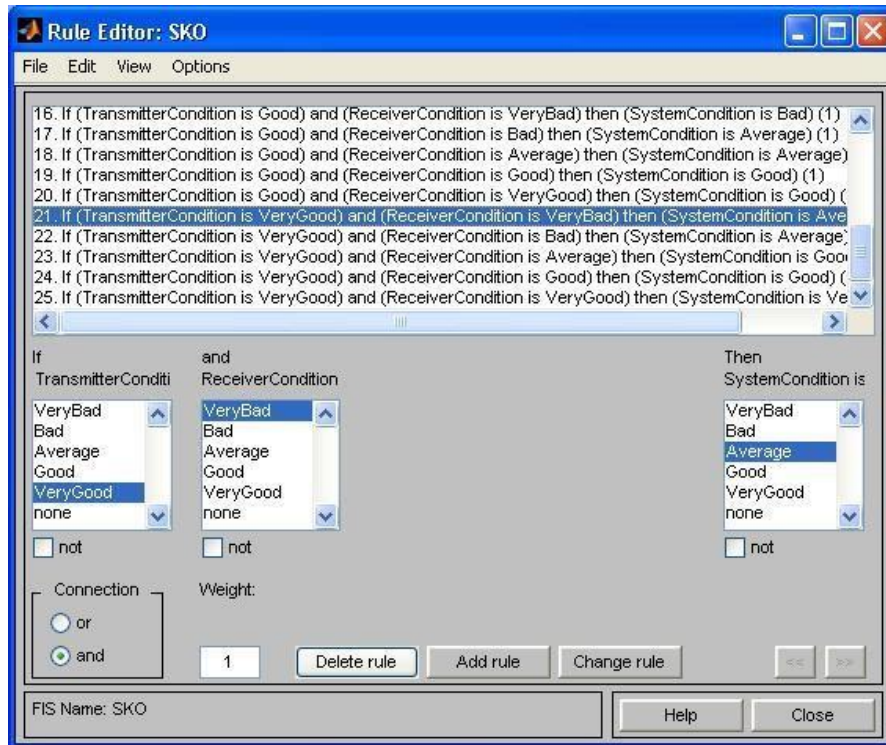


Рисунок 4.14 – База знань в RuleEditor

На рисунку 4.15 приведено вікно візуалізації нечіткого логічного виводу. Були вказані значення входних змінних, які відповідають станам приймача і передавача.

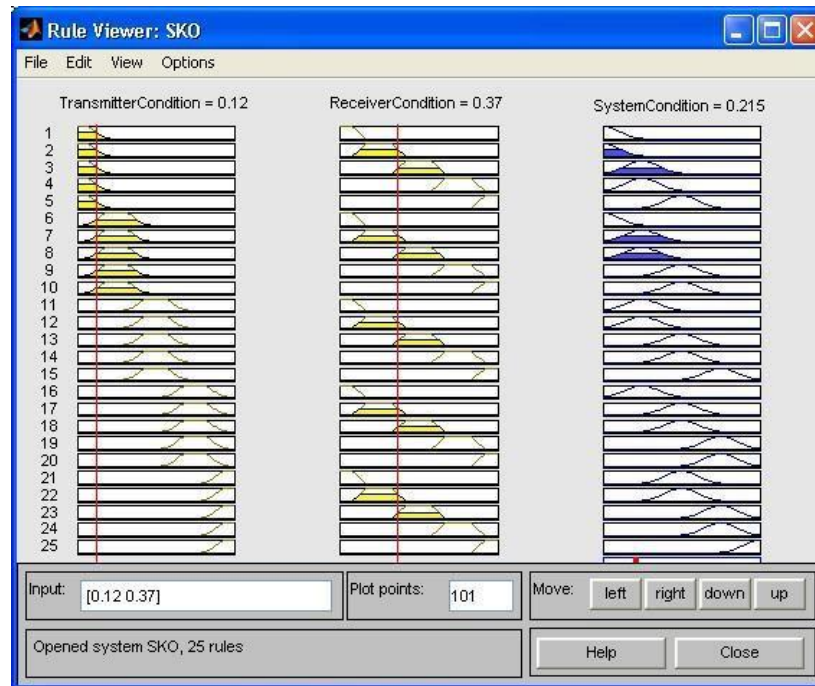


Рисунок 4.15 - Візуалізація нечіткого логічного виводу в RuleViewer

Таблиця 4.2 - Агрегування підумов правил

TransmitterCondition	ReceiverCondition	SystemCondition
0	0	0.0825 – Very Bad
0.25	0.25	0.253 – Bad
0.5	0.5	0.492 – Average
0.75	0.75	0.735 – Good
1	1	0.899 – Very Good
0.12	0.37	0.215 – Bad
0.12	0.64	0.368 – Bad
0.34	0.63	0.39 – Bad
0.37	0.11	0.368 – Bad
0.81	0.62	0.611 – Average
0.80	0.81	0.742 – Good

На рисунку 4.16 приведена поверхня "входи-вихід", відповідна синтезованій нечіткій системі.

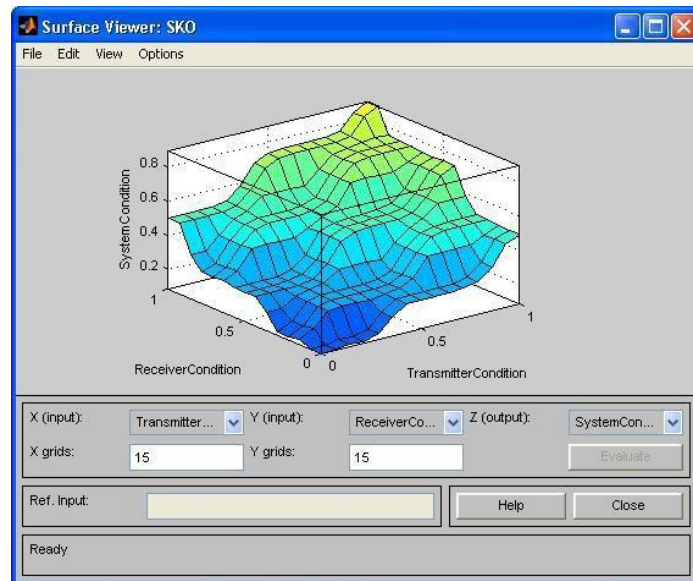


Рисунок 4.16 - Поверхня "входи-вихід" у вікні SurfaceViewer

В ході даного експерименту відповідно до вхідних даних були задані функції належності для вхідних змінних `TransmitterCondition` та `ReceiverCondition` і для вихідної змінної `SystemCondition`. Була складена таблиця правил логічного висновку, на основі даної таблиці була сформульована сукупність правил, які були введені в систему. Для вхідних значень, визначених відповідно до вхідних даних, було виконано моделювання. В результаті моделювання було отримано значення 0.789, що характеризує високий показник стану системи. Була отримана поверхня "входи-вихід", відповідна синтезованій нечіткій системі.

#### 4.4 Створення нейро-нечіткої мережі в середовищі Matlab

Створимо нейро-нечітку мережу в середовищі Matlab, здійснимо її навчання, а також використання налаштованої мережі для отримання результатів нечіткого виведення.

В середовищі MATLAB виберемо кількість та тип функцій належності.

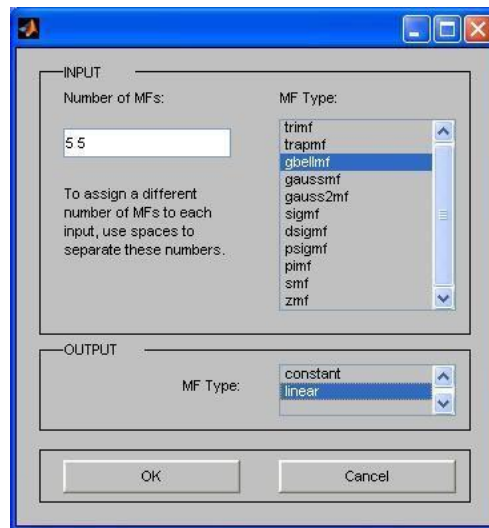


Рисунок 4.17 – Вибір кількості та типу функцій належності

Був підготовлений файл "train.dat" з навчальними даними, який містять 11 рядків. Значення якості приймача і передавача є однаковими для всіх рядків.

Зміст файлу з навчальними даними наведено в таблиці 4.3.

Таблиця 4.3 – Файл "train.dat"

0	0	0.0825
0.12	0.64	0.368
0.34	0.63	0.39
0.37	0.11	0.215
0.81	0.62	0.611
0.80	0.81	0.742
0.25	0.25	0.253
0.5	0.5	0.492
0.75	0.75	0.735
1	1	0.899
0.12	0.36	0.215

Файл з навчальними даними був завантажений в систему. В результаті було отримано результат, зображений на рисунку 4.18.

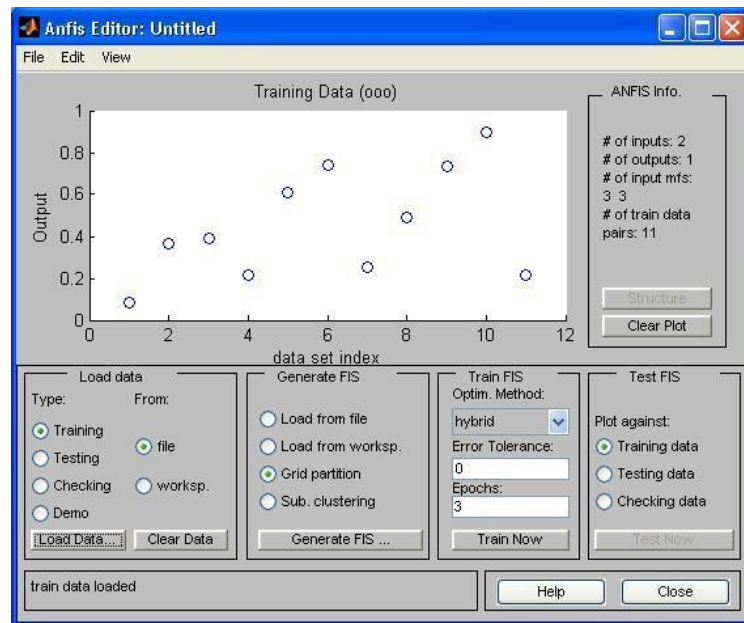


Рисунок 4.19 — Графічний інтерфейс ANFIS після завантаження файлу "train.dat"

ANFIS підтримує тільки системи типу Сугено. Крім того, системи нечіткого виводу повинні мати такі властивості:

- мати один вихід, отриманий з використанням дефазифікації за методом середньозваженого значення;
- бути системою Сугено першого або нульового порядку;
- мати однакову вагу для всіх правил;
- всі вихідні функції належності повинні бути одного типу і бути linear або constant;
- кількість вихідних функцій належності має дорівнювати кількості правил.

Була отримана система Сугено, яка задовольняє цим вимогам. Дана система була завантажена в ANFIS.

Структура нейро-нечіткої мережі представлена на рисунку 4.20.

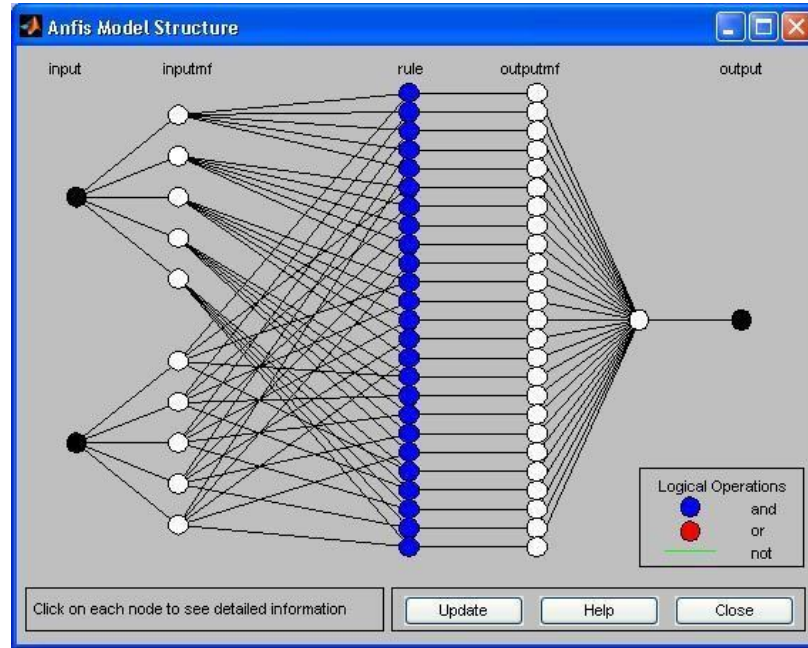


Рисунок 4.20 — Структура нейро-нечіткої мережі

Були виставлені параметри навчання нейронної мережі: метод навчання (гібридний), рівень помилки (0), кількість циклів навчання (30). В результаті навчання мережі було отримано графік зміни помилки в ході виконання циклів навчання, зображений на рисунок 4.21.

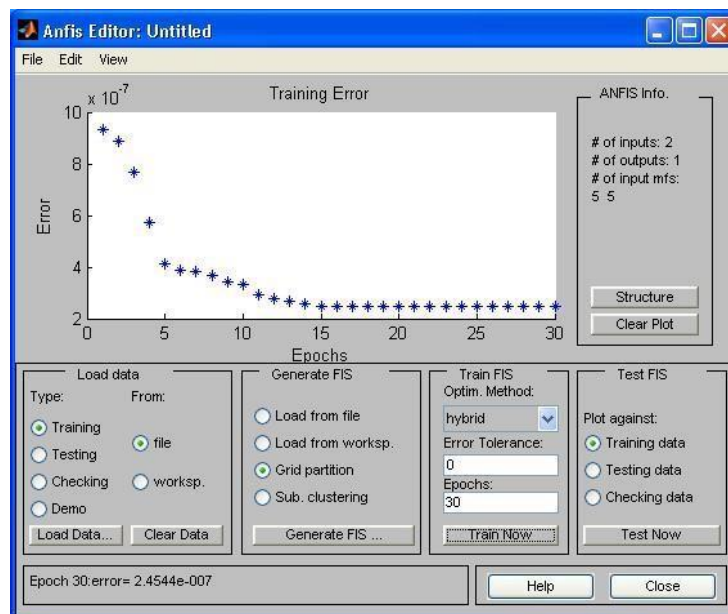


Рисунок 4.21 – Графік залежності помилки навчання від кількості циклів навчання

Можна бачити, що значення помилки навчання навіть на першому циклі навчання незначно відрізняється від 0.

Був підготовлений файл "testing.dat" з перевірочними даними в 11 рядків. Зміст файлу з перевірочними даними наведено в таблиці 4.4.

Таблиця 4.4 – Файл "testing.dat"

0.02	0.02	0.0966
0.76	0.77	0.744
0.96	0.99	0.86
0.1	0.39	0.229
0.14	0.6	0.338
0.33	0.65	0.421
0.35	0.14	0.229
0.22	0.23	0.246
0.48	0.52	0.475
0.79	0.64	0.655
0.79	0.78	0.746

Файл з перевірочними даними був завантажений в систему. В результаті було отримано графічний результат, зображений на рисунку 4.22.

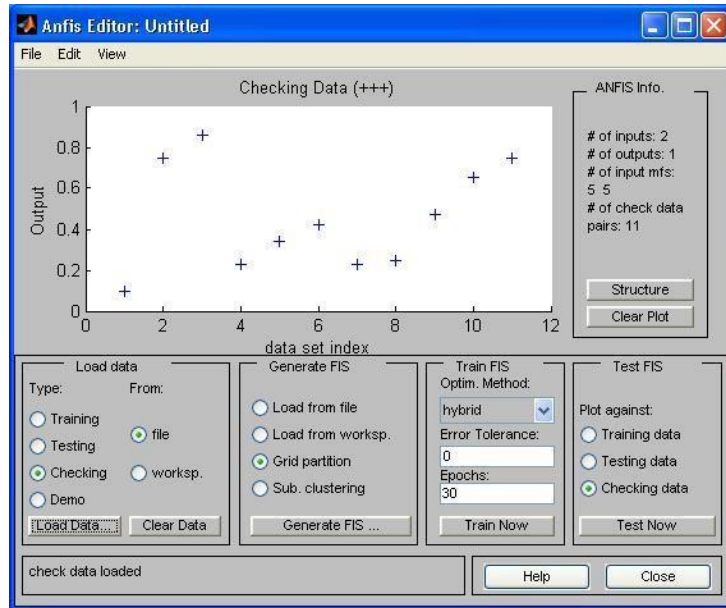


Рисунок 4.22 — Графічний інтерфейс ANFIS після завантаження файлу "testing.dat"

Результати перевірки нейронної мережі на даних з файлу "testing.dat" представлені на рисунку 4.23.

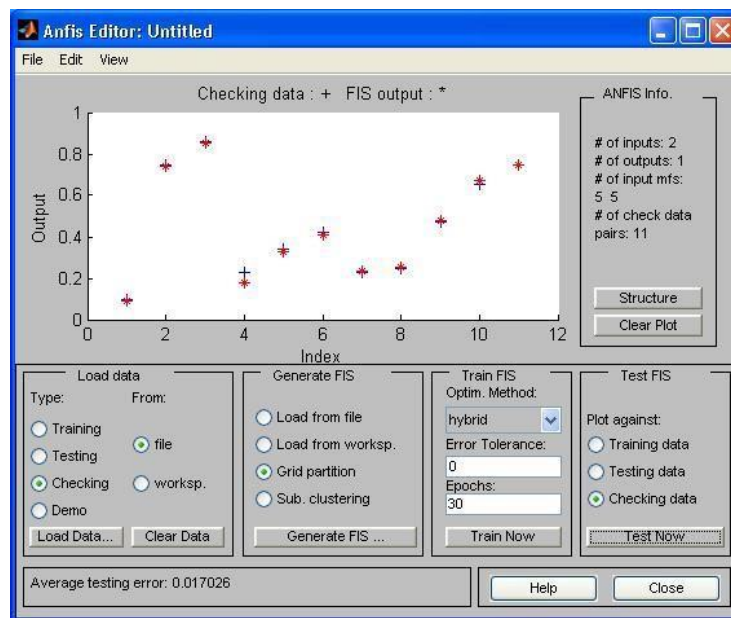


Рисунок 4.23 — Результати перевірки нейронної мережі на даних з файлу "testing.dat"

В ході даного експерименту було складено і завантажено в ANFIS файл з навчальними даними для нейронної мережі. У адаптивну систему нечіткого

виводу була завантажена FIS-система типу Сугено. Було виконано навчання нейронної мережі на навчальних даних. Був складений і завантажений в адаптивну систему нечіткого виводу файл з перевірочними даними. Результати перевірки нейронної мережі показують, що середнє значення помилки при тестуванні дорівнює 0.017. Була отримана структура розробленої нейро-нечіткої мережі.

## ВИСНОВКИ

В даній атестаційній роботі було проведено дослідження існуючих технологій для збору інформації з використанням бездротових сенсорних мереж та інтелектуальних датчиків, а також подальшої її відправки до віддаленої базової станції.

В результаті виконання даної роботи запропоновано оптимальний спосіб моніторингу параметрів навколишнього середовища, який може подолати недоліки, існуючі в сучасних системах бездротових сенсорних мереж. Використання метода Вороного призводить до відчутного збільшення тривалості функціонування і скорочення вартості розгортання подібної мережі для розміщення сенсорних вузлів в межах контрольованої території.

На основі цієї роботи може бути розроблена БСС для виробничого підприємства. Мережа допоможе оптимізувати витрати на електрику, поліпшити робочі умови персоналу і підвищити безпеку майна підприємства. Дана мета буде досягнута завдяки використанню ZigBee модулів. Датчики на їх основі визначають температуру і вологість повітря в приміщенні, освітленість робочих місць, рівень забрудненості повітря і наявність диму, ці дані дозволяють оптимізувати рівень вентиляції, обігріву та освітлення приміщення. Датчики руху і відкриття, закриття дверей і вікон сприяють затриманню зловмисника в разі проникнення на підприємство. Розумні замки обмежують доступ персоналу в різні приміщення. Надалі можлива установка систем управління, на основі ZigBee. З їх допомогою можливо буде автоматично регулювати закриття, відкриття жалюзі на вікнах, зміна рівня освітленості і вентиляції, а так же обігрів приміщення.

## ПЕРЕЛІК ПОСИЛАНЬ

1. C. Chong and S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges", Proceedings of the 6th International Conference on Mobile Computing and Networking (Mobicom), Boston, USA, pp. 56 - 67, August 2000.
2. Вишнеvский В. М. Теоретические основы проектирования компьютерных сетей. – М. : Техносфера, 2003.
3. K. Sohraby, D. Minoli, and T. Znati, Wireless sensor networks: technology, protocols, and applications. John Wiley & Sons, 2007.
4. Пушкарев О. А. ZigBee-модули MaxStream новые возможности / О. А. Пушкарев // Журн. Новости Электроники. – 2007. – Т. 9, №2, С. 27–32.
5. K. Sohraby, D. Minoli, and T. Znati, Wireless sensor networks: technology, protocols, and applications. John Wiley & Sons, 2007.
6. Пушкарев О. А. Передача данных в ZigBee-сети с помощью модулей XBee ZNet 2.5 / Пушкарев О. А. // Журнал Новости Электроники. – 2008. Т.10, №3, С. 27.
7. C. Schurgers. Energy efficient routing in wireless sensor networks, Proceedings of the IEEE Military Communications Conference – USA, 28 - 31 October 2001. – 357 – 361p.
8. M. Ilyas, "The Handbook of Ad Hoc Wireless Networks", CRC Press, Boca Raton, USA, pp. 14 - 2 to 14 - 3, 2003.
9. J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey", IEEE Wireless Communications, vol. 11, no. 6, pp. 6 28, Dec.2004.
10. Вороной, Георгий Феодосьевич [Электронный ресурс] / Материал из Википедии — свободной энциклопедии. – Режим доступа: [www/ URL: https://ru.wikipedia.org/wiki/Вороной,\\_Георгий\\_Феодосьевич](http://www/ URL: https://ru.wikipedia.org/wiki/Вороной,_Георгий_Феодосьевич) – Загл. с экрана.

11. Акимова И. Я. Взвешенные разбиения Вороного / И. Я. Акимова, А. Н. Акимов // Изв. АН СССР. Техн, кибертехника, 1984. – № 2. – С. 102 – 109.
12. Акимова И. Я. Применение диаграммы Вороного в комбинаторных задачах / И. Я. Акимова, А. Н. Акимов // Изв. АН СССР. Техн, кибертехника, 1988. – № 3. – С. 185 – 190.
13. Астраков С. Н. Построение эффективных моделей покрытия при мониторинге протяженных объектов / С. Н. Астраков, А. Е. Ерзин // Вычислительные технологии, 2012. – Т. 17, № 1. – С. 26 – 34.
14. Препарата Ф. О. Вычислительная геометрия. Введение [Текст] / Ф. О. Препарата, М. Шеймос –М. : Мир, 1989. – 59 с., 260 с., 234 с.
15. Alcalá R. A Fast and Scalable Multiobjective Genetic Fuzzy System for Linguistic Fuzzy Modeling in HighDimensional Regression Problems / R. Alcalá, M.J. Gacto, F. Herrera // IEEE Transaction on Fuzzy Systems. – 2011. – Vol. 19, No 4. – P. 666–681.
16. W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks, Proceedings of the 5th International Conference on Mobile Computing and Networking – USA, August 1999. – 174 – 185p.
17. Gorzalczany M. B. A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability / M.B. Gorzalczany, Rudzinski F. // Applied Soft Computing. – 2016. – Vol. 40. – P. 206–220.
18. Горбунов И. В. Методы построения трехкритериальных паретооптимальных нечетких классификаторов / И. В. Горбунов, И.А. Ходашинский // Искусственный интеллект и принятие решений. – 2015. – № 2. – С. 75–87.