

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут телекомунікаційних систем

(повна назва інституту/факультету)

Кафедра телекомунікацій

(повна назва кафедри)

«На правах рукопису»

УДК _____

До захисту допущено

В.о. завідувача кафедри

_____ Явіся В.С.
(підпис) (ініціали, прізвище)

“ _____ ” _____ 2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 172 Телекомунікації та радіотехніка,

(код і назва)

спеціалізація Апаратно-програмні засоби електронних комунікацій

на тему: Розвиток методів маршрутизації в безпроводових сенсорних мережах

Виконав: студент 2 курсу, групи ТЗ-71мп

(шифр групи)

Козик Юрій Леонідович

(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник Лисенко Олександр Іванович

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант _____

(назва розділу)

_____ (науковий ступінь, вчене звання, прізвище, ініціали)

_____ (підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

(підпис)

Київ – 2018 рік

РЕФЕРАТ

Методика порівняльного аналізу методів маршрутизації і вибір найоптимальнішого методу з урахуванням характеристик та особливостей побудови мережі є надзвичайно важливою частиною розробки сенсорної мережі, адже лише змодельовавши і порівнявши основні варіанти вирішення проблем ми можемо вирішити якою буде топологія мережі, алгоритм маршрутизації і т.д.

Метою дипломної роботи є розробка програмного забезпечення для продовження терміну роботи безпроводової сенсорної мережі завдяки вибору оптимального методу маршрутизації для кожної ділянки безпроводової сенсорної мережі на підставі порівняльного аналізу існуючих методів маршрутизації.

Об'єкт досліджень – безпроводові сенсорні мережі.

Предмет досліджень – методи маршрутизації в безпроводових сенсорних мережах.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- дослідження основних особливостей та вимог до БСМ;
- дослідження та класифікація способів маршрутизації в БСМ;
- аналіз основних алгоритмів маршрутизації в БСМ;
- порівняльний аналіз протоколів маршрутизації в БСМ;
- розробка програмного забезпечення

При виконанні досліджень та прийнятті рішень використовувались такі підходи: цілісності, ієрархічності, структуризації, комплексний підхід.

В даній роботі розглянуто основні особливості та вимоги до процесу маршрутизації в безпроводових сенсорних мережах. Проаналізовані параметри маршрутизації в залежності від призначення мережі. Наведена класифікація способів маршрутизації та розглянуто найвідоміші протоколи маршрутизації. Виконано порівняння цих протоколів та вказані переваги

кожного з них. В ході дослідження були визначені основні вимоги до побудови БСМ, було обрано оптимальну структуру мережі та проведено порівняльний аналіз методів маршрутизації, що дало змогу на основі проведених досліджень розробити програмне забезпечення для визначення оптимальної топології та протоколу маршрутизації для кожної окремої ділянки великої безпроводової сенсорної мережі.

Основні наукові результати наукової студентської роботи полягають у розробці програмного забезпечення для вибору нового протоколу маршрутизації на кожному з проміжних вузлів ретрансляторів. Цей метод, на відміну від існуючих, базується на багатокритеріальному порівнянні із пріоритетом енергоефективності окремих вузлів мережі. Це дало змогу продовжити термін роботи безпроводової сенсорної мережі на 25%.

Текстова частина магістерської атестаційної роботи: 94 с., 28 рис., 3 табл. та 24 джерела.

Ключові слова: безпроводові сенсорні мережі, маршрутизація, сенсорні вузли, доставка даних, ретрансляція повідомлень, балансування навантаження трафіку.

ABSTRACT

The work includes 94 pages, 28 pictures, 3 tables and 24 information sources.

The purpose of the work is research of routing methods in wireless sensor networks.

This work examines the main principles and features of routing in the wireless sensor network. Routing options have been analyzed depending on the purpose of the network. The classification of routing methods has been given and the most popular protocols have been considered. The comparison of the protocols has been done and benefits of all protocols have been indicated.

Key words: wireless sensor networks, routing, sensor nodes, data transmission, message relaying, traffic load balancing.

Development of routing methods in wireless sensor networks

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	8
ВСТУП	10
1. БЕЗПРОВОДОВІ СЕНСОРНІ МЕРЕЖІ	12
1.1. Загальні відомості	12
1.2. Особливості БСМ.....	14
1.3. Технічні вимоги до БСМ.....	15
1.4. Використання БСМ.....	18
Висновки до розділу 1	19
2. ОСОБЛИВОСТІ МАРШРУТИЗАЦІЇ В БЕЗПРОВОДОВИХ СЕНСОРНИХ МЕРЕЖАХ.....	21
2.1. Основні принципи та задачі маршрутизації.....	21
2.2. Аналіз параметрів маршрутизації в БСМ.....	24
2.2.1. Кількість проміжних вузлів	24
2.2.2. Енергія.....	25
2.2.3. Надійність	29
2.2.4. Якість обслуговування	30
Висновки до розділу 2	34
3. ПРОТОКОЛИ МАРШРУТИЗАЦІЇ В БЕЗПРОВОДОВИХ СЕНСОРНИХ МЕРЕЖАХ.....	36
3.1. Класифікація протоколів маршрутизації.....	36
3.2. Маршрутизація, орієнтована на дані	38
3.2.1. Протокол SPIN	39
3.2.2. Протокол Directed Diffusion.....	43
3.2.3. Протокол Rumor Routing.....	45
3.2.4. Протокол GBR.....	47

					КПІ ім. Сікорського 1321-с 22.ТЗ-71мп.2018.ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розроб.		Козик Ю.Л.			Розвиток методів маршрутизації в безпроводових сенсорних мережах Пояснювальна записка	Літ.	Арк.	Акрушів
Перевір.		Лисенко О.І.					6	68
Реценз.		Мошинська А.В.						
Н. Контр.		Петрова В.М.						
Затверд.		Явіся В.С.						

	6
3.2.5. Протокол OLSR.....	48
3.2.6. Протокол AODV	50
3.3. Ієрархічні протоколи маршрутизації	52
3.3.1. Протокол LEACH.....	54
3.3.2. Протокол PEGASIS.....	56
3.4. Маршрутизація на основі місцеположення.....	57
3.4.1. Протокол GAF.....	58
3.4.2. Протокол GEAR	60
3.5. Порівняльний аналіз протоколів маршрутизації для БСМ.....	61
Висновки до розділу 3	63
4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	67
4.1. Програмні засоби використані при розробці	67
4.2. Основні вимоги до алгоритму вибору протоколу маршрутизації	68
4.2. Розробка логіки прийняття рішень	72
4.3. Створення об'єктів та середовища моделювання.....	74
4.4. Розрахунок енерговитрат	75
4.5. Результати моделювання.....	78
Висновки до розділу 4	82
ВИСНОВКИ.....	84
ПЕРЕЛІК ПОСИЛАНЬ.....	87
Додаток А. ЛІСТИНГ СКРИПТА ПРОГНОЗУВАННЯ.....	90
Додаток Б. ЛІСТИНГ СКРИПТА МОДЕЛЮВАННЯ.....	92
Додаток В. ЛІСТИНГ СКРИПТА СЕРЕДОВИЩА ТА ОБ'ЄКТІВ	94

					КПІ ім. Сікорського 1321-с 22.ТЗ-32.2017.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ПЕРЕЛІК СКОРОЧЕНЬ

- AODV** (Ad-hoc On-demand Distance Vector) – однорангова дистанційна маршрутизація по вимозі
- BER** (Bit Error Rate) – коефіцієнт помилкових бітів
- CH** (Cluster Head) – голова кластеру
- CM** (Cluster Members) – член кластеру
- DD** (Directed Diffusion) – спрямована дифузія
- GAF** (Geographic Adaptive Fidelity) – географічний адаптивний протокол
- GBR** (Gradient-Based Routing) – маршрутизація заснована на градієнтах
- GEAR** (Geographic and Energy Aware Routing) – протокол географічної і енергозберігаючої маршрутизації
- GPS** (Global Positioning System) – система глобального позиціонування
- IP** (Internet Protocol) – мережевий протокол
- LEACH** (Low-Energy Adaptive Clustering Hierarchy) – ієрархічний алгоритм адаптивної кластеризації з низьким рівнем використання енергії
- MPR** (Multi-point Relays) – багато точкові ретранслятори
- OLSR** (Optimized Link State Routing) – оптимізована маршрутизація за стану каналів зв'язку
- PEGASIS** (Power-Efficient Gathering in Sensor Information Systems) – протокол енергоефективного збору інформації в сенсорних мережах
- QoS** (Quality of Service) – якість обслуговування
- RR** (Rumor routing) – маршрутизація за рахунок поширення слухів
- RREP** (Route Response) – відповідь маршруту
- RREQ** (Route Request) – запит маршруту
- SPIN** (Sensor Protocols for Information via Negotiation) – сенсорні протоколи для інформації через переговори
- TC** (Topology Control) – контроль за топологією
- TDMA** (Time Division Multiple Access) – доступ з розділенням у часі
- TTL** (Time To Live) – час життя запиту

WSN (Wireless Sensor Network) – безпроводова сенсорна мережа

БС – базова станція

БСМ – безпроводова сенсорна мережа

ВСТУП

Останнім часом активний розвиток отримали технології безпроводного зв'язку, що в поєднанні з досягненнями в області мікропроцесорної та виміральної техніки зробило можливим створення нового класу систем передачі даних – безпроводових сенсорних мереж. Безпроводова сенсорна мережа (БСМ) являє собою розподілену, самоорганізуючу і стійку до відмови мережу з великою кількістю (до декількох десятків тисяч) автономних електронних вузлів, здатних обмінюватися повідомленнями і ретранслювати їх по безпроводному каналу зв'язку.

Безпроводові сенсорні мережі можуть бути ефективно використані для вирішення різних прикладних задач, пов'язаних з розподіленим збором, обробкою і аналізом інформації в таких сферах, як автоматизація комплексів, промислова автоматика, безпека і оборона, моніторинг навколишнього середовища, охорона здоров'я і т.п., володіючи при цьому наступними перевагами:

- відсутність необхідності в прокладанні кабелів для електроживлення і передачі даних;
- низька вартість монтажу і технічного обслуговування системи;
- зручність розміщення автономних безпроводових вузлів в різноманітних точках простору;
- можливість впровадження і модифікації мережі на експлуатуємому об'єкті при мінімальному втручанні в процес його функціонування;
- надійність і відмовостійкість всієї системи в цілому при виході з ладу окремих вузлів або при порушенні зв'язку між ними.

У загальному випадку безпроводові сенсорні мережі є одноранговими мережами, в яких всі вузли рівноправні, мають автономні джерела живлення та можуть виступати в ролі ретранслятора пакетів інформації. Внаслідок цього для БСМ надзвичайно актуальним постає вирішення наступних завдань:

- завдання пошуку оптимальних маршрутів. Оптимальним вважається маршрут доставки пакетів інформації від вузла-відправника до вузла-отримувача, що вимагає мінімальних сумарних витрат ресурсів (наприклад, енергії) вузлів, що задіяні у цьому маршруті.
- завдання маршрутизації із забезпеченням максимального часу життя мережі. Під часом життя мережі розуміється термін її експлуатації до першого виходу з ладу одного з вузлів через виснаження автономного джерела живлення.

Отже, ефективність функціонування безпроводових сенсорних мереж багато в чому визначається вибором протоколу маршрутизації пакетів. Неправильний вибір протоколу маршрутизації призводить до нерівномірного завантаженості вузлів задіяних в ретрансляції пакетів, а це в свою чергу призводить до швидкої витрати заряду батарей цих вузлів.

Таким чином, вибір правильного способу маршрутизації є одним із головних факторів при розробці безпроводової сенсорної мережі.

1. БЕЗПРОВОДОВІ СЕНСОРНІ МЕРЕЖІ

Завдяки нещодавнім технологічним досягненням та швидким темпам розвитку технологій виробництво малих і недорогих датчиків стало технічно і економічно вигідним. Чутливі електронні датчики вимірюють навколишні умови, пов'язані з навколишнім середовищем і перетворюють данні в електронний сигнал і передають зібранні данні по безпроводному каналу. В результаті, після обробки даного сигналу можливо отримати інформацію про властивості об'єктів або подій, що відбуваються поблизу датчика. Велика кількість таких датчиків, які здатні контролювати велику територію, обмінюючись даними по безпроводному зв'язку, складають єдину безпроводову сенсорну мережу.

1.1. Загальні відомості

Безпроводова сенсорна мережа – це розподілена мережа, яка стійка до відмови окремих елементів. Водночас загальна кількість елементів може сягати від сотні до десятків тисяч сенсорних вузлів [1]. Сенсорні вузли обмінюються інформацією не тільки один з одним, а й з базовою станцією, що дозволяє їм надати зібрані данні для дистанційної обробки, аналізу та збереження. При цьому можлива функція ретрансляції повідомлень між різними елементами мережі, що дозволяє збільшити зону покриття до декількох кілометрів. Отже, узагальнену структурну схему безпроводової сенсорної мережі можна зобразити у вигляді двох груп датчиків, які контролюють дві різні ділянки простору та підключені до Інтернету за допомогою базових станцій (рис. 1.1).

Можливості сенсорних вузлів в БСМ можуть змінюватися в широких межах. Наприклад, прості вузли датчиків можуть контролювати один фізичний параметр, в той час як в більш складних датчиках можуть бути

об'єднані різні методи зондування (наприклад, акустичні, оптичні, магнітні). Вони також можуть відрізнятися за можливостями передачі даних, наприклад, за допомогою ультразвуку, інфрачервоних або радіочастотних технологій з різною швидкістю передачі даних і затримками. У той час як прості датчики можуть тільки збирати і передавати інформацію, більш потужні пристрої з кращими обчислювальними та енергетичними ресурсами, можуть самостійно проводити обробку та агрегацію даних. Такі пристрої часто дозволяють реалізувати додаткові функції в БСМ, наприклад, вони можуть утворювати першочергові шляхи зв'язку для з'єднання сенсорних вузлів з обмеженими ресурсами з базовою станцією. Нарешті, деякі пристрої можуть мати доступ до додаткових допоміжних технологій, наприклад, глобальних систем позиціонування (GPS) для визначення координат вузлів [2]. Однак, додаткові функції часто призводять до надмірних витрат енергії джерел живлення. А це стає причиною до зменшення часу життя мережі. Оскільки кожен елемент має власне автономне джерело живлення, то у випадку його виснаження, елемент відключається.

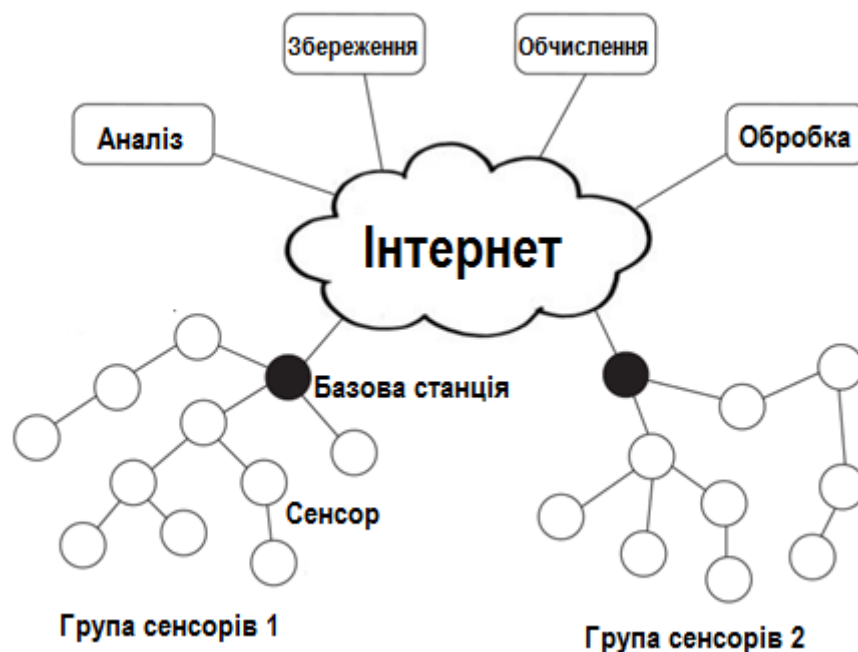


Рисунок 1.1 Безпроводова сенсорна мережа

Отже, однією з основних цілей проектування БСМ є здійснення обміну даними при одночасному прагненні продовжити термін служби мережі і запобігти погіршенню з'єднання шляхом використання методів управління енергією, зокрема, вибору доцільного способу маршрутизації.

1.2. Особливості БСМ

У той час як безпроводові сенсорні мережі мають багато спільних проблем з іншими розподіленими системами, вони мають ряд ключових відмінностей від інших типів безпроводових мереж передачі інформації, таких як локальні безпроводові мережі і мобільні епізодичні мережі.

Перелічимо основні особливості БСМ [3]:

- великі масштаби мережі – кількість вузлів в мережі може досягати десятків тисяч;
- складна топологія – в загальному випадку мережа має багатокоміркову топологію, де всі або більшість вузлів є нерухомими;
- обмежені ресурси вузлів – ємність автономного джерела живлення, обчислювальна потужність і пам'ять мікропроцесора, пропускна здатність каналів зв'язку та ін. дуже обмежені;
- види трафіку – в залежності від поточної прикладної задачі здійснюється підтримка типів трафіку «багато-до-одного», «один-до-багатьох» і «багато-до-багатьох»;
- розміщення вузлів – розташування вузлів в просторі може бути випадковим або детермінованим, їх розподіл по площі покриття мережі може бути як рівномірним, так і нерівномірним;
- самоорганізація і відмовостійкість – вузли можуть самостійно налаштовуватися на етапі розгортання системи, а також в процесі роботи адаптуватися до умов навколишнього простору та поточного режиму експлуатації;

- масштабованість – кількість службового мережевого трафіку і необхідний обсяг пам'яті вузлів майже не залежить від загального розміру мережі;
- модель генерації повідомлень – вузли можуть виконувати передачу пакетів за часом (періодично), за подією або, за запитом від зовнішнього отримувача інформації, при цьому можливі різні комбінації перерахованих варіантів;
- різноманітність вузлів і з'єднань – вузли можуть мати різні енергетичні ресурси, обсяги пам'яті і т.п., а безпроводні канали відрізняються швидкістю передачі даних, надійністю, відстанню зв'язку і т.п..

1.3. Технічні вимоги до БСМ

Завдяки очевидним перевагам, безпроводні сенсорні мережі є доволі затребуваною технологією. Це сприяє швидким темпам розвитку БСМ. Проте основними завданнями при розробці безпроводних сенсорних систем залишається створення малогабаритних, дешевих і більш ефективних пристроїв. Але подібні вимоги створюють жорсткі обмеження для багатofункціональності БСМ. Оскільки сенсорні вузли з невеликим споживанням енергії мають дуже малу обчислювальну потужність, яку можна зіставити із комп'ютерними системами минулого десятиліття [4]. Невеликий розмір датчику і низьке споживання енергії також забороняє інтеграції багатьох бажаних функцій та компонентів, наприклад, GPS приймачів.

Проблема енергоефективності пов'язана з тим фактом, що сенсорні вузли працюють з обмеженим запасом енергії. Як правило, вони розраховані на живлення від акумуляторів, які повинні бути замінені або заряджені одразу після їх розрядки. Водночас для деяких вузлів жоден з варіантів не підходить. В такому випадку вони будуть просто відкинуті, як тільки їх

джерело енергії буде вичерпано. Насамперед це негативно впливає на якість та кількість отриманої інформації і ефективність використання обладнання. Можливість заряду акумулятора також істотно впливає і на стратегію споживання енергії при виконанні збору і передачі інформації. Для одноразових батарей, сенсорний вузол повинен бути здатний працювати до закінчення часу його місії, або до моменту, коли батарея може бути замінена.

Як наслідок, першим і зазвичай найбільш важливим завданням при проектуванні БСМ є саме енергетична ефективність. Ця вимога відбивається на розробці як сенсорного вузла, так і при проектуванні всієї мережі.

Безпроводові сенсорні мережі зазвичай використовують у віддалених районах і суворих кліматичних умовах, без належної інфраструктури, можливості обслуговування та ремонту системи. Це не тільки негативно впливає на технічний стан обладнання, а й ускладнює швидке відновлення працездатності системи, у разі виникнення несправності. Таким чином, сенсорні вузли повинні бути стійкими до зовнішніх чинників впливу та самоорганізованими щодо налаштування власних параметрів, взаємодії з іншими вузлами, адаптації до збоїв, змін навколишнього середовища, а також зміни факторів, які впливають на роботу системи, без втручання людини. Тобто, всі вищеперелічені функції, повинні виконуватись в автономному режимі. При цьому важливою функцією є оптимізація, яка відноситься до здатності пристрою робити моніторинг і варіювати використання власних ресурсів. Нарешті, здатність самостійно відновлюватися дозволяє сенсорним вузлам ідентифікувати і реагувати на мережеві збої. В енергетично обмежених сенсорних мережах всі ці функції самоорганізації повинні бути розроблені і впроваджені таким чином, щоб вони не викликали значних додаткових витрат енергії.

Використання безпроводових мереж для зв'язку між вузлами створює ряд проблем при розробці БСМ. Наприклад, ослаблення сигналу обмежує відстань поширення радіосигналів, тобто, радіочастотний сигнал

зменшується в силі, поки він поширюється через середовище і при проходженні через перешкоди.

Велика кількість елементів мережі та енергетичні обмеження багатьох безпроводових сенсорних мереж не дозволяють завжди покладатися на централізовані алгоритми управління топологією та маршрутизацією, які б можливо було виконати на базовій станції. Замість цього, сенсорні вузли повинні співпрацювати з сусідніми до них вузлами для прийняття локальних рішень, без використання глобальних відомостей. Як наслідок, результат цих децентралізованих алгоритмів не буде оптимальним, але може бути більш енергетично ефективним, ніж централізовані рішення, за рахунок зменшення об'єму переданої службової інформації управління. Хоч даний метод негативно впливає на ефективності системи в цілому, але дає змогу подовжити час життя мережі.

Багато безпроводових сенсорних мереж застосовується для збору конфіденційної інформації. Проте дистанційна та автономна робота сенсорів підвищує ризик шкідливих мережевих атак. Крім того, безпроводова передача даних спрощує перехоплення інформації при передачі даних від датчиків. Наприклад, однією з найбільших загроз для функціонування мережі є атаки, метою яких є порушення нормальної роботи сенсорної мережі. Це може бути досягнуто з використанням різних атак, насамперед за допомогою приглушення сигналу, що погіршує якість зв'язку між вузлами. Наслідки можуть бути дуже серйозними і залежать від сфери застосування сенсорних мереж. Існують численні методи вирішення проблеми безпеки для розподілених систем, які запобігають атакам або обмежують їх вплив, проте більшість з них вимагає значних обчислювальних ресурсів. Ці вимоги часто не можуть бути виконані через обмеженість ресурсів сенсорних вузлів. Як наслідок, сенсорні мережі вимагають нових рішень для створення аутентифікації вузлів і шифрування інформації.

На основі викладеного вище можна зробити висновок, що багато рішень при розробці БСМ відрізняється від проектних рішень інших систем і

мереж. Ряд додаткових проблем можуть впливати на конструкцію сенсорних вузлів і планування безпроводових сенсорних мереж. Наприклад, деякі датчики можуть бути встановлені на рухомих об'єктах, таких як транспортні засоби, що призводить до безперервної зміни топології мережі. У таких системах потрібна оперативна адаптація на декількох рівнях системи, в тому числі маршрутизації (наприклад, зміна списків сусідніх вузлів), управління доступом до середовища (наприклад, тривалості заняття каналу), і агрегації даних. Вузлам датчиків буде необхідно більше апаратних ресурсів, якщо їх завдання вимагає більше обчислень і збереження даних, або якщо вони відповідають за збір та обробку даних отриманих від певної кількості інших датчиків в мережі. Крім того, деякі сенсори можуть мати специфічні вимоги до продуктивності і якості даних.

1.4. Використання БСМ

Сфера використання безпроводових сенсорних мереж може бути різноманітною: від датчиків руху на охоронюваних об'єктах, до системи контролю за вологістю повітря у бібліотечних архівах. БСМ можуть бути використані на будь-яких ділянках, де необхідний збір інформації про стан предметів, середовища, явища або процесу [5].

БСМ ідеально підходять для роботи на віддалених ділянках, де неможливо здійснювати постійне пряме втручання людини.

БСМ широко використовуються у системах для моніторингу і захисту цивільної інфраструктури, національної енергосистеми і трубопровідної інфраструктури. Мережі із тисяч вузлів та датчиків вже використовуються для моніторингу великих природних територій, моделювання і прогнозування забруднення навколишнього середовища і повеней, збору інформацію про цілісність структури на мостах за допомогою датчиків вібрації, контролю

використання води, добрив і пестицидів для поліпшення здоров'я і кількість врожаю.

Один з підходів до вирішення проблеми заторів є використання розподілених сенсорних систем. Ці системи збирають інформацію про щільність, розмір і швидкості транспортних засобів на дорогах, сигналізують про затори і пропонують водіям деякі альтернативні маршрути.

Широке застосування БСМ знайшли у системі «Розумний дім». Дана система полягає в відслідковуванні усіх показників помешкання (наприклад, температура повітря або освітленість) і самостійному прийнятті рішень по їх зміні, у відповідності із встановленими нормами. Також можливе дистанційне керування пристроями, які підключенні до даної системи.

Висновки до розділу 1

У даному розділі були розглянуті загальні відомості про безпроводові сенсорні мережі.

БСМ – це розподілена мережа із великої кількості датчиків, які об'єднанні між собою за допомогою радіоканалу, стійка до відмови окремих елементів, з можливістю самоорганізації. Зона покриття БСМ може складати від декількох метрів до десятків кілометрів за рахунок функції ретрансляції повідомлень від одного вузла до іншого.

Проаналізовано принцип обміну інформації між вузлами мережі та розглянуто узагальнену структурну схему безпроводової сенсорної мережі.

Детально розглянуто особливості безпроводових сенсорних мереж, порівняно з іншими типами розподілених мереж.

Переваги безпроводових сенсорних мереж:

- велика кількість елементів мережі;
- автономність джерела живлення для кожного вузла;
- підтримка різних видів трафіку та протоколів маршрутизації;

- можливість реалізації різноманітних схем розташування вузлів;
- самоорганізація;
- відмовостійкість;
- невеликі розміри датчиків;
- різнорідність вузлів і з'єднань.

На етапі проектування мережі, накладається багато обмежень на технічні характеристики кожного елементу мережі і модель мережі в цілому. Здебільшого вони викликані обмеженими енергетичними ресурсами датчиків. Однією з основних цілей проектування БСМ є здійснення обміну даними при одночасному прагненні продовжити термін служби мережі і запобігти погіршенню з'єднання. Одним із найефективніших методів управління енергії є вибір оптимального способу маршрутизації.

Розглянуто сфери використання безпроводових сенсорних мереж. Наведено приклади використання БСМ у повсякденному житті людини, для охорони об'єктів, у різних галузях виробництва, медицині, для оцінки стану навколишнього середовища, моніторингу масштабних систем, для оцінки технічного стану конструкцій.

2. ОСОБЛИВОСТІ МАРШРУТИЗАЦІЇ В БЕЗПРОВОДОВИХ СЕНСОРНИХ МЕРЕЖАХ

Дані зібрані датчиками в безпроводових сенсорних мережах, як правило, поширюються у напрямку до базової станції, яка пов'язує БСМ з іншими мережами, де дані можуть бути оброблені і проаналізовані. Процес встановлення шляху передачі повідомлення від джерела до приймача (наприклад, базової станції) через один або більше ретрансляторів називається маршрутизацією і є основним обов'язком мережевого рівня набору протоколів зв'язку.

2.1. Основні принципи та задачі маршрутизації

У малих сенсорних мережах, де датчики розташовані близько один до одного, може здійснюватись прямий зв'язок між усіма сенсорними вузлами з базовою станцією. На рис. 2.1 зображено принцип прямого зв'язку, коли всі сенсорні вузли можуть зв'язатися з приймачем без ретрансляції повідомлень через інші вузли. Ця модель прямого зв'язку – найпростіша реалізація, де всі дані здійснюють один стрибок для досягнення мети.

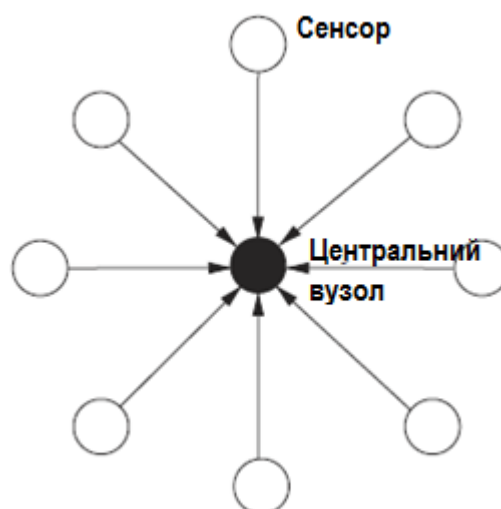


Рисунок 2.1 Модель прямого зв'язку

Тим не менше, більшість БСМ складаються з великої кількості сенсорних вузлів, які покривають масштабну територію, що потребує використання непрямого зв'язку (рис. 2.2). Тобто, сенсорні вузли повинні не тільки генерувати і поширювати їх власну інформацію, але також виконувати функцію ретрансляції повідомлень від інших сенсорних вузлів. В даному випадку, вирішальним завданням для мережевого рівня всіх сенсорних вузлів, полягає, в визначенні шляху від сенсора до приймача через безліч інших сенсорних вузлів, які діють як проміжні вузли.

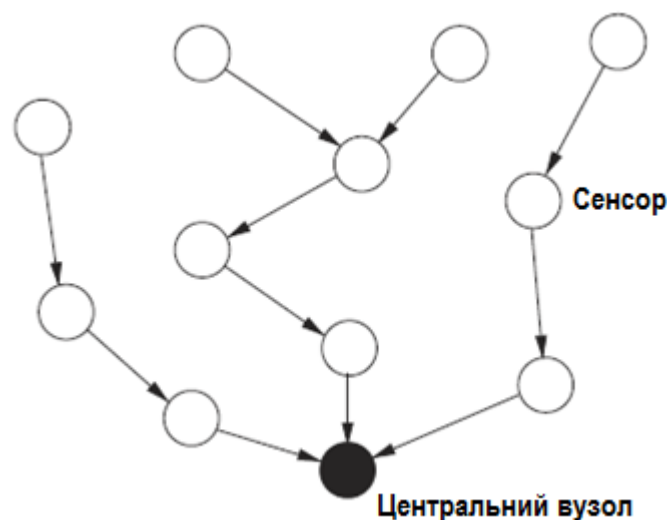


Рисунок 2.2 Модель непрямого зв'язку

Цей спосіб маршрутизації протоколу є складним через унікальність характеристик БСМ, як брак ресурсів або ненадійність бездротового середовища. Наприклад, обмежена обробка даних, зберігання, пропускна здатність і енергоємність, вимагають рішень маршрутизації, які є легкими, в той час як можливі зміни в топології мережі, вимагають рішень маршрутизації які легко пристосовуються і є варіативними.

Коли вузли БСМ розгорнуті детермінованим способом (тобто, вони розташовані в певних визначених місцях), для зв'язку між ними і базовою станцією будуть використовуватись визначені маршрути маршрутизації. Проте, якщо вузли розташовані випадком чином, то топологія мережі є

неоднорідною і заздалегідь непередбачуваною. В даному випадку, першочерговим завданням вузлів є самоорганізація, тобто, вони повинні співпрацювати, щоб визначити своє місце розташування у просторі, визначити сусідні до них вузли, і знайти шлях до базової станції [6].

В області БСМ одним з найбільш важливих параметрів є час життя мережі, який визначається, як термін експлуатації мережі до першого виходу з ладу одного з вузлів через виснаження автономного джерела живлення.

Простий пошук маршрутів з мінімальною вартістю може призвести до нерівномірного розподілу мережевого навантаження між вузлами мережі, бо вузли, що знаходяться на часто використовуваних оптимальних маршрутах, будуть частіше виконувати ретрансляцію пакетів. Отже, можуть утворюватися точки, які будуть задіяні для одночасної ретрансляції великих обсягів інформації. Поява таких точок із підвищеним навантаженням на вузли, сприяє виникненню проблем з пропускнуою спроможністю і передчасним виснаженням джерел енергії елементів. Тому для забезпечення високих показників ефективності та тривалості життя мережі в цілому, необхідно виконувати так зване балансування навантаження. Тобто розподілення потоків трафіку між вузлами для більш рівномірного використання їх ресурсів (в першу чергу, енергії). Таким чином, завдання пошуку маршрутів з балансуванням навантаження полягає в оптимізації загальних енергетичних витрат мережі, а не окремо для кожного з маршрутів.

Отже, для безпроводових сенсорних мереж можна виділити два завдання маршрутизації, які будуть відрізнятись критеріями пошуку маршрутів:

- завдання пошуку оптимальних маршрутів – оптимальним маршрутом вважається шлях доставки пакетів інформації від вузла-відправника до вузла-призначення, що вимагає мінімальних сумарних витрат ресурсів вузлів цього шляху;
- завдання маршрутизації із забезпеченням максимальної тривалості життя мережі.

2.2. Аналіз параметрів маршрутизації в БСМ

В залежності від призначення, безпроводові сенсорні мережі широко відрізняються у своїх обмеженнях і характеристиках, що має братися до уваги при розробці протоколу маршрутизації. Наприклад, більшість БСМ будуть обмежені в енергетичних ресурсах, продуктивності, ємності накопичувача. Сенсорні мережі можуть широко відрізнятись в масштабах та площах географічних зон, які вони покривають. Всесвітні програми адресації (такі як IP адреси, які використовуються в інтернеті) можуть бути недоступними і навіть недоцільними, зокрема в мережах з різнотипними вузлами і мобільними вузлами. Нарешті, в залежності від призначення інформації, сенсорні дані будуть збиратися різними способами, що відрізняються один від одного. У схемах керованих часом (наприклад, моніторинг навколишнього середовища), вузли періодично поширюють зібрані ними дані до приймача. У схемах керованих подіями (наприклад, виявлення пожеж руйнівної сили), вузли повідомляють про зібрану ними інформації тільки, коли відбувається цікава для них подія. Нарешті, в схемах керованих запитами, приймач розсилає запит на збір даних та проводить збір даних від сенсорів тільки, коли необхідно [7].

Отже, параметри маршрутизації використовуються, що б охарактеризувати різні цілі протоколів маршрутизації, з урахуванням використання цих ресурсів. Розглянемо найважливіші параметри та критерії при розробці способу маршрутизації.

2.2.1. Кількість проміжних вузлів

Найбільш поширеним показником, що використовуються в протоколах маршрутизації – мінімальний хоп (або найкоротший хоп), тобто протокол маршрутизації намагається знайти шлях від відправника до одержувача, що

вимагає найменшу кількість проміжних вузлів. В цьому простому алгоритмі, кожна ланка має певну вартість і протокол маршрутизації вибирає шлях, який мінімізує загальну вартість для поширення даних від джерела до місця призначення. Основна ідея цієї метрики полягає в тому, що використання найкоротшого шляху призведе до зменшення часу передачі і низьких витрат ресурсів, тому що буде задіяне якнайменше передавальних вузлів. Однак, оскільки цей підхід не враховує фактичної доступності ресурсів на кожному вузлі, результуючий маршрут, ймовірно, буде неоптимальним з точки зору затримки енергії та запобігання заторів. В багатьох безпроводових сенсорних мережах може знадобитися використання вузлів з різними апаратними можливостями, наприклад, потужністю, ємністю джерела енергії, пропускною здатністю, відстанню передачі. Тому в процесі маршрутизації необхідно враховувати різноманітність елементів мережі.

Проте, метрика мінімального хопу використовується в багатьох протоколах маршрутизації, завдяки своїй простоті і практичності.

2.2.2. Енергія

Без сумніву, ключовий аспект маршрутизації в БСМ – це енергоефективність. Як правило, серед елементів вузла, найбільше енергії споживає приймач, тому головний спосіб зменшити середнє енергоспоживання вузла полягає в мінімізації активності в радіоканалі (передача і прийом даних, прослуховування каналу). З огляду на те, що кожен вузол є не тільки джерелом або одержувачем інформації, а й у разі необхідності проміжним ретранслятором пакетів, оптимізація обсягів і напрямів потоку трафіку є важливим завданням рівня маршрутизації.

Існує не один унікальний енергетичний показник, який може бути застосований до задачі маршрутизації. Існують різні інтерпретації енергоефективності, проте, до однієї із ключових можна віднести ідею

мінімальної витрати енергії на пакет. Це найпростіша концепція енергетичної ефективності, тобто, її мета полягає в тому, щоб мінімізувати загальну кількість енергії, що витрачається для поширення одного пакету від джерела до місця призначення. Повна енергія – сумарна енергія, що споживається кожним вузлом уздовж маршруту для прийому і передачі пакета. На рис. 2.3 зображено приклад невеликої сенсорної мережі, коли вузол джерела хоче передати пакет до вузла призначення з використанням маршруту, який мінімізує накладені витрати енергії пакета. Число на кожній лінії зв'язку вказує на вартість пакета, що поширюється за цим напрямком. Як наслідок, цей пакет буде передаватись через вузли А – D – G із загальною вартістю 5. Можна зауважити, що цей маршрут відрізняється від варіанту з використанням мінімального хопу маршруту (В – G).

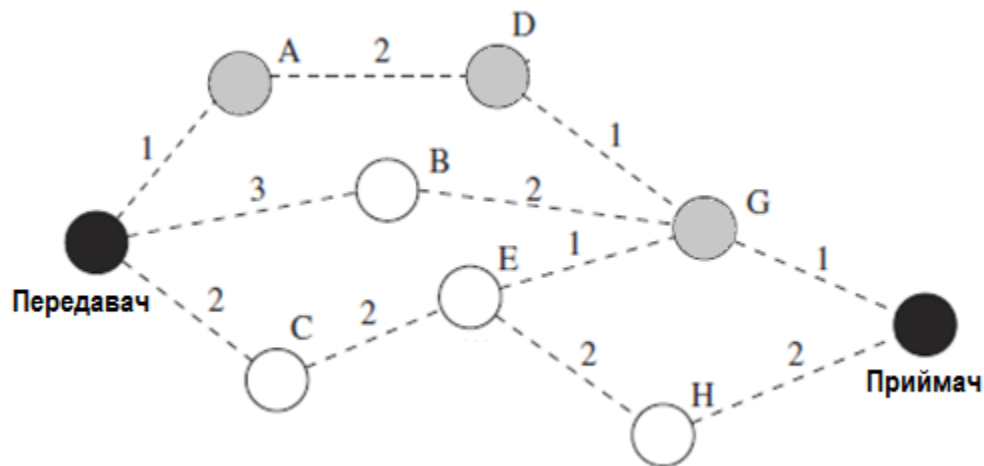


Рисунок 2.3 Вибір маршруту за умови мінімальної витрати енергії на пакет

Концепція максимального часу поділу мережі полягає в поділі всієї мережі на кілька менших підмереж. Це відбувається, коли останній вузол, який пов'язує дві частини мережі, виходить з ладу. Як наслідок, підмережі можуть бути недоступні, а робота сенсорних вузлів в цій під мережі є марною. Таким чином, завдання полягає в тому, щоб зменшити споживання енергії на вузлах, які мають вирішальне значення для підтримки мережі, в якій кожен сенсорний вузол, може бути, досягнутий за допомогою

щонайменше одного маршруту. Наприклад, мінімальний набір вузлів, видалення яких призведе до поділу мережі, можна знайти, використовуючи теорему про максимальний потік і мінімальний розріз. Після того, як протокол маршрутизації визначив ці вирішальні вузли, можна спробувати збалансувати навантаження в мережі. На рис. 2.4 вузол D є вирішальним вузлом. Наприклад, якщо батарея цього вузла вичерпалася, вузли F, I, J стануть недоступними для будь-якого іншого вузла в мережі.

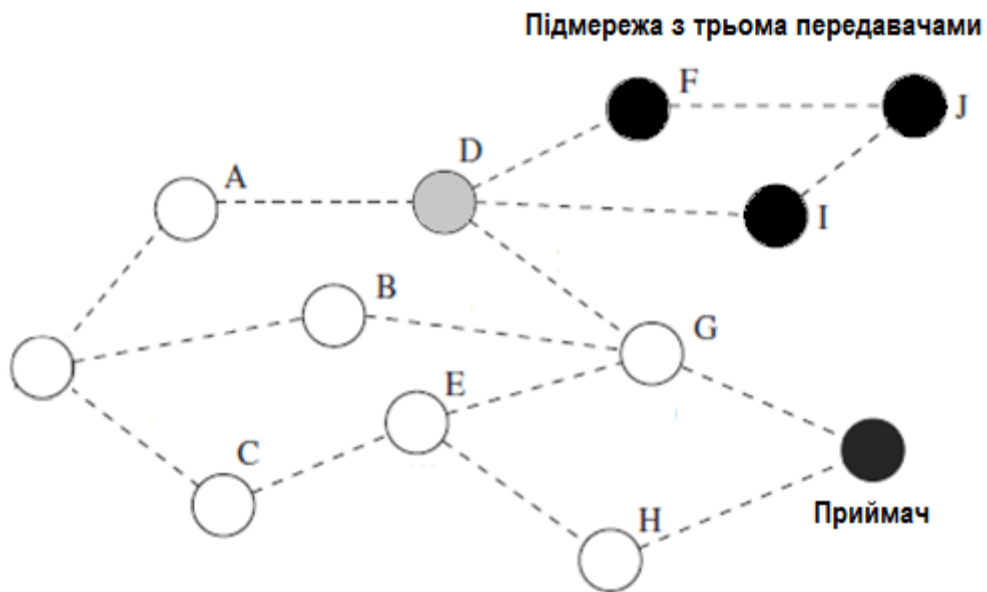


Рисунок 2.4 Вибір маршруту за критерієм максимального часу поділу мережі

У випадку використання концепції мінімальної різниці в рівнях потужності вузлів, всі вузли в мережі вважаються однаково важливими і завдання полягає в тому, щоб розподілити споживання енергії на всіх вузлах мережі якомога більш рівномірно. Мета такого підходу може полягати в тому, щоб максимізувати термін служби всієї мережі, наприклад, замість того, щоб деякі вузли виснажувались раніше, ніж інші, що тим самим постійно б зменшувало розмір мережі, можна прагнути до збереження більшої кількості робочих вузлів, так довго, наскільки це можливо. В ідеальному випадку, всі вузли мають вимкнутись одночасно, але подібний сценарій є практично нездійсненним.

Використання способу із визначенням середньої енергоємності. В цьому підході, ключовим моментом є не енергетична вартість поширення пакетів, а енергоємність (наприклад, поточний рівень заряду батареї) вузлів. Протокол маршрутизації, який використовує цю метрику, прокладає маршрут, який має найбільшу сумарну енергоємність вузлів, від джерела до одержувача. На рисунку 2.5, цифри під вузлами вказують на енергоємність вузлів. У цьому прикладі, протокол маршрутизації може вибрати шлях С – Е – Н, який має найбільшу сумарну потужність. Протокол маршрутизації, який використовує цю метрику, повинен бути ретельно розроблений, щоб уникнути небезпеки вибору надмірно довгих маршрутів для того, щоб максимізувати загальну пропускну здатність енергії. Зміна цього показника полягає в максимізації середньої енергоємності, що дозволяє уникнути цієї проблеми.

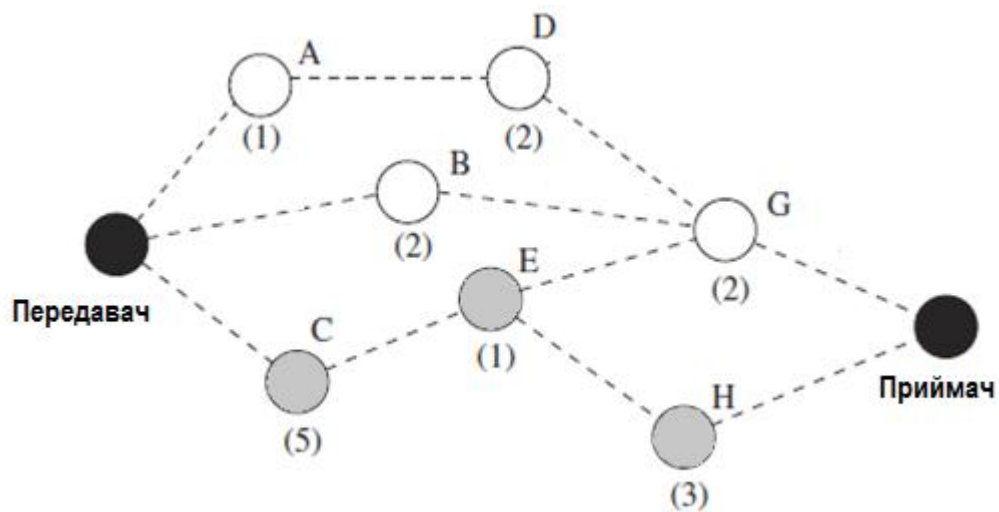


Рисунок 2.5 Вибір маршруту за середньою енергоємністю

У принципі найбільшого мінімального обсягу енергії, замість максимізації енергетичних потужностей на всьому шляху, основна мета маршрутизації полягає у виборі шляху з найбільшою мінімальною енергоємністю. Цей метод також допомагає маршрутам з великими енергетичними запасами, але і захищає вузли малої потужності від передчасного закінчення терміну їх дії. На рис. 2.6, протокол, який

використовує цей показник, вибирає маршрут В – G, так як мінімальна ємність по цьому маршруту становить 2, що більше, ніж мінімальна потужність вузлів у всіх інших можливих маршрутах.

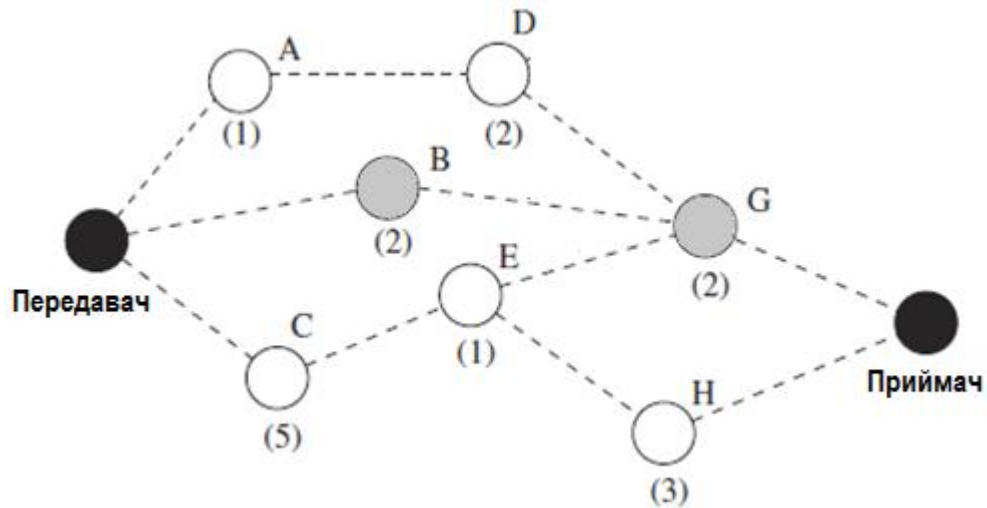


Рисунок 2.6 Вибір маршруту за найбільшим мінімальним обсягом енергії

Ці різні формулювання способів підвищення економії енергії призводять до дуже різних реалізацій протоколів, що відрізняються за своїми результатами і витратами. Наприклад, щоб визначити мінімальну енергію, яка споживається для кожного пакета, вартість для прийому і передачі пакета може залежати від розміру пакета. З іншого боку, енергетична потужності змінюються з плином часу, а отже, протокол маршрутизації з використанням метрики на основі ємності повинен періодично оновлювати ці дані.

Функцію забезпечення енергоефективності можна записати наступним чином:

$$(\mathbf{R}'', T'') = f(\mathbf{R}', T', L) \quad (2.2.2.1)$$

де \mathbf{R}'' - вектор зв'язків між вузлами, доповнений новими маршрутами,
 T'' - множина транзитних вузлів з урахуванням змінених і доданих на цьому етапі,

$f(x)$ - функція перерозподілу транзитних вузлів з метою забезпечення енергоефективності

$$y_e = g(\mathbf{R}'' , T'' , \mathbf{Q}) \quad (2.2.2.2)$$

$y_e \in \{0,1\}$ – результат перевірки енергоефективності мережі,

$g(x)$ – функція перевірки на енергоефективність.

На сьогоднішній день універсального визначення поняття енергоефективності немає – воно залежить від конкретної предметної області, тим не менш часто використовується наступний підхід: енергоефективність визначається тим, наскільки раціонально в деякій системі використовується надається їй ззовні енергія. Дається таке визначення коефіцієнта енергоефективності [13]:

$$E = \frac{W_p}{W_p + W_{NP}} \quad (2.2.2.3)$$

де W_p – корисно використана енергія;

W_{NP} – непродуктивні витрати.

У роботах, пов'язаних з БСМ, поняття часу автономної роботи часто не відрізняється від поняття енергоефективності. Тобто вважається, що більша енергоефективності забезпечує більший час автономної роботи.

Однак для спростування даного факту можна навести наступний простий приклад: мережа, в якій відсутній передача корисних даних, а енергія витрачається тільки на паразитні процеси, очевидно, має на порядок більший час автономної роботи, ніж мережу, яка транслює дані. Але оскільки енергія не витрачається в корисних цілях, коефіцієнт ефективності дорівнює нулю.

2.2.3. Надійність

Передбачається, що умови експлуатації БСМ можуть бути жорсткими, тому існуватиме вірогідність виходу з ладу вузлів і порушення з'єднань між

ними. Отже, для забезпечення високої надійності всієї системи в цілому, метод маршрутизації повинен автоматично формувати нові маршрути в обхід виключеним вузлам, витрачаючи на переконфігурацію маршруту якомога менші ресурси.

В багатьох системах будуть вибиратися маршрути, які залишаться стабільними на довгий проміжок часу. Для цього вузол може змінити і оцінити якість каналу у порівнянні з його сусідами і потім вибрати сусіда для наступного хопу, що збільшує ймовірність успішної передачі. Протокол маршрутизації може ідентифікувати кілька шляхів для мінімального хопу і потім вибрати той, який гарантує кращу якість серед усіх маршрутів. У мережах з мобільними вузлами, протокол маршрутизації може також використовувати метрику стійкого зв'язку, яка вимірює, наскільки канал буде доступним в майбутньому. Ці показники можуть бути використані, щоб виконувати вибір серед більш надійних маршрутів і стаціонарних вузлів.

2.2.4. Якість обслуговування

Термін якість обслуговування (QoS) відноситься до певних показників ефективності в мережах, зокрема для виявлення затримки при передачі пакету, пропускної здатності рівня і коефіцієнта помилок.

Вибір метрики QoS залежить від типу програми. Сенсорні мережі, що здійснюють виявлення та супроводження цілей вимагають низьких затримок при передачі термінових сенсорних даних, в той час як мережі з інтенсивним обміном даними (наприклад, мультимедійні сенсорні мережі) вимагають високої пропускної здатності.

Коли метод маршрутизації обирає маршрут передачі даних, оптимальний з точки зору деяких показників якості обслуговування, то він покладається в своїй оцінці на набір параметрів стану вузлів і каналів зв'язку. Ці параметри відображають поточну картину розподілу трафіку в мережі,

тому їх аналіз дозволяє вибирати відповідні маршрути передачі даних, наприклад, в обхід заторів трафіку в мережі або з метою мінімізації затримки передачі пакету.

Ресурсом мережі є буквально будь-який ресурс, котрий споживається під час виконання завдань пошуку відповідного маршруту, встановлення і підтримки сесій передачі даних, підтримки таблиць маршрутизації. Нижче наведено класифікацію мережевих ресурсів.

Ресурси мережі, необхідні для забезпечення якості маршрутизації:

- процесорний час. Хоча з кожним роком мобільні пристрої мають все більш потужні процесори, вони все ще обмежені в своїх обчислювальних можливостях і не можуть конкурувати з традиційними настільними комп'ютерами. Однак, це найменш критичний фактор, оскільки методи маршрутизації зазвичай не створюють велике навантаження на процесор;
- обсяг пакетного буфера вузла. Неминучий факт, що під час роботи мережі кілька вузлів одночасно передають пакети даних або маршрут до будь-якого вузла ще не був виявлений, тому необхідно тимчасово поміщати пакети в буфер. Більш того, коли буфери стають цілком заповнені, всі вхідні пакети будуть відкинуті, що веде до підвищення частки втрачених пакетів;
- пропускна здатність каналу зв'язку. Зазвичай вимірюється в бітах на секунду і свідчить безпосередньо про швидкість передачі даних і затримку. Однак, оскільки безліч вузлів одночасно використовують канал зв'язку, нам необхідно якимось чином виразити частку пропускної здатності, яку отримує кожен з вузлів. Спосіб вираження частки залежить від методу доступу до середовища передачі. При змагальному методі доступу, жодному вузлу не гарантується отримання доступу до каналу в необхідний йому момент часу, оскільки одночасно з ним інші вузли також можуть почати передачу даних. Тому, доступ до середовища передачі надається лише з певною

ймовірністю. Проте, більшість методів маршрутизації базуються на тому, що доступ до середовища передачі надається за змагальною схемою.

Для оцінки ефективності роботи методів маршрутизації з точки зору якості обслуговування використовуються наступні показники:

- середня затримка передачі пакету. Визначається як час, що минув з моменту відправки пакета вузлом до моменту прийому його вузлом призначення, для всієї мережі в цілому за весь час роботи;
- частка втрачених пакетів для всієї мережі. Вимірюється у відсотках, як частка втрачених пакетів від загальної кількості відправлених пакетів за весь час роботи мережі;
- частка прийнятих сесій передачі даних. Визначається у відсотках. Якщо метод маршрутизації не зміг знайти потрібний маршрут для передачі даних з урахуванням якості обслуговування, то він може відмовити у встановленні сесії. Цей показник відображає, який відсоток сесій був успішно прийнятий. Його значення показує ефективність методу маршрутизації і якість зв'язку. Очевидно, що даний показник неможливо застосувати для методів, які прагнуть поліпшити параметри якості обслуговування для всієї мережі в цілому;
- частка успішних сесій. Визначається у відсотках. Відображає, який відсоток додатків був успішно опрацьовано після того, як метод маршрутизації встановив сесію передачі даних. Даний показник з точки зору кінцевого користувача може бути важливіше, ніж частка прийнятих сесій передачі даних;
- загальна пропускна здатність мережі. Обсяг даних, переданих всією мережею за одиницю часу;
- пропускна спроможність вузла. Середня пропускна здатність, досягнута цим вузлом мережі за час роботи. Метод маршрутизації повинен визначати поточну пропускну здатність каналу зв'язку і оцінювати його придатність за цим параметром;

- затримка виявлення маршруту. Середній час, що проходить між моментом, коли вузол запитує встановлення сесії і моментом, коли протокол здійснює цю сесію;
- енергоефективність протоколу маршрутизації. Визначаються як заряд батареї живлення, витрачений на обробку та передачу одного пакета;
- нормалізована навантаження маршрутизації. Визначає відсоток керуючих пакетів маршрутизації серед всіх пакетів даних. Цей показник дає можливість оцінити накладні витрати на маршрутизацію (передачу контрольних повідомлень);
- час відновлення маршруту. Цей показник вказує час, який проходить від моменту збою або виходу одного з проміжних вузлів на маршруті до моменту, коли протокол маршрутизації встановить обхідний маршрут для даної сесії. Зазвичай характеризує стійкість протоколу маршрутизації до збоїв мережі і можливість балансувати навантаження.

При розробці БСМ необхідно знайти баланс між впровадженням якості обслуговування конкретних програм і енергоефективністю мережі в цілому.

При проектуванні методу маршрутизації для безпроводової сенсорної мережі, розробнику доводиться часто вибирати між різними можливостями в залежності від мети і призначення мережі. Наприклад, важливим є вибір між пропускною здатністю мережі та затримкою передачі пакету. У БСМ ці чинники є взаємовиключними. Якщо знизити вимоги до затримки передачі пакету, то можливо збільшити загальний обсяг трафіку, що передається мережею, шляхом збільшення розміру пакетів і резервування каналів зв'язку на довший час. Крім того, можна використовувати наступний алгоритм маршрутизації, якщо програма не накладає взагалі ніяких обмежень на затримку передачі пакету. Джерело може розділити пакети однієї сесії і відправити їх різним сусіднім вузлам. Ці вузли, при потраплянні в зону прямої взаємодії з вузлом-отримувачем, направляють йому збережені пакети. За рахунок поділу маршрутів досягається велика щільність передачі даних, але при цьому пакети можуть приходити з абсолютно різною затримкою. В

якості іншої стратегії можна спробувати знизити затримку передачі, якщо направити пакети відразу по декількох маршрутах одночасно. Але такий метод знижує ефективність використання пропускної здатності мережі в цілому, через відправки кількох копій одного пакета. Очевидно, що такий метод також призводить до збільшення енерговитрат на передачу повідомлення.

Отже, з точки зору проектування безпроводової сенсорної, метою оптимізації процесу маршрутизації є задоволення вимог якомога більшої кількості користувачів, тобто поліпшення показників якості обслуговування для всієї мережі в цілому, або для окремих сесій передачі даних.

Висновки до розділу 2

У даному розділі було розглянуто маршрутизацію у безпроводових сенсорних мережах. Розглянуті основні варіанти зв'язку між вузлами мережі.

Ефективність функціонування безпроводових сенсорних мереж багато в чому визначається вибором протоколів маршрутизації пакетів. При проектуванні БСМ необхідно прагнути до зменшення і вирівнювання навантаження вузлів, що досягається завдяки протоколу маршрутизації.

Виділяють дві основні задачі маршрутизації:

- завдання пошуку оптимальних маршрутів;
- завдання маршрутизації із забезпеченням максимальної тривалості життя мережі.

В залежності від призначення, розмірів та технічних особливостей, безпроводові сенсорні мережі широко відрізняються у своїх обмеженнях і характеристиках, що має братися до уваги при розробці протоколу маршрутизації. Для охарактеризування можливостей протоколів маршрутизації, з урахуванням споживання ресурсів, необхідної

продуктивності та потужності обладнання, використовують наступні основні показники протоколів маршрутизації:

- кількість проміжних вузлів;
- енергоефективність;
- надійність;
- необхідна якість обслуговування.

Детально проаналізовані фактори, які впливають на якість передачі інформації та ресурси мережі, необхідні для її покращення. Оскільки більшість методів маршрутизації прагне поліпшити тільки один з показників якості обслуговування, наприклад, середню затримку передачі пакета або загальну пропускну здатність мережі. Покращення декількох показників одночасно, негативно впливає на інші параметри і збільшує енерговитрати мережі. Тому при розробці БСМ постає задача оптимізації ресурсів, з метою знайти баланс між покращенням ключових параметрів і енергоефективністю мережі в цілому.

3. ПРОТОКОЛИ МАРШРУТИЗАЦІЇ В БЕЗПРОВОДОВИХ СЕНСОРНИХ МЕРЕЖАХ

3.1. Класифікація протоколів маршрутизації

Існують різні способи класифікації протоколів маршрутизації. На рис. 3.1 представлено класифікацію протоколів маршрутизації за організацією мережі. Більшість протоколів маршрутизації чітко відносяться до одного з трьох класів [8]. У плоских протоколах маршрутизації передбачається, що всі вузли мають рівні функції і ролі. А в ієрархічних протоколах маршрутизації, навпаки, різні вузли, мають різні ролі в процесі маршрутизації, тобто деякі вузли можуть направляти дані від імені інших, в той час, як інші вузли лише генерують і поширюють дані отримані від їх власних сенсорів. Протокол маршрутизації, який базується на місцезнаходженні, покладається на місце розташування інформації отриманої від вузлів, щоб прийняти рішення про подальшу маршрутизацію.



Рисунок 3.1 Класифікація протоколів за способом організації мережі

Протоколи маршрутизації несуть відповідальність за визначення або відкриття маршруту від відправника до необхідного одержувача. Цей процес можна використовувати, щоб розрізняти різні види протоколів маршрутизації. На рис. 3.2 зображено класифікацію способів маршрутизації

за способом дослідження маршруту. Наприклад, реактивні протоколи можуть створювати маршрут на вимогу, тобто в будь-який час, коли відправник хоче надіслати дані приймачу і ще не має встановленого маршруту. У той час, як відкриття реактивного маршруту спричиняє певні затримки до здійснення передачі даних, проактивний протокол маршрутизації встановлює маршрути, до того, як в них виникне необхідність. Деякі протоколи демонструють характеристики реактивних і проактивних протоколів і тому належать до категорії гібридних протоколів маршрутизації.

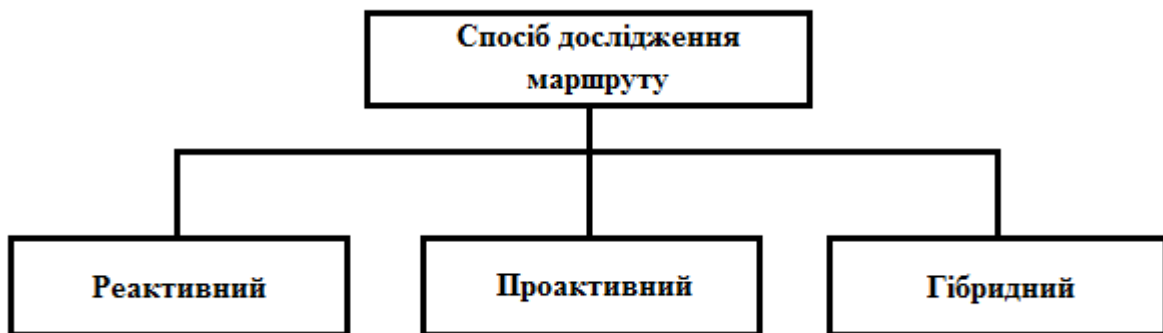


Рисунок 3.2 Класифікація протоколів за способом дослідження маршруту

Зрештою, протоколи маршрутизації також відрізняються за тими функціями, які вони здійснюють (рис. 3.3). Наприклад, метою протоколів заснованих на узгодженні, є зниження надлишкової передачі даних, спираючись на обмін узгоджених повідомлень між сусідніми сенсорними вузлами до виникнення фактичної передачі даних. До цієї категорії належать SPIN протоколи. Багатопроменеві (multipath) протоколи використовують безліч маршрутів одночасно, щоб досягти високої продуктивності або стійкості до помилок. У протоколи маршрутизації із запитами, сенсорні вузли відправляють дані тільки після отримання запиту від вузла приймача. Метою протоколів маршрутизації QoS є задоволення показників якості обслуговування, таких як затримка, низьке споживання енергії, або низькі втрати пакетів. Протоколи маршрутизації також відрізняються тим, як вони

підтримують обробку даних в мережі. Когерентні протоколи виконують лише мінімальний обсяг обробки (наприклад, усуваючи дублікати, позначки часу), перш ніж вони відправлять дані. Проте, в некогерентних протоколах, вузли можуть виконувати значну місцеву обробку вихідних даних, перед тим як передати їх до інших вузлів для їх подальшої обробки.



Рисунок 3.3 Класифікація протоколів за принципом роботи

Більшість протоколів маршрутизації орієнтовані на одноадресну маршрутизацію, тобто, спрямовують сенсорні дані до певного приймача. В той час, як багатоадресні і ширококомвні підходи маршрутизації поширюють дані на певну кількість вузлів.



Рисунок 3.4 – загальна класифікація протоколів маршрутизації БСМ

Кожен з протоколів адаптований для деякого класу бездротових мереж. Вибір деякого протоколу маршрутизації безпосередньо впливає на продуктивність мережі, яка може бути визначена різними способами, в залежності від розв'язуваної задачі [8].

3.2. Маршрутизація, орієнтована на дані

Першою категорією протоколів маршрутизації є протоколи з плоским способом організації мережі [9].

Для створення ряду спеціалізованих методів маршрутизації в БСМ була врахована наступна їх особливість: вузли БСМ виконують однаковий набір функцій і взаємодіють один з одним для виконання єдиної задачі збору даних від безлічі датчиків. Якщо, наприклад, сенсорні вузли вимірюють будь-які фізичні параметри навколишнього середовища, то велика ймовірність того, що близько розташовані вузли зафіксують схожі значення, тому передавати до базової станції свідчення від кожного окремого вузла буде недоцільно. В результаті була запропонована нова концепція маршрутизації – маршрутизація, орієнтована на дані.

У маршрутизації орієнтованої на дані, акцент робиться на пошук і поширення інформації певного типу або описується певними атрибутами, на відміну від збору даних від окремих датчиків. Базова станція передає в певні області території покриття мережі запити, в яких описано якого роду інформація або події її цікавлять. Вузли, які знаходяться в цих областях, зібравши необхідну інформацію або зафіксувавши дані необхідної події, на локальному рівні обмінюються один з одним наявними даними для підвищення достовірності та зменшення надмірності, а потім передають базовій станції тільки одне повідомлення з узагальненими показниками. Такий принцип дозволяє значно скоротити обсяг мережевого трафіку і, як наслідок, зменшити енергоспоживання. Класичними прикладами реалізації

цього підходу є група протоколів Sensor Protocols for Information via Negotiation (SPIN) і Directed Diffusion (DD).

Очевидно, що області застосування методів маршрутизації на основі орієнтації на дані, обмежені мережами з типом трафіку «багато-до-одного», і їх можна розглядати як спеціалізовані рішення для мереж розподіленого збору даних з великою надмірністю сенсорних вузлів. Тому їх недоцільно застосовувати в тимчасових мережах з типом трафіку «багато-до-багатьох», а також в системах, в яких важливі повідомлення від кожного вузла.

3.2.1. Протокол SPIN

SPIN – сімейство протоколів, що забезпечує доставку даних на основі процедури «переговорів». Відноситься до методів маршрутизації з одноранговими вузлами без гарантованої доставки повідомлень, виконується облік витрат енергії вузлів. Добре підходить для БСМ з динамічною топологією з мобільними вузлами. Використовується адаптивний варіант простої техніки лавинного розповсюдження (flooding), що значно підвищує ефективність маршрутизації в порівнянні з прототипом. При цьому, щоб уникнути непотрібних повідомлень, перед передачею даних проходить опитування між сусідніми вузлами. Повідомлення від кожного вузла розповсюджуються по всій мережі, що дозволяє досить простим способом отримати інформацію з будь-якого вузла за запитом з негайною її доставкою.

Механізм взаємодії базується на трьох типах повідомлень:

- ADV – для інформування про наявність нових даних у вузлі, містить їх опис у вигляді так званих «метаданих»;
- REQ – для запиту даних;
- DATA – для пересилання безпосередньо самих даних.

Принцип алгоритму полягає в наступних трьох фазах (рис 3.4). При отриманні нових даних вузол проводить попередні «переговори» з сусідніми

вузлами, посилаючи повідомлення ADV і чекаючи від них повідомлення-відповіді REQ. Після отримання ADV повідомлення, вузол-приймач перевіряє, чи були раніше вже отримані ці дані. Якщо ні, то вузол-приймач відповідає з проханням про передачу повідомлення з даними, використовуючи службове повідомлення REQ. Отримавши запит на передачу REQ, вузол-передавач відсилає зібрані дані – повідомлення DATA. Кожен сусід, отримавши нові дані, аналогічним чином веде переговори відповідно зі своїми сусідами і розсилає ці дані тим з них, які в цих даних зацікавлені, тобто які їх ще не отримали і мають можливість транслювати їх далі по мережі. Як зображено на рис. 3.5, тільки вузли, які ще не мають копію оголошених даних відповідають на ADV повідомлення. Більш того, після отримання DATA повідомлення від А, вузли В і D, можуть об'єднати ці дані зі своїми і розіслати зібрані дані своїм сусідам.

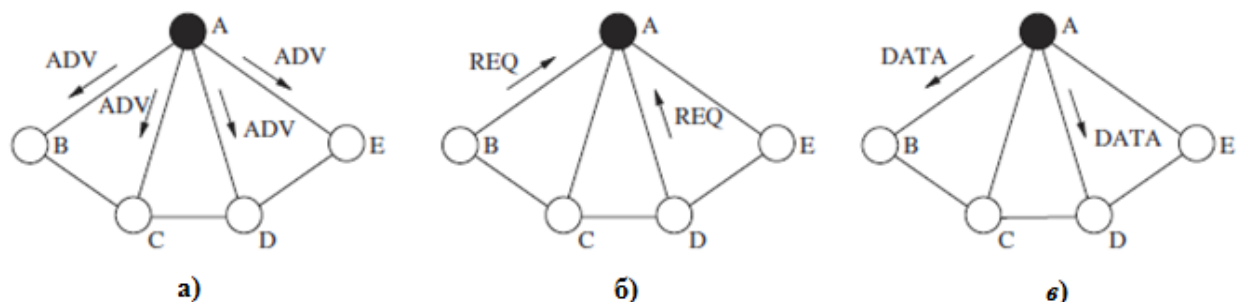


Рисунок 3.5 Алгоритм протоколу SPIN: а – фаза інформування, б – фаза запиту, в – передача інформації

Метадані служать свого роду ідентифікатором реальних даних, ґрунтуючись на яких, вузол має можливість уникнути повторної передачі одних і тих же даних. Таким чином, знижується надмірність мережевого трафіку і зменшується даремна витрата енергії в порівнянні з класичним методом лавинної маршрутизації.

Основні протоколи, які входять до сімейства SPIN:

- SPIN-PP;
- SPIN-BC;

- SPIN-EC;
- SPIN-RL.

Протокол SPIN-EC – різновид базового протоколу SPIN-PP. Основна відмінність полягає в введенні контролю за енерговитратами. Поки всі вузли мають достатню енергію, вони беруть участь в трьохфазовому алгоритмі SPIN-PP. Однак, як тільки енергія вузла наближається до певного низького рівня, він зменшує свою активну участь в роботі протоколу. Тобто, вузол буде приймати участь у всіх трьох фазах, тільки якщо є гарантія, що це не призведе до виснаження джерела живлення. Тому, вузол відповідає на оголошення тільки, якщо він володіє достатньою енергією, щоб здійснити запит і отримати запитані дані. Крім того, вузол ініціює обмін даними зі своїми сусідами, тільки якщо він вважає що зможе закінчити протокол, навіть якщо всі сусіди запросять копію даних.

Протокол SPIN-BC покращує попередні протоколи, використовуючи характеристики ширококомовної передачі. У ширококомовній передачі, кожне повідомлення від відправника буде отримано усіма вузлами в радіодіапазоні відправника. SPIN-BC використовує дешевий, спосіб зв'язку «один-до-багатьох» і вузли можуть координувати та використовувати свої ресурси більш ефективно, тому що вони можуть перехопити всі операції в своєму радіодіапазоні. SPIN-BC використовує три фази обміну ADV, REQ і DATA повідомлень, але з трьома важливими відмінностями в порівнянні з алгоритмом SPIN-PP:

- всі повідомлення направлені на ширококомовну передачу, тобто, всі вузли в межах діапазону передачі відправника отримують копію цього повідомлення.
- після отримання повідомлення ADV, вузол перевіряє, чи хоче він отримати копію прорекламованих даних, якщо так, він встановлює випадковий таймер. Тільки після того як таймер закінчиться, вузол подає повідомлення REQ на ширококомовну адресу. Якщо вузол зчитає повідомлення REQ до того, як його власний таймер закінчиться, вузол-

відправник скасовує свій таймер і не відправляє своє власне REQ повідомлення. Випадковий таймер необхідний, щоб уникнути колізій REQ повідомлень від різних сусідів і дозволити вузлам уникнути відправки REQ, коли інший вузол вже відправив REQ.

- вузол-відправник передає запит отримання даних на широкомовна адресу тільки один раз, тобто, він буде ігнорувати дублікати REQ повідомлень для одних і тих же даних.

Остаточним варіантом є протокол SPIN-RL. Він базується на модифікованій версії SPIN-BC, з асиметричним зв'язком. Спочатку кожен вузол продовжує приймати REQ повідомлення і якщо він не отримує відповідне DATA повідомлення в певний проміжок часу, він передбачає, що повідомлення REQ або повідомлення DATA не надходило. В цьому випадку вузол подає повторний запит на отримання даних, передаючи повідомлення REQ, вказавши ідентифікатор випадково обраного вузла серед вузлів, які раніше рекламували ці дані в заголовку повідомлення. Крім того, SPIN-RL обмежує частоту відправлення DATA повідомлень. Тобто, як тільки вузол відправляє DATA повідомлення, він буде чекати протягом заданого часу, перш ніж відповісти на будь-які інші запити для тих же даних.

Ключовою перевагою протоколів SPIN є їх простота. Оскільки вузлу потрібно знати тільки своїх найближчих сусідів, які знаходяться на відстані одного стрибка. Цей протокол був розроблений, щоб зменшити втрати в середовищі з симетричними лініями зв'язку. Вузли можуть компенсувати втрати повідомлень ADV, періодично повторно передаючи запит на наявність даних, а для втрачених повідомлень REQ і повідомлень DATA здійснюють повторний запит даних, якщо такі дані не надходять через певні інтервали часу.

3.2.2. Протокол Directed Diffusion

Directed Diffusion – це один із протоколів, орієнтованих на дані. Головна ідея алгоритму DD, полягає в тому, що вузли запитують дані, відправляючи запит в іменованих повідомленнях [10]. Таке поширення інтересів встановлює градієнти всередині мережі, які використовуються для направлення сенсорних даних до одержувача, а проміжні вузли вздовж шляхів передачі даних можуть об'єднувати дані з різних джерел, щоб усунути надмірність і зменшити кількість передач.

Протокол DD не залежить від глобально діючих ідентифікаторів вузлів, замість цього використовується пара атрибут-значення, щоб описати задачу зчитування і управління процесом маршрутизації.

Кожен вузол підтримує локальний кеш запитів, в якому зберігаються записи про стан активних запитів. Для кожного запиту в кеші зберігаються додаткові дані, такі як тимчасова мітка, тобто значення таймера, що фіксує момент отримання запиту, градієнти на кожен сусідній вузол з вказівкою на швидкість обміну, тривалості існування і час життя. Градієнт фактично вказує можливий напрямок маршрутизації і може мати свою вагу, яка використовується в залежності від конкретних вимог щодо маршрутизації. Описана структура кеша запитів є основою для оптимальної маршрутизації.

На рисунку 3.6 зображено процес передачі повідомлень за допомогою протоколу DD. Вузол-приймач спочатку розповсюджує повідомлення про готовність прийняти дані. Між ним та сусідніми вузлами встановлюються градієнти для прокладання маршруту. Алгоритм передачі повідомлень встановлює певні правила, спрямовані на підтримання вмісту кеша в актуальному стані в процесі обробки одержаних запитів. Запити від центрального вузла періодично повторюються, з метою забезпечення надійності доставки і працездатності загальної схеми маршрутизації при відмовах в мережі.

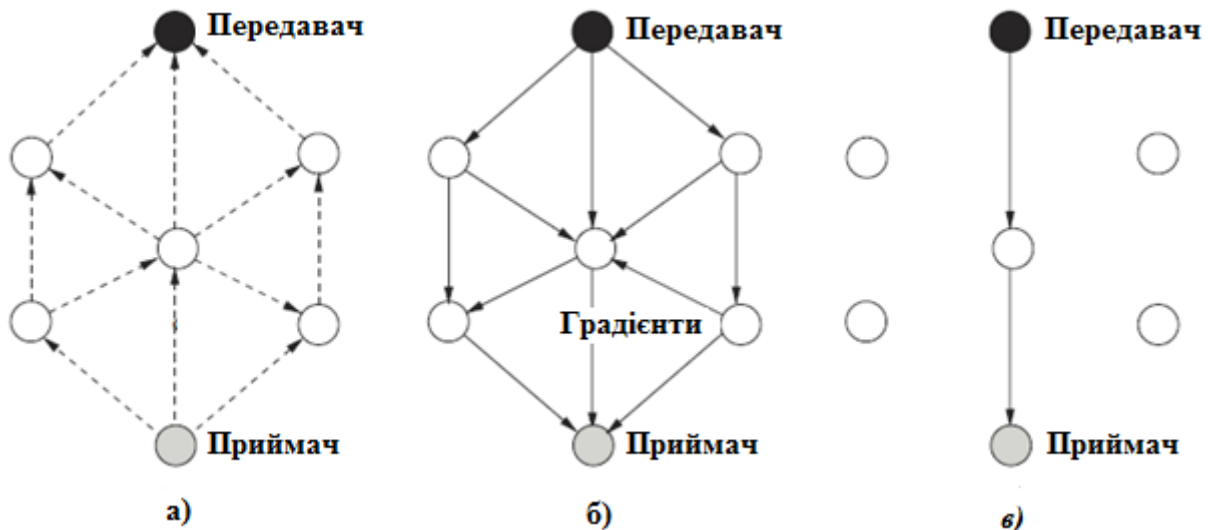


Рисунок 3.6 Протокол DD: а – розповсюдження зацікавленості,
б – встановлення градієнтів, в – передача даних

При виникненні подій, які цікавили центральний вузол, починається зворотна маршрутизація повідомлень від джерела інформації до центрального вузла. Повідомлення можуть надходити з певною частотою, протягом всього часу існування запиту. Для даних, що передаються у відповідь, використовуються ті ж пари атрибут-значеннь, що і у запиті. Також для зворотної трансляції використовуються ті ж встановленні відомості, що показують для кожного конкретного вузла оптимальний напрямок передачі. В результаті для потоку повідомлень вибирається найбільш підходящий шлях з точки зору заданих критеріїв маршрутизації (наприклад, найкоротший або з мінімізацією енергоспоживання).

Протокол DD відрізняється від SPIN тим, що запити відправляються на вимогу приймачів, а не рекламуються самими джерелами, як в SPIN. Весь зв'язок здійснюється від сусіда до сусіда, усуваючи необхідність у схемі адресації, і дозволяє кожному вузлу виконувати агрегацію і кешування сенсорних даних, що може сприяти зменшенню споживання енергії.

Недолік цього протоколу полягає в тому, що його недоцільно застосовувати в системах, де потрібна безперервна передача даних.

3.2.3. Протокол Rumor Routing

Rumor routing є варіацією попереднього алгоритму DD. Оптимізує схему маршрутизації для тих мереж, в яких кількість подій невелика, але величезна кількість запитів. В алгоритмі маршрутизації RR, кожен вузол підтримує список своїх сусідів і таблицю з інформацією про подію.

При виникненні подій, відомості про них заносяться в таблицю і генеруються спеціальні повідомлення, названі «агентами», які містять локальну інформацію про подію. Агент – довговічний пакет, який відправляється по мережі, щоб поширити інформацію про цю подію та інші події, які виникають уздовж його шляху до віддалених вузлів. При отриманні таких повідомлень віддалені вузли поповнюють свою таблицю подій і передають агента сусіднім вузлам до тих пір, поки він не вичерпає свій час життя TTL (time to live) [11].

Наприклад, на рис. 3.7, таблиця для вузла А містить інформацію про події E1 і E2, до прибуття агента із вузла Е. Коли агент прибуває до вузла А (за допомогою вузла G), А бачить, що прийняти дані від вузла Е можна за допомогою сусіда G, при цьому використовуючи коротший маршрут, ніж той який зараз зберігається в його таблиці. Тому вузол А оновлює свою таблицю з урахуванням інформації від агента Е.

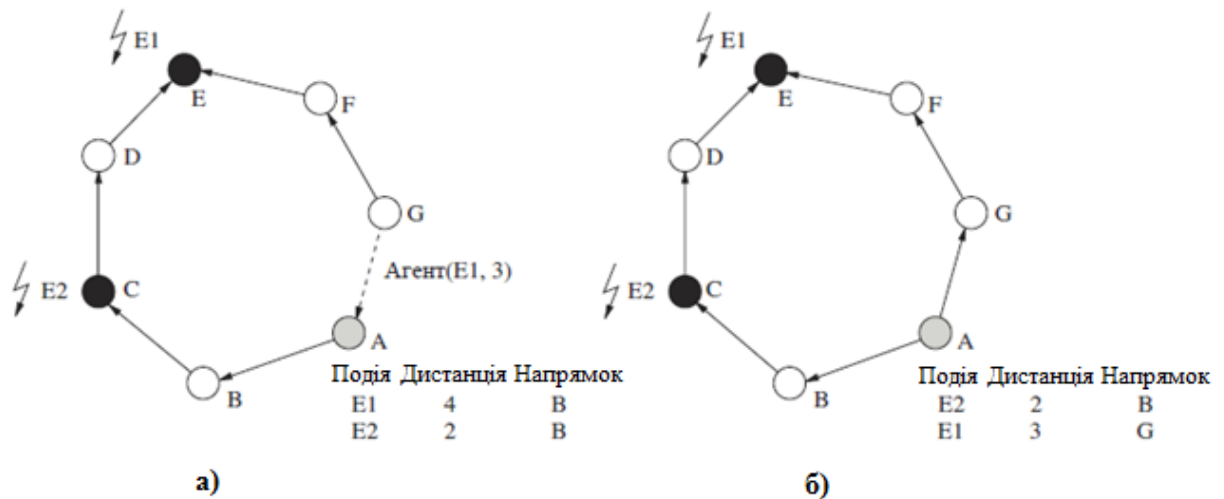


Рисунок 3.7 Протокол RR: а – маршрутизація до прибуття агента, б – після прибуття агента

Отже, коли вузол хоче подати запит, спрямований на певну подію, він спочатку перевіряє, чи є у нього маршрут до необхідного вузла. Якщо це так, то він пересилає запит сусідові, вказаному у таблиці події. Якщо відсутні відомості про маршрут, вибирається випадковий сусід і запит передається йому. Цей процес триває на кожному вузлі, причому повідомлення запиту містить список нещодавно задіяних вузлів, щоб уникнути повторної передачі. Хоча вузол може не знати напрямку до необхідного вузла, направляючи його до випадкового сусіда, запит може дістатися до вузла, який має доступ до інформації про бажану подію.

В цілому, спосіб маршрутизації RR намагається знайти компроміс між запитом і флудингом події. Хоч даний алгоритм маршрутизації використовує підхід на основі запитів, він намагається встановити зв'язок тільки з вузлами, які спостерігали конкретну подію, замість того щоб здійснювати флудинг по всій мережі. Крім того, розмір таблиці вузла зростає з числом подій, які він оброблює, тому, в мережах з великою кількістю подій, вартість зберігання і підтримки таких таблиць буде проблематичною.

3.2.4. Протокол GBR

Gradient-Based Routing (GBR) – спосіб маршрутизації за допомогою градієнтів. Ще один варіант алгоритму Directed Diffusion. Дана модифікація має ряд суттєвих відмінностей. У процесі поширення запиту з центрального вузла по всій мережі враховується кількість ретрансляцій від вузла до вузла (hops). У кожному вузлі обчислюється параметр, названий «висотою» вузла, який вказує на мінімально можливе число хопів в ланцюжку маршруту від даного вузла до центрального. Для кожного з сусідніх напрямків в вузлі позначається градієнт – різниця між висотою даного вузла і висотою його сусіда. Для маршрутизації вибирається напрям з найбільшим значенням градієнта. У тих випадках, коли градієнти для різних напрямків рівні, вибір здійснюється випадковим чином.

Наведена схема маршрутизації може доповнюватися правилами, які враховують економію енергії. Так, при падінні поточного рівня запасу енергії в вузлі нижче певного значення, вузол може автоматично збільшити свою «висоту» з метою зменшення навантаження з боку інших вузлів. Сусідні вузли аналізують, що використовувати для передачі повідомлень даний вузол стає все більш не вигідно, тому перестають використовувати цей напрямок, надаючи перевагу більш оптимальним маршрутам. Такий механізм забезпечує хорошу адаптивність алгоритму в плані балансування і оптимізації енергоживлення.

Крім того, алгоритм може забезпечувати балансування навантаження трафіку за допомогою аналізу вже існуючих маршрутів. Відповідно для нових потоків створюється маршрут таким чином, щоб в нього входили тільки ті вузли, котрі не беруть участі в інших маршрутах.

3.2.5. Протокол OLSR

Протокол Optimized Link State Routing (OLSR) є проактивним протоколом маршрутизації для безпроводових сенсорних мереж. Він відноситься до проактивних протоколів маршрутизації. Тобто, встановлення маршруту відбувається ще до того, як це необхідно. В будь-який існує готовий маршрут до вузла призначення. Принцип роботи протоколу полягає в зменшенні кількості широкомовних повідомлень в мережі за рахунок передачі цих повідомлень тільки через спеціальні вузли – багатоточкові ретранслятори (Multi-point Relays, MPR).

Ідея полягає в мінімізації розсилки широкомовних повідомлень за рахунок зниження пересилання надлишкових повідомлень в межах однієї області. Кожен вузол в мережі вибирає набір сусідніх вузлів серед найближчих в межах одного хопу, які відповідають за передачу його пакетів. Вони називаються багатоточковими ретрансляторами. Решта сусідніх вузлів, що не входять в кількість багато точкових ретрансляторів, читають і обробляють пакети даного вузла, але не передають їх далі. Вузли MPR для кожного центрального вузла вибираються таким чином, щоб вони покривали всі сусідні вузли в радіусі двох хопів від центрального вузла. Чим менше MPR, тим менше потрібно передавати керуючих повідомлень.

Протокол OLSR при маршрутизації пакетів, як і при передачі контрольних повідомлень, покладається на вибрані багатоточкові ретранслятори і обчислює маршрути до всіх вузлів призначення через ці вузли MPR. Вони стають проміжними вузлами на маршруті. Для цього кожен вузол підтримує інформацію про безліч сусідів, які обрали його як MPR. У протоколі OLSR використовується два типи керуючих повідомлень. Вузол періодично розсилає повідомлення «Hello», в яких оголошує, хто є його сусідами, а вузол може отримувати і детектувати сигнал від них. Повідомлення «Hello» розсилаються широкомовно всім сусіднім вузлам, але сусідні вузли в свою чергу не передають їх далі. Таким чином, повідомлення

«Hello» дозволяють кожному вузлу дізнаватися про своїх сусідів в радіусі двох хопів. На основі цієї інформації вузол і робить вибір багатоточкових ретрансляторів. Вони позначені особливим статусом в повідомленнях «Hello», тому кожен вузол також може дізнатися, хто вибрав його в якості багато точкового. Потім вузол записує адресу сусіднього вузла, список вузлів в межах двох хопів, з якими можна зв'язатися через даний вузол, відмітка, чи є даний вузол багатоточковим ретранслятором в таблицю інформації про найближчих сусідів. Кожен запис в таблиці має свій час актуальності. Принцип дії протоколу OLSR зображено на рис. 3.7.

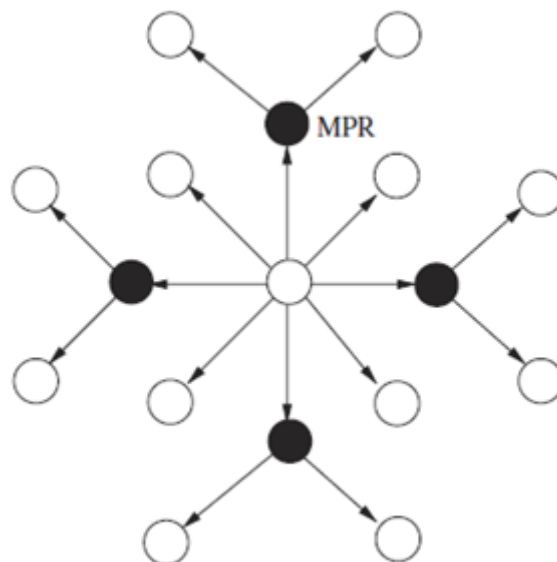


Рисунок 3.8 Протокол OLSR

Для підтримки маршрутизації пакетів кожен вузол ще розсилає керуючі повідомлення другого типу: Topology Control (TC). Повідомлення TC періодично розсилаються кожним вузлом, щоб оголосити, які вузли обрали даний вузол в якості MPR. Ця інформація допомагає вузлам будувати топологію мережі і таблиці маршрутизації. Інформація з таблиць маршрутизації дозволяє протоколу OLSR обчислювати маршрути передачі даних. При цьому єдиним показником оцінки маршруту є кількість проміжних вузлів.

Безумовною перевагою протоколу є миттєвість встановлення маршруту, оскільки кожен з вузлів завжди підтримує таблицю маршрутизації, де є маршрути до будь-якого вузла мережі. Однак це збільшує трафік, що генерується для передачі контрольних повідомлень, і підвищує вимоги до обсягу пам'яті вузлів.

Головним недоліком протоколу є той факт, що маршрути завжди проходять через вузли MPR, тому вони не завжди є найбільш оптимальні з точки зору завантаження. Крім того, при виході з ладу вузла на маршруті або іншому збої, проходить певний час, поки інші вузли дізнаються про цей факт через контрольні повідомлення.

3.2.6. Протокол AODV

Протокол Ad-hoc On-demand Distance Vector (AODV) – проактивний протоколом маршрутизації для безпроводових сенсорних мереж з встановленням з'єднання на вимогу. Процедура пошуку підходящого маршруту починається після запиту від центрального вузла. Маршрути підтримуються в таблиці маршрутизації до тих пір, поки вони використовуються [12].

AODV будує маршрути, використовуючи цикл "запит-відповідь". Коли програма створює запит на передачу даних до вузла, до якого ще немає маршруту в таблиці, вузол ширококомовно розсилає запит на пошук маршруту. Вузол, який отримав даний пакет, передає його далі ширококомовно, записуючи при цьому в свою таблицю маршрутизації шлях до вузла-передавача через вузол, від якого він отримав вихідний циркулярний запит. Таким чином, вузли передають один одному запит, поки він не досягне вузла призначення, попутно дізнаючись про маршрут до джерела. Якщо якийсь із проміжних вузлів вже знає про маршрут до вузла призначення, він

відправляє повідомлення-відповідь до вузла-передавача, щоб припинити пересилання запиту.

Як тільки вузол призначення отримує запит, він відправляє відповідь вузлу-передавачу. Оскільки всі проміжні вузли вже знають маршрут до джерела, вони просто пересилають пакет відповідно до записаним маршрутам. При цьому вони записують маршрут до вузла-призначення через вузол, від якого вони тільки що отримали даний пакет з відповіддю. Таким чином, при передачі запиту від вузла-передавача до вузла призначення і при передачі відповіді від вузла призначення до передавача, проміжні вузли попутно записують маршрути до них через сусідні вузли. Коли відповідь досягає вузла-передавача, всі проміжні маршрутизатори вже знають маршрут для передачі даних в обидві сторони. При цьому вузол-передавач може отримати безліч відповідей. Вибирається маршрут з найменшою кількістю проміжних вузлів.

Для зменшення обсягу контрольних повідомлень, протокол AODV використовує спеціальні лічильники. Кожен запит має послідовний номер, проміжні вузли використовують цей номер, щоб виявляти запити, які вони вже передали далі.

Обмін повідомленнями в протоколі AODV показаний на рис. 3.8. Для моніторингу та виявлення сусідніх вузлів відбувається розсилка повідомлень для запиту маршруту RREQ (Route Request). Після отримання відповідних повідомлень RREP (Route Response) від сусідніх вузлів, дані перенаправляється оптимальним маршрутом між джерелом і вузлом призначення.

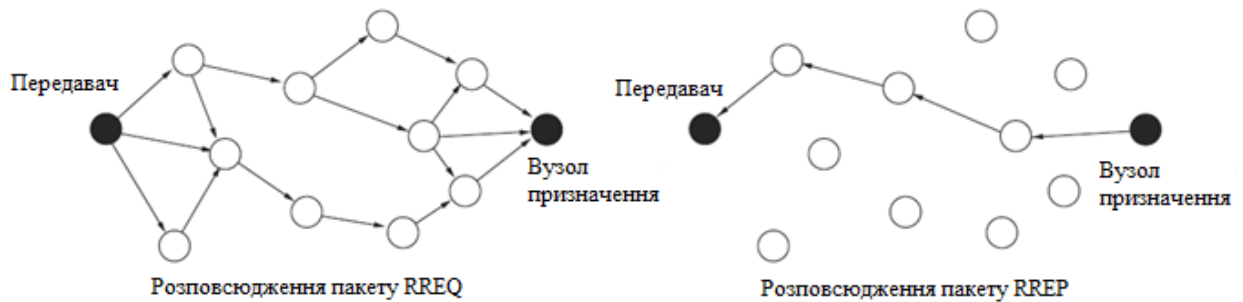


Рисунок 3.9 Обмін повідомлень в протоколі AODV

Для зменшення обсягу контрольних повідомлень, протокол AODV використовує спеціальні лічильники. Кожен запит має послідовний номер, проміжні вузли використовують цей номер, щоб виявляти запити, які вони вже передали далі.

Основною перевагою AODV є те, що він не створює додаткового трафіку, коли немає передачі даних. Однак протокол AODV має і ряд істотних недоліків. По-перше, якщо на маршруті стався збій (наприклад, вузол призначення більше не доступний), один з вузлів повинен ініціювати видалення маршруту з усіх таблиць маршрутизації проміжних вузлів. Це може привести до передачі застарілої маршрутної інформації. Крім того, для виявлення дублікатів контрольних повідомлень використовуються послідовні номери, тому кожен вузол повинен підтримувати спеціальну таблицю для зберігання безлічі нещодавно переглянутих номерів. Нарешті, оцінка оптимального маршруту відбувається по кількості проміжних вузлів.

3.3. Ієрархічні протоколи маршрутизації

Протоколи ієрархічної маршрутизації засновані на групуванні вузлів в кластери, з метою збільшення масштабованості і ефективності мережі. Основна ідея ієрархічної маршрутизації полягає в тому, що сенсорні вузли зв'язуються тільки безпосередньо з головним вузлом в своєму власному кластері – головою кластеру.

Ці головні вузли кластеру, можуть бути більш потужними і менш енерговитратними пристроями, ніж звичайні сенсорні вузли, відповідальні за транспортування сенсорних даних приймача. Цей підхід може істотно зменшити зв'язок і навантаження на сенсорні вузли, в той час як кластерні голови здатні витримати більшу завантаженість, ніж звичайні сенсорні вузли.

На рис. 3.10 зображено два варіанти кластеризації. Перший варіант – всі кластерні голови зв'язуються на пряму з вузлом приймачем. В другому варіанті, коли голови кластеру не пов'язуються на пряму з приймачем протокол маршрутизації заснований на кластері повинен також встановити мультихоп маршрути від всіх кластерних глав до приймача через голови сусідніх кластерів.

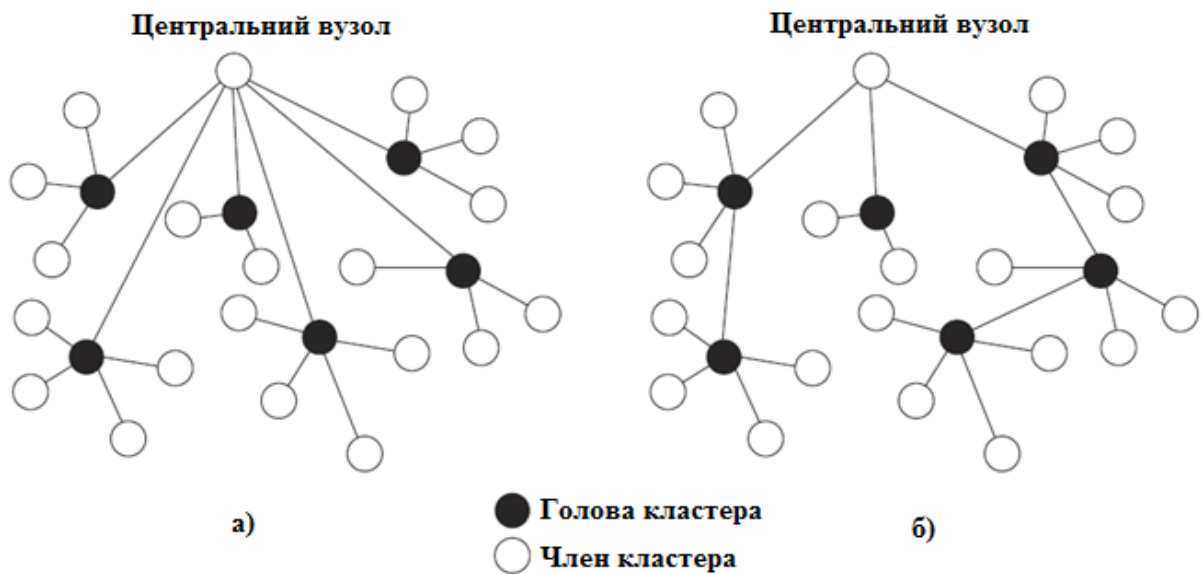


Рисунок 3.10 Способи кластеризації: а – з прямим зв'язком голів кластерів з центральним вузлом, б – мультихоповий зв'язок голів кластерів з центральним вузлом

Труднощі в роботі протоколів ієрархічної маршрутизації полягають в виборі кластерних голів, формування кластерів та адаптацію до динаміки мережі. У порівнянні з підходом плоскої маршрутизації, ієрархічні рішення можуть зменшити колізії в безпроводовому середовищі і полегшити робочий

цикл сенсорних вузлів для збільшення енергоефективності. Поділ на кластери може також полегшити процес маршрутизації, але може спричинити появу довгих маршрутів. Крім цього, поділ на кластери також полегшує агрегацію сенсорних даних усередині мережі, тому що дані, які приходять від поруч розташованих сенсорів, найімовірніше поступають на одну і ту саму голову кластеру.

$$\text{МЧВЦ} = \text{ST} + \text{TT} + \text{RT} + \text{IPT} \quad (3.3.1)$$

де МЧВЦ – мінімальна тривалість виконання циклу;

ST – тривалість відправлення;

TT – тривалість передачі;

RT – тривалість отримання;

IPT – тривалість простою задачі.

Отже, ієрархічна маршрутизація підтримує всі типи трафіку в БСМ, але доставка пакетів виконується не по оптимальним маршрутам, а також мають місце додаткові витрати ресурсів на організацію ієрархічної структури мережі.

3.3.1. Протокол LEACH

Low-Energy Adaptive Clustering Hierarchy (LEACH) відноситься до класу ієрархічних методів маршрутизації з балансуванням енергетичного навантаження. Оскільки LEACH є базовим протоколом ієрархічної маршрутизації, в нього закладені основні принципи алгоритму ієрархічної маршрутизації [13].

$$t = \begin{cases} n \cdot t_r \cdot \text{int}_1, n < n_1 \\ n_1 \cdot t_r \cdot \text{int}_1 + (n - n_1) \cdot t_r \cdot \text{int}_2, n_1 < n < n_2 \\ n_1 \cdot t_r \cdot \text{int}_1 + (n_2 - n_1) \cdot t_r \cdot \text{int}_2 + (n - n_1 - n_2) \cdot t_r \cdot \text{int}_3, n_2 < n < n_3 \\ \dots \end{cases} \quad (3.3.1.1)$$

В цій формулі n_i - кількість хопів зони;

int_i - інтервал оновлення зони;

t_r - мінімальний інтервал оновлення.

Сусідні вузли, що знаходяться в одній зоні радіодоступу, об'єднуються в окрему комунікаційну групу – кластер. Кластер складається зі звичайних вузлів нижчого рангу (Cluster Members - CM) і одного вузла вищого рангу (Cluster Head - CH), що виконує функції маршрутизації для всіх інших вузлів кластера. Безліч головних вузлів утворюють основу комунікаційної інфраструктури мережі.

$$T(n) = \begin{cases} P \\ 1 - P * (r \bmod n), n \in G \\ 0, \notin G \end{cases} \quad (3.3.1.2)$$

де: r – поточний раунд;

P – ймовірність вибору головним вузлом;

n – вказівник на поточний вузол.

Якщо згенерований символ менше $T(n)$, то вузол стає головним вузлом.

Вузли, які були clusterhead в нульовому раунді, не можуть бути ними знову протягом раундів $1/P$. Після $1/P - 1$, $T = 1$. Після цього всі вузли можуть ставати головним вузлом. У фазі steady-state дані передаються базової станції. У порівнянні з фазою setup, steady-state більш тривала для того, щоб мінімізувати витрату енергії.

Взаємодія всередині кластера відбувається безпосередньо між звичайними вузлами і головою кластера, далі передачу повідомлень вузлам інших кластерів беруть на себе глави кластерів (рис. 3.10). Голова кластера вибирається випадковим чином і періодично виконується зміна головних вузлів для рівномірного розподілу навантаження по всіх вузлах мережі, оскільки функції маршрутизації, покладені на главу кластера, є причиною інтенсивної витрати енергії на ретрансляцію повідомлень від всіх вузлів кластера. Кількість голів кластерів зазвичай складає 5% від загальної кількості вузлів в кластері.

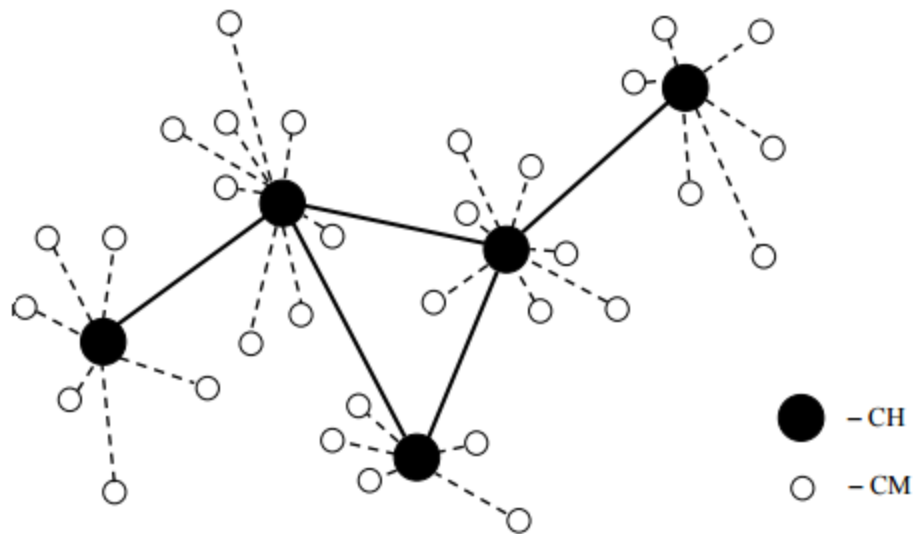


Рисунок 3.11 Протокол LEACH

Маршрутизація виконується в 2 етапи. Спочатку кінцеві вузли передають пакети головному вузлу свого кластера, а той уже виконує агрегацію даних і відправляє результат безпосередньо базовій станції, що дозволяє значно скоротити обсяг можливого трафіку. Додаткова економія енергії досягається завдяки використанню всередині кластера доступу з тимчасовим поділом каналів, який встановлює голова кластера. А для зменшення колізій між кластерами застосовується кодове розділення каналів.

Однак протокол LEACH має ряд недоліків. По-перше, передбачається, що обрані головні вузли кластера здатні безпосередньо передавати дані базовій станції, але в більшості практичних ситуацій це неможливо через обмежену відстань безпроводових каналів зв'язку. По-друге, випадковий вибір головних вузлів не гарантує їх рівномірного розподілу по мережі, тому можлива поява кінцевих вузлів, в радіусі яких немає жодного головного вузла кластера. По-третє, накладні витрати на періодичне формування кластерів зменшують ефект від використання методів зниження енергоспоживання.

3.3.2. Протокол PEGASIS

Power-Efficient Gathering in Sensor Information Systems (PEGASIS) – протокол енергоефективного збору інформації в безпроводових сенсорних мережах. Оптимізована версія протоколу LEACH на основі ланцюгів.

Основна відмінність від протоколу LEACH полягає в тому, що для кожного вузла відбувається обмін пакетів з близькими сусідами і вони по черзі несуть відповідальність за передачу пакетів базових станцій [14]. З цією метою, вузли організовуються в ланцюг, наприклад, використовуючи жадібні алгоритми, що ініціюються конкретним вузлом або базовою станцією, яка потім розсилає інформацію ланцюга до решти вузлів мережі. Поки дані передаються по ланцюгу, вони можуть бути об'єднані з іншими даними для подальшої передачі на базову станцію. Протокол передбачає, що всі вузли можуть зв'язуватись з базовою станцією.

Застосування протоколу PEGASIS дозволяє суттєво збільшити час життя кожного вузла. Термін роботи мережі із використанням протоколу PEGASIS в два рази більший, ніж у аналогічній мережі з використанням протоколу LEACH [15]. PEGASIS досягає високої енергетичної ефективності, за рахунок того, що кожен вузол взаємодіє тільки з своїм найближчим сусідом, а іноді використовується як передавач до базової станції. Тим самим дозволяючи вузлу зменшити свою потужність передачі до мінімально необхідної, щоб досягти своїх сусідів в ланцюзі.

Недоліком в PEGASIS є те, що пакети можуть передаватись із суттєвими затримками, особливо, якщо вони передаються від далеких вузлів в ланцюзі.

3.4. Маршрутизація на основі місцеположення

Принцип маршрутизації на основі інформації про місцеположення полягає в використанні інформації про місцезнаходження вузлів в просторі для визначення маршруту передачі пакета. На кожному кроці процесу маршрутизації поточний вузол вибирає серед своїх сусідів вузол, який найближче розташований до вузла призначення, і передає йому пакет даних. Відстань між двома вузла ми обчислюється на основі їх координат за допомогою деякої метрики. Процес послідовної мінімізації відстані триває до тих пір, поки пакет не досягне вузла призначення або серед сусідів поточного вузла не буде вузла, який розташований ближче до вузла призначення.

У одноадресній маршрутизації, заснованій на локалізації, пакети відправляються безпосередньо в одне місце призначення, яке ідентифікується його розташуванням. Тобто, відправник повинен знати не тільки своє місце розташування, але також місце розташування місця призначення. Це місце розташування може бути отримано або за допомогою запиту (наприклад, можна подати запит з проханням отримати відповідь від місця призначення) або брокера місцезнаходження, тобто, служби, яка відображає ідентичність вузла до місця розташування. При ширококомунікаційній передачі або при підході багатадресної маршрутизації на основі відомостей про географічне місцеположення вузлів, один і той же пакет повинен бути відправлений в декількох напрямках. З метою мінімізувати споживання ресурсів за рахунок скорочення надлишкових зв'язків, багатадресні протоколи використовують відомості про розташування місця призначення [16].

Ідентичність сенсорного вузла, як правило, менш важлива, ніж його місце розташування, тобто, дані можуть передаватися всім вузлам, які розташовуються в межах певної географічної зони. Цей підхід може бути використаний для поширення запитів в конкретній зоні інтересів, замість того що б розповсюджувати їх по всій мережі, що значно знижує пропускну

здатність і енергетичні вимоги до вузлів. Після того, як пакет досягає бажаної зони, він повинен бути поширений серед усіх вузлів в цій мережі.

Головною перевагою алгоритмів маршрутизації на основі місцеположення є висока масштабованість, так як вузлам необхідно зберігати інформацію тільки про географічне розташування сусідніх вузлів в межах одного хопу і немає необхідності підтримувати таблиці маршрутизації або встановлювати маршрут між усіма вузлами., тому службові витрати не залежать від загальної кількості вузлів в мережі. Саме тому принцип геометричної маршрутизації розглядається як найбільш перспективний спосіб реалізації маршрутизації типу «багато-до-багатьох» в БСМ, які відрізняються від інших типів мереж великими масштабами при дуже обмежених ресурсах вузлів.

Однак важливо відзначити, що при географічній маршрутизації не гарантована сувора оптимальність маршрутів, але на практиці знайдені маршрути дуже близькі до найкращих, і одним з критеріїв ефективності є ступінь цієї близькості між вузлами.

3.4.1. Протокол GAF

Протокол Geographic Adaptive Fidelity (GAF) є ще одним прикладом енергоефективного протоколу маршрутизації заснованого на визначенні місця розташування. Але в першу чергу він призначений для мереж з мобільними вузлами.

Область мережі ділиться на віртуальну сітку, де тільки один пристрій в кожному квадраті сітки служить в якості передавального вузла в будь-який момент часу. Цей вузол надалі відповідає за передачу даних до базової станції, в той час як всі інші вузли можуть перейти в сплячий режим. Крім того, протокол GAF передбачає, що для двох суміжних квадратів А і В, всі вузли в А можуть зв'язатись з усіма вузлами в квадраті В і навпаки (рис.

3.11). Розміри сітки і квадрата можуть бути заздалегідь задані, дозволяючи кожному вузлу (за умови, що він знає своє місце розташування) визначити квадрат до якого він належить [17]. Це означає, що у більшості вузлів в мережі будуть сусіди у всіх чотирьох напрямках.

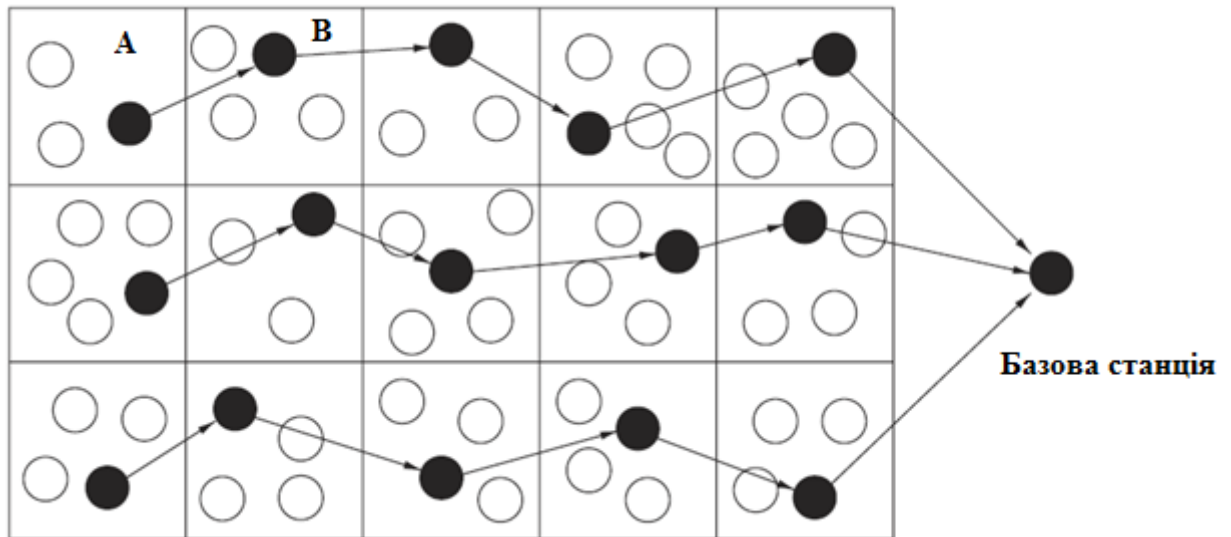


Рисунок 3.12 Принцип віртуальної сітки в протоколі GAF

Вузли в GAF здійснюють передачу в трьох різних станах. Спочатку, кожен вузол входить в стан виявлення, де він прослуховує повідомлення від інших вузлів в своєму квадраті. Він також встановлює таймер на певний час, і як тільки таймер спрацює, вузол передає повідомлення про виявлення та входить в активний стан. Вузол використовує інший таймер, щоб повторно увійти в стан виявлення, як тільки спрацює другий таймер. Поки вузол перебуває в активному стані, він періодично ретранслює своє повідомлення про входження у стан виявлення. Крім того, вузол може перейти в сплячий режим, якщо він визначить, що ще якийсь інший вузол може впоратися з передачею пакетів. Це досягається за допомогою використання процедури узгодження, наприклад, на основі очікуваного терміну служби вузла. Вузли в активному стані виграють процес узгодження, над вузлами, які знаходяться в

режимі виявлення. Вузли, що входять в сплячий режим періодично повторно переходять у стан виявлення, щоб повторити процес передачі даних.

$$C(r) = \sum_{i=0}^{k-1} (p(u_i, u_{i+1}) + c) \quad (3.4.1.1)$$

де $u = u_0$;

$p(u, v) = t.d(u, v)^n$ – потужність, яка необхідна для передачі даних з використанням цього протоколу маршрутизації

Отже, алгоритм GAF дозволяє скорочувати витрати енергоспоживання мережі за рахунок можливості тимчасового переходу вузлів у сплячий режим.

3.4.2. Протокол GEAR

Geographic and Energy Aware Routing (GEAR) – протокол географічної і енергозберігаючої маршрутизації є прикладом протоколу орієнтованого на місцеположення, де передача пакетів відбувається лише в межах конкретної цільової зони.

Протокол GEAR розбиває процес передачі даних на дві фази:

- пакети передаються до цільової області, використовуючи географічний алгоритм і алгоритм енергоефективного вибору сусіднього вузла. Тобто вибирається сусідній вузол, що має найбільшу енергію;
- пакети розсилаються серед вузлів в цільовій області, використовуючи алгоритм рекурсивно-географічної передачі.

Як тільки пакет добереться до цільової зони, можна використовувати просту схему флудингу з придушенням дублікатів, що б поширити пакети серед всіх вузлів в зоні. Проте, через значні енерговитрати та неефективність флудингу, GEAR покладається на процес рекурсивно-географічної передачі, зображена на рис. 3.13 [18]. Припустимо що цільова зона – це великий прямокутник і вузол S отримав пакет для цільової зони. Тоді, S створює

чотири нові копії, пов'язані з чотирма меншими підобластями великої цільової зони. Для кожної підобласті, GEAR повторює процес передачі і поділу до тих пір поки пакет не досягне вузла, який є єдиним у своїй підобласті.

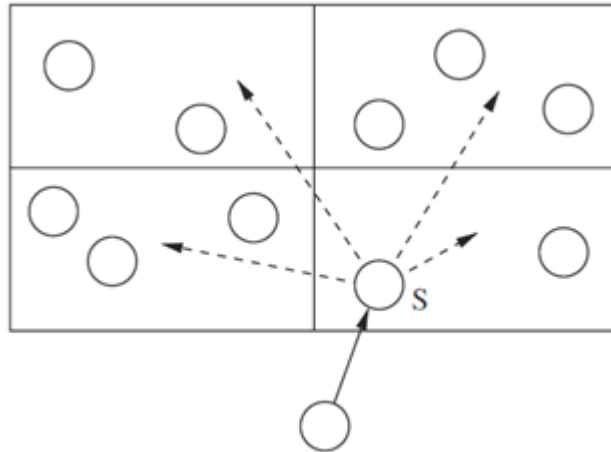


Рисунок 3.13 Процес рекурсивної маршрутизації в протоколі GEAR

Перевага протоколу GEAR полягає в тому, щоб обмежити кількість запитів, використовуючи запити з урахуванням певного регіону, а не розсилати запити всій мережі, що сприяє підвищенню економії енергії.

3.5. Порівняльний аналіз протоколів маршрутизації для БСМ

Були розглянуті найвідоміші методи маршрутизації для безпроводових сенсорних мереж.

Враховуючи особливості розгортання та призначення мереж, кожен із протоколів буде оптимальним для тої, чи іншої мережі. Тому основними параметрами при порівнянні слугують показники енергоефективності та якості обслуговування. Розробники методів маршрутизації для безпроводових сенсорних мереж закладають можливості для підвищення деяких показників якості обслуговування, щоб не просто забезпечувати доставку пакетів, але і робити це ефективно з точки зору якості

обслуговування. Проте слід пам'ятати, що зменшення витрат енергії, спричиняє погіршення інших показників та накладає обмеження на функціональність мережі [19].

Отже рішення щодо вибору способу маршрутизації є унікальним для кожної окремої мережі.

Основні особливості підсумовані в таблиці 3.1 і таблиці 3.2.

Таблиця 3.1

Назва протоколу	Класифікація топології	Мобільність вузлів	Масштабованість мережі
SPIN	плоска	можлива	обмежена
DD	плоска	обмежена	обмежена
RR	плоска	дуже обмежена	велика
GBR	плоска	обмежена	обмежена
OLSR	плоска	дуже обмежена	низька
AODV	плоска	дуже обмежена	низька
LEACH	ієрархічна	фіксована БС	велика
PEGASIS	ієрархічна	фіксована БС	велика
GAF	географічна	обмежена	велика
GEAR	географічна	обмежена	обмежена

Таблиця 3.2

Назва протоколу	Використання енергії	Агрегація даних	Multipath	На основі запитів
SPIN	середнє	так	так	так
DD	середнє	так	так	так
RR	неефективне	так	ні	так
GBR	неефективне	так	ні	так
OLSR	середнє	так	ні	ні

AODV	середнє	так	ні	ні
LEACH	обмежене	так	ні	ні
PEGASIS	обмежене	ні	ні	ні
GAF	обмежене	ні	ні	ні
GEAR	обмежене	ні	ні	ні

Висновки до розділу 3

В даному розділі була розглянута класифікація протоколів маршрутизації в безпроводових сенсорних мережах за трьома основними критеріями:

- способом організації мережі;
- способом дослідження маршруту;
- принципом роботи протоколу.

Була проведена класифікація протоколів за способом організації мережі:

- протоколи для мереж з плоскою топологією;
- протоколи для мереж з ієрархічною топологією;
- протоколи для мереж з маршрутизацією на основі місцезнаходження вузлів.

Було детально проаналізовано принцип роботи наступних протоколів: SPIN, DD, RR, GBR, OLSR, AODV, LEACH, PEGASIS, GAF, GEAR. Розглянуті особливості їх використання в залежності від призначення мережі. Проведено порівняльний аналіз протоколів за різноманітними показниками, на основі якого можна зробити висновок, що кожен із протоколів буде оптимальним лише для певної мережі, в той час, як для мережі з іншою кількістю та розміщенням вузлів, використання цього протоколу є неефективним або взагалі неможливим. Проте, для багатьох протоколів існують оптимізовані версії з меншими енерговитратами, але і гіршими

показниками якості обслуговування. Але якщо при функціонування мережі не накладається жорстких вимог до якості обслуговування, доцільніше використовувати енергоефективні протоколи маршрутизації.

4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

На етапі розробки програмного забезпечення необхідно реалізувати алгоритм вибору наступного вузла при передачі даних у безпроводовій сенсорній мережі, базуючих лише на інформації про стан найближчих до нього вузлів та вектор-напрямок до кінцевого вузла отримувача. При цьому важливою особливістю даного алгоритму вибору маршруту має стати постійне уявне розбиття мережі на зони та побудова окремої топології для кожної з підмереж. Такий підхід дозволяє використовувати найоптимальніший існуючий протокол маршрутизації для кожної з ділянок маршруту, які є незалежними одна від одної з урахуванням відстані між вузлами, топології ділянки та енергетичного запасу акумулятора вузла.

4.1. Програмні засоби використані при розробці

Для моделювання було використано програмне забезпечення з відкритим вихідним кодом. Програмне забезпечення повністю написане на мові програмування Python. Python представляє собою універсальну мову і має численну кількість різноманітних стандартних бібліотек та фреймворків, що дозволяє використовувати велику кількість допоміжних модулів та методів. Скрипт мови програмування Python був розроблений для автоматизації процесів і генерування випадкових даних, таких як координати вузлів, координати точки відправлення та точки отримання даних, енергетичний потенціал кожного з вузлів.

Зокрема для розробки програмного забезпечення були використані такі вузкопрофільні бібліотеки, як SciPy, NumPy, GPy, Matplotlib. Основне призначення цих бібліотек – це наукові та інженерні розрахунки, моделювання процесів та їх графічне відтворення в окремому вікні програми.

Так Matplotlib – бібліотека для візуалізації двохвимірних та трьохвимірних зображень та являється гнучким, конфігурованим пакетом. Важливою особливістю цього пакету є можливість візуалізувати анімоване зображення, що буде постійно оновлюватись на кожній ітерації, тобто, при кожному перерахунку маршруту на кожному з проміжних вузлів ретрансляторів.

Бібліотеки NumPy та SciPy дозволяють відтворити функції MATLAB у вигляді програмного коду на мові програмування Python.

4.2. Основні вимоги до алгоритму вибору протоколу маршрутизації

Головною вимогою до програмного забезпечення є ефективність його використання. Оскільки всі обчислення будуть відбуватись безпосередньо на кожному з вузлів безпроводової сенсорної мережі, це програмне забезпечення має також не витратити багато розрахункових потужностей вузлів, що в свою чергу може підвищити енерговитрати та збільшити час виконання усіх операцій через високе навантаження на систему. Тобто алгоритм має бути максимально простим, з мінімальною кількістю проміжних операцій та якомога меншою кількістю системних повідомлень між окремими сусідніми вузлами мережі. Також код має бути простим для виконання на малопотужному обладнанні, яким є вузли ретранслятори з поганими обчислювальними можливостями.

Для зменшення додаткових системних повідомлень між вузлами, дуже вигідною є реалізація, коли вся система є статичною. В такому разі можливо заздалегідь внести координати усіх вузлів у систему. Тоді відпаде необхідність виконувати додатковий обмін системними повідомленнями з метою отримання даних про координати сусідніх вузлів на кожній з ітерацій та перераховувати як всю мапу розташування вузлів, так і топологію окремих ділянок.

В даній реалізації програмного забезпечення загальний вигляд мапи розташування вузлів безпроводової сенсорної мережі будується лише при першій передачі повідомлення, оскільки прийнято, що вузли мають статичне положення. При динамічній зміні їх координат, буде використовуватись реалізація з постійними запитами до сусідів з метою отримання їх місцезнаходження. Для даного програмного забезпечення використовуються стандартні методи мови програмування Python для досягнення випадкового розміщення вузлів. Даний підхід використовується для незалежної оцінки якості роботи програмного забезпечення, відкидання людського фактору та зміни вигляду мережі при кожному новому запуску програмного забезпечення. Це дозволяє розмістити всі вузли у вигляді точок на всій площині програмного вікна.

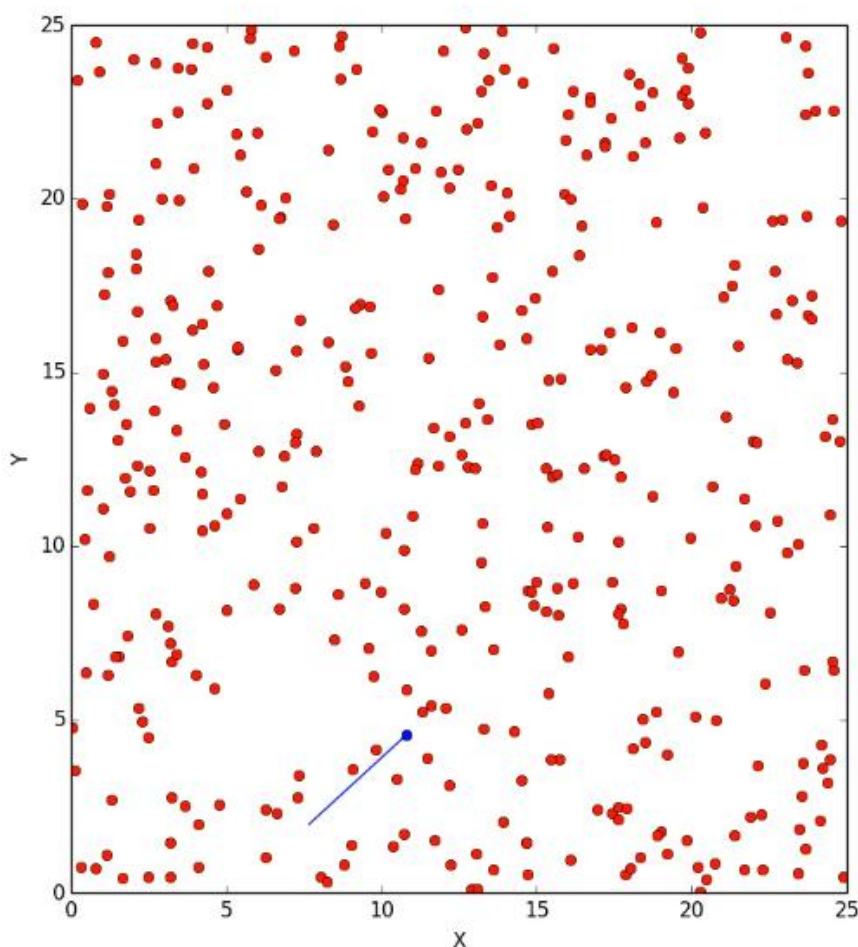


Рисунок 4.1 Приклад випадкового розташування вузлів

Такі параметри як зміна кількості вузлів та зміна розмірів ділянки, яка буде впливати на густину розташування вузлів, доступні для редагування перед кожним запуском програмного забезпечення. Це зроблено для гнучкості та варіативності програмного забезпечення.

Важливу роль відіграє принцип розбиття мережі на окремі ділянки та побудова локальної топології на кожній ітерації передачі даних. При цьому активною ділянкою вважається ділянка мережі, яка складається з вузлів у радіусі двох хопів від активного вузла ретранслятора. Інформацію про сусідні вузли, активний вузол отримує шляхом запитів до сусідніх вузлів. Відповідь на цей запит, який інші вузли ретранслятори відправляють активному містить такі відомості:

- рівень заряду батареї живлення;
- координати вузла.

Координати вузла дають змогу розрахувати вектор-відстань між вузлами та побудувати топологію ділянки.

Найважливішою частиною програмного забезпечення є порівняння розрахованої топології активної ділянки мережі з еталонними топологіями для кожного з існуючих протоколів маршрутизації. Тут і постає необхідність у виборі параметрів, які можуть характеризувати протокол та їх чисельну оцінку. Оцінка буде проводитись по п'ятибальній шкалі, від нуля до п'яти, де п'ять – найвищий бал. Розрахована топологія мережі також буде оцінюватись, а згодом чисельна оцінка буде зіставлятись з еталонними показниками для кожного з протоколів маршрутизації. Наприклад, спочатку відбувається оцінка рівня заряду батареї. Якщо він є критично низьким, то одразу параметру рівня енергії присвоюється п'ять балів. Тобто цей показник є критично важливим і пріоритетним при виборі протоколу маршрутизації. Після визначення пріоритетного показника відбувається відсіювання протоколів маршрутизації, які не відповідають цьому показнику. Серед списку енергоефективних протоколів відбувається другий етап відсіювання, при якому визначається другий за пріоритетом показник і за таким самим

алгоритмом дій, серед списку вибирається найкращий протокол. Всього проводиться три етапи відсіювання за трьома основними показниками:

- енергоефективність;
- відстань між вузлами;
- топологія ділянки мережі.

Таким чином на кожному з вузлів ретрансляторів буде переглядатись топологія окремої ділянки мережі в радіусі двох вузлів по відношенню до активного вузла та прийматись нове незалежне рішення стосовно вибору протоколу маршрутизації для цієї невеликої проміжної ділянки, а отже і наступного вузла, в залежності від відповідності існуючих енергетичних показників, розташування вузлів та топології ділянки, до оптимальних еталонних моделей кожного з протоколів маршрутизації. На кожному хопі протокол маршрутизації буде переглядатись по вищеописаному алгоритму.

Така реалізація дозволить значно продовжити термін роботи великої безпроводової сенсорної мережі, де кожен з вузлів має різний заряд акумулятора. Адже якщо рівень заряду батареї живлення є низьким, то всі не енергоефективні протоколи будуть відсіянні ще на першому етапі вибору. Якщо ж рівень заряду акумулятора вузла не є критичним, то оптимальніше вибрати наступний вузол-отримувач з урахуванням відстані між вузлами та топології, що зменшить час передачі інформації. А параметр рівня заряду батареї живлення буде найменш пріоритетним показником, тобто буде враховуватись лише на третьому етапі вибору

Цей алгоритм вибору проміжних протоколів маршрутизації доцільно застосовувати в безпроводових мережах, які не мають чітке симетричне розташування вузлів, велику кількість ретрансляторів, та вже якийсь час були в роботі, що спричинило нерівномірну зміну заряду батареї на вузлах. Тобто є ризик, що певні вузли можуть вийти з робочого стану набагато раніше за інші.

Розрахунок маршруту на кожній з ітерації та його моделювання відбувається одночасно. Тобто всі зміни в режимі реального часу моделюються в вікні програми.

4.3. Розробка логіки прийняття рішень

Основні етапи роботи програми на кожній ітерації (під ітерацією слід розуміти новий хоп та зміна активного вузла):

- активний вузол опитує сусідів з метою отримати інформацію про їх рівень заряду батареї;
- активний вузол виконує розрахунок вектор-відстаней до сусідніх вузлів (при умові статичного положення вузлів, цей розрахунок проводиться лише одного разу і результати зберігаються на вузлі для їх подальшого використання без перерахунку);
- визначення пріоритетів показників;
- вибір протоколу маршрутизації;
- передача даних (активним вузлом стає вузол, який отримав дані на попередній ітерації).

Весь цей алгоритм повторюється на кожній ітерації, доки дані не дійдуть до кінцевого вузла-отримувача.

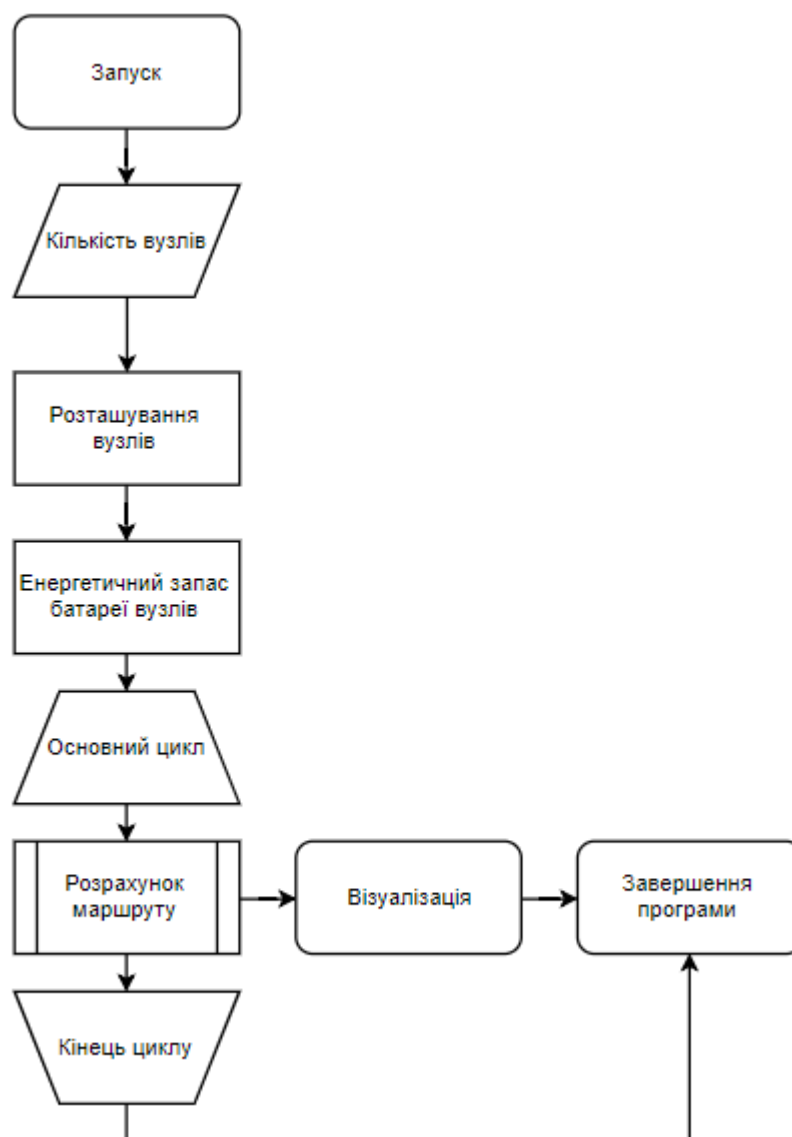


Рисунок 4.2 Блок-схема скрипта

Візуалізація відбувається паралельно з виконанням скрипта та обчисленням. Очевидно, що на потужних сучасних комп'ютерах можливо змоделювати та виконати весь скрипт для величезної кількості вузлів за лічені доли секунди. Але для наглядності зображення, результати виконання обчислень на кожній ітерації поступово передаються в анімоване програмне вікно візуалізації, див. додаток А.

4.4. Створення об'єктів та середовища моделювання

Для візуалізації було створено два програмних вікна моделювання:

- допоміжне програмне вікно розташування вузлів та ідеальний маршрут передачі. Ідеальним маршрутом називається маршрут, який є еталонним напрямком у якому відбувається передача даних. Це програмне вікно слугує для візуалізації розташування вузлів, положення яких є статичним і визначається бібліотекою Random мови програмування Python;
- реальний маршрут, який розраховується на кожній ітерації та зображується у вигляді ламаної лінії.

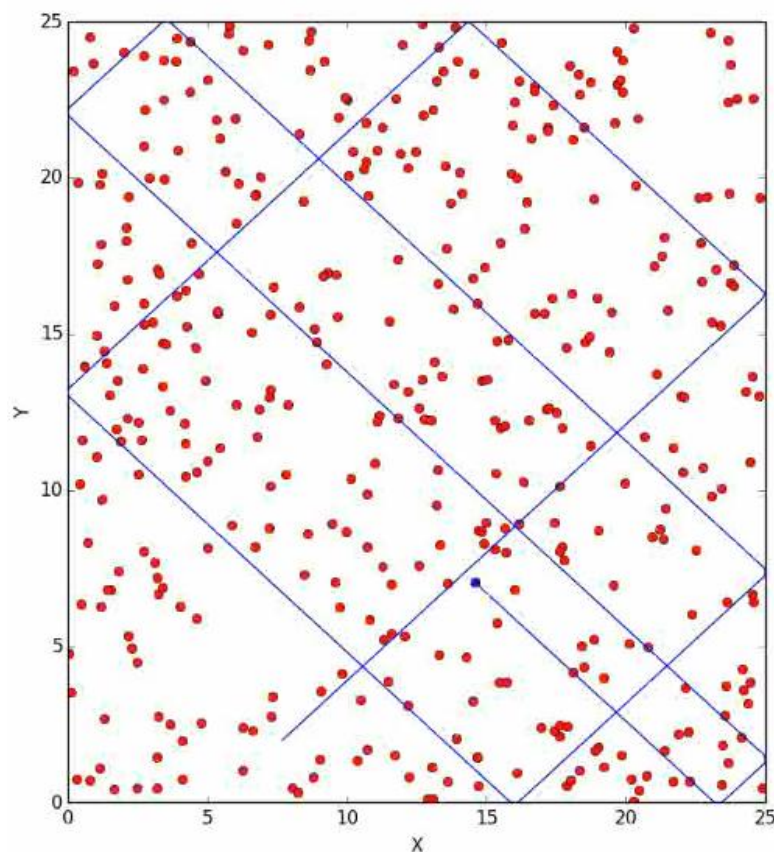


Рисунок 4.3 Програмне вікно розташування вузлів та ідеального маршруту

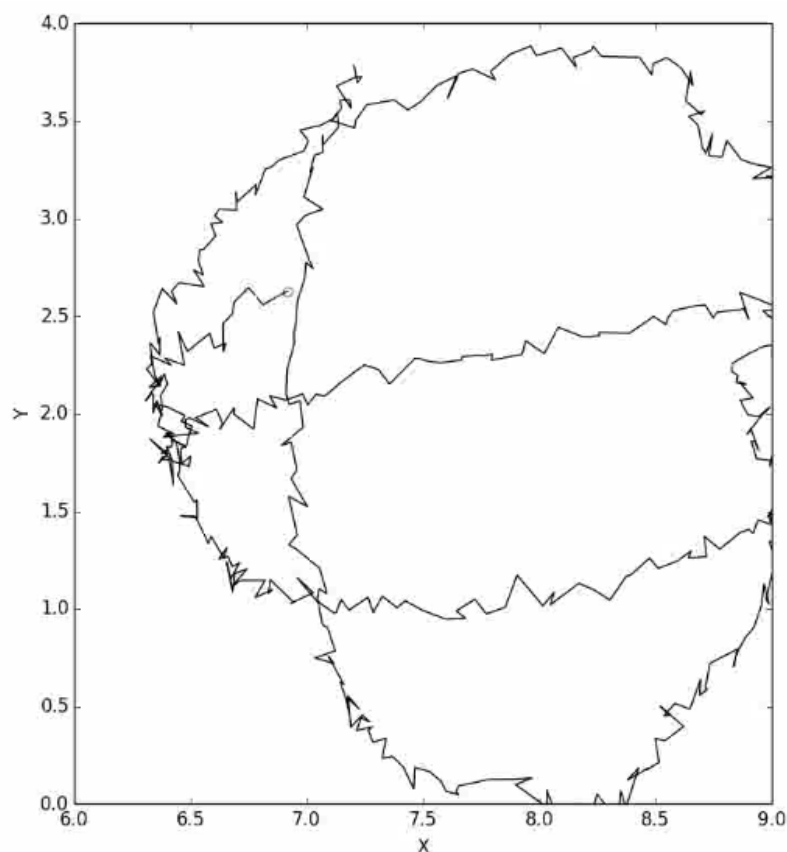


Рисунок 4.4 Реальний маршрут

Для коректної роботи скрипта моделювання та специфіки функціоналу бібліотеки Matplotlib мови програмування Python, масштаб та поворот візуалізації вікна моделювання, вираховується за першою вектор-відстанню першого хопу.

Можна виділити два основні об'єкти:

- вузли ретранслятори;
- рухома точка, яка демонструє напрям передачі інформації.

4.5. Розрахунок енерговитрат

Формула споживаної потужності для кінцевого пристрою має вигляд:

$$P_{ed} = \frac{P_f t_f + P_a t_a + P_s (t_c - t_f - t_a)}{t_c} \quad (4.4.1)$$

де t_c – тривалість одного циклу роботи пристрою [с];

P_f – середня потужність в процесі передачі даних і подальшого прийому підтвердження [Вт];

t_f – час витрачений на передачу даних та обмін підтвердження [с];

P_a – потужність, що споживається в режимі обробки даних (зчитування показань) [Вт];

t_a – сумарний час витрачений на зчитування показань з датчиків, їх обробку та підготовку до передачі [с];

P_s – споживана потужність в режимі очікування [Вт].

Вважається, що у кінцевого пристрою є достатньо часу на перехід сплячий режим. У формулі (4.4.1) величини P_a і P_s є постійними і визначаються особливостями конкретної апаратної реалізації пристрою. Час t_a залежить як від характеристик використовуваного процесора, датчиків, так і від складності та програмної реалізації алгоритму обробки даних [23].

Час t_f , а також потужність P_f залежать від використовуваного стандарту безпроводної передачі, розміру переданих даних, а також ймовірності помилок при передачі. Розглянемо останню залежність більш докладно.

На практиці передача по безпроводному каналу завжди пов'язана з можливими втратами як при передачі кадру з корисним навантаженням, так і при відправці підтвердження. В результаті пристрій змушений ініціювати одну або декілька повторних передач, що затримують перехід в сплячий режим. Введемо додатковий параметр p_e – ймовірність помилки при передачі [24].

Якщо вважати, що ймовірність помилки на (n – й передачі не залежить від кількості попередніх передач n , отримаємо, що ймовірність того, що передача пройде за один раз, буде дорівнює $(1 - p_e)$, за два рази – $p_e (1 - p_e)$. Враховуючи, що максимальна кількість передач як правило завжди програмно обмежена деяким значенням N , отримаємо формулу для

середнього часу, що витрачається на передачу кадру в умовах повітряних передач:

$$t_{real} = t_f [(1 - p_e) \sum_{i=1}^{N-1} i p_e^{N-1} + N p_e^{N-1}] \quad (4.4.2)$$

де t_f – час передачі даних з підтвердженням без урахування повторних передач;

N – максимальна кількість передач.

Параметр p_e залежить як від зовнішніх по відношенню до мережі чинників (відношення сигнал/шум та ін.), так в ряді випадків і від внутрішніх особливостей побудови мережі. Наприклад, при випадковому доступі до загального середовища передачі, ймовірність виникнення помилки буде залежати від кількості вузлів, що знаходяться в зоні покриття передавачів, від інтенсивності потоків даних на кожному з них. Підставляючи p_e замість P в формулу (4.4.2), можна оцінити час автономної роботи кінцевого пристрою.

Для оцінки точності в імітаційну модель був включений клас Validation. Оцінка точності здійснюється в два етапи. Перший етап включає в себе перевірку нормальності розподілу отриманих даних за допомогою критерію Колмогорова. Згідно з цим критерієм міра відхилення емпіричної функції розподілу $F_n^*(x)$ від гіпотетичної функції розподілу $F(x)$ визначається таким чином:

$$D_n = D\{F_n^*, F\} = \sup_x |F_n^*(x) - F(x)| \quad (4.4.3)$$

На другому етапі обчислюється інтервал довіри для отриманих даних. Для цього використовується вибіркова дисперсія для вибірки (X_1, X_2, \dots, X_n) :

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2 \quad (4.4.4)$$

Тоді значення математичного очікування отриманого значення лежить в інтервалі з кінцями в точках і процентилі розподілу Стюдента, можуть

бути отримані з їх таблиць. Оцінка точності проводиться безпосередньо в процесі імітаційного моделювання.

4.6. Результати моделювання

Завдяки можливості змінювати кількість вузлів, можливо провести моделювання маршрутизації безпроводової сенсорної мережі з надзвичайно великою щільністю вузлів. В такому випадку критичний рівень заряду батареї живлення окремих вузлів буде не так суттєво впливати на остаточний маршрут і відхилення змодельованого маршруту будуть мінімальними. При цьому основним пріоритетним параметром виступає вектор-відстань між вузлами і перш за все відбувається розрахунок топології ділянки мережі, використовуються звичайні не енергоефективні методи маршрутизації.

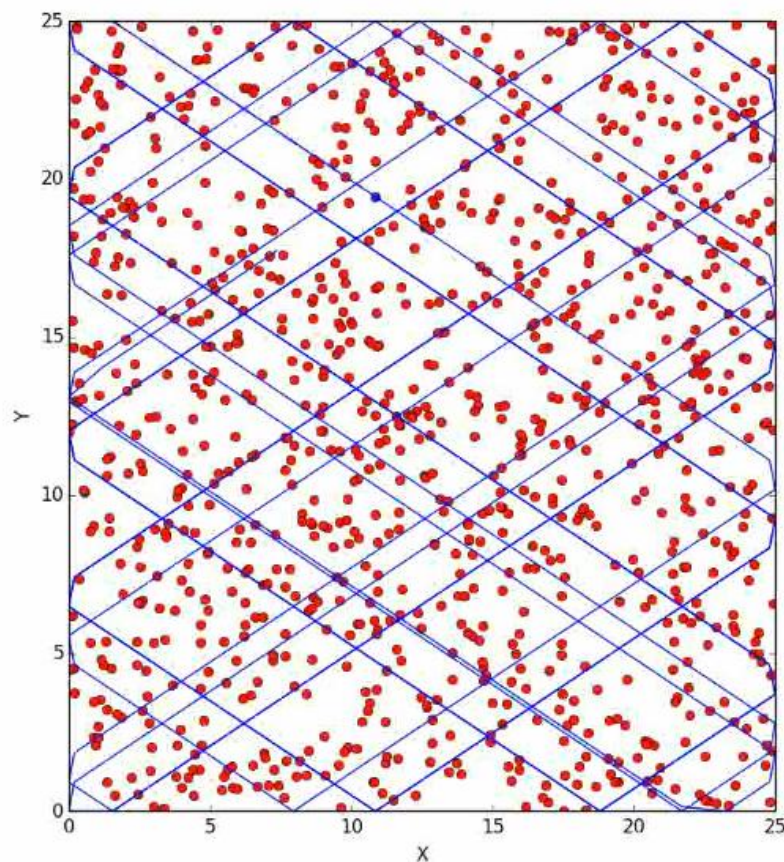


Рисунок 4.5 Приклад мережі з великою кількістю вузлів

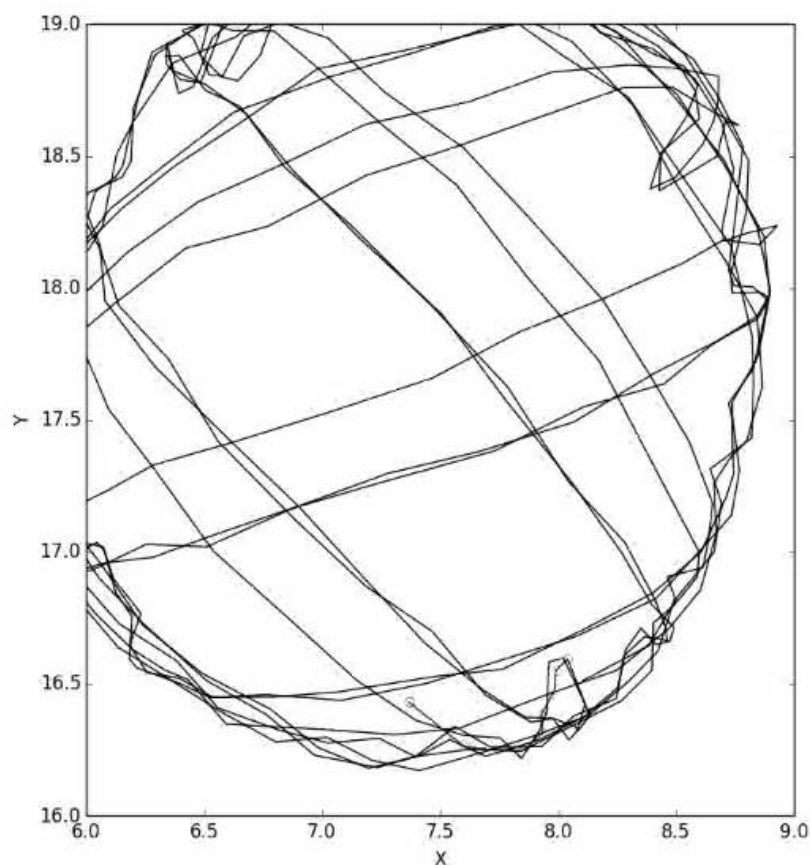


Рисунок 4.6 Результати моделювання для мережі з великою кількістю вузлів

Дуже цікавим та показовим є результат моделювання при умові повного відключення частини мережі, тобто ціла ділянка безпроводової сенсорної мережі стає недоступною та вимикається через недостатній рівень заряду батареї живлення. В цій ситуації скрипт буде продовжувати опитувати сусідні вузли, які можуть виявитись на суттєвій відстані від активного вузла ретранслятора, але через відсутність альтернативного робочого вузла, за логікою скрипта, маршрутизація буде виконуватись через дальні вузли з достатнім зарядом батареї живлення.

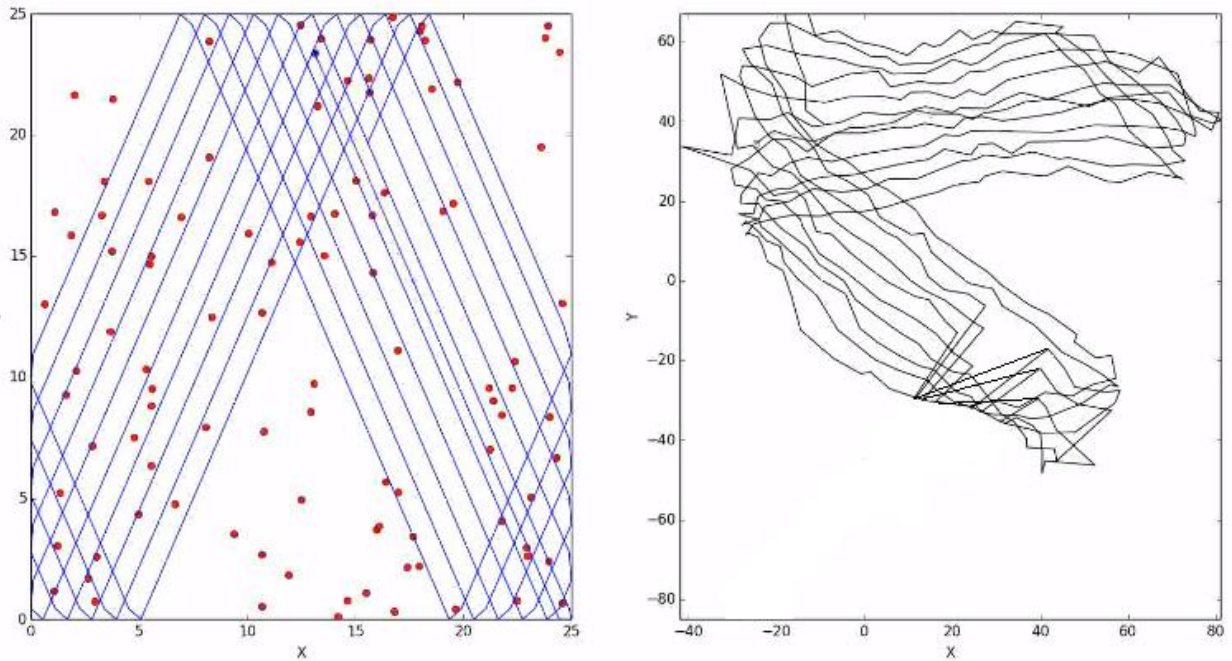


Рисунок 4.5 Моделювання за умови критично низького рівня заряду батареї для певної ділянки мережі

Постійна зміна рівня заряду окремих вузлів при кожній передачі даних та просто в режимі очікування, спричиняє постійний пере вибір наступного вузла, який має вищий рівень заряду акумулятора в порівнянні з сусідніми вузлами. саме тому не можна досягти точних спів падань маршрутів при кожному новому колі. Це можна добре помітити при цьому моделюванні:

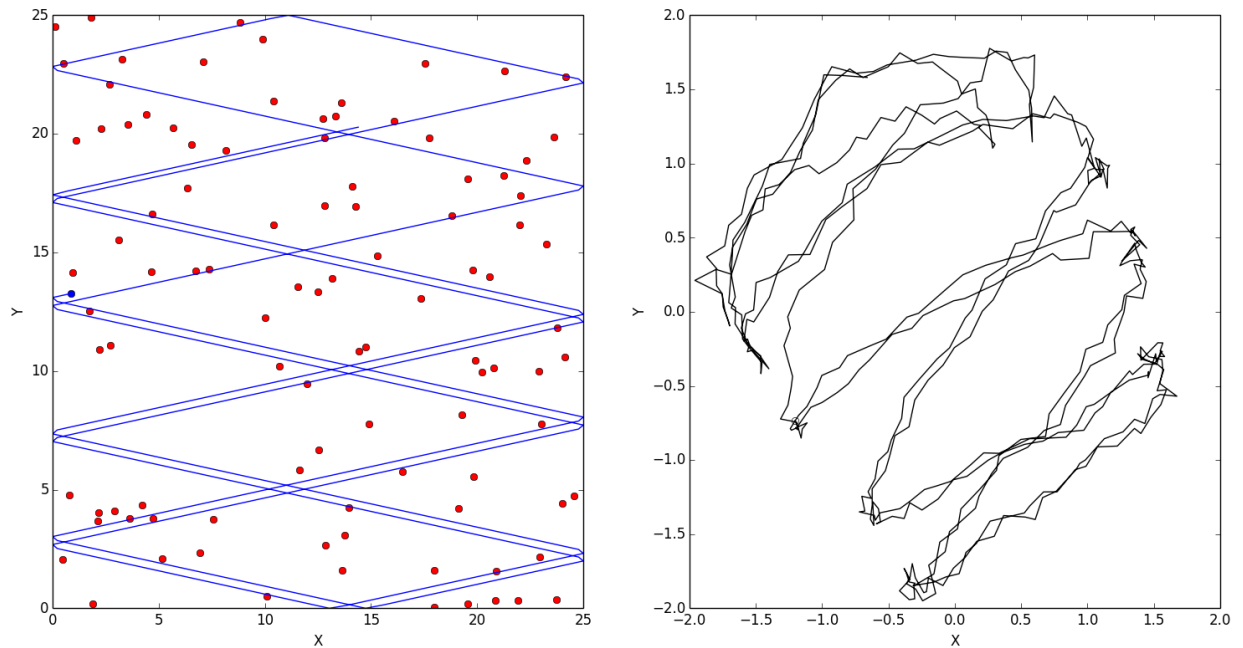


Рисунок 4.6 Моделювання в умовах постійної зміни рівня заряду акумулятора

Для оцінки ефективності розробленого програмного забезпечення було проведено ряд моделювань в яких протокол маршрутизації вибирався тільки одного разу на першій ітерації і вся подальша передача даних відбувалась саме за цим протоколом. Кількість ітерацій, а отже і кількість хопів, цих моделювань порівнювались з кількістю ітерацій для аналогічної мережі, але за умови вибору протоколу маршрутизації з використанням повноцінного скрипта.

Результати дослідження внесені до таблиці 4.1. Усі результати були отримані експериментальним шляхом. Скрипт виконувався по 10 разів для кожного варіанту розташування вузлів, а потім вираховувалась середня кількість ітерацій до моменту повної зупинки скрипта. Зупинка скрипта відбувалась у момент повного виснаження більшої частини вузлів, що не давало змогу продовжувати передачу даних.

Таблиця 4.1

Назва протоколу за яким відбувалась маршрутизація	Середня кількість ітерацій до відключення мережі
SPIN	1450
DD	1280
RR	890
GBR	850
OLSR	1000
AODV	1360
LEACH	1370
PEGASIS	1080
Скрипт	1700

Якщо порівнювати середнє значення кількості ітерацій за умови використання одного фіксованого протоколу маршрутизації протягом усього процесу передачі інформації з результатом кількості ітерацій при використанні скрипта, різниця кількості ітерацій буде сягати приблизно 25%. Ці дані також підтверджуються середньою кількістю повних кіл передачі інформації, яка на четверть більша при маршрутизації з використанням представленого скрипта.

Висновки до розділу 4

В цьому розділі розглядалось створення програмного забезпечення для підвищення тривалості роботи великої безпроводової сенсорної мережі з нерівномірним рівнем заряду батареї живлення вузлів на основі характеристик існуючих протоколів маршрутизації.

Були виконані та розглянуті основні етапи розробки необхідного програмного забезпечення:

- визначення жорстких вимог до програмного забезпечення;
- побудова блок-схеми;
- розробка алгоритму прийняття рішень скрипта;
- розглянуті основні функціональні елементи скрипта;
- представлені результати моделювання;
- проведено розрахунок ефективності використання скрипта.

Результатом виконаної роботи є підвищення тривалості роботи експериментальної безпроводової мережі на 25%.

ВИСНОВКИ

У моїй дипломній роботі розглянуто розвиток способів маршрутизації у безпроводових сенсорних мережах.

Поставлена мета роботи досягнена, а отримані результати відповідають сформульованим завданням та задовольняють їх.

Для досягнення поставленої мети були вирішенні наступні задачі:

1) У першому розділі було детально розглянуто особливості функціонування безпроводових сенсорних мереж.

Безпроводова сенсорна мережа – розподілена, самоорганізуюча і стійка до відмови мережа з великою кількістю автономних електронних вузлів, здатних обмінюватися повідомленнями і ретранслювати їх по безпроводному каналу зв'язку.

Основні переваги безпроводових сенсорних мереж:

- відсутність необхідності в прокладанні кабелів для електроживлення і передачі даних;
- низька вартість монтажу і технічного обслуговування системи;
- зручність розміщення автономних безпроводових вузлів в різноманітних точках простору;
- можливість впровадження і модифікації мережі на експлуатуємому об'єкті при мінімальному втручанні в процес його функціонування;
- надійність і відмовостійкість всієї системи в цілому при виході з ладу окремих вузлів або при порушенні зв'язку між ними.

Завдяки практичності, багатофункціональності, дешевизні та можливості функціонувати у віддалених регіонах з поганими кліматичними умовами, БСМ набули широкого застосування у різноманітних сферах.

2) У другому розділі розглядався процес маршрутизації у безпроводових сенсорних мережах. Були наведені параметри, покращення яких

досягається вибором оптимального способу маршрутизації, з використанням доступних ресурсів мережі.

Основні задачі маршрутизації:

- завдання пошуку оптимальних маршрутів;
- завдання маршрутизації із забезпеченням максимальної тривалості життя мережі.

Маршрутизація дозволяє покращити тільки декілька необхідних показників, які безпосередньо впливають на якість і швидкість передачі інформації та ефективність використання обладнання, за рахунок погіршення менш значущих параметрів для даної конкретної мережі.

Тому вибір способу маршрутизації для кожної мережі є індивідуальним і залежить від її призначення, розмірів, розміщення вузлів та технічних можливостей обладнання.

- 3) В третьому розділі досліджена класифікація протоколів маршрутизації в безпроводових сенсорних мережах. Детально описаний алгоритм роботи таких протоколів, як SPIN, DD, RR, GBR, OLSR, AODV, LEACH, PEGASIS, GAF, GEAR. Всі розглянуті протоколи маршрутизації широко використовуються у БСМ і є базовими протоколами з можливістю подальшої оптимізації та модифікації. Проаналізована доцільність їх використання в залежності від специфічних особливостей конкретної мережі. Виконано порівняльний аналіз ефективності протоколів. Наведені переваги та недоліки кожного з розглянутих протоколів маршрутизації.
- 4) В четвертому розділі розглядався процес розробки програмного забезпечення для визначення оптимального протоколу маршрутизації на основі проведеної багатокритеріальної порівняльної оцінки основних існуючих протоколів маршрутизації в безпроводових сенсорних мережах з метою підвищення енергоефективності використання обладнання та продовження тривалості роботи мережі. Були представлені основні вимоги до програмного забезпечення та їх

реалізація, приведені результати моделювання та проведен комплексний експериментальний аналіз тривалості роботи мережі при використанні одного фіксованого протоколу з тривалістю роботи аналогічної мережі при використанні розробленого скрипта.

ПЕРЕЛІК ПОСИЛАНЬ

1. Шахнович И.А. Современные технологии беспроводной связи. – М.: Техносфера, 2006. – 288 с.
2. Смелянский Р. Л. Компьютерные сети. В 2 томах. Том 1. Системы передачи данных. – М.: Академия, 2011. – 304 с
3. Dargie W., Poellabauer C. Fundamentals of wireless sensor networks: theory and practice. — John Wiley and Sons, 2010. — 330 p.
4. Вишневский В.М. Широкополосные беспроводные сети передачи информации / В.М. Вишневский, А.И. Ляхов, С.Л. Портной, И.В. Шахнович. - М.: Техносфера, 2005. - 592с.
5. Выборнова А.И. Модели беспроводных сенсорных сетей для различных применений / Выборнова А.И. // Электросвязь. 2013. № 1. С. 24-27
6. Дорошенко А.Е., Жереб К.А., Шевченко Р.С. О моделировании сенсорных сетей средствами высокого уровня // Проблемы програмування. — 2006. — Т. 2-3.
7. Баскаков С. С. Исследование способов повышения эффективности маршрутизации по виртуальным координатам в беспроводных сенсорных сетях // Вестник МГТУ им. Н. Э. Баумана. Сер. Приборостроение. 2009. № 2. С. 112–124.
8. Гуйда О.Г., Петрова В.М., Бондарук О.А. Вибір протоколу маршрутизації за допомогою імітаційного моделювання безпроводних сенсорних мереж // Науковий журнал «Вчені записки ТНУ імені В.І. Вернадського – 2017 – С. 3-6
9. Бершадский, А. М. Обзор методов маршрутизации в беспроводных сенсорных сетях / А. М. Бершадский, Л. С. Курилов, А. Г. Финогеев // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2012. – № 1 (21). – С. 47–57.

10. Kulik, J. Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks / J. Kulik, W. R. Heinzelman, H. Balakrishnan // Wireless Networks, 2002. – V. 8. – P. 169–185.
11. Intanagonwiwat, C. Directed Diffusion for Wireless Sensor Networking / C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva // IEEE/ACM Transactions on Networking. – 2003. – V. 11, № 1. – P. 2–16.
12. Braginsky, D. Rumor Routing Algorithm for Sensor Networks / D. Braginsky, D. Estrin // Proceedings of the First ACM International Workshop on Sensor Networks and Applications (WSNA), October 2002. – Atlanta GA, 2002. – P. 22–31.
13. C.Siva Ram Murthy, B.S.Manoj —Ad Hoc Wireless Networks: Architectures and Protocols, Prentice Hall, 2016, 880p.
14. Biradar, R. Inter-Intra Cluster Multihop-LEACH Routing In Self-Organizing Wireless Sensor Networks / R. Biradar, S. R. Sawant, R. Mudholkar, V. C. Patil // International Journal of Research and Reviews in Computer Science (IJRRCS). – 2011. – March. – V. 2, № 1. – P. 124–130.
15. S. Lindsey, C. Raghavendra, “PEGASIS: Power-Efficient Gathering in Sensor Information Systems”, IEEE Aerospace Conference Proceedings, 2016, Vol. 3, 9-16 pp. 1125-1130.
16. V. Rodoplu and T. H. Meng, “Minimum Energy Mobile Wireless Networks”, IEEE Journal Selected Areas in Communications, vol. 17, no. 8, Aug. 1999, pp. 1333-1344.
17. Geographic routing without location information / A. Rao [et al.] // Proceedings of the 9th annual international conference on mobile computing and networking. San Diego (USA), 2003. P. 96-108.
18. Roychowdhury, S. Geographic Adaptive Fidelity and Geographic Energy Aware Routing in Ad Hoc Routing / S. Roychowdhury, C. Patra // International Journal of Computer and Communication Technology (IJ CCT) for International Conference on Advances in Computer, Communication

- Technology and Applications (ACCTA-2010) (August 2010). – 2010. – V. 1, № 2–4. – P. 309–313.
19. Schurgers, C. Energy Efficient Routing in Sensor Networks / C. Schurgers, M. Srivastava // The MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, McLean VA, 2001. – P. 357–361.
 20. W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS '00), January 2000.
 21. Distributed energy adaptive routing for wireless sensor networks / C. Ok [et al.] // Proceedings of the IEEE international conference on automation science and engineering. Scottsdale (USA), 2007. P. 905-910.
 22. Puccinelli D., Sifakis E., Haenggi M. A cross-layer approach to energy balancing in wireless sensor networks // Workshop on networked embedded sensing and control. Notre Dame (USA), 2005. P. 1-17.
 23. Dai H., Han R. A node-centric load balancing algorithm for wireless sensor networks // Proceedings of the IEEE global telecommunications conference. San Francisco (USA), 2003. Vol. 1. P. 548-552.
 24. Beacon vector routing: scalable point-to-point routing in wireless sensor networks / R. Fonseca [et al.] // Proceedings of the 2nd symposium on networked systems design and implementation. Boston (USA), 2005. P. 329-342.

Додаток А. ЛІСТИНГ СКРИПТА ПРОГНОЗУВАННЯ

```
import matplotlib
matplotlib.use('TKAgg')
import SLAC.network.wireless as wsn
import SLAC.environment.world as env
import SLAC.simulation.controllers as contr
import numpy as np
from SLAC.simulation.animation import PlaybackAnimation
import GPY as gp

config = {
    'movingNodes': 1,
    'fixedNodes': 100,
    'xMax': 25,
    'yMax': 25,
}

world=env.World(config['xMax'], config['yMax'])

node =[wsn.FixedAP(max =world.getMaxX(), max=world.getMaxY()) for x in
range(0, config['fixedNodes'])]

nodes.extend([wsn.Fixed(max=world.getMaxX(), max=world.getMaxY()) for x in
range(0, config['movingNodes'])])

user=wsn.MovingAP(max=world.getMaxX(), max=world.getMaxY())

nodes.append(user)

controller=contr.NetworkController(world, nodes)
```

```
controller.initialize()
```

```
Y = np.array([n.getSignalStrengthAtLocation(*user.getPosition(), noise=0) for n in  
nodes if n != user])
```

```
for i in range(0, 100):
```

```
    controller.iterate()
```

```
    rssi = [n.getSignalStrengthAtLocation(*user.getPosition(), noise=0) for n in  
nodes if n != user]
```

```
    Y = np.vstack((Y, rssi))
```

```
predX = np.array(predX) + user.trace[0][0]
```

```
predY = np.array(predY) + user.trace[0][1]
```

```
anim = PlaybackAnimation(nodes[:-1], user, predX, predY)
```

```
anim.show()
```

Додаток Б. ЛІСТИНГ СКРИПТА МОДЕЛЮВАННЯ

```
import SLAC.network.wireless as wsn
import SLAC.environment.world as env
import SLAC.simulation.controllers as contr
from SLAC.predictor.probabilistic import predict
import SLAC.simulation.animation as anim
import numpy as np

config = {
    'movingNodes': 1,
    'fixedNodes': 100,
    'xMax': 25,
    'yMax': 25,
}

world=env.World(config['xMax'], config['yMax'])
node =[wsn.Moving(max=world.getMaxX(), max=world.getMaxY()) for x in
range(0, config['fixedNodes'])]
nodes.extend([wsn.Fixed(max=world.getMaxX(), max=world.getMaxY()) for x in
range(0, config['movingNodes'])])
user=wsn.Moving(max=world.getMaxX(), max=world.getMaxY())
nodes.append(user)

controller =contr.NetworkController(world, nodes)
controller.initialize()
```

```
animation = anim.NetworkAnimation(controller)  
animation.show()
```

Додаток В. ЛІСТИНГ СКРИПТА СЕРЕДОВИЩА ТА ОБ'ЄКТІВ

```
import numpy as np

import math

import numpy as np

from abc import ABCMeta, abstractmethod

import matplotlib

import matplotlib.pyplot as plt

from matplotlib.lines import Line2D

import matplotlib.animation as animation

class World(object):

    def __init__(self, dimX, dimY):

        self.dimX = dimX

        self.dimY = dimY

    def getRandomPosition(self):

        return [np.random.rand() * self.dimX, np.random.rand() * self.dimY]

    def getBoundingBox(self):

        return (self.dimX, self.dimY)

    def getMaxX(self):

        return self.dimX

    def getMaxY(self):

        return self.dimY

class Node(object, metaclass=ABCMeta):
```

```
def __init__(self, x = 0, y = 0):
    self.x = x
    self.y = y
def initialize(self, x, y):
    self.x = x
    self.y = y
@abstractmethod
def iterate(self):
    pass
def getPosition(self):
    return (self.x, self.y)
def setPosition(self, x, y):
    self.x = x
    self.y = y
def getDistance(self, x, y):
    return math.sqrt(math.pow((x - self.x),2) + math.pow((y - self.y),2))
```

```
class MovingNode(Node, metaclass=ABCMeta):
```

```
def __init__(self, x = 0, y = 0):
    super().__init__(x,y)
    self.predX = x
    self.predY = y
    self.trace = [(x,y)]
def initialize(self, x, y):
    self.trace = [(x,y)]
```

```

        Node.initialize(self, x, y)

    @abstractmethod
    def move(self):
        pass

    def moveToPosition(self, x, y):
        self.trace.append((x, y))
        self.x = x
        self.y = y

class BouncingNode(MovingNode):
    def __init__(self, maxX = 100, maxY = 100, x = 0, y = 0, ):
        super().__init__(x, y)
        self.maxX = maxX
        self.maxY = maxY
        self.r = 0
        self.s = 1

    def setMotion(self, angle, speed):
        self.r = angle
        self.s = speed

    def initialize(self, x, y):
        super().initialize(x, y)
        r = (2 * np.random.random()) * math.pi
        s = (2 * np.random.random()) + 0.1
        self.setMotion(r, s)

```



```

def RSSIraw(dist, n, txPower):
    return -(10 * n) * math.log10(dist) + txPower

def RSSI(dist, n, txPower, sd = 1):
    return RSSIraw(dist, n, txPower) + np.random.normal(0, n)

def plotAccessPoint(node, xDim, yDim, precission):
    X = np.linspace(0, xDim, precission)
    Y = np.linspace(0, yDim, precission)
    Z = np.array([node.getSignalStrengthAtLocation(x,y) for x in X for y in
Y]).reshape(precission, precission)
    X,Y = np.meshgrid(X,Y, indexing='ij')
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.plot_surface(X, Y, Z, rstride=2, cstride=2, cmap=cm.coolwarm,
        linewidth=0, antialiased=True)
    plt.show()

class WirelessEntity(metaclass=ABCMeta):
    def __init__(self, txPower = -59, n = 2):
        self.txPower = txPower
        self.n = n
    @abstractmethod
    def getDistance(self, x, y):
        pass
    def getSignalStrengthAtLocation(self, x, y, noise = 1):

```

```
return RSSI(self.getDistance(x,y), self.n, self.txPower, noise)
```

```
class FixedAP(WirelessEntity, Node):
    def __init__(self, maxX, maxY, txPower = -59, n = 2):
        WirelessEntity.__init__(self, txPower, n)
        Node.__init__(self)
    def iterate(self):
        pass
    def getDistance(self, x, y):
        return Node.getDistance(self, x, y)
```

```
class MovingAP(WirelessEntity, BouncingNode):
    def __init__(self, maxX, maxY, txPower = -59, n = 2):
        WirelessEntity.__init__(self, txPower, n)
        BouncingNode.__init__(self, maxX, maxY)
    def getDistance(self, x, y):
        return BouncingNode.getDistance(self, x, y)
```

```
class PlaybackAnimation(animation.TimedAnimation):
    def __init__(self, nodes, user, Xpred, Ypred):
        fig = plt.figure()
        combAx = fig.add_subplot(1,1,1)
        self.nodesPlt, = plt.plot([], [], 'ro')
        self.userPlt, = plt.plot([], [], 'bo')
        self.predPlt, = plt.plot([], [], 'wo')
```

```
combAx.add_line(self.nodesPlt)
combAx.add_line(self.predPlt)
combAx.add_line(self.userPlt)
combAx.set_xlabel('X')
combAx.set_ylabel('Y')
combAx.set_xlim(0, 25)
combAx.set_ylim(0, 25)
self.iteration = 0
self.nodes = nodes
self.user = user
self.Xpred = Xpred
self.Ypred = Ypred
super().__init__(fig, interval=100, blit=True, repeat=True, repeat_delay=500)
```

```
def show(self):
```

```
    plt.show()
```

```
def _draw_frame(self, iteration):
```

```
    x,y = zip(*[n.trace[iteration] if isinstance(n, MovingAP) else n.getPosition()
for n in self.nodes])
```

```
    self.nodesPlt.set_data(x,y)
```

```
    x,y = self.user.trace[iteration]
```

```
    self.userPlt.set_data([x], [y])
```

```
    x = self.Xpred[iteration]
```

```
    y = self.Ypred[iteration]
```

```
    self.predPlt.set_data([x], [y])
```

```
self._drawn_artists = [self.predPlt, self.nodesPlt, self.userPlt]
```

```
def _init_draw(self):
```

```
    self.nodesPlt.set_data([], [])
```

```
    self.userPlt.set_data([], [])
```

```
    self.predPlt.set_data([], [])
```

```
def new_frame_seq(self):
```

```
    self.iteration = 0
```

```
    return iter(range(len(self.user.trace) - 2))
```

```
class NetworkAnimation(animation.TimedAnimation):
```

```
    def __init__(self, networkController, interval = 50, iterations = 100):
```

```
        fig = plt.figure()
```

```
        combAx = fig.add_subplot(1,1,1)
```

```
        self.nodesPlt, = plt.plot([], [], 'ro')
```

```
        self.predPlt, = plt.plot([], [], 'wo')
```

```
        combAx.add_line(self.nodesPlt)
```

```
        combAx.add_line(self.predPlt)
```

```
        combAx.set_xlabel('X')
```

```
        combAx.set_ylabel('Y')
```

```
        combAx.set_xlim(0, 25)
```

```
        combAx.set_ylim(0, 25)
```

```
        self.iteration = 0
```

```
        self.network = networkController
```

```
super().__init__(fig, interval=interval, blit=True)
```

```
def show(self):
```

```
    plt.show()
```

```
def _draw_frame(self, framedata):
```

```
    self.network.iterate();
```

```
    x,y = zip(*[n.getPosition() for n in self.network.nodes])
```

```
    xPred,yPred = zip(*[(50,50) for n in self.network.nodes])
```

```
    self.nodesPlt.set_data(x,y)
```

```
    self.predPlt.set_data(xPred,yPred)
```

```
    self._drawn_artists = [self.predPlt, self.nodesPlt]
```

```
def _init_draw(self):
```

```
    self.nodesPlt.set_data([], [])
```

```
    self.predPlt.set_data([], [])
```

```
def new_frame_seq(self):
```

```
    return iter(range(self.network.iteration + 1))
```

```
class Controller(object):
```

```
    def __init__(self):
```

```
        self.iteration = 0
```

```
        self.outputEnabled = True
```

```
def iterate(self):  
    self.iteration += 1  
    self.output("Iteration " + str(self.iteration))  
  
def setOutput(self, flag):  
    self.outputEnabled = flag  
  
def output(self, msg):  
    if(self.outputEnabled):  
        print(msg)
```

```
class NetworkController(Controller):  
    def __init__(self, world, nodes):  
        Controller.__init__(self)  
        self.world = world  
        self.nodes = nodes  
  
    def initialize(self):  
        for n in self.nodes:  
            n.initialize(*self.world.getRandomPosition())  
  
    def iterate(self):  
        super().iterate()  
        for n in self.nodes:  
            n.iterate()
```

```
class GP:
```

```

def __init__(self,X,Y,kernel=None,parameter_priors=None):
    self.N = Y.shape[0]
    self.setX(X)
    self.setY(Y)
    if kernel==None:
        self.kernel = kernels.RBF_full(-1,-np.ones(self.Xdim))
    else:
        self.kernel = kernel
    if parameter_priors==None:
        self.parameter_prior_widths = np.ones(self.kernel.nparams+1)
    else:
        assert parameter_priors.size==(self.kernel.nparams+1)
        self.parameter_prior_widths = np.array(parameter_priors).flatten()
    self.beta=0.1
    self.update()
    self.n2ln2pi = 0.5*self.Ydim*self.N*np.log(2*np.pi)

def setX(self,newX):
    self.X = newX.copy()
    N,self.Xdim = newX.shape
    assert N == self.N,
    self.xmean = self.X.mean(0)
    self.xstd = self.X.std(0)
    self.X -= self.xmean
    self.X /= self.xstd

```

```

def setY(self,newY):
    self.Y = newY.copy()
    N,self.Ydim = newY.shape
    assert N == self.N,
    self.ymean = self.Y.mean(0)
    self.ystd = self.Y.std(0)
    self.Y -= self.ymean
    self.Y /= self.ystd

def find_kernel_params(self,itters=1000):
    new_params = fmin_cg(self.ll,np.hstack((self.kernel.get_params(),
np.log(self.beta))),fprime=self.ll_grad,maxiter=itters)
    final_ll = self.ll(new_params) # sets variables - required!

def update(self):
    self.K = self.kernel(self.X,self.X)
    self.K += np.eye(self.K.shape[0])/self.beta
    self.L = np.linalg.cholesky(self.K)
    self.A = linalg.cho_solve((self.L,1),self.Y)

def update_grad(self):
    self.Kinv =
np.linalg.solve(self.L.T,np.linalg.solve(self.L,np.eye(self.L.shape[0])))
    self.alphalK = np.dot(self.A,self.A.T)-self.Ydim*self.Kinv

```



```

def marginal(self):

    return -self.Ydim*np.sum(np.log(np.diag(self.L))) -
    0.5*np.trace(np.dot(self.Y.T,self.A)) - self.n2ln2pi

def predict(self,x_star):

    x_star = (np.asarray(x_star)-self.xmean)/self.xstd

    k_x_star_x = self.kernel(x_star,self.X)

    k_x_star_x_star = self.kernel(x_star,x_star)

    means = np.dot(k_x_star_x, self.A)

    means *= self.ystd

    means += self.ymean

    v = np.linalg.solve(self.L,k_x_star_x.T)

    variances = (np.diag( k_x_star_x_star -
    np.dot(v.T,v)).reshape(x_star.shape[0],1) + 1./self.beta) *
    self.ystd.reshape(1,self.Ydim)

    return means,variances

```