Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод забезпечення надійності полінгової мережі у поставарійному стані високомобільного вузла рою БПЛА

(тема)

Виконав:

студент II курсу, групи СПм-22-1

Шостак А.Р.
(прізвище, ініціали)

Спеціальність

123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

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

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: доц. Ткачов В.М.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

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

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Шостаку Антону Романовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Метод забезпечення надійності полінгової мережі у поставарійному стані високомобільного вузла рою БПЛА.

затверджена наказом по університету від " 06 " листопада 2023 р. № 1299 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 січня 2024 р.

3. Вхідні дані до роботи 1) Комунікаційні архітектури; 2) Протоколи маршутизації; 3) Стратегії забезпечення надійності мережі; 4) Методи забезпечення надійності мережі.

4. Перелік питань, що потрібно опрацювати у роботі

1) огляд комунікаційних архітектур рою БПЛА;
2) огляд протоколів маршрутизації рою БПЛА;
3) огляд стратегій забезпечення надійності мережі;
4) огляд існуючих методів забезпечення надійності мережі;
5) вибір та обгрунтування методики та засобів дослідження;
6) проведення експериментальних досліджень;
7) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 14 слайдів

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд методів забезпечення надійності		
	мережі	07.11.23-22.11.23	
2	Вибір та обґрунтування дослідження методу	23.11.23-24.11.23	
3	Вибір інструментальних засобів	25.11.23-26.11.23	
4	Проведення моделювання	27.11.23-04.12.23	
5	Оформлення матеріалів кваліфікаційної роботи	05.12.23-01.01.24	
6	Подання кваліфікаційної роботи керівникові та		
	її попередній захист	02.01.24-07.01.24	
7	Подання кваліфікаційної роботи на рецензу-	12.01.24	
	вання		

Дата видачі завдання 06 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Ткачов В.М.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 102 с., 32 рис., 1 табл., 14 дод., 11 джерел.

РІЙ БПЛА, ПОЛІНГОВА МЕРЕЖА, КОМУНІКАЦІЙНІ АРХІТЕКТУРИ, ПРОТОКОЛИ МАРШРУТИЗАЦІЇ, НАДІЙНІСТЬ МЕРЕЖІ.

Метою кваліфікаційної роботи є дослідження методу забезпечення надійності полінгової мережі у поставарійному стані високомобільного рою БПЛА.

У ході виконання кваліфікаційної роботи було досліджено комунікаційну архітектуру рою БПЛА, визначено їх плюси та мінуси, досліджено протоколи маршрутизації, оглянуто існуючі стратегії забезпечення надійності мережі, а також проаналізувати основні аспекти, які можуть зробити БПЛА чутливими до пошкоджень. Після чого було розглянуто декілька існуючих методів, які спроможні забезпечити надійність мережі й відновити з’єднання у поставарійному стані. Після дослідження було обрано один метод з огляду на його унікальні можливості та переваги над іншими методами, який буде розглянуто в моделюючій частині й порівняно з іншими методами.

ABSTRACT

Master’s thesis: 102 pages, 32 figures, 1 tables, 14 appendices, 14 sources.

UAV SWARM, POLING NETWORK, COMMUNICATION ARCHITECTURES, ROUTING PROTOCOLS, NETWORK RELIABILITY.

The major goal of this thesis is research of the method of ensuring the reliability of the polling network in the post-emergency state of a highly mobile UAV swarm.

In the course of the qualification work, the communication architecture of the UAV swarm was investigated, their pros and cons were determined, routing protocols were investigated, existing strategies for ensuring network reliability were reviewed, and the main aspects that could make UAVs sensitive to damage were analyzed. After that, several existing methods were considered, which are able to ensure the reliability of the network and restore the connection in the original state. After the study, one method was chosen due to its unique capabilities and advantages over other methods, which will be discussed in the modeling part and compared to other methods.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AODV- Ad hoc On Demand Distance Vector

APAR- Ant colony optimization-based Polymorphism-Aware Routing

B.A.T.M.A.N. - Better Approach to Mobile Ad hoc Networking

BeeAdhoc - Bee colony algorithm based Ad hoc network

DCR - Data Centric Routing

DOLSR - Directional Optimized Link State Routing

DSDV - Destination-Sequenced Distance Vector

DSR - Dynamic Source Routing

GGF - Greedy Geographic Forwarding

GLSR - Geographic Load Share Routing

GPMOR - Geographic Position Mobility Oriented Routing

GPSR - Greedy Perimeter Stateless Routing

G-T-G - Group-to-Group

HRP - Hybrid Routing Protocol

LCAD - Load Carry and Deliver Routing

MPR - Multiple Point Relay

OLSR - Optimized Link State Routing

PRP - Proactive Routing Protocol

RGR - Reactive-Greedy-Reactive

RREP - Route Reply

RREQ - Route Requests

RRP - Reactive Routing Protocol

SI - Swarm Intelligence

TC - Topology Control

TORA - Temporarily Ordered Routing Algorithm

UAV - Unmanned Aerial Vehicle

UE-DSR - UAV Energy Dynamic Source Routing

USNET - UAV Swarm Network

U-T-I - UAV-to-Infrastructure

ZRP - Zone Routing Protocol

ВСТУП

Рій безпілотних літальних апаратів (БПЛА) — це набір повітряних роботів, тобто дронів, які працюють разом для досягнення певної мети. Кожен дрон у групі приводиться в рух певною кількістю роторів і має можливість вертикального зависання, зльоту та посадки. Управління польотом дронів здійснюється як вручну, тобто за допомогою дистанційного керування, так і автономно за допомогою процесорів, встановлених на дронах. Загальне призначення дронів – військове, але останнім часом все більше уваги привертає їх цивільне застосування. Дійсно, недорогі безпілотні літальні апарати та їхні зграї є багатообіцяючою платформою для інноваційних дослідницьких проектів і майбутніх комерційних застосувань, які допоможуть людям у їхній роботі та повсякденному житті. У кожному рої кожен безпілотник може мати спеціальні завдання зі збору та обробки даних із достатньою обчислювальною потужністю для виконання цих завдань у режимі реального часу. Його центральна обробка відбувається на більш продуктивному сервері/базовій станції або навіть у хмарі.

Зв'язок відіграє важливу роль у контролі та координації зграї БПЛА. Комунікаційна архітектура визначає спосіб обміну інформацією між БПЛА або між БПЛА та центральним центром управління. Протоколи маршрутизації допомагають забезпечити надійну наскрізну передачу даних. Тому особливо важливо розробити архітектуру зв’язку роя БПЛА та протоколи маршрутизації з високою продуктивністю та стабільністю.

Рій БПЛА був представлені як багатообіцяюча парадигма для проведення моніторингу та взаємозв’язку завдань у неконтрольованих або навіть ворожих середовищах. Однак суворий сценарій розгортання може зробити БПЛА чутливими до великомасштабних пошкоджень і, таким чином, погіршити підключення та продуктивність мережі. Жодна з існуючих технологій не може ефективно реорганізувати вцілілі БПЛА в групі серйозно

пошкоджених БПЛА в єдину мережу роїв БПЛА.

Метою роботи є дослідження методу забезпечення надійності полінгової мережі високомобільного рою БПЛА.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Комунікаційні архітектури

Комунікаційна архітектура відіграє важливу роль в інтелектуальному управлінні та автономній співпраці роїв БПЛА. На початковому етапі розвитку групи БПЛА центральна станція є доступною та достатньо потужною для зв’язку з усією групою БПЛА. Тому централізований підхід було розширено з традиційної системи з одним БПЛА до архітектури зв’язку з роєм БПЛА. У міру того, як рій зростає в розмірах, пропонується децентралізований підхід. Децентралізований підхід має більш складну структуру та організацію, але він зменшує залежність рою від центральних станцій.

1.1.1 Централізована комунікаційна архітектура

Архітектура централізованого зв’язку розвинулась із систем з одним БПЛА та була реалізована для технології рою БПЛА. Як показано на (рисунок 1.1), існує центральний вузол (тобто фіксована мережева інфраструктура), до якого підключені всі БПЛА в групі. Кожен БПЛА формує зв’язок «один-до-одного» з інфраструктурою та безпосередньо отримує команди управління від інфраструктури. Цей тип централізованої комунікаційної архітектури є відносно стабільним, використовує простіші алгоритми маршрутизації та має менший масштаб. Якщо розмір групи БПЛА та зона покриття місії невеликі, а місія відносно проста, цей тип архітектури є більш придатним. Одним із застосувань централізованої комунікаційної архітектури є «UAV-GCS Centralized Data-Oriented Communication Architecture», яка призначена для спостереження.

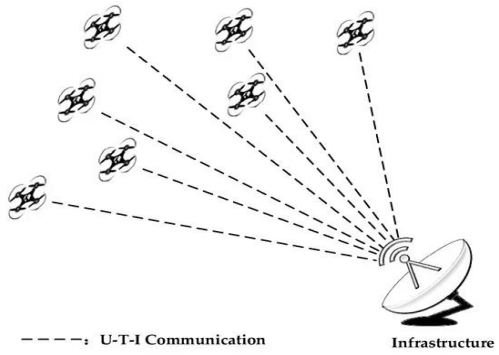


Рисунок 1.1 – Схематичне зображення архітектури централізованого зв’язку

Однак зв'язок між БПЛА потребує інфраструктури для передачі. Відстань U-T-I набагато більша, ніж відстань U-T-U, що спричиняє триваліші затримки. Крім того, враховуючи високу мобільність БПЛА та вимоги до покриття ройових програм, ця архітектура, очевидно, не є стабільною. Інфраструктура як центральний вузол також спричиняє паралізацію всієї комунікаційної мережі, коли наземна станція чи супутник виходять із ладу або зазнають атаки. Таким чином, архітектура централізованого зв’язку має недолік – єдину точку відмови (SPOF). У результаті архітектуру централізованого зв'язку можна вважати ненадійною.

1.1.2 Децентралізована комунікаційна архітектура

Оскільки UVA працюють на високих швидкостях і місії можуть охоплювати великі території, БПЛА часто підключаються до мережі та від’єднуються від неї. Як наслідок, найкращим вибором вважаються спеціальні мережі БПЛА. У рамках децентралізованої архітектури зв’язку БПЛА здійснюють інтерактивний зв’язок у режимі реального часу в режимі ad hoc, тим самим усуваючи залежність від інфраструктури та обмеження на діапазон зв’язку.

1.1.2.1 Single-Group Swarm Ad hoc Network

У «single-group swarm Ad hoc network» (рисунок 1.2) внутрішній зв’язок рою не залежить від інфраструктури. Зв’язок між роєм та інфраструктурою є одноточковим зв’язком, що покладається на певний БПЛА, який називається шлюзовим БПЛА. У цій системі інші БПЛА є ретрансляційними вузлами, які передають дані всередині зграї. Використовуючи цей метод, БПЛА в групі можуть обмінюватися інформацією про ситуацію в режимі реального часу для оптимізації спільного контролю та підвищення ефективності. Подібним чином, взаємний зв’язок між шлюзовим БПЛА та інфраструктурою також дозволяє завантажувати та завантажувати інформацію про рій, включаючи навчальну інформацію. Що стосується шлюзового БПЛА, необхідні два додаткових типи приймачів-передавачів: один для зв’язку з іншими БПЛА на низькій потужності та на коротких відстанях, а інший – для зв’язку з інфраструктурою на великій потужності та на великій відстані. Як наслідок, іншим БПЛА в зграї потрібно мати лише недорогі та легкі трансивери малого радіусу дії. Це не тільки значно розширює дальність зв’язку мережі, щоб забезпечити великі вимоги до покриття, але також робить невеликі БПЛА з меншим корисним навантаженням більш корисними. Однак, щоб гарантувати послідовне підключення зграї, архітектура «спеціальної мережі одногрупової зграї» вимагає, щоб моделі польоту (швидкість і орієнтація курсу) усіх БПЛА в зграї були однаковими. Таким чином, група БПЛА малого розміру підходить для сценаріїв, які використовують комунікаційну архітектуру. Одним із прикладів «спеціальної мережі з одногруповим роєм» є хмарний рівень БПЛА в трирівневій архітектурі стійкої до стихійних лих для довгострокової еволюції громадської безпеки (DR-PSLTE).

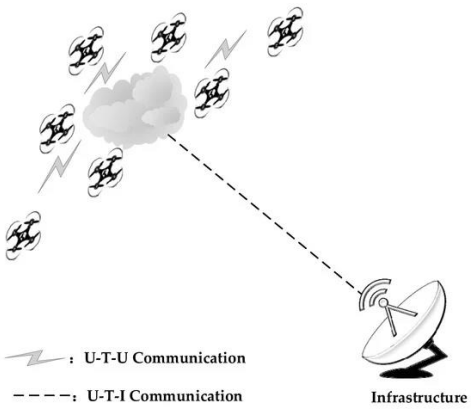
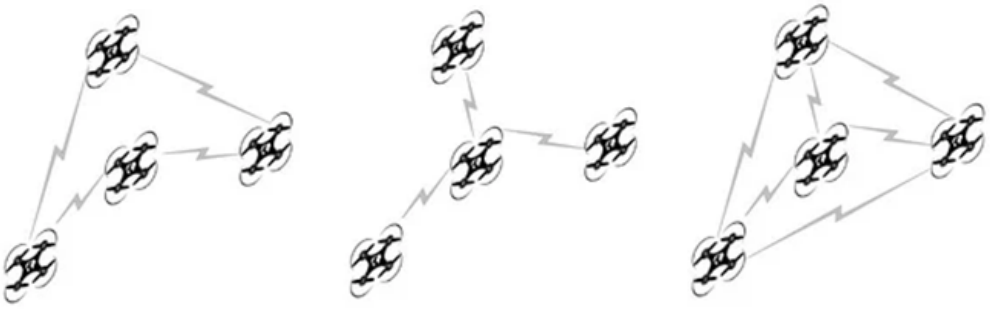


Рисунок 1.2 - Схематичне зображення децентралізованої архітектури Single-Group Swarm Ad hoc Network

Відповідно до вимог місії архітектура внутрішнього зв’язку може зазнавати кількох змін. На рисунку 1.3 показано три загальні архітектури зв’язку між роями.



а) б) в)

Рисунок 1.3 – Архітектура зв’язку всередині рою: а) кільцева архітектура; б) зіркова архітектура; в) сітчаста архітектура

БПЛА в кільцевій архітектурі (рисунок 1.3) утворюють замкнутий контур зв’язку через двонаправлене з’єднання. Будь-який БПЛА може діяти як шлюзовий вузол для зграї. Якщо прямий зв’язок між двома сусідніми БПЛА виходить з ладу, цільовий БПЛА може повернутися через петлю зв’язку. Тому кільцева архітектура має певну стабільність. Однак очевидно,

що цій архітектурі бракує масштабованості.

Комунікація організована у формі зіркової архітектури (рисунок 1.3). Шлюз БПЛА знаходиться посередині, і він спілкується не тільки з інфраструктурою, але й з усією групою БПЛА. Неважко побачити, що зіркова архітектура має властивий недолік SPOF. Тобто, якщо вузол шлюзу виходить з ладу, це призведе до збою всієї системи.

Сітчаста архітектура є комбінацією кільцевої та зіркоподібної архітектур і має переваги обох систем (рисунок 1.3). Усі вузли БПЛА в групі мають однакові можливості, як з термінальними вузлами, так і з функціями маршрутизації. Інформаційний потік від одного вузла до іншого може бути реалізований у різних формах, і будь-який БПЛА може діяти як шлюзовий вузол для рою. Сітчаста архітектура тепер стала стандартом для систем зв’язку між роями.

1.1.2.2 Multi-Group Swarm Ad hoc Network

Щоб усунути недоліки мережі «Single-Group», мережа «Multi-Group» (рисунок 1.4) об’єднує як централізовану архітектуру, так і архітектуру «Single-Group». Різні типи груп мають різне застосування залежно від місії. Міжгруповий зв’язок, тобто зв’язок між групами (G-T-G), здійснюється через інфраструктуру, тому шлюзові БПЛА все ще відповідають за зв’язок з інфраструктурою. Коли сценарії місії вимагають різноманітних БПЛА, можна використовувати архітектуру «спеціальної мережі з багатьма групами». Однак слід звернути увагу на надійність цієї архітектури, оскільки архітектура «багатогрупової мережі Ad hoc» є напівцентралізованою. У той же час для зв’язку G-T-G між двома БПЛА в різних групах архітектура «спеціальної мережі багатогрупового рою» все ще страждає від високої затримки. Одним із конкретних застосувань багатогрупової архітектури є спільні операції на кількох театрах військових операцій. Центральний центр управління спілкується з різними роями БПЛА, і рої спрямовуються в зону

місії з різних напрямків, згідно з інструкціями управління.

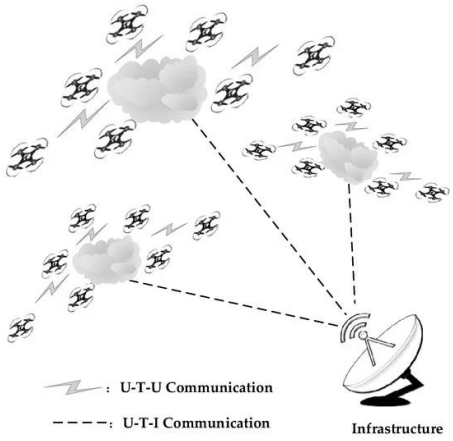


Рисунок 1.4 – Схематичне зображення децентралізованої архітектури Multi-Group Swarm Ad hoc Network

1.1.2.3 Multi-layer Swarm Ad hoc Network

Архітектура «Multi-layer Swarm» є ще одним типом архітектури, який підходить для масової різноманітності рою (рисунок 1.5). Ця архітектура є значно вдосконаленою порівняно з архітектурою «спеціальної мережі багатогрупового рою». Група суміжних БПЛА одного типу утворює спеціальну мережу, яка є першим рівнем архітектури зв’язку. Різні типи груп БПЛА покладаються на шлюзові БПЛА для здійснення зв’язку G-T-G, який складається з другого рівня. Найближчий шлюз БПЛА спілкується з інфраструктурою, яка є третім рівнем архітектури. Зв'язок між будь-якими двома БПЛА в архітектурі «Multi-layer» не потребує ретрансляції інфраструктури. Взаємне спілкування БПЛА в одній групі відбувається на першому рівні. Зв'язок між БПЛА в різних групах маршрутизується через шлюз БПЛА. Пакети даних проходять по черзі через перший і другий рівні. Таким чином, мережа немає SPOF, і цей тип архітектури є надійним.

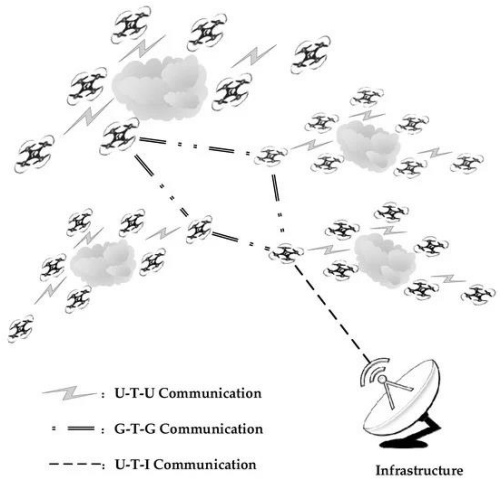


Рисунок 1.5 – Схематичне зображення децентралізованої архітектури Multi-layer Swarm Ad hoc Network

Під час місії, коли кількість БПЛА змінюється, архітектура мережі «Multi-layer Swarm Ad hoc» компенсує збільшення або зменшення вузлів БПЛА та швидко здійснює реконструкцію мережі. Таким чином, архітектура мережі підходить для сценаріїв, коли місії рою БПЛА є складними: існує величезна кількість БПЛА, які виконують місію, топологія мережі змінюється, а зв’язок між вузлами БПЛА є частим. З дослідженнями та розробкою роїв БПЛА та вдосконаленням комунікаційних технологій одне з можливих покращень включає кількість рівнів, необхідних для формування спеціальної мережевої архітектури рою з більшою кількістю БПЛА, що призводить до більшого покриття завдань і більш надійної мережі.

1.1.3 Полінгова мережа

Полінгова мережа для рою безпілотних літальних апаратів (БПЛА) - це архітектурна система управління та комунікації, в якій головний вузол (найчастіше центральний БПЛА або земна станція) взаємодіє з підлеглими БПЛА за допомогою запитів і відповідей. Основною характеристикою полінгової мережі є те, що ініціатива щодо комунікації і контролю

знаходиться в головному вузлі, який періодично запитує підлеглі вузли (інші БПЛА) для збору інформації та керування їхньою поведінкою.

Полінгові мережі для роїв БПЛА можуть бути використані в різних сферах, включаючи військові додатки, цивільні розвідувальні операції, агропередачу, моніторинг навколишнього середовища та багато інших. Вони дозволяють забезпечити ефективне та централізоване управління роєм БПЛА.

У деяких випадках полінгова мережа може бути самостійною мережею, де головний вузол (центральний БПЛА або земна станція) взаємодіє лише з підлеглими БПЛА в режимі полінгу. Такий підхід може використовуватися в ситуаціях, де не потрібна широка мережева комунікація або де вимагається велика автономність рою. В цьому випадку полінгова мережа відносится до централізовано комунікаційної архітектури.

В інших сценаріях полінгова мережа може бути однією з складових багатовидової мережі, де існують інші види комунікації та мережеві протоколи. Наприклад, великі рої БПЛА можуть використовувати полінгову мережу для централізованого управління та координації, а також мережу передачі даних для обміну інформацією між БПЛА.

Залежно від конкретних потреб і вимог застосування, полінгова мережа може існувати окремо або співіснувати з іншими мережами, створюючи комплексну систему управління та комунікації для рою БПЛА.

1.2 Протоколи маршутизації

Протокол маршрутизації відіграє вирішальну роль у надійній наскрізній передачі даних. Це робить протокол маршрутизації привабливою темою для дослідження в області зв’язку з роєм БПЛА. Однак через швидку мобільність БПЛА, нестабільність повітряних з’єднань, обмежені ресурси та мінливість вимог до якості обслуговування (QoS) маршрутизація в мережі зв’язку роя БПЛА стала життєво важливим завданням. Крім того, традиційні

протоколи маршрутизації Ad hoc не можуть використовуватися в зв’язку з роєм БПЛА без удосконалення Тому розробка протоколів маршрутизації, які відповідають більшості сценаріїв місій і характеристик БПЛА, залишається проблемою для дослідників.

У цьому розділі представлено загальну технологію маршрутизації зв’язку БПЛА, яка є основою розробки та реалізації протоколу маршрутизації. Потім класифікується кожен протокол маршрутизації клас. Класифікація заснована на технологіях, за якими йдуть протоколи маршрутизації. Крім того, надається детальний огляд кожного класу протоколів маршрутизації.

1.2.1 Технології маршрутизації

Технології маршрутизації забезпечують підтримку реалізації протоколів маршрутизації. Реалізація протоколу маршрутизації зазвичай є вдосконаленням основних технологій маршрутизації або комбінацією кількох технологій маршрутизації. Спеціальна мережа роя БПЛА розроблена на основі традиційної спеціальної мережі. Таким чином, звичайні технології маршрутизації також можуть бути використані в мережі БПЛА спеціального типу після аналізу, вдосконалення та комбінування. На рисунку 1.6 показано шість загальних технологій маршрутизації спеціальної мережі БПЛА: Store-Carry-Forward, Greedy Forward, Path Discover, Single-path, Multi-path, Predictive routing.

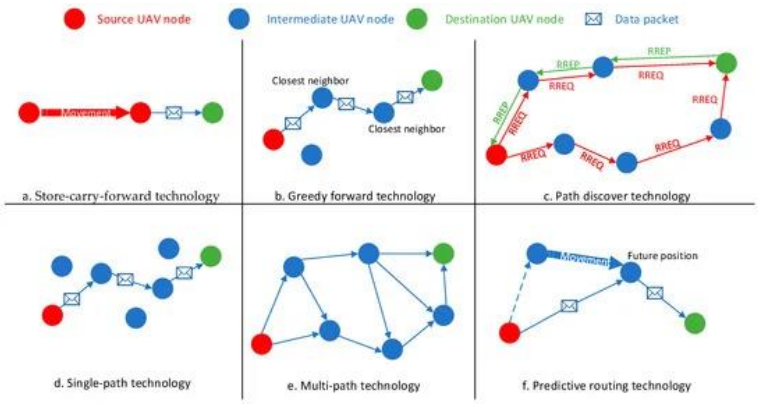


Рисунок 1.6 – Обґрунтування загальних технологій маршрутизації мережі БПЛА: a) Store-Carry-Forward; b) Greedy Forward; c) Path Discover; d) Single-path; e) Multi-path; f) Predictive routing

Store-Carry-Forward. Якщо в певний час не вдається знайти вузол ретрансляції, поточний вузол зберігатиме та переноситиме дейтаграму, доки не знайде вузол пересилання. Ця технологія підходить для переривчастої мережі, але недоліком є те, що вона створює велику затримку.

Greedy forward. Принцип пересилання полягає у виборі сусіднього вузла, найближчого до пункту призначення, як вузла ретрансляції, доки дейтаграма не буде надіслана до пункту призначення. Його можна використовувати в сценаріях, коли розгортання БПЛА є інтенсивним. Однак, коли вузол знаходиться найближче до пункту призначення і немає шляху до сусіднього вузла, це спричинить збій. У цьому випадку її слід поєднувати з іншими технологіями, щоб підвищити надійність технології.

Path Discover. Ядро здійснюється за допомогою переповнення запиту маршрутизації (RREQ), що максимізує доступність шляху. Ця техніка зменшує ймовірність перерв зв’язку, коли поточний вузол втрачає цільове географічне розташування. Але недоліком є те, що він надмірно споживає ресурси пропускної здатності.

Single-path. Характеризується використанням єдиного шляху для передачі даних. Підходить для надзвичайно обмежених ресурсів пропускної

здатності. Однак він має недолік - низьку міцність. Немає альтернативного шляху для збою мережі, тому легко втратити пакети. Ця техніка рідко використовується в сценаріях зв’язку з роєм БПЛА.

Multi-path. Технологія багатошляхового розповсюдження може ефективно підвищити надійність з’єднання. Коли одна ланка виходить з ладу, інші ланки можуть взяти її на себе. Він підходить для сценаріїв, де потрібна висока надійність зв’язку. Недоліком є те, що коли багатошляхові перехрестя виходять з ладу, мережа матиме петлю, яка блокуватиме мережу.

Predictive routing. Технологія прогнозної маршрутизації передбачає майбутнє положення вузла за його поточним положенням, швидкістю та напрямком, а потім вибирає наступний оптимальний вузол переходу. Ця технологія застосовна до сценаріїв, коли положення вузлів швидко змінюються, тому вона широко використовується в мережі БПЛА Ad hoc.

1.2.2 Класифікація протоколів маршрутизації

Більшість традиційних протоколів маршрутизації не підходять для зв’язку з роєм БПЛА. З одного боку, вузли БПЛА мають властивості, відмінні від традиційних вузлів, такі як швидка мобільність. З іншого боку, різні сценарії завдань мають різні вимоги до протоколів маршрутизації. Тому, щоб отримати протокол маршрутизації для групи БПЛА в спеціальних мережах, дослідники вдосконалили деякі традиційні протоколи маршрутизації. У той же час було запропоновано низку нових, виділених протоколів маршрутизації для рою БПЛА (рисунок 1.7).

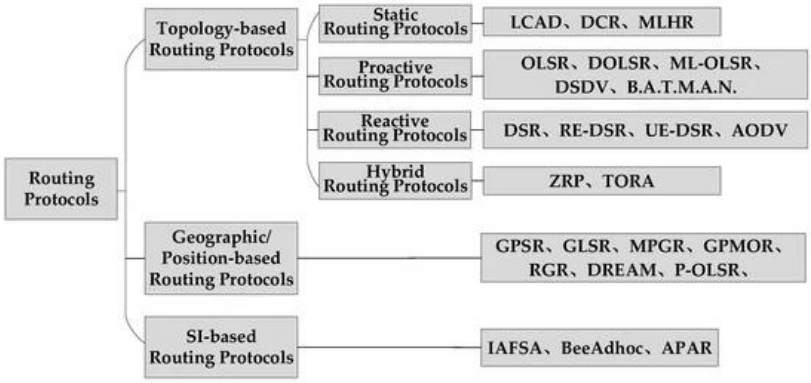


Рисунок 1.7 – Класифікація протоколів маршрутизації

1.2.3 Topology-Based Routing Protocols

Протоколи маршрутизації на основі топології використовують адреси Internet Protocol (IP) для визначення вузлів і використовують наявну інформацію про посилання для пересилання пакетів через відповідні шляхи. Протоколи маршрутизації на основі топології можна розділити на статичні протоколи маршрутизації, проактивні протоколи маршрутизації, реактивні протоколи маршрутизації та гібридні протоколи маршрутизації.

1.2.3.1 Static Routing Protocols

Статичні протоколи маршрутизації мають статичні та неоновлювані таблиці маршрутизації. Вони в основному використовуються в ситуаціях із фіксованою топологією та без оновлень завдань. Через недостатню відмовостійкість і адаптованість до динамічних середовищ традиційні статичні протоколи маршрутизації мають обмежене застосування для групових систем БПЛА. Нижче наведено три протоколи статичної маршрутизації:

Протокол Load Carry and Deliver Routing (LCAD) більш раннім протоколом маршрутизації, що використовується в централізованій архітектурі зв'язку. Інфраструктура передає пакети даних до БПЛА, який

потім переносить і доставляє пакети даних до місця призначення. Метою LCAD є максимізація пропускної здатності мережі з одночасним підвищенням безпеки. Це мінімізує кількість стрибків для досягнення пакетної передачі даних і вимог до допустимих затримок. Крім того, недоліком LCAD є те, що зі збільшенням відстані зв’язку між БПЛА значно зростає затримка.

Data Centric Routing (DCR). Для практичних застосувань вимога щодо передачі даних «один до багатьох» існує в системах з кількома БПЛА. DCR добре працює з «single-group swarm Ad hoc network», в якій шлюз БПЛА відповідає за розподіл інформації іншим вузлам. Типовим застосуванням є модель «публікація-підписка», яка може реалізовувати автоматичні з’єднання між видавцями даних і передплатниками, а також збирати трафік. Через мінімальну допомогу між БПЛА, DCR є кращим, коли система має обмежену кількість БПЛА та заплановану траєкторію польоту.

Multilevel Hierarchical Routing (MLHR) переміщена з мобільної мережі транспортних засобів, що вирішує проблему масштабованості великомасштабних мереж транспортних засобів. Як описано в «спеціальній мережі багаторівневих роїв», зграї БПЛА можна розділити на групи на основі розміру та функції БПЛА. Тому комунікаційні мережі організовані як ієрархічна структура. Тільки шлюзові БПЛА відповідають за зв'язок G-T-G. БПЛА шлюзу надсилає дані іншим вузлам БПЛА в групі за допомогою широкомовної передачі (рисунок 1.8). У поточній літературі запропоновано використання алгоритму кластеризації для вибору шлюзових БПЛА на основі географічної інформації. Ця модель маршрутизації ефективно вирішує проблему масштабованості та значно підвищує стабільність динамічної мережі.

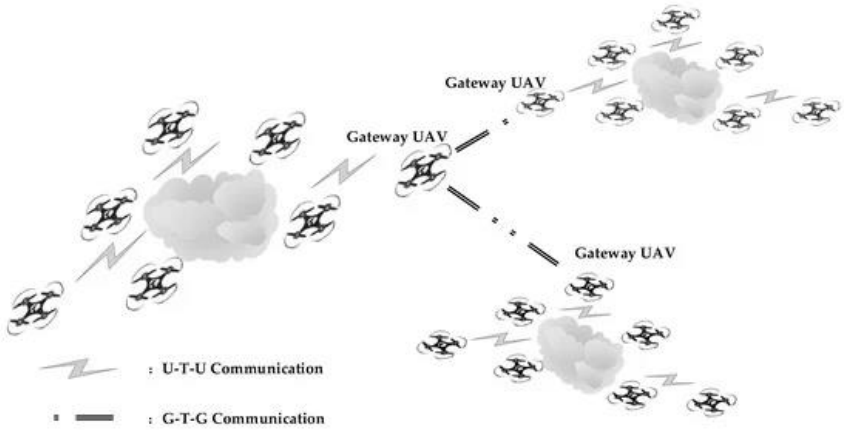


Рисунок 1.8 – Multilevel Hierarchical Routing

1.2.3.2 Proactive Routing Protocols

Proactive Routing Protocols (PRP) може періодично оновлювати таблицю маршрутизації, щоб підтримувати її в актуальному стані. Переваги полягають у тому, що він може точно відображати топологію мережі, а затримка в отриманні маршрутів мінімальна. Таким чином, PRP підходить для додатків з вимогами реального часу. Однак, щоб періодично оновлювати таблиці маршрутизації, кількість повідомлень постійно обмінюється між вузлами. Ось чому PRP не підходять для ройової мережі БПЛА, вузли якої рухаються швидко. Коли масштаб мережі та мобільність збільшуються до певного ступеня, більшість схем PRP не застосовуються. На даний момент найпопулярнішими протоколами проактивної маршрутизації є Optimized Link State Routing (OLSR), Destination Sequenced Distance Vector (DSDV) та їхні варіації.

Протокол Optimized Link State Routing (OLSR) — це протокол маршрутизації, оптимізований за станом зв’язку, спочатку для плоскої топології. Шляхом передачі стану зв’язку сусідів іншим вузлам відбувається обмін інформацією про топологію між вузлами. Вузли використовують алгоритм найкоротшого шляху до місць призначення, щоб вибрати наступний стрибок. K Singh та ін. застосували маршрутизацію OLSR у

літаючих ad hoc мережах, а потім проаналізували застосовність OLSR для літаючої ad hoc мережі. Деякі інші дослідники також порівнювали OLSR з іншими протоколами маршрутизації. У мережі БПЛА Ad hoc розташування вузлів і стан зв'язків швидко змінюються. Це призведе не тільки до повних втрат, але також створить навантаження на вже обмежену пропускну здатність. Оптимізація передбачає відмову від традиційного підходу невибіркового пересилання пакетів керування топологією (TC) і натомість прийняття методу вибору сусідів як multiple point relays (MPR). Завдяки цьому механізму OLSR зменшує накладні витрати на протокол. Технологія MPR, стратегія TC і технологія оптимізації маршрутизації стали основними засобами для розробки ефективних протоколів OLSR. Таким чином, в останні роки протоколи OLSR і алгоритм оптимізації MPR були широко вивчені, ставши найбільш часто використовуваними алгоритмами маршрутизації для ad hoc мереж. Хоча накладні витрати, викликані стратегією періодичного затоплення, зменшуються завдяки MPR, такі проблеми, як перерахунок маршрутів і вибір MPR, все ще мучать дослідників. Тому було розроблено та запропоновано багато варіантів. A.I. та ін. в запропонував міжрівневу конструкцію для мереж зв'язку БПЛА та далі запропонував протокол направленого OLSR (DOLSR) на основі спрямованих антен для спеціальних мереж БПЛА. DOLSR мінімізує затримку, зменшуючи кількість учасників MPR.

DSDV. У протоколі DSDV мережа має вузли для кожного шляху від джерела до місця призначення до того, як з'явиться попит. Кожен вузол у мережі має таблицю маршрутизації, у якій перераховані всі дійсні вузли призначення та кількість вузлів, які можуть досягти вузла призначення. Кожен запис у таблиці маршрутизації вузла призначення позначається серійним номером вузла. Щоб інформація в таблиці маршрутизації була узгодженою з топологією, що динамічно змінюється, її необхідно часто оновлювати. Оновлення таблиці маршрутизації DSDV поділяються на «full-dump» і «incremental-dump». «Full-dump» надсилає всю таблицю

маршрутизації своїм сусідам, що вимагає кількох пакетів; «incremental-dump» надсилає своїм сусідам лише змінені частини інформації про маршрутизацію, тому потрібен лише один пакет. Коли мережа відносно стабільна, «incremental-dump» може уникнути додаткового трафіку та зменшити навантаження на мережу. Однак у мережі з топологією, що швидко змінюється, як-от ройова мережа БПЛА, процес оновлення різко збільшує пропускну здатність мережі, а також ставить протокол DSDV у невигідне становище. У автори проаналізували адаптивність протоколу DSDV для ad hoc мереж БПЛА та порівняли продуктивність з іншими протоколами.

На додаток до OLSR і DSDV, є також деякі відносно нові протоколи проактивної маршрутизації. Better Approach to Mobile Ad hoc Networking (B.A.T.M.A.N.) проактивно зберігає інформацію про всі вузли та визначає найкращий наступний стрибок для кожного пункту призначення. D.S. Sandhu та ін. переглянули порівняльні дослідження B.A.T.M.A.N. Результати показують, що B.A.T.M.A.N. метод і OLSR поводяться подібно для невеликих розмірів мережі та мереж з обмеженою пропускною здатністю. Але в сценаріях великої кількості вузлів B.A.T.M.A.N. перевершує OLSR. Це може стати новим методом для протоколів маршрутизації спеціальної мережі БПЛА.

1.2.3.3 Reactive Routing Protocols

Reactive Routing Protocols (RRP) також називається протоколом маршрутизації на вимогу, оскільки він запускає процедуру виявлення маршруту лише тоді, коли потрібно надіслати повідомлення. Тому необхідний кеш інформації маршрутизації невеликий. Коли мережа не перевантажена, витрати на керування RRP набагато менші, ніж PRP. Однак, оскільки вихідний вузол повинен виконати виявлення маршруту перед передачею даних, RRP страждають від великих затримок передачі. Типові RRP включають Dynamic Source Routing (DSR), Ad hoc On-Demand Distance

Vector (AODV) тощо.

Dynamic Source Routing (DSR) є широко використовуваним і стандартизованим методом RRP для спеціальних мереж. Найбільш очевидною перевагою протоколу DSR є те, що він не має спеціальних вимог до мережевої інфраструктури. Цей протокол дозволяє мережі самоорганізовуватися та самоналаштовуватися. Кожен пакет даних у протоколі DSR містить повний список вузлів маршрутизації, тому всі вузли, які пересилають або прослуховують цей пакет, зберігатимуть інформацію маршрутизації як резервну копію. Таким чином, незалежно від того, як вузли рухаються і топологія мережі змінюється, вони можуть підтримувати продуктивність. Існує багато подальших досліджень адаптивності DSR у спеціальних мережах БПЛА.

Ad hoc On-Demand Distance Vector (AODV) - це метод RRP, заснований на DSDV і DSR. Немає підтримки інформації про глобальну маршрутизацію для всієї мережі, а лише про адреси призначення в пакетах даних. Тому накладні витрати низькі. Перед тим, як вузол надсилає дані, він спочатку шукає таблицю маршрутизації. Якщо існує один шлях до кінцевого вузла, пакет даних надсилається до наступного переходу відповідно до таблиці маршрутизації. Однак, якщо немає доступного шляху, вихідний вузол транслює Route Request (RREQ) сусіднім вузлам. Після отримання неповторюваного RREQ сусідні вузли встановлюють або оновлюють зворотні маршрути. Потім вони продовжують транслювати RREQ, доки вузол призначення не отримає. Якщо поточний вузол містить дійсний маршрут до вузла призначення, процес виявлення маршруту також може завершитися. Потім відповідь маршруту повертається до вихідного вузла по зворотному шляху. Коли відповідь повертається, встановлюється запис прямого маршруту, а коли вихідний вузол отримує RREP, він встановлює маршрут від джерела до вузла призначення. В останні роки було проведено багато досліджень із застосування AODV до мереж зв’язку БПЛА, і деякі також запропонували варіанти AODV.

1.2.3.4 Hybrid Routing Protocols

Щоб подолати величезні накладні витрати на контрольні повідомлення в PRP і велику затримку процесу виявлення маршруту в RRP, були введені Hybrid Routing Protocols (HRP). HRP розділяє великомасштабні мережі Ad hoc на зони. PRP використовується між сусідніми вузлами в одній зоні, тоді як RRP використовується між вузлами в різних зонах. Налаштування стратегії маршрутизації відповідно до характеристик мережі може не тільки зменшити накладні витрати на маршрутизацію, але й зменшити велику затримку виявлення маршруту. Однак дуже динамічний характер спеціальних мереж БПЛА ускладнює HRP отримання та підтримку інформації. Найбільш типовими HRP є Zone Routing Protocol (ZRP) і Temporarily Ordered Routing Algorithm (TORA).

ZRP - це протокол маршрутизації, який повністю використовує топологію зони маршрутизації. Діапазон застосування PRP обмежений зоною, яка вимірюється кількістю стрибків. Відповідно, зменшується кількість і поширення пакетів керування маршрутизацією. При спілкуванні з вузлами поза зоною вибирається RRP. У порівнянні з простим PRP, ZRP може значно зменшити витрати на маршрутизацію. У порівнянні з RRP його збіжність часових маршрутів і затримки передачі відносно малі. У разі великого навантаження на мережу продуктивність ZRP не сильно знижується, що робить його протоколом маршрутизації з великим потенціалом розвитку. В даний час радіус зони в ZRP зазвичай вказується заздалегідь, що обмежує адаптивність протоколу. Таким чином, удосконалення ZRP з радіусами адаптивних зон було гарячою точкою дослідження.

TORA є гібридним протоколом розподіленої маршрутизації з високою адаптивністю. Розподілений алгоритм втілюється в обслуговуванні інформації про маршрутизацію. Кожен вузол лише оновлює інформацію про

маршрути сусідніх вузлів. У відповідь на високу мобільність БПЛА TORA мінімізує чутливість до змін топології, щоб обмежити поширення керуючих повідомлень. Варто зазначити, що алгоритм найкоротшого шляху не є вибором TORA. Замість цього він зазвичай використовує довші маршрути, щоб швидко знаходити нові маршрути в разі збою зв’язку. У TORA різні значення «висоти» використовуються для позначення вузлів БПЛА. На основі цього будується спрямована структура ациклічної маршрутизації для досягнення ефективної переадресації трафіку.

1.2.4 Geographic/Position-Based Routing Protocols

Через високу мобільність вузлів у ad hoc мережі БПЛА важко підтримувати таблицю маршрутизації. У той же час традиційні протоколи маршрутизації призведуть до значних накладних витрат на повторний пошук маршрутизації перед надсиланням пакетів. У відповідь на ці проблеми дослідники запропонували протоколи маршрутизації на основі інформації про географічне положення. У цьому типі протоколу вузли можуть використовувати служби визначення місцезнаходження, такі як reactive location services (RLS), grid location services (GLS) або hierarchical location services (HLS). Протокол маршрутизації на основі географічного розташування більше підходить для високодинамічних мереж, включаючи мережі зв'язку БПЛА.

GPSR є протоколом маршрутизації на основі позиції, який спочатку був запропонований для мобільних ad hoc мереж. У порівняльному дослідженні автори порівняли проактивний, реактивний і протоколи маршрутизації на основі позиції в групі БПЛА. Результати показують, що продуктивність протоколу GPSR краща за інші. Shirani та ін. досліджував протоколи маршрутизації на основі позиції, включаючи GPSR для літаючих мереж Ad hoc. Вони роблять висновок, що протокол GPSR застосовний для щільно розгорнутої мережі БПЛА. Однак узагальнення та надійність цього

протоколу маршрутизації ще не покращено, що призвело до появи багатьох варіантів.

Geographic Load Share Routing (GLSR), запропонована Medina та ін. є розширенням GPSR. Основний внесок полягає в тому, що GLSR використовує «відстань випередження», щоб вибрати відповідний наступний стрибок із кількох шляхів від вузла джерела до вузла призначення, таким чином роблячи шлях більш надійним.

Базуючись на тому ж принципі, що й GPSR, Lin et al. запропонував Mobility Prediction-based Geographic Routing (MPGR). Інновацією цього протоколу є використання методу прогнозування руху на основі функції Гауса. Результат передбачення руху можна використовувати для оцінки зв'язку вузлів БПЛА, а також для вибору надійного наступного стрибка. В іншому дослідженні було запропоновано маршрутизацію, Geographic Position Mobility Oriented Routing (GPMOR). Вузли БПЛА можуть отримувати власну інформацію про місцезнаходження через глобальну систему позиціонування (GPS). Кожен БПЛА періодично обмінюється інформацією про місцезнаходження з сусідніми вузлами. GPMOR також використовує «модель мобільності Гаусса-Маркова» для прогнозування руху БПЛА.

Shirani та ін. поєднує протокол AODV з технологією Greedy Geographic Forwarding (GGF) і додатково запропонованим протоколом маршрутизації Reactive-Greedy-Reactive (RGR). Щоб отримати адресу призначення, RGR використовує інформацію про місцезнаходження БПЛА. Структура AODV RREQ / RREP і технологія GGF доповнюють одна одну. Протокол спочатку знаходить початковий шлях і здійснює передачу даних за способом протоколу AODV. Якщо передача випадково обривається, стратегія перемикається в режим GGF. Цей протокол покращує коефіцієнт доставки пакетів і зменшує затримку. Однак через високу мобільність БПЛА, якщо інформація про географічне положення наступного стрибка не оновлюється вчасно, виникне помилка втрати пакетів. Тим не менш, RGR має великий потенціал застосування для відстеження, пошуку та багатозадачності роїв

БПЛА.

1.2.5 Swarm Intelligence-Based Routing Protocols

Swarm intelligence (SI) — багатоагентна система, що самоорганізується. Натхненні моделями роїння риб, птахів, комах тощо, ми можемо застосувати SI до мобільних роботів. Крім того, SI пропонує нові ідеї для оптимізації складних спільних завдань. У протоколі маршрутизації ройових систем БПЛА також є варіанти SI. Щоб вирішити проблеми розширення діапазону зв'язку та витоку інформації, які викликані збільшенням масштабу рою, одне дослідження запропонувало Improved Artificial Fish-Swarm Algorithm (IAFSA). Шляхом коригування топології групи запропонована модель бездротового зв'язку досягає цілей безпечного зв'язку. Крім того, існують інші протоколи маршрутизації, спеціально розроблені для зв'язку БПЛА, такі як спеціальна мережа Bee colony algorithm-based Ad hoc (BeeAdhoc) і маршрутизація з урахуванням поліморфізму на основі оптимізації колонії мурашок (APAR).

1.3 Стратегії забезпечення надійності мережі

Мережа БПЛА є відносно новою галуззю досліджень, і це дуже особливий вид мережі з багатьма технічними проблемами. Незважаючи на те, що БПЛА часто використовується як мобільний вузол для відновлення з'єднання WSN, і були проведені деякі дослідження проблеми стійкості до пошкоджень у WSN та інших сферах, але досі немає повідомлень про проблему відновлення серйозно пошкоджених мереж USNET.

Існує дві стратегії для вирішення проблеми стійкості до пошкоджень мережі: проактивна та реактивна (рисунок 1.9).

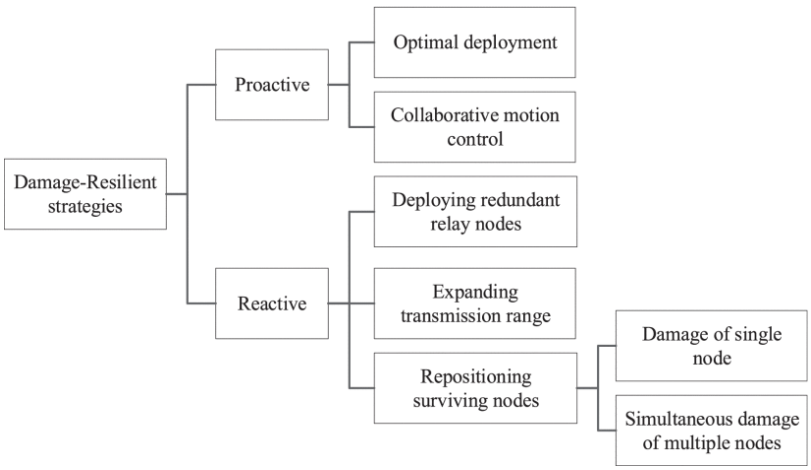


Рисунок 1.9 – Класифікація стратегій надійності до пошкоджень

1.3.1 Proactive Strategy

Проактивна стратегія зменшує ймовірність розбиття мережі за рахунок покращення або підтримки підключення до мережі шляхом оптимального розгортання або спільного керування рухом вузлів, тим самим покращуючи стійкість мережі до пошкоджень. Наприклад, Han et al. представили алгоритм для покращення підключення MANET шляхом інтелектуального розгортання та руху БПЛА. Більшість існуючих стійких до пошкоджень механізмів USNET використовують проактивну стратегію, яка зосереджується на тому, як підтримувати зв’язок USNET. Ajorlou та ін. запропонували клас розподілених законів керування на основі потенціалу для уникнення роз’єднання краю в графі потоку інформації. Дутта та ін. представили децентралізований контролер для кількох БПЛА, щоб створити орієнтоване на ціль утворення, зберігаючи задану алгебраїчну зв’язність. Еспозіто та ін. запропонував закон керування, заснований на потенціалі, щоб направляти рій роботів від початкової позиції до кінцевої позиції, зберігаючи бажані зв’язки протягом усього руху. Однак у згаданій вище роботі вивчалося лише те, як підтримувати з’єднання з мережею, коли вузли добре працюють, і не впоралося з відключенням мережі, спричиненим пошкодженням масових вузлів.

1.3.2 Reactive Strategy

Реактивна стратегія зосереджена на відновленні з’єднання роз’єднаних підмереж, і її можна класифікувати на три категорії: розгортання надлишкових ретрансляційних вузлів між роз’єднаними підмережами, розширення діапазону передачі вузлів для об’єднання роз’єднаних підмереж і зміна розташування вцілілих вузлів для відновлення з’єднання.

1.3.2.1 Deploying Redundant Relay Nodes

Lee та ін. запропонував алгоритм відновлення підключення з гарантованою відмовостійкістю (CRAFT) для відновлення розділеного WSN і формування двозв’язаної топології між розділами, яка є толерантною до відмови одного вузла. Метою CRAFT є мінімізація максимальної довжини шляху між парами розділів і розгортання найменшої кількості вузлів ретрансляції. Ця категорія методів використовує переваги маневреності та гнучкості БПЛА, але розгортання резервних вузлів ретрансляції значно збільшить вартість USNET через відносно високу вартість вузлів БПЛА. Крім того, резервні резервні вузли можуть бути пошкоджені одночасно, коли мережа серйозно пошкоджена.

1.3.2.2 Expanding Transmission Range

Ця категорія методів зазвичай вимагає додаткового обладнання. У односпрямовані антени використовуються для розширення дальності зв’язку вузлів, тим самим покращуючи зв’язок WSN. Tian та ін. запропонував алгоритм відновлення зв’язку для мереж БПЛА, який використовує кооперативну комунікаційну технологію для встановлення зв’язку на великій відстані між розділеними частинами мережі, щоб зменшити рух вузлів. Якщо

спільні канали зв’язку не можуть бути встановлені, вузли можуть заздалегідь переміститися до кращих місць для встановлення зв’язків. Однак вузлу групи БПЛА важко нести додаткові апаратні пристрої, такі як односпрямовані антени, через обмежене корисне навантаження, тому ця категорія методів може бути незастосовною.

1.3.2.3 Repositioning Surviving Nodes

Технологія відновлення пошкоджених мереж зі збереженими ресурсами була вивчена в WSN, і ця категорія методів відносно більше підходить для USNET. Відповідно до кількості пошкоджених вузлів, які можна прийняти, цю категорію методів можна розділити на дві підкатегорії: перша може мати справу лише з проблемою розбиття мережі, спричиненою пошкодженням одного вузла. Другий може переносити одночасне пошкодження кількох вузлів.

Методи першої підкатегорії відновлюють зв’язок шляхом переміщення вцілілого вузла до місця пошкодженого вузла зрізаної вершини. Методи відновлення зв’язку, мають подібну ідею, яку можна підсумувати як пошук вузлів розрізної вершини в мережі, а потім, якщо вузол розрізаної вершини виявлено недоступним, відновлення зв’язку каскадним рухом відповідних вузлів. Однак каскадний рух спричиняє багато накладних витрат на зв’язок, оскільки кожен рухомий вузол транслює повідомлення своєму сусідові перед переміщенням. Шарма та ін. запропонував схему виявлення та відновлення на основі зони (ZBFR), яка враховувала як підключення, так і покриття. Процес відновлення прагне спільно відновити покриття та підключення шляхом рекурсивного переміщення деяких мобільних вузлів і перевірки резервних вузлів. Мі та ін. запропонував стратегію відновлення зв’язку з уникненням перешкод (OCRS), яка відновлює зв’язок, вибираючи резервний вузол для кожного можливого вузла розсікаючої вершини та спрямовуючи резервний вузол до місця розташування несправного вузла, коли можливий

вузол розсікаючої вершини виходить з ладу . На відміну від інших робіт, OCSR розглядав динаміку вузла під час виконання процесу відновлення. Однак наведені вище методи на основі вирізання вершин не можуть впоратися з одночасним виходом з ладу кількох вузлів, у той час як USNET, що працюють у жорстких середовищах, може мати велику кількість пошкоджених вузлів одночасно. Методи першої підкатегорії відновлюють зв'язок шляхом переміщення вцілілого вузла до місця пошкодженого вузла зрізаної вершини. Методи відновлення зв'язку, які можна підсумувати як пошук вузлів розрізної вершини в мережі, а потім, якщо вузол розрізаної вершини виявлено недоступним, відновлення зв'язку каскадним рухом відповідних вузлів. Однак каскадний рух спричиняє багато накладних витрат на зв'язок, оскільки кожен рухомий вузол транслює повідомлення своєму сусідові перед переміщенням. Шарма та ін. запропонував схему виявлення та відновлення на основі зони (ZBFR), яка враховувала як підключення, так і покриття. Процес відновлення прагне спільно відновити покриття та підключення шляхом рекурсивного переміщення деяких мобільних вузлів і перевірки резервних вузлів. Мі та ін. запропонував стратегію відновлення зв'язку з уникненням перешкод (OCRS), яка відновлює зв'язок, вибираючи резервний вузол для кожного можливого вузла розсікаючої вершини та спрямовуючи резервний вузол до місця розташування несправного вузла, коли можливий вузол розсікаючої вершини виходить з ладу . На відміну від інших робіт, OCSR розглядав динаміку вузла під час виконання процесу відновлення. Однак наведені вище методи на основі вирізання вершин не можуть впоратися з одночасним виходом з ладу кількох вузлів, у той час як USNET, що працюють у жорстких середовищах, може мати велику кількість пошкоджених вузлів одночасно.

Методи другої підкатегорії відновлюють підключення шляхом переміщення вузлів/розділів до попередньо визначеної точки зустрічі або узгодження рішення для відновлення в точці зустрічі вузлами-переговорниками. Джоші та ін. представили підхід розподіленого

відновлення ресурсних обмежень (RCR) у випадку, коли вцілілих мобільних вузлів недостатньо для формування стабільної міжсегментної топології. Коли мережа розділена на кілька сегментів, кожен сегмент заповнює ретрансляційний вузол до точки зустрічі (припустімо, що кожен сегмент має принаймні один мобільний вузол). Потім вузли ретрансляції поділяються на стаціонарні вузли ретрансляції та мобільні збирачі даних (MDC) на основі попередньо визначеного критерію (наприклад, залишкова енергія). MDC використовуються для забезпечення періодичного з'єднання між сегментами. Представлений алгоритм розподіленого автономного відновлення (AuR) для вирішення проблеми розбиття мережі через відмову кількох вузлів. AuR імітує міжмолекулярну взаємодію, щоб поширити розділ у напрямку втрати, щоб мати можливість з'єднатися з іншими розділами, і розділ буде переміщуватися каскадним способом до точки зустрічі, якщо він не зустрічається з іншими розділами. Розділ повторює процес AuR, доки не досягне точки зустрічі або не буде підключено до вузла, розташованого як точка зустрічі. Схема AuR може впоратися з одночасним збоєм кількох вузлів, але вона вимагає, щоб коли вузол рухається, його сусіди залишалися нерухомими, таким чином подовжуючи час відновлення з'єднання. Крім того, кожен рухомий вузол повинен надіслати свою позицію своєму сусіду перед переміщенням, що збільшує накладні витрати на зв'язок. Шрівастав та ін. запропонував підхід для відновлення зв'язку WSN за допомогою переговорів за круглим столом (RTN). Метою RTN є мінімізація часу для повторного підключення, а також мінімізація кількості розгорнутих вузлів і загальної пройденої відстані. Ідея RTN полягає у виборі вузла з кожного розділу, який є найближчим до точки зустрічі, як учасник переговорів. Потім ці учасники переговорів переміщуються в кругову зону навколо місця зустрічі (називається круглим столом), щоб узгодити найкоротші шляхи повторного підключення та вузли заміни. Нарешті учасники переговорів повертаються до своїх початкових місць і об'єднують вузли розділу, щоб перемістити в потрібне положення для відновлення з'єднання. Підхід RTN

вимагає багатьох комунікаційних і обчислювальних ітерацій, що призводить до відносно високої складності часу та накладних витрат на зв’язок. Крім того, він не враховує вплив затримки мережі.

1.4 Висновок до розділу

Протоколи статичної маршрутизації не підходять для мереж з динамічною топологією, як-от Ad hoc мережа UAV swarm через фіксовані таблиці маршрутизації та меншу масштабованість. Протоколи проактивної маршрутизації мають великі накладні витрати на підтримку таблиць в актуальному стані. Крім того, їх повільна реакція на зміни топології призводить до затримок. Основним недоліком протоколів реактивної маршрутизації є висока затримка при пошуку маршрутизації. Маршрутизація джерела погано масштабується, оскільки великі накладні витрати мережі можуть збільшитися через великий розмір заголовка. Гібридні протоколи маршрутизації поєднують проактивний і реактивний протоколи, щоб подолати обмеження. Однак у мережах БПЛА динамічні вузли та поведінка каналів ускладнюють отримання та підтримку інформації. Таким чином, протоколи маршрутизації на основі топології не підходять для сценаріїв, що характеризуються високим динамічним рухом і великою кількістю вузлів. Протоколи маршрутизації на основі географічного розташування додають атрибут географічного розташування до вузлів БПЛА. Це робить його видатним у роботі з високою мобільністю вузлів і частою зміною топології в мережах БПЛА.

Більшість існуючих досліджень ігнорують вплив факторів зв’язку, таких як затримка мережі та втрата пакетів, але ці фактори мають великий вплив на механізм відновлення, особливо бездротові канали зв’язку чутливі до навколишнього середовища або сигналів на одній частоті. Оскільки частий зв’язок може призвести до великої кількості колізій, що призведе до великої затримки або втрати пакетів. Крім того, оскільки виявлені підмережі

будуть об’єднані під час виконання процесу відновлення, алгоритм повинен враховувати динаміку вузла та контролювати рух вузлів у режимі реального часу, а також слід враховувати вплив маршрутизації через те, що для цього потрібно а для встановлення маршрутизації між зустрічними підмережами. Крім того, основною метою більшості існуючих досліджень є мінімізація відстані пересування вузлів для зменшення споживання енергії. Однак, оскільки енергія, споживана вузлами БПЛА в стані висіння та в стані руху, не сильно відрізняється, зменшення відстані подорожі не має особливого сенсу.

2 МОДЕЛЬ ПОШКОДЖЕННЯ ТА ФОРМУЛЮВАННЯ ПРОБЛЕМИ

2.1 Модель пошкодження мереж USNET

USNET, який розглядається, має попередньо визначену мережеву структуру та траєкторію польоту (включаючи ряд маршрутних точок) відповідно до своєї місії. Розподілений алгоритм виборів, такий як Bully, використовується для визначення головного вузла. Подібним чином, якщо USNET розділено на кілька підмереж, кожна підмережа автоматично вибере головний вузол, який буде керувати підмережею. Головний вузол відповідає за різноманітні функції, такі як сприйняття станів інших вузлів у підмережі, обчислення розташування підмережі та об'єднання підмереж.

Нехай змінний у часі неорієнтований граф $\mathrm{G(t)} = \{\mathrm{U(t)}, \mathrm{E(t)}\}$ є мережею USNET, яка виконує певну місію, де $U(t) = \{u_i \mid i = 1,2, \dots, n\}$ позначають n вузлів UAV USNET у момент часу t, $\mathrm{E(t)} = \{e_{ij} \mid u_i \in \mathrm{U(t)}, u_j \in \mathrm{U(t)}\}$ позначають двонаправлені широкосмугові бездротові з'єднання між вузлами в момент часу t. Передбачається, що кожен вузол несе модуль позиціонування, такий як система глобального позиціонування (GPS), щоб отримати його поточне положення, і вузол може самостійно переміщатися в певне місце за допомогою літака. Крім того, за допомогою датчиків вузол може відчувати зовнішні ситуації і не стикається з іншими вузлами під час руху. Нехай $q_i(\mathrm{t}) \in \mathbb{R}^3$ позначає положення u_i вузла БПЛА в момент часу t, $d_{ij}(\mathrm{t}) = \|q_i(\mathrm{t}) - q_j(\mathrm{t})\|$ позначає ейлерову відстань між двома вузлами u_i та u_j в момент часу t. Край $e_{ij}(\mathrm{t}) \in \mathrm{E(t)}$ існує тоді і тільки тоді, коли $d_{ij}(\mathrm{t}) \leq$ R, де R – дальність передачі вузла БПЛА.

У добре працюючому USNET усі вузли підключені, а вся мережа рухається по траєкторії польоту. Перш ніж зануритись у формулювання

проблеми визначимо кілька термінів та символів.

1 Нехай $c_{i,j}(\mathrm{t})$ буде булевою змінною, яка показує, чи є u_i та u_j сусідами чи ні. Тобто,

$$c_{i,j} = \begin{cases} 1 & if\ e_{ij} \in E(t) \\ 0 & otherwise \end{cases}, \tag{2.1}$$

2 Нехай $N_i(t)$ — набір сусідів UAV u_i у момент часу t. Тобто,

$$N_i(t) = \{u_j | c_{i,j} = 1\}, \tag{2.2}$$

3 Набір $S(t)$ вважається підключеною підмережею в момент часу t, якщо $S(t) \subseteq U(t)$ і $\forall u_i \in S(t)$ можуть досягати інших вузлів $u_i \in S(t)$.

4 Набір $S(t)$ називається максимально підключеною підмережею, якщо не існує іншої підключеної підмережі $S'(t)$, такої, що $S'(t) \supset S(t)$. Нехай $\mathbb{S} = \{S_i(t) \mid i = 1,2,\dots,m\}$ - набір із m максимальних підключених підмереж, $\cup_{i=1}^{m} S_i(t) = U(t)$. Для простоти виразу всі згадані нижче підмережі стосуються максимальної підключеної підмережі.

Нехай час $t_1 < t_2 < \cdots < t_n$ позначає дискретний часовий ряд, $f_p(t_n) \in \mathbb{R}^3$ позначає попередньо визначену маршрутну точку в момент часу t_n. USNET повинен літати вздовж маршрутних точок у цілому під час місії. Припустимо, що USNET летить по прямій лінії з рівномірною швидкістю між двома попередньо визначеними точками маршруту, безперервна функція попередньо визначеної траєкторії польоту $f_p(t_n) \in \mathbb{R}^3$ може бути отримана з послідовності точок шляху, як показано в (3).

$$f_p(t) = f_p(t_{k-1}) + \frac{(t-t_{k-1})[f_p(t_k)-f_p(t_{k-1})]}{t_k-t_{k-1}}, \tag{2.3}$$

де $t_{k-1} \leq t \leq t_k$;

5 Нехай δi(t) позначає відстань між підмережею Si(t) і точкою шляху в момент часу t, вона визначається як мінімальна відстань між усіма вузлами $u_j \in S_i(t)$ і точка шляху в момент часу t. Тобто,

$$\sigma_i(t) = \min_{u_i \in S_i(t)} \left\| q_i(t) - f_p(t) \right\|, \tag{2.4}$$

6 USNET вважається добре працюючим USNET у момент часу t, якщо він задовольняє такі умови:

1) U(t) – максимальна підключена підмережа, що включає всі збережені вузли;

2) Відстань між U(t) і маршрутною точкою в момент часу t не перевищує $\frac{R}{2}$, тобто

$$\min_{S_i(t) \in \mathbb{S}} \sigma_i(t) \leq \frac{R}{2}, \tag{2.5}$$

Добре працюючий USNET означає, що всі збережені вузли з’єднані між собою, а відстань між будь-яким вузлом і маршрутною точкою не перевищує $\frac{R}{2}$. Під час ініціалізації USNET перебуває в нормальному робочому стані, і після серйозного пошкодження він розділений на кілька непересічних підмереж довільної форми. Стан, у якому перебуває пошкоджений USNET, називається пошкодженим станом. На рис. 2.1a наведено приклад добре працюючої мережі USNET, де чорне суцільне коло позначає вузол, що вижив, порожнє коло позначає пошкоджений вузол, заштрихована сіра область позначає підключену підмережу, що складається з вузлів, що вижили, а червоний квадрат позначає поточна маршрутна точка. На рис. 2.1b, рис. 2.1c і рис. 2.1d показано деякі можливі структури сильно пошкодженої USNET.

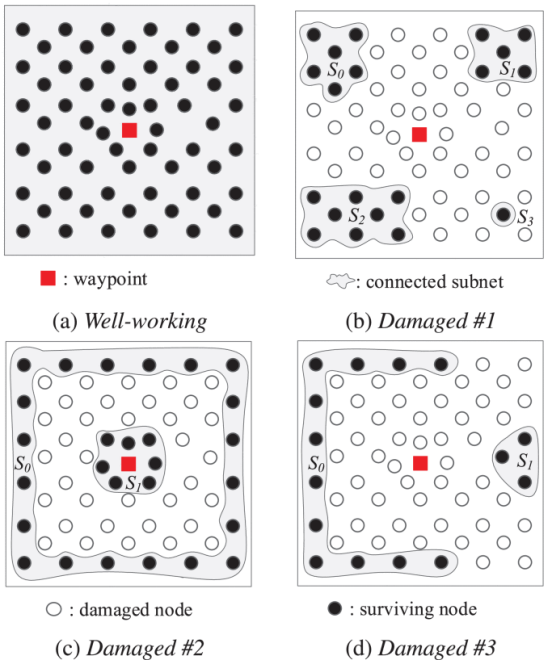


Рисунок 2.1 – Приклад USNET у жорстких умовах

2.2 Формулювання проблеми

Коли мережа серйозно пошкоджена, вузли можуть переміщатися у відповідне місце зі швидкістю $v_i(t)$, щоб швидко об'єднати декілька непересічних підмереж у з'єднану мережу, де $v_i(t) \in \mathbb{R}^3$ - вектор, який позначає швидкість вузла u_i на час t . Динаміка u_i регулюється такою формулою:

$$\begin{cases} q_i(t) = v_i(t) \\ 0 \le \|v_i(t)\| \le V_{max} \end{cases}, \tag{2.6}$$

де $V_{max} \in \mathbb{R}$ – максимальна швидкість польоту UAV.

У цьому досліджені передбачається, що максимальна швидкість польоту визначається відповідно до зовнішніх сил, що діють на БПЛА, тобто вітер та інші сили були враховані при розрахунку V_{max}. Тим часом передбачається, що зовнішні сили на всіх вузлах в одній підмережі в

основному однакові, тому максимальна швидкість польоту кожного вузла в одній підмережі майже не відрізняється; таким чином вузли в підмережі не будуть відставати під час польоту в строю.

Висока стійкість USNET до пошкоджень означає здатність усіх уцілілих вузлів відновлювати USNET після серйозного пошкодження. Час відновлення $T_{recovery}$ є ключовим показником для вимірювання здатності USNET до пошкоджень, і він відноситься до часу відновлення мережі з пошкодженого стану до нормального робочого стану (що задовольняє умови визначення 2.6).

Нехай $t_{damaged}$ позначає час, коли USNET пошкоджено, $t_{recovery}^{i}$ позначає час, коли і-та підмережа відновлюється до нормального робочого стану, час, потрібний USNET для завершення відновлення:

$$T_{recovery} = \max_{S_i(t)\in\mathbb{S}} (t_{recovery}^{i} - t_{damaged}), \tag{2.7}$$

Метою дослідження є розгляд та порівняння методів керування вузлів БПЛА під час пост аварійного стану. Іншими словами, кожен уцілілий вузол повинен рухатися в певне місце з відповідною швидкістю, щоб USNET можна було відновити до нормального робочого стану. Метою цієї роботи є:

$$v(t) = \arg\min_{v(t)} T_{recovery}, \tag{2.8}$$

Наступні обмеження повинні бути виконані для досягнення цілі, показаної в (2.8).

Першим обмеженням є час відновлення T, тобто час, необхідний для відновлення зв’язку між усіма вузлами, що вижили після пошкодження. T пов’язане з розміром USNET, зоною покриття та максимальною швидкістю польоту вузла UAV. Це обмеження показано в (2.9).

1) T-обмежене зв'язне обмеження: задані часи $t_1 \le t_2 \le \cdots \le t_{k-1}$, для $\forall u_{i_1}, u_{i_k} \in U(t)$, $\exists u_{i_2}, u_{i_3}, \ldots, u_{i_{k-1}} \in U(t)$, такий як

$$\begin{cases} \prod_{j=1}^{k-1} c_{i_j, i_{j+1}}(t_j) = 1 \\ |t_{k-1} - t_1| < T \end{cases}, \tag{2.9}$$

Друге обмеження вимагає, щоб усі вцілілі вузли відстежували траєкторію польоту під час процесу відновлення. Обмеження виражається у (2.10);

2) Обмеження доріжки: за часів $t_{k-1} \le t \le t_k$ для $\forall u_i \in U(t)$, він завжди має компонент швидкості $v_f^i(t)$ для відстеження траєкторії польоту. Тобто,

$$v_f^i(t) = \frac{f_p(t_k) - f_p(t_{k-1})}{t_k - t_{k-1}}, \tag{2.10}$$

Щоб дозволити вузлам UAV виконувати операції відновлення мережі під час відстеження траєкторії польоту, можна зробити припущення, що швидкість відстеження траєкторії польоту менше максимальної швидкості польоту V_{max}. Тобто $\left\| v_f^i(t) \right\| < V_{max}$.

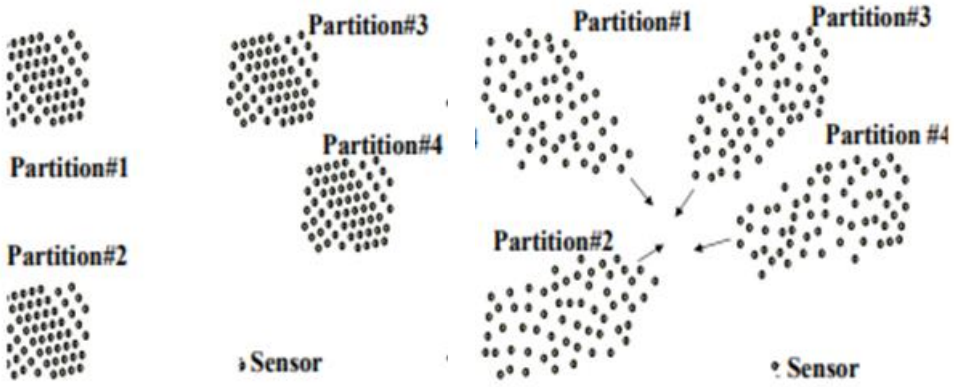
3 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ МЕРЕЖІ

3.1 Autonomous Recovery

Autonomous Recovery (Aur) - це розподілений підхід, який використовує мобільність вузлів для відновлення міцного зв’язку в мережі шляхом переміщення непересічних блоків вузлів один до одного та до центру області розгортання. Принцип розробки AuR базується на моделюванні зв’язності між сусідніми вузлами як модифікованої електростатичної взаємодії на основі закону Кулона між зарядами. Сусіди невдалих вузлів керують процесом відновлення, розподіляючись у зоні розгортання, щоб знайти інші вузли та тягнучи свої блоки до центру області. Відновлення локалізоване за допомогою вузлів, які взаємодіють лише зі своїми безпосередніми сусідами. Ефективність AuR перевірена моделюванням і показала ефективність повторного підключення всіх вузлів, пом’якшуючи частину втраченого покриття.

Якщо мережа буде розділена через збій групи вузлів, кожен сусід SN цих вузлів застосовуватиме AuR. Очевидно, що ці вузли будуть знаходитися на периферії свого розділу. Таким чином, ці периферійні вузли в кожному розділі застосовуватимуть AuR і рухатимуться назовні, а інші вузли всередині розділу слідуватимуть за ними. Сукупний ефект нагадує розтягування топології внутрішнього розділу до несправних вузлів, як показано на малюнку 3.1. Потім SN виявить що він не міг з’єднатися зі своїми колишніми сусідами з 1-хопом і знову застосувати AuR. Коли топологія не може розтягнутися далі, вузли рухатимуться до центру області. Оскільки AuR застосовуватиметься одночасно всіма розділами, розділи поступово просуваються всередину, доки не з’єднаються з іншим розділом або не досягнуть центру зони розгортання. Як тільки розділ досягає центру або коли він знову з’єднується з іншим розділом, під’єднаним до центру,

виконується остаточне саморозповсюдження для максимального покриття. Варто зазначити, що територія, охоплена об’єднаною мережею, утвореною AuR, швидше за все, буде більшою, ніж об’єднані регіони, охоплені окремими розділами. Це можна пояснити тим фактом, що AuR вирішує мінімізувати відстань подорожі шляхом саморозповсюдження до та в кінці процесу відновлення. Таким чином, топологія розтягується, а перекривання покриття між вузлами зменшується. Ми точно визнаємо, що AuR може відновити покриття у внутрішній частині мережі за рахунок зовнішньої частини. Однак AuR максимізує охоплення з урахуванням доступного набору вузлів.



а) б)

Рисунок 3.1 – Ілюстрація того, як AuR відновлює зв’язок у розділеній мережі: а) розподілення вузлів всередині окремих блоків у напрямку до недоступних частин мережі; б) переміщення розподілених вузлів до центру зони розгортання

Детальні кроки алгоритму:

1) початкове налаштування мережі та список сусідів 1-Hop: під час налаштування мережі кожен вузол транслює повідомлення серцевого ритму, щоб представити себе своїм сусідам, і створює список 1-hop, що складається з вузлів у межах його діапазону зв’язку. Цей список оновлюється під час

роботи мережі, щоб відобразити зміни в топології мережі та статусі вузла. Кожен запис у списку містить ідентифікатор вузла, положення та ступінь вузла. Перед переміщенням вузол повідомить сусідам про своє нове розташування, щоб їх таблиця була точною. Для кожного вузла в списку включено два додаткові атрибути: стан підключення вузла та центральна сила. Ці атрибути використовуються лише під час відновлення мережі. Стан підключення вузла відстежує, чи підключений він до центру області розгортання безпосередньо чи через своїх сусідів. Щоразу, коли вузол підключається або досягає центру, він встановлює свій стан підключення та інформує своїх сусідів, які роблять те саме. Атрибут центральної сили відстежує, чи можливе для вузла розповсюдження в його мережі з 1 стрибком. Якщо встановлено, це означає, що сили, які діють на вузол, збалансовані, і він не може рухатися далі, і залежно від кроку процесу відновлення він спричиняє різні ефекти, які будуть детально пояснені пізніше;

2) виявлення збою та ініціювання процесу відновлення: вузли надсилають періодичні повідомлення своїм сусідам, постійні промахи вказують на збій. Процес відновлення починається з того, що кожен сусід несправного вузла оновлює свій список 1-хопів, щоб відобразити збій. Ступінь вузла ділиться на дві частини; поточний ступінь вузла «β», який є кількістю живих сусідів з 1-хопом, і ступінь невдалого вузла «P», який є кількістю невдалих вузлів з 1-хопом, які втратили зв’язок;

3) порядок переміщення з саморозповсюдженням: кожен вузол виконує AuR, коли виявляє збій одного або кількох своїх сусідів. Оскільки одночасний неузгоджений рух вузлів може призвести до небажаних розривів існуючих зв’язків між вузлами, коли один вузол рухається, його сусіди повинні залишатися нерухомими. Початковий порядок переміщення визначається змаганням, причому переможець переміщається першим, а інші залишаються на місці в очікуванні своєї черги. Пріоритет надається в такому порядку.

1 Кількість втрачених сусідів: несправність багатьох сусідів вказує на те, що в цьому напрямку можуть бути інші непересічні блоки. Таким чином, краще рухатися в цьому напрямку, щоб мати більше шансів відновити з'єднання з іншими вцілілими вузлами, а також відновити покриття в області, яка зараз позбавлена вузлів. Кожен вузол обчислює кількість втрачених сусідів «P» і кількість живих сусідів «β» у своїй області 1-хопу, транслює свої значення «P» і «β» своїм здоровим сусідам з 1-хопом і порівнює свої «P» і значення 'β' до значення його сусідів. Вузол, який має найвище значення «P» у своєму околиці з 1-хопом, отримує право першим переміститися та лідирує у відновленні, і надалі його називають провідним вузлом;

2 Поточний ступінь вузла: вузли без втрачених вузлів мають P =0, і їхній пріоритет визначається на основі поточного ступеня вузла «β». Більший «β» означає вищий пріоритет. Обґрунтування полягає в тому, що вузли з вищим β затягнуть за собою багато сусідів і підвищать рівень саморозповсюдження та, як наслідок, прискорять відновлення;

3 Ідентифікатор вузла: якщо два вузли мають однаковий P або β, вузол із нижчим ідентифікатором рухається першим.

Мережа, зображена на (рисунок 3.2), відчуває одночасні збої вузлів S_0, S_5, S_6 і S_7. S1 має втрачений ступінь вузла два (P=2), оскільки він був у прямому контакті з несправними вузлами S_6 і S_7, тоді як його сусід S_9 має втрачений ступінь вузла один (P=1) через те, що він є сусідом S_0. Тому S_1 має найвищий пріоритет переміщення, за яким слідує S_9, і стає провідним вузлом. Хоча S_{10}, S_{12} і S_{13} мають однаковий ступінь вузла (β=2), S_{10} отримує вищий пріоритет, оскільки він має найменший ідентифікатор вузла. AuR прагне відновити зв'язок із загубленими вузлами, такими як S_4, які не мають живих сусідів, переміщуючи їх до центру зони розгортання, поки вони не досягнуть центру або не вступлять у контакт з іншими вузлами. Обґрунтування переходу до центру полягає в тому, що вцілілі вузли зрештою зійдуться там, отже, забезпечуючи осиротілому вузлу найкращі шанси на повторне підключення. Якщо вузол-сирота вступає в контакт з іншими

вузлами, він переміщується на основі рейтингу в межах свого відповідного околиці з 1-хопом;

4 Закони, що керують рухом вузла: AuR імітує міжмолекулярну взаємодію та використовує принцип електростатичного притягання та відштовхування, щоб поширюватися в напрямку втрати, щоб мати найкращий шанс з'єднатися з будь-якими розділеними блоками, а також мінімізувати втрату покриття через невдача. Кожен вузол використовує близькість до своїх сусідів для обчислення сил, що діють на нього, і малює віртуальний шлях у напрямку комбінованої сили до досягнення рівноваги. Потім вузол переміщується на нове місце. Ця функція дозволяє AuR відповідати обмеженню підключення, оскільки вузол не може бути далі, ніж діапазон зв'язку «R» від свого найдальшого сусіда. Подібно до закону Кулона, сила між двома вузлами залежить від відстані між ними, і на вузол можуть діяти кілька сил. Загальна сила, що діє на вузол, визначається як:

$$\overrightarrow{F_{comp}} = \overrightarrow{F_1} + \overrightarrow{F_2} + \overrightarrow{F_3} + \overrightarrow{F_4} \dots\dots, \tag{3.1}$$

Сила притягання на провідний вузол S_a через мертвий вузол S_f визначається наступним чином:

$$F_{af} = R, \tag{3.2}$$

Лідерні вузли розраховують напрямок свого руху виключно на основі сил притягання та рухаються в напрямку комбінованої сили, доки вони не опиняться на максимальній відстані «R» від свого найдальшого сусіда. Лідерний вузол S_1 відчуває сили притягання F_{16} і F_{17} до S_6 і S_7 відповідно (рисунок 3.2). S_1 розраховує складову силу та рухається, як показано на (рисунок 3.2) доки не буде на одиницях «R» від свого сусіда S_9. Якщо вузол вступає в контакт з новим вузлом, він доповнює свій список сусідів і скидає свій втрачений ступінь вузла до нуля, оскільки він увійшов у контакт з іншим

вузлом у цьому напрямку, потім перераховує свій пріоритет переміщення на основі свого оновленого списку 1-hop. Сила відштовхування між двома живими вузлами «S_a» і «S_b», розділених відстанню d_{ab}, визначається наступним чином:

$$F_{ab} = \begin{cases} R - d_{ab}, & if\ R > d_{ab} \\ 0, & if\ R \leq d_{ab} \end{cases}, \tag{3.3}$$

Нелідерні вузли переміщуються за допомогою цього методу. Композитна сила, що діє на вузол, обчислюється на основі його живих сусідів з 1 стрибком і змушує вузол рухатися якомога далі від своїх сусідів, зберігаючи зв'язок із ними. Це призводить до розкиданої топології та може призвести до повторного підключення деяких непересічних блоків вузлів. Наприклад, S_9 відчуває силу відштовхування F_{19} від S_1 і F_{10-9} через S_{10} (рисунок 3.2). S_9 обчислює сумарну силу відштовхування, що діє на нього через S_1 і S_{10}, і прокладає віртуальний шлях до місця, де сумарна сила, що діє на нього з боку всіх його сусідів, дорівнює нулю, а потім переміщується туди. Тим часом вихідний вузол S_4 рухається до центру зони розгортання. Саморозповсюдження S_{10} і S_{12} через відштовхування (рисунок 3.2) Вузол на периферії розділу розглядатиме межу зони розгортання як бар'єр, щоб він не переміщався за межі регіону, де покриття становить інтерес під час саморозповсюдження;

5 Рух до центру: щоб забезпечити остаточну конвергенцію, AuR виконує каскадне переміщення до центру після початкового саморозповсюдження. Розрізнений розділ рухається до центру області розгортання, доки він не стикнеться з іншим розділом або не досягне центру. Вузол, який не може рухатися далі через саморозповсюдження, інформує своїх сусідів з 1-хопом, і його центральна сила ввімкнена в таблиці сусідів з 1-хопом. Якщо всі вузли в 1-хоп-таблиці S_i відчувають цю силу, це означає, що всі вузли розповсюдилися і не можуть рухатися далі. Це запускає

каскадний рух у мережі з 1 переходом, і порядок переміщення перераховується на основі відстані до центру. Вузол, найближчий до центру зони розгортання в межах своєї мережі з 1 стрибком, стає лідером і відчуває силу тяжіння F_{center} до центру та поступово рухається одиницями «$\frac{R}{2}$», доки не досягне центру або не зайде в контакт з іншими вузлами, які мають стан підключення як підключений, що означає, що вони є частиною підключеної мережі навколо центру. Перед переміщенням провідний вузол інформує всіх своїх сусідів про своє нове положення та чекає, поки всі вони будуть у зоні дії, перш ніж продовжити переміщення. Сусіди чекають, доки вони не отримають сповіщення від усіх вузлів, ближчих до центру, ніж вони самі, у своїй таблиці 1-хопу, перш ніж переміститися.

Каскадний рух до центру має на меті зберегти розширену топологію, досягнуту раніше. Для досягнення цієї мети провідний вузол включає свою поточну позицію (x_{old}, y_{old}) і майбутню позицію (x_{new}, y_{new}) у сповіщенні, яке він надсилає своїм сусідам перед переміщенням. Сусіди обчислюють напрямок руху через нахил лінії та продовжують рухатися в цьому напрямку, поки вони не будуть на однаковій відстані від найдальшого вузла, з якого було отримано повідомлення, перш ніж почнеться рух до центру.

$$Slope = \frac{y_{new} - y_{old}}{x_{new} - x_{old}}, \tag{3.4}$$

На рисунку 3.2(d) показано рух до центру для блоку вузлів після саморозповсюдження. Оскільки S_1 є найближчим до центру вузлом у його околиці з 1 стрибком, він рухається першим. Отримавши сповіщення про рух від S_1, S_9 обчислює нахил лінії, уздовж якої рухається S_1, і використовує цей нахил для визначення напрямку руху. Це гарантує, що вузли рухаються паралельно та підтримують розтягнуту мережеву топологію, досягнуту завдяки саморозширенню. У випадку, якщо вузол є частиною більш ніж одного блоку, він отримуватиме різні повідомлення щодо напрямку, у якому він має рухатися, від своїх різних сусідів з 1-хопом. У цьому випадку вузол

обчислює загальну суму сил, що діють на нього, і рухається в складеному напрямку, доки не вийде в контакт з усіма вузлами, від яких він отримав повідомлення, зберігаючи ту саму відстань від найдальшого сусіда, з яким він був пов’язаний у кінець саморозповсюдження.

Якщо вузол виявляє нового сусіда під час каскадного переміщення, він додає його до свого списку сусідів, скидає його втрачений ступінь вузла до нуля та вимикає його центральну силу, щоб забезпечити саморозповсюдження перед продовженням каскадного переміщення. Коли вузол досягає центру або стає частиною підключеної до центру мережі, він змінює свій статус на «відновлений», вимикає свою центральну силу та інформує всіх своїх сусідів з 1-хопом про зміну, і вони рекурсивно застосовують ті самі зміни. На рисунку 3.2(d) каскадний рух у напрямку до центру продовжується до тих пір, поки головний вузол розділу S_1 не вступить у контакт із S_4, статус якого «відновлюється»;

6) Остаточне саморозповсюдження: рух до центру змушує вузли всередині сегмента наближатися, якщо вони є частиною двох різних розділів. Крім того, каскадне переміщення та злиття сегментів призводить до того, що вузли наближаються один до одного, що зменшує зону покриття. Тому, як тільки всі вузли в околиці з 1-хопом відновлюються, виконується ще один етап саморозширення для збільшення покриття. Остаточна отримана топологія показана на рисунку 3.2(e). Вузол припиняє роботу алгоритму AuR, коли його статус «відновлюється» і його центральна сила ввімкнена, що означає, що сили, що діють на нього, збалансовані;

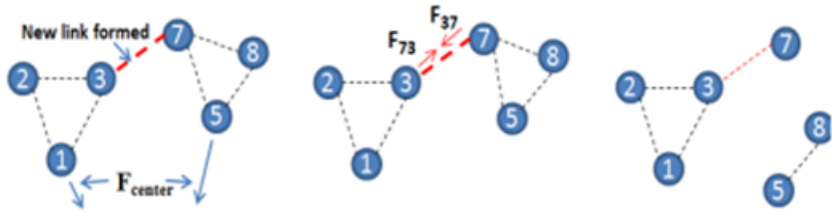
а) б) в) г) ґ)

Рисунок 3.2 – AuR відновлює зв’язок: а) вирішуючи порядок переміщення; б) рух через саморозгортання та сирітливі вузли; в) рух нелідерних вузлів; г) рух до центру; ґ) остаточна відновлена топологія після саморозгортання

7 Злиття сегментів: під час переміщення до центру деякі вузли сегментів увійдуть у контакт один з одним, перш ніж досягнуть центру. На рисунку 3.3(a) рух до центру робить S_3 і S_7 з'єднаними, рисунку 3.3(b) ілюструє саморозповсюдження S_3 і S_7. Перед встановленням зв'язку з S_3 S_7 перемістився лише після отримання сповіщень від своїх сусідів з 1-хопом S_5 та S_8, які мали вищий пріоритет через те, що вони були ближче до центру. Після контакту з S_3, S_7 має чекати сповіщення від S_3 перед тим, як переміститися, оскільки S_3 знаходиться ближче до центру, ніж S_7.

У наступній ітерації S_7 має центральну силу, тоді як ведучий вузол S_5 рухається, за ним слідує S_8. S_7 отримує сповіщення як від S_5, так і від S8, але тепер також потребує сповіщення від свого нового сусіда S_3, перш ніж він зможе рухатися. У випадку, якщо сусіди S_1, S_2 і S_3 з 1 стрибком все ще знаходяться на стадії саморозповсюдження, тоді S7 не отримає сповіщення від S_3 і, отже, не зможе рухатися. S_7 знає, що не може стежити за S_5, і тому він розриває зв'язки, які раніше мав із S_5 та S_8, і видаляє їх зі свого списку сусідів.

Коли S_5 досягає нового місця, він чекає, поки сусіди з нижчим рангом увійдуть у зону дії, у результаті чого він дізнається, що більше не контактує з S_7. S_5 чекає заздалегідь визначену кількість циклів у надії, що S_7 увійде в зону дії, після чого розриває зв'язок і продовжує свій рух до центру. Якщо сценарій відбувається з наступними вузлами, наприклад, коли S_1, S_2 і S_3 рухаються, але S_7 не може слідувати, S_3 розірве зв'язок із S_7, коли виявить, що він більше не контактує з S_7, коли прибуває на нове місце розташування. Варто зазначити, що AuR гарантовано сходиться до підключеної мережі, оскільки всі вузли з часом рухаються до центру. Основна мета AuR — прискорити відновлення з'єднання та обмежити накладні витрати, які відчувають окремі вузли.



а) б) в)

Рисунок 3.3 – Приклад для ілюстрації сценарію розриву посилання

3.2 Round-table negotiation

Підхід Round-table negotiation (RTN), починаючи з підключеної WSN, у якій кожен датчик здатний до мобільності. Через обмеження діапазону зв’язок відбувається між напряму підключеними вузлами (сусідами з 1 переходом) за допомогою повідомлень, щоб перевірити стан з’єднання. Це імпульсні повідомлення, які надсилаються через періодичні проміжки часу. Розглядається сценарій одночасної втрати кількох вузлів через якийсь неочікуваний інцидент. Втрата вузла означає фізичне пошкодження вузла, його каналів зв'язку та всього бортового обладнання та інформації. Мета полягає в тому, щоб за короткий час відновити зв’язок між кластерами за допомогою вузлів, що вижили.

3.2.1 Ідентифікація кластерів (самовідкриття) та збір інформації

Це перша фаза алгоритму, яка запускається після розділення WSN. Роз’єднані кластери не мають інформації про власний розмір і вузли, що вижили. Кожен вузол у кластері одночасно починає процес ідентифікації. Цей етап складається з чотирьох основних завдань.

3.2.1.1 Ідентифікація кластера

Кластер C_i з вузлів N_i, тобто $|C_i| = N_i$, визначається як

$$C_i = \{M_1^i, M_2^i, M_3^i, \dots, M_{N_{i-2}}^i, M_{N_{i-1}}^i, M_{N_i}^i\}, \quad (3.5)$$

Вузли в кластері представлені як M_p^i, де p позначає їх індивідуальний ідентифікатор, а і позначає відповідний ідентифікатор кластера. Кожен такий вузол генерує власний набір сусідів кластера протягом ітерацій, оскільки потік інформації відбувається лише між сусідами з 1-хопом. Цей набір для вузла в першій ітерації містить лише власну інформацію та оновлюється протягом кожної ітерації, як,

$$S_p^i(t+1) = S_p^i(t) \cup \left[\cup_{q \in N_p^1}\{S_q^i(t)\}\right], \quad (3.6)$$

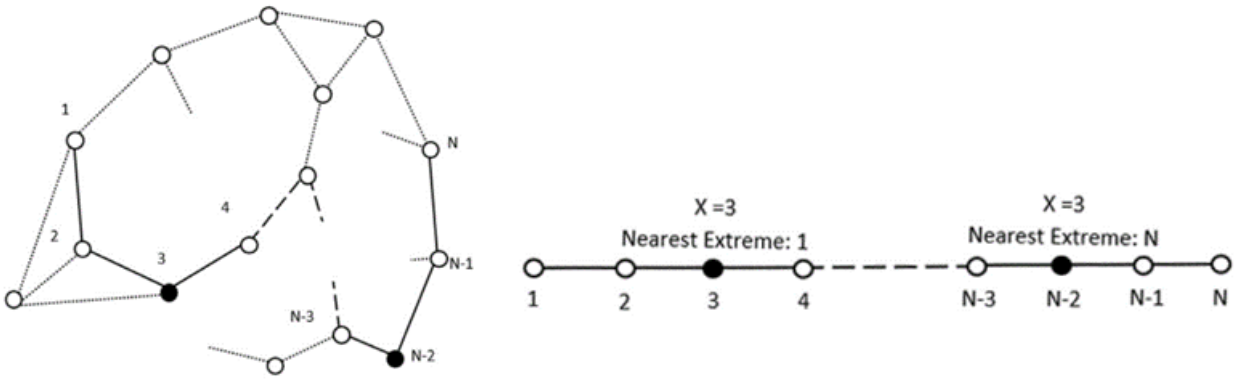
де $S_p^i(t)$ і $S_p^i(t+1)$ визначаються як набори досяжності вузла M_p^i кластера C_i на t-й і $(t+1)$-й ітераціях відповідно, а $S_q^i(t)$ визначається як набір досяжності вузла Mi q на t-й ітерації. Зверніть увагу, що кожен вузол у множині досяжності вузла знаходиться в околиці δ-стрибка цього вузла для деякого кінцевого δ. Цей процес припиняється, коли інформація в наборах сусідів кластера для всіх вузлів у кластері збігається, тобто інформація стає ідентичною.

$$S_p^i(t) = S_p^i(t-1) \; \forall \; p \in C_i, \quad (3.7)$$

Інформація, якою обмінюються під час процесу самовиявлення, складається в основному з поточного досяжного набору вузла. Однак для реалізації алгоритму також обмінюється іншою інформацією, такою як розташування вузла та його відстань від точки зустрічі.

Інформація в наборах сусідів кластера збігається, коли рівняння для кластера виконується (3.7). Припустимо, що рівняння (3.7) не виконується

для кластера для певного t, тоді $S_p^i(t) = S_p^i(t-1)$ для деяких вузлів M_p^i, що означає, що набори сусідів кластера цих вузлів не мають все ж зійшлися. Отже, потрібні подальші ітерації. Крім того, якщо рівняння (3.7) задовольняється для деякого t, подальші ітерації (t + 1 і далі) призводять до тих самих наборів сусідів кластера, що й для конвергентної інформації кластера (при t).



а) б)

Рисунок 3.4 – а) Приклад мережі; б) простий шлях у мережі для отримання кількості ітерацій

3.2.1.2 Встановлення максимальної кількості ітерацій для самоідентифікації кластера

Щоб отримати максимальну кількість ітерацій, необхідну для самоідентифікації кластера переглянути найкоротші прості шляхи між кожною парою вузлів у мережі. Приклад мережі показано на рисунку 3.4(a), де розглядається простий шлях довжиною N у мережі. Цей шлях показано окремо на рисунок 3.4(b), як,

$$C = \{1, 2, 3, 4, \dots, N-2, N-1, N\}, \tag{3.8}$$

Загалом кожен вузол буде залежати лише від своїх сусідів з 1-хопом для оновлення інформації. Це означає, що на кожній ітерації у прикладі мережі до набору сусідів кластера буде додано один або два нові вузли на основі оновлень від сусідів. Посиланням X для вузла вважається його положення від найближчого екстремуму шляху (наприклад, як вузли 3, так і $(N-2)$, на рисунку 3.4(b), мають $X=3$ від відповідних найближчих екстремумів).

Спочатку набір сусідів кластера $S_i(0)$ для вузла і містить лише власну інформацію. Потім цей набір, $S_i(t)$, оновлюється протягом ітерацій, використовуючи інформацію про набори сусідів кластера елементів $S_j(t)$, де $j \in N_j^1$. Для X-го вузла оновлення міститиме два нові елементи (інформацію про вузли) на кожній ітерації для $(X-1)$ ітерацій. Потім на кожній ітерації буде додано лише один новий елемент, оскільки не буде оновлень з коротшої сторони шляху, тобто найближчого екстремуму. Тоді для будь-якої ітерації t набір сусідів кластера для вузла і визначається як,

$$S_i(t) = \{(i-k): k \in [-t;t];\ i,(i-k) \in C_i; k,t \in \mathbb{Z}^+\}, \tag{3.9}$$

Цей процес триває доки,

$$S_p(t) = S_p(t-1)\ \forall\ p \in C_i \tag{3.10}$$

З цього можна зробити висновок, що X-й вузол скомпілює інформацію про всіх підключених вижили після $(N-X)$ ітерацій, на основі кількості записів оновлення з найдальшого крайнього простого шляху. Для прикладу мережі на рисунку 3.4(b) кількість ітерацій, необхідних для різних положень вузлів, буде:

1) Центральний вузол: (N парне), тобто $X=\frac{N}{2}$, загальна кількість ітерацій буде дорівнювати $N-X=\frac{N}{2}$;

2) Центральний вузол: (N непарне), тобто $X = \frac{(N+1)}{2}$, загальна кількість ітерацій буде дорівнювати $N - X = \frac{(N-1)}{2}$;

3) Крайній вузол: (Вузли 1 і N), тобто $X = 1$, загальна кількість ітерацій буду дорівнювати $N - X = N - 1$;

4) Проміжний вузол: $1 < X < \frac{N}{2}$, загальна кількість ітерацій лежить між $\frac{(N-1)}{2}$ (коли N непарне) або $\frac{N}{2}$ (коли N парне) і $N - 1$.

Центральний вузол(и) сходяться першими, і кількість ітерацій, необхідних для зближення, поступово збільшується, коли ми рухаємося до крайніх кінців. Зроблено висновок, що мережева інформація буде скомпільована $(N-1)$-ю ітерацією, оскільки максимальна кількість ітерацій становить $N - 1$ для крайніх вузлів простого шляху.

LSSP у кластері диктує максимально можливу кількість ітерацій, необхідних для конвергенції мережевої інформації. Щоб довести це припустимо, що довжина LSSP дорівнює N, а довжина якогось іншого шляху дорівнює P, так що $N > P$, як показано на рисунку 3.5. Максимальна кількість ітерацій для конвергенції інформації кластера через будь-який із цих шляхів буде $N - 1$ або $P - 1$ (як описано вище). Оскільки $(N-1) > (P-1)$, в кінцевому підсумку буде $N - 1$, тобто інформація кластера зійдеться на $(N-1)$-й ітерації або перед нею, доводячи, що довжина LSSP визначає максимальна кількість ітерацій, необхідних для збіжності.

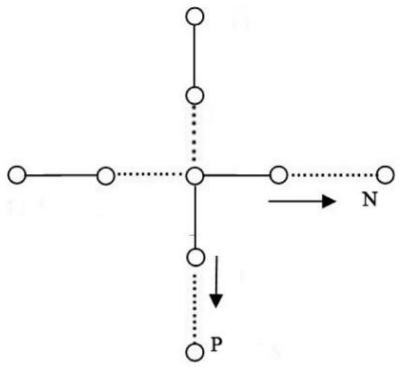


Рисунок 3.5 – Приклад сценарію для доведення методу знаходження максимальної к-ті ітерацій

Порядкова складність для самостійного виявлення кластера з N вузлами дорівнює $O(N)$. З наведеного вище методу можна зробити висновок, що LSSP є найгіршим можливим з'єднанням з точки зору кількості ітерацій, необхідних для компіляції інформації про мережу, яка займає $(N-1)$ ітерацій, і, отже, складність порядку $O(N)$.

3.2.1.3 Ідентифікація прикордонних вузлів

Під час ідентифікації кластера вузли, які втратили одного чи кількох сусідів з 1-хопом, ідентифікують себе як прикордонні вузли та передають власну інформацію підключеним сусідам. Після ідентифікації кластера список усіх таких вузлів складається для кожного вузла в кластері. Ці вузли, швидше за все, будуть найближчими до інших кластерів, що вижили, оскільки вони можуть бути розташовані на крайньому краю кожного кластера. Вузли ідентифікують себе як прикордонні вузли, порівнюючи інформацію після розділу з інформацією початково підключеної мережі наступним чином. Якщо для будь-якого вузла $p \in C_i$, $N_p^{1'} \neq N_p^1$, тоді $p \in B_i$, де B_i - список граничних вузлів для кластера C_i, N_p^1 і $N_p^{1'}$ - це набори сусідів з 1 стрибком вузла $p \in C_i$ до і після розділу відповідно. Оскільки лише деякі з сусідів цих вузлів втрачаються і нові з'єднання не встановлюються, $N_p^{1'} \subseteq N_p^1$.

3.2.1.4 Відбір учасників переговорів

Оскільки розташування точки зустрічі та круглого столу доступне для всіх вузлів у мережі (припущення 3), кожен вижив вузол у кластері знає свою відстань від точки зустрічі, яку він передає всім вижилим у цьому кластері. під час процесу самопізнання. Після того, як інформація про вузли в кластері

скомпільована, кожен вузол знає відстані всіх вузлів кластера від точки зустрічі. Ця інформація використовується кожним вузлом, щоб визначити, чи він сам найближче до точки зустрічі. Якщо так, то вузол знає, що він є учасником переговорів. Якщо ні, то він визначає вузол, найближчий до точки зустрічі, як учасник переговорів. Це одночасне рішення, прийняте кожним вузлом у кластері. Для кластера C_i учасник переговорів позначається M_i^n і призначається як,

$$M_i^n = p \; if \; D(p,m) = \min_{x \in C_i}\{D(x,m)\}, \tag{3.11}$$

де $D(x,m)$ - відстань між розташуванням вузла x і точкою зустрічі m;

У разі наявності двох або більше вузлів, рівновіддалених від точки зустрічі m для кластера, будь-який з них може бути учасником переговорів. Вибір можна зробити на основі попереднього взаємно узгодженого правила (наприклад, вищого значення ідентифікатора вузла).

Ці вузли переміщуються для обміну інформацією та беруть участь у процесі прийняття рішень, а потім повертаються у вихідне положення. До кінця цієї фази всі кластери ідентифікували себе, а також свої прикордонні вузли та учасників переговорів.

3.2.2 Переговори та рішення про підключення

Кожен учасник переговорів проходить відстань $\left[D(M_i^n,m) - (\frac{R}{2})\right]$ і досягає місця зустрічі, де $D(M_i^n,m)$ - відстань від початкового місця розташування учасника переговорів до місця зустрічі (м) і R - дальність зв'язку. Після того, як усі ці учасники переговорів збираються за столом, здійснюється обмін інформацією та переговори для вирішення шляхів повторного підключення та вузлів, які будуть використовуватися для повторного підключення. Цей етап складається з наступних трьох кроків.

3.2.2.1 Обмін інформацією

Вузли переговорників, прибувши до місця переговорів, виявляють інших учасників переговорів у межах свого діапазону зв'язку та підключаються. Кожен учасник переговорів несе інформацію свого кластера (ідентифікатори вузлів у своєму кластері, їх розташування та канали зв'язку, а також прикордонні вузли). Незважаючи на те, що учасники переговорів продовжують мати власні ідентифікатори вузлів, а їхні кластери ідентифікуються ідентифікаторами учасника переговорів, для легкого розуміння алгоритмів ідентифікатори N_i та C_i призначаються учасникам переговорів і відповідним кластерам у порядку прибуття на круглий стіл, з і, що йде від 1 до η, де η – загальна кількість кластерів. Кожен учасник переговорів прибуває до круглого столу з наміром чекати протягом часу, що не перевищує T_w. Однак обмін інформацією розпочинається, як тільки один із учасників переговорів (фактично перший учасник переговорів, який прибув до круглого столу) закінчує час очікування. Обмін інформацією здійснюється через механізм трансляції, за якого кожен учасник переговорів передає інформацію про свій кластер усім іншим учасникам переговорів за круглим столом. Таким чином, після того, як перший учасник переговорів завершить свій час очікування та транслює інформацію про свій кластер, інші учасники переговорів за круглим столом припиняють очікування та транслюють інформацію про свій відповідний кластер. Наприклад, розглянемо трьох учасників переговорів, які прибувають до круглого столу в моменти часу t_1, t_2 і t_3 (з $t_1 \leq t_2 \leq t_3$), кожен з наміром чекати щонайбільше T_w часу для прибуття всіх можливих учасників переговорів. Але, як тільки перший учасник переговорів (прибувши в t_1) завершує свій час очікування (тобто в момент часу $t_1 + T_w$), він транслює інформацію про свій кластер, а всі інші учасники переговорів припиняють очікування та починають транслювати інформацію про свій відповідний кластер. Зауважте, що

ідентифікатора вузла достатньо для створення додаткової інформації про їх розташування та канали зв'язку. Остаточний набір інформації для кожного узгоджувача після обміну є ідентичним, який потім використовується для прийняття рішення про шляхи повторного з'єднання та вузли заміни.

/* At the time of arrival of the first negotiator */
1: $(i = 1)$ ID: M_1^n, cluster ID: C_1, Neg_List = $\{M_1^n\}$
2: **while** $t <$ Waiting time **do**
3: At the arrival of the next negotiator
4: $i = i + 1$
5: ID: M_i^n, cluster ID: C_i
6: Neg_List = Neg_List $\cup$ $\{M_i^n\}$ and passes to all M_i^n
7: **end while**
8: Each M_i^n broadcasts its cluster information to all $M_j^n \in$ Neg_List$\backslash\{M_i^n\}$

Рисунок 3.6 – Алгоритм обміну інформацією

Учасники переговорів, які прибувають на місце зустрічі, повинні чекати не більше часу $T_w = \widehat{N}T + \frac{D-\left(\frac{R}{2}\right)}{v}$, перш ніж розпочнеться обмін інформацією, щоб забезпечити прибуття всіх інших учасників переговорів з кожного кластера, що вижив. Тут $\widehat{N}$ - загальна кількість вузлів у початковій підключеній WSN, T - час, необхідний для кожного циклу оновлення, D - відстань найдальшої вершини від точки зустрічі, а v -швидкість вузла, доки він мобільний.

З моменту, коли відбувається розділення WSN, кожен кластер, що вижив, ідентифікує себе до моменту $\widehat{N}T$, а максимальний час, потрібний будь-якому вузлу (і, отже, учаснику переговорів), щоб досягти місця переговорів, визначається як $\frac{D-\left(\frac{R}{2}\right)}{v}$,. Таким чином, T_w є верхньою межею часу для компіляції інформації про виживший кластер у загальній мережі та учасників переговорів для переміщення до місця переговорів, після чого може бути розпочато обмін інформацією та процес прийняття рішень.

3.2.2.2 Вирішення шляхів перез'єднання

На наступному кроці шляхи повторного підключення визначаються одночасно кожним учасником переговорів. Для цього вибирається найближчий кластер до кожного з кластерів і найкоротший шлях, що з'єднує їх. Ці передбачувані шляхи повторного підключення становлять розташування мертвих вузлів, оскільки вони є відомими місцями з бажаним покриттям. Вузли повторного підключення розміщуються в цих місцях, щоб відновити зв’язок між кластерами. Це допомагає покращити покриття повторно підключеної мережі.

Для визначення цих шляхів кожен граничний вузол у кластері перераховує найближчий відповідник у цільовому кластері та відповідний шлях у термінах кількості переходів (рисунок 3.7). Найкоротший із цих шляхів, заданих $\min_P\{L(P_{xy})\}$, вибирається між парою кластерів (C_i та C_j) як кінцевий шлях, з початковим і кінцевим вузлами як $x \in B_i$ та $y \in B_j$. У разі рівності шлях із найменшою декартовою відстанню вибирається, щоб мінімізувати відстань подорожі. Для з’єднання n кластерів потрібно всього $(n-1)$ шляхів. Отже, ці шляхи, вибрані кластерами, потім сортуються за відстанню, а зайві шляхи (наприклад, шляхи від C_2 до C_1 і C_1 до C_2) пропускаються. Це робиться протягом кількох ітерацій шляхом створення шляхів, віртуального злиття кластерів і перевірки загального зв’язку вцілілих, доки всі кластери не будуть з’єднані.

```
   /* Find n − 1 paths for n clusters */
        For all i and j, B_i, C_i ∈ Info of(M_i^n),  B_j, C_j ∈ Info of(M_j^n),
 1: for k = 1 to length(B_i) do /* Paths from C_i to some C_j */
 2:     for l = 1 to length(B_j) do
 3:         [path hop_dist] = shortestpath(B_i(k), B_j(l))
 4:     end for
 5:     Store path
 6: end for
 7: Choose shortest path between C_i and C_j
 8: Final_paths = List of all shortest paths
    /* Omitting redundant paths */
 9: if Two clusters have chosen each other then
10:     Omit one of these paths from Final_paths
11: end if
    /* Checking for overall connection */
12: for k = 1 to rowcount(Final_paths) do
13:     Merge the to be connected clusters
14:     if Merged cluster = All survivor nodes then
15:         End path decision process
16:     else
17:         Define merged clusters and borders
18:         Repeat from step 1 and add paths
19:     end if
20: end for
```

Рисунок 3.7 – Алгоритм визначення шляхів повторного підключення

3.2.2.3 Заміну вузлів

Після визначення шляхів повторного з’єднання уздовж цих шляхів витягується список бажаних розташувань мертвих вузлів, які потрібно відновити. На кожному визначеному шляху повторного підключення бажані розташування вузлів на початку та в кінці належать кластерам, які потрібно з’єднати, і зайняті вузлами, що вижили. Заміни для цих місць розташування вузлів не потрібно призначати. Для інших подібних бажаних місць вузли, які будуть розгорнуті на шляху (вузли заміни), призначаються за допомогою вижилих вузлів із кластерів на обох кінцях шляху. Ці завдання виконуються з урахуванням наступних міркувань:

1) вузол заміни не повинен бути вузлом шляху на визначеному шляху повторного підключення;

2) переміщення вузла заміни не повинно спричиняти внутрішньокластерне роз’єднання після виходу з початкового місця розташування;

3) замінний вузол повинен мати мінімальну кількість сусідів з 1-хопом,

щоб забезпечити найменшу втрату зв’язку;

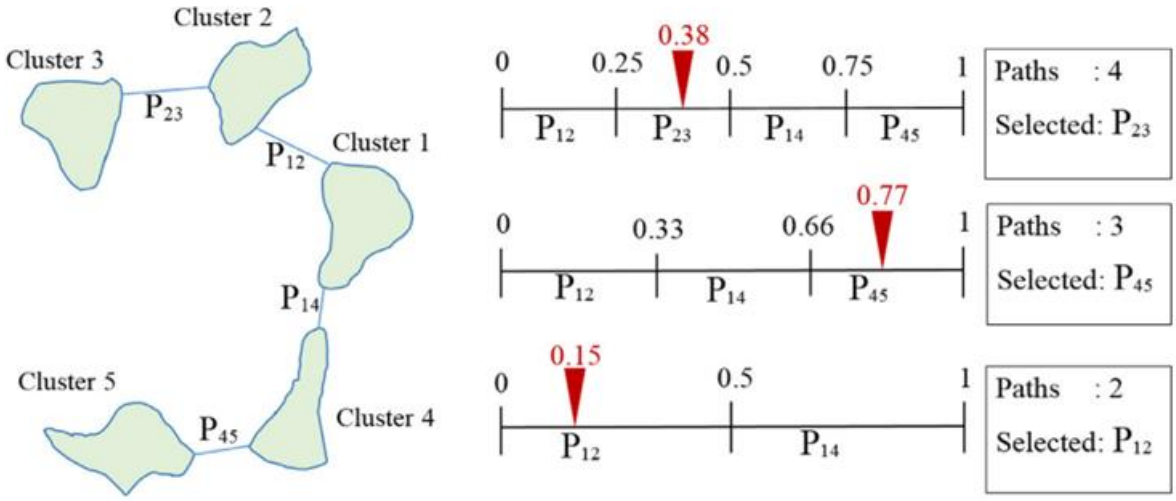
4) якщо (1) і (2) пункти задовольняються, то вузол заміни повинен бути на найкоротшій відстані від бажаного місця мертвого вузла, щоб мінімізувати відстань подорожі.

Порядок шляхів, які необхідно вирішити, визначається випадковим вибором учасників переговорів. Для цього діапазон ймовірностей 1/n надається кожному з n шляхів, які потрібно вирішити, і генерується рівномірно розподілене випадкове число між 0 і 1 (цей процес буде називатися випадковим жеребкуванням). Шлях, діапазон якого включає випадковий результат розіграшу, вказано в списку для вирішення першим. Процес повторюється для всіх шляхів, що залишилися, доки не буде отримана впорядкована послідовність цих шляхів, якою слідувати під час призначення вузлів заміни. Як показано в прикладі на рис. 3.6(a), чотири шляхи (P_{12}, P_{23}, P_{14} і P_{45}) повинні бути заповнені замінними вузлами для відновлення зв’язку. На першому етапі генерації випадкових чисел, як показано на рис. 3.6(b), для шляхів було визначено чотири інтервали з діапазоном 0,25 кожен. На основі результату випадкового жеребкування серед учасників переговорів (вихід 0,38, тобто P23), шлях P_{23} вибирається для вирішення першим. Подальші етапи випадкового жеребкування показані на рис. 3.6(b), які призводять до порядку шляху P_{23}, P_{45}, P_{12} і P_{14}. Цей процес наведено на рисунку 3.8.

Після цього кожен учасник переговорів паралельно визначає вузли заміни для шляху, використовуючи вузли з кластерів на обох кінцях шляху як потенційних кандидатів. Вузли заміни вибираються на основі наведених вище критеріїв, а завершені заміни видаляються зі списку потенційних варіантів заміни. Цей процес триває, доки не буде прийнято рішення про заміну всіх бажаних місць мертвих вузлів. Після завершення визначення шляхів і вузлів учасники переговорів повертаються до вихідних кластерів з усією інформацією про рішення.

```
/* Sequence in which the paths will be resolved (Decided by
M_1^n) */
Interval pool 1/n for n paths to resolve
Random draw to choose the path (See Fig 4)
Repeat step 1 to 2 till an order is defined
/* Desired node locations */
for j = 1 to rowcount(Final_paths) do
    if Path_node ∉ Survivor_Nodes then
        Add that 'Path_node' to 'Desired_node'
    end if
end for
/* Replacements */
for k1 = 1 to length(Desired nodes for path P_ij) do
    Options = setdiff((C_i ∪ C_j), (End_node ∪ Used_node))
    for k2 = 1 to length(Options) do
        if Options(k2) does not partition its own cluster then
            List such node as 'Choice'
        end if
    end for
    Sort 'Choice' by neighbor count and distance, choose 'replacement'
    Used_node = {Used_node, replacement}
end for
```

Рисунок 3.8 – Алгоритм заміни вузлів



а) б)

Рисунок 3.9 – Вибір шляху шляхом випадкового жеребкування: а) Приклад WSN із шляхами; б) умова та результати розіграшу

3.2.3 Заміна та повторне підключення

На цьому останньому етапі учасники переговорів повертаються до

своїх початкових місць, і рішення про повторне підключення передається до вузлів у кластерах. Призначені замінні вузли потім переміщуються до бажаних мертвих вузлів, яким вони були призначені, і підключаються до іншого шляху та вузлів кластера в межах їхнього діапазону зв’язку. Крім того, відновлено зв’язок загальної мережі тих, хто вижив. Оскільки для повторного підключення використовуються найкоротші шляхи між кластерами (через розташування мертвих вузлів), ціль досягається за рахунок мінімальної кількості розгорнутих вузлів і мінімальної відстані шляху. Також досягнуто основної мети – скоротити час повторного підключення.

3.3 Swarm Intelligence-Based Damage-Resilient

Коли пошкоджено велику кількість вузлів USNET, уцілілі вузли можуть утворити кілька непересічних підмереж. У цей момент USNET може бути частково або фрагментовано пошкоджений. Традиційні методи керування топологією ускладнюють об’єднання вцілілих вузлів в єдину мережу. На щастя, вузли USNET зазвичай мають можливості зберігання бездротового зв'язку, обчислень, позиціонування та мобільності.

Кожен вузол БПЛА повинен зберігати місію, структуру мережі, ресурси та інформацію про відстеження USNET у своєму сховищі. Це змушує всіх агентів досягти консенсусу, що також є основою для появи інтелекту. Можливість бездротового зв’язку дозволяє БПЛА безперервно шукати вцілілі вузли або підмережі навколо себе та постійно об’єднувати підмережі. Обчислювальна здатність дозволяє БПЛА визначати найкращий маршрут для реконструкції USNET за найкоротший час, дотримуючись траєкторії польоту. Здатність мобільності дозволяє агентам переміщатися у відповідну позицію. Таким чином, комбінація цих можливостей може бути використана для розподіленого створення інтелекту роя, стійкого до пошкоджень.

Метод Swarm Intelligence-Based Damage-Resilient (SIDR) складається з

3 етапів.

1) Етап правильної роботи: використовуючи можливості зберігання, кожен вузол запам’ятовує структуру, конфігурацію та інформацію про відстеження попередньої мережі USNET, отриману від головного вузла. Усі вузли зграї відстежують траєкторію польоту USNET в єдиному напрямку та швидкості.

2) Етап ідентифікації пошкодження: головний вузол визначає, чи пошкоджена мережа чи ні, і якщо мережа пошкоджена, буде розпочато процес відновлення мережі. Якщо пошкоджено лише окремий вузол і він не впливає на підключення до мережі, головний вузол налаштує його відповідно до політики (у цьому документі це питання не розглядається), і процес агрегації не буде розпочато.

3) Етап агрегації мережі: головний вузол кожної підмережі відповідає за коригування поведінки вузлів у підмережі. Поведінка польоту визначається двома факторами: один полягає в тому, щоб рухатися до поточної точки маршруту, а інший - слідувати заздалегідь визначеному шляху. У процесі руху до маршрутної точки підмережа об’єднається з зустрічними підмережами та усуне надлишковий головний вузол. USNET знову перейде до нормальної робочої фази, якщо завершиться фаза агрегації.

3.3.1 Добре робоча фаза

На етапі ініціалізації USNET він має унікальний головний вузол, визначений передвиборним або іншим методом. Головний вузол відповідає за сприйняття ресурсів і розташування інших вузлів, направляє USNET на політ уздовж траєкторії польоту та інформує всі вузли про інформацію, якою потрібно поділитися, таку як інформація про місію, структуру мережі, траєкторію польоту, тощо.

Головний вузол надсилає опитувальні повідомлення всім підлеглим вузлам у USNET кожного разу T_{poll} під час виконання місії. Повідомлення

опитування містить поточний стан ресурсів і конфігурацій. Кожен підлеглий вузол вкладає інформацію про свій стан у повідомлення підтвердження під час отримання повідомлення опитування. Таким чином, головний вузол може отримувати останню глобальну інформацію мережі. Крім того, підлеглі вузли також можуть активно повідомляти свою інформацію головному вузлу за допомогою повідомлення TRAP. За допомогою операцій «POLL & TRAP» усі вузли в мережі досягають консенсусу щодо набору вузлів добре працюючого стану S_w і траєкторії польоту f_p.

Крім того, всі вузли повинні відстежувати траєкторію польоту в добре робочій фазі, таким чином, швидкість вузла u_i дорівнює $v_f^i(t)$, тобто $v_i(t) = v_f^i(t)$.

3.3.2 Етап ідентифікації пошкоджень

Після пошкодження великої кількості вузлів необхідно швидко визначити пошкодження та розпочати процес відновлення. У цьому розділі представлено алгоритм для ідентифікації пошкоджень.

Нехай RTO буде часом очікування повторної передачі для операції опитування. Головний вузол повторно передасть повідомлення опитування, якщо він не отримає очікуване повідомлення підтвердження від підлеглого вузла протягом часу RTO. Нехай $k > 0$ — максимальна кількість повторних передач. Головний вузол позначатиме підлеглий вузол як пошкоджений, якщо після k разів спроби не буде отримано повідомлення про підтвердження. Якщо підлеглий вузол не отримує жодного повідомлення опитування від головного вузла в $T_{poll} + k * RTO$, він спонтанно ініціює певний алгоритм вибору (наприклад Bully), щоб визначити новий головний вузол поточної підмережі. Кожна підмережа матиме унікальний головний вузол після процесу розподіленого вибору. Новообраний головний вузол збиратиме інформацію про вузли підмережі та обчислюватиме напрямок

руху та швидкість підмережі відповідно до розташування вузлів та історичної інформації USNET.

Головний вузол оновить набір вузлів $S(t)$ поточної підмережі після завершення повного процесу опитування або отримання повідомлення TRAP і обчислить відстань між підмережею та маршрутною точкою. Відхилення відбувається, якщо відстань перевищує $\frac{R}{2}$. У цьому випадку наступна дія буде виведена на основі кількості пошкоджених вузлів, яку можна виразити як $|S_\omega| - |S(t) \cap S_\omega|$. Зазначається, що згадана тут кількість пошкоджених вузлів стосується кількості відключених вузлів, яка не дорівнює кількості фактично пошкоджених вузлів. Наприклад, відключення кількох вузлів може бути спричинене пошкодженням вузла розрізаної вершини. Якщо $|S_\omega| - |S(t) \cap S_\omega| = 1$, пошкоджений вузол не впливає на підключення до мережі. Якщо пошкоджений вузол є головним вузлом, інші підлеглі вузли ініціюють процес розподіленого вибору, а потім новообраний головний вузол перезапустить процес ідентифікації пошкодження. Щоб уникнути ініціювання високовартісного процесу агрегування, відстань між підмережею та маршрутною точкою може бути скоригована головним вузлом, наприклад, перемістивши підмережу до маршрутної точки в цілому, або вибравши вцілілий вузол для заміни пошкодженого вузла. Якщо $|S_\omega| - |S(t) \cap S_\omega| > 1$, мережа може бути розділена. У цьому випадку головний вузол переходить у *damage* стан і записує час t_{damage}, а потім ініціює процес агрегації мережі (рисунок 3.11). Процес ідентифікації пошкоджень, який виконує головний вузол, показано на рисунку 3.10.

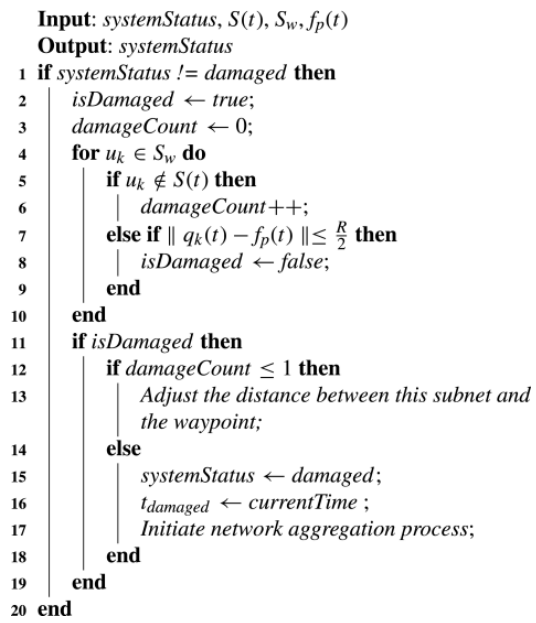


Рисунок 3.10 – Алгоритм процесу ідентифікації пошкоджень, що виконується головним вузлом

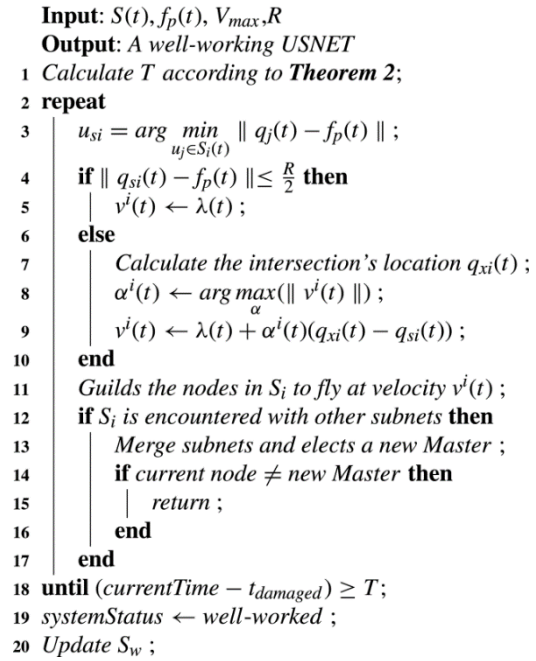


Рисунок 3.11 – Алгоритм процесу агрегування на основі потенційного поля, який виконує головний вузол підмережі S_i

В алгоритмі процесу ідентифікації пошкоджень systemStatus представляє поточний стан USNET. На кроці 1 головний вузол перевіряє, чи

поточна система знаходиться в пошкодженому стані, процес ідентифікації пошкодження буде ініційовано, лише якщо поточна система не знаходиться в пошкодженому стані. Крок 4-10 підрахувати кількість пошкоджених вузлів і перевірити, чи поточна підмережа відхиляється від маршруту польоту. Кроки 11-19 надають метод обробки, коли підмережа відхиляється від траєкторії польоту. Якщо кількість пошкоджених вузлів не більше одного, головний вузол регулює відстань між підмережею та маршрутною точкою. В іншому випадку головний вузол встановлює пошкоджений стан поточної системи та записує поточний час, а потім ініціює процес агрегації мережі.

3.3.3 Фаза агрегації мережі

Алгоритм агрегації потрібен для об’єднання всіх підмереж у підключену мережу після виявлення пошкодження. Цей розділ спочатку представляє основну ідею дизайну алгоритму, ілюструє кілька лем процесу агрегації, а потім детально описує мережевий алгоритм агрегації на основі потенційних полів.

3.3.3.1 Основна ідея агрегації мережі

Агрегація мережі відноситься до процесу, під час якого всі підмережі наближаються до поточної маршрутної точки. USNET може бути розділена на кілька непересічних підмереж довільної форми після серйозного пошкодження. Метод потенційного поля використовується для вирішення проблеми агрегації мережі. На рисунку 3.12 наведено приклад агрегації мережі на основі потенційного поля. USNET розділений на п'ять непересічних підмереж, кожна з яких може бути представлена віртуальним вузлом. Нехай $\vec{F}_f$ позначає силу віртуального вузла для відстеження траєкторії польоту, $\vec{F}_a$ позначає доцентрову силу для агрегування підмереж, тобто притягання маршрутної точки до підмережі. Швидкість підмережі

може бути визначена спільною силою $\vec{F}_f$ і $\vec{F}_a$. Нехай $\upsilon_f \in \mathbb{R}^3$ і $\upsilon_a \in \mathbb{R}^3$ позначають компоненти швидкості, породжені $\vec{F}_f$ і $\vec{F}_a$ відповідно. Тоді швидкість підмережі становить $\upsilon_f + \upsilon_a$. Оскільки швидкість польоту не може перевищувати V_{max}, має виконуватися наступна формула:

$$\left\|\upsilon_f + \upsilon_a\right\| \leq V_{max}, \tag{3.14}$$

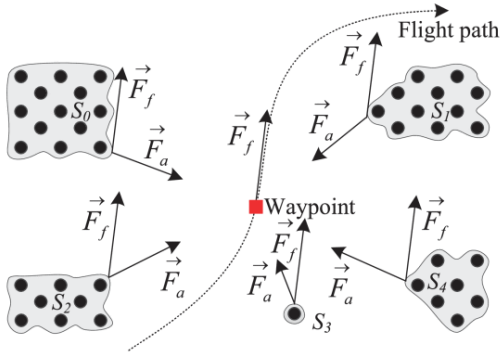


Рисунок 3.12 – Приклад агрегації мережі на основі потенційних полів

Усі підмережі будуть об’єднані в підключену підмережу, якщо відстань від усіх підмереж до маршрутної точки не перевищує $\frac{R}{2}$. Для $\forall S_1(t)$, $S_2(t) \leq S(t)$ припустимо, що $u_1 \in S_1(t)$, $u_2 \in S_2(t)$ є вузлами, які є найближчими до маршрутної точки в $S_1(t)$ і $S_2(t)$ відповідно. Якщо відстані від $S_1(t)$, $S_2(t)$ до маршрутної точки не перевищують $\frac{R}{2}$, отже маємо $\delta_1 = \left\|q_1(t) - f_p(t)\right\| \leq \frac{R}{2}$ і $\delta_2 = \left\|q_2(t) - f_p(t)\right\| \leq \frac{R}{2}$. Нехай d_{12} представляє відстань між u_1 і u_2. $d_{12} \leq \delta_1(t) + \delta_2(t) \leq R$, що вказує на те, що u_1 і u_2 можуть спілкуватися один з одним напряму, тому $S_1(t)$ і $S_2(t)$ будуть об’єднані в зв’язаний підмережа (рисунок 3.13).

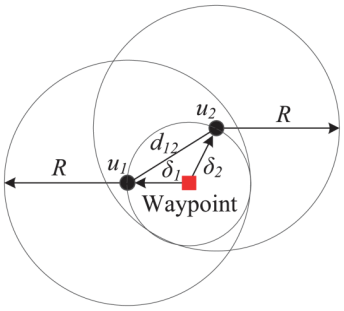


Рисунок 3.13 – Об'єднання двох підмереж

3.3.3.2 Алгоритм агрегації на основі потенційного поля

Оскільки кожен уцілілий вузол БПЛА зберігав попередньо визначену траєкторію польоту, об’єднання підмереж може бути досягнуто шляхом об’єднання підмереж у точку маршруту під час відстеження траєкторії польоту. Зокрема, кожна підмережа Si розглядається як сутність. Кожен головний вузол у кожній підмережі визначає швидкість своєї підмережі, а підлеглі вузли в підмережі слідують за головним вузлом за принципом «лідер-слідувач».

Основна ідея полягає в тому, що головний вузол u_l^i підмережі S_i збирає розташування всіх вузлів у підмережі та знаходить вузол u_{si} у підмережі S_i, який є найближчим до маршрутної точки в момент часу t. Тоді нехай розташування $q_{si}(t)$ вузла u_{si} представляє розташування підмережі, і визначте потенційну функцію $\phi^i(t)$ відповідно до розташування підмережі та маршрутної точки. Швидкість поточної підмережі vi обчислюється відповідно до $\phi^i(t)$:

$$v^i = \dot{q}_{si}(t) = -\frac{\vartheta \phi^i(t)}{\vartheta q_{si}(t)}, \tag{3.15}$$

До кожної підмережі застосовуються дві сили: сила тяжіння $\overrightarrow{F_f}$, викликана відстеженням наступної маршрутної точки, і доцентрова сила $\overrightarrow{F_a}$,

викликана агрегацією мережі (рисунок 3.12). Таким чином, потенційна функція, показана в (3.16), визначена для створення сил, необхідних для відновлення USNET.

$$\phi(t) = \phi_f(t) + \phi_a(t), \tag{3.16}$$

де $\phi_f(t) \in \mathbb{R}$ представляє потенційну функцію відстеження траєкторії польоту, а $\phi_a(t) \in \mathbb{R}$ представляє потенційну функцію агрегування підмереж;

Потенційна функція відстеження траєкторії польоту $\phi_f(t)$ дорівнює:

$$\phi_f(t) = \sum_{i=1}^{m} \phi_f^i(t), \tag{3.17}$$

де $\phi_f^i(t) \in \mathbb{R}$ представляє потенційну функцію підмережі S_i для відстеження траєкторії польоту, і вона визначається як:

$$\phi_f^i(t) = -[q_{si}(t)]^T * \lambda(t), \tag{3.18}$$

де $\lambda(t) \in \mathbb{R}^3$ являє собою швидкість відстеження траєкторії польоту в момент часу t. Нехай $t_{k-1} \leq t \leq t_k$, можна отримати, що

$$\lambda(t) = \frac{f_p(t_k) - f_p(t_{k-1})}{t_k - t_{k-1}}, \tag{3.19}$$

Якщо місцезнаходження наступної маршрутної точки таке ж, як і поточної, $\phi_f^i(t) = 0$. Ця функція дозволяє підмережі рухатися по траєкторії польоту, таким чином задовольняючи Track constraint.

Доцентрова сила, створювана $\phi_a(t)$, дозволяє всім підмережам наближатися до поточної маршрутної точки та скорочує відстань між кожною підмережею та маршрутною точкою. Усі підмережі будуть об’єднані в підключену мережу, якщо відстань від усіх підмереж до маршрутної точки

не перевищує $\frac{R}{2}$. Нехай $q_{si}(t)$ буде найближчим розташуванням підмережі S_i до поточної маршрутної точки, а $q_{xi}(t)$ буде місцем перетину кола з маршрутною точкою $f_p(t)$ як його центром і $\frac{R}{2}$ як його радіусом і лінією з'єднуючи $q_{si}(t)$ і маршрутну точку (рисунок 3.14), $\phi_a(t)$ визначається як:

$$\phi_a(t) = \sum_{i=1}^{m} \phi_a^i(t) = \sum_{i=1}^{m} \alpha^i(t) h_a^i(t) \tag{3.20}$$

де $\phi_a^i(t) \in \mathbb{R}$ – зважена потенційна функція агрегованих підмереж, а вага $\alpha^i(t) \geq 0$. Швидкість підмережі може задовольняти $\left\|v^i(t)\right\| \leq V_{max}$ шляхом регулювання $\alpha^i(t)$. $h_a^i(t) \in \mathbb{R}$ є потенційною функцією до зважування, тобто

$$h_a^i(t) = \begin{cases} \frac{[q_{si}(t) - q_{xi}(t)]^T * [q_{si}(t) - q_{xi}(t)]}{2} & if\ \delta_i(t) \geq \frac{R}{2} \\ 0 & otherwise \end{cases} \tag{3.21}$$

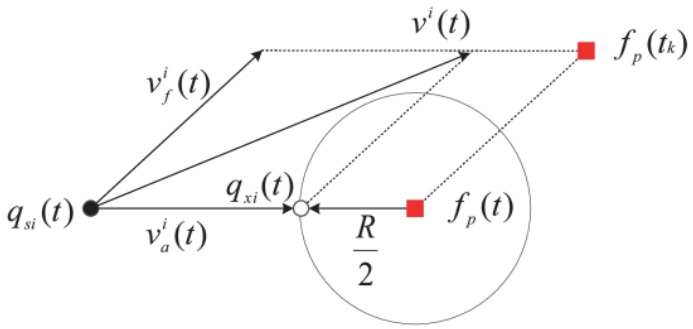


Рисунок 3.14 – Складові швидкості підмережі S_i

Коли відстань між підмережею S_i та точкою шляху $\delta_i(t) \leq \frac{R}{2}$, $h_a^i(t) = 0$.

Потенційна функція дозволяє всім підмережам рухатися до поточної маршрутної точки. Коли зустрічаються дві непересічні підмережі, вони будуть об'єднані в одну підмережу, і на основі прийнятого алгоритму

виборів буде обрано нового головного вузла, а потім новообраний головний вузол перерахує потенціал об'єднаної підмережі. Поки швидкість агрегації контролюється належним чином, зіткнень між підмережами не буде.

Швидкість $v^i(t)$ підмережі S_i може бути розкладена на дві складові швидкості: складову швидкості $v_f^i(t)$, що генерується потенційною функцією відстеження траєкторії польоту, і складову швидкості $v_a^i(t)$, що генерується потенційною функцією агрегування підмереж (рисунок 3.14).

Щоб у кожній підмережі можна було відстежувати попередньо визначену траєкторію польоту, швидкість кожної підмережі для відстеження маршрутної точки має бути узгодженою. Коли швидкість $\|v^i(t)\|$ підмережі S_i перевищує V_{max}, $\|v^i(t)\|$ можна зменшити шляхом зменшення розміру компонента швидкості $v_a^i(t)$. Щоб швидко об'єднати розосереджені підмережі в пов'язану підмережу, необхідно максимізувати значення $\alpha^i(t)$, як показано в (3.22).

$$\begin{cases} v^i(t) = v_f^i(t) + v_a^i(t) \\ 0 \le \|v^i(t)\| \le V_{max} \\ \alpha^i(t) = \arg\max_{\alpha}(\|v^i(t)\|) \end{cases}, \tag{3.22}$$

Згідно з наведеними вище результатами аналізу можна визначити швидкість кожної підмережі для відновлення мережі.

Коли час, витрачений на процес агрегації, перевищить T, процес відновлення буде припинено, а набір вузлів well-working стану S_ω, що відповідає поточному рою UAV, буде оновлено.

3.3.4 Фаза завершення процесу відновлення

У цьому розділі аналізується теоретична верхня межа T часу завершення процесу відновлення, тобто головний вузол, який ініціює процес

відновлення, повинен чекати щонайбільше T часу, щоб переконатися, що мережу відновлено до well-working стану.

Час T, витрачений на процес відновлення мережі, включає час, необхідний для виявлення пошкодження та агрегування підмереж, які наведені нижче.

Коли USNET зазнає серйозних пошкоджень, пошкодження можна ідентифікувати щонайбільше за час $T_{\text{identify}}^{max}$.

$$T_{\text{identify}}^{max} = T_{poll} + 2k + RTO + \xi, \tag{3.23}$$

де $\xi > 0$ – максимальний час виборчого процесу, який визначається обраним алгоритмом виборів;

Коли головний вузол виявляє пошкодження та ініціює процес агрегування мережі, для об'єднання всіх уцілілих вузлів у підключену підмережу потрібно щонайбільше $T_{aggregate}^{max}$ часу, а відстань між підмережею та точкою маршруту менше $\frac{R}{2}$. $T_{aggregate}^{max}$ показано таким чином:

$$T_{aggregate}^{max} = T_{move}^{max} + RTT + \tau, \tag{3.24}$$

де T_{move}^{max} визначено в (3.25), і воно представляє максимальний час для переміщення підмережі до кругової області навколо маршрутної точки;

$\tau > 0$ представляє максимальний час, необхідний для об'єднання підмереж;

τ – це емпіричне значення, пов'язане з поточною маршрутизацією. протокол і розмір мережі.

$$T_{move}^{max} = \frac{\max\limits_{u_i \in U_{(t_{damaged})}} \left\| q_i(t_{damaged}) - f_p(t_{damaged}) \right\| - \frac{R}{2}}{V_{max} - \|v_f\|}, \tag{3.25}$$

Коли процес відновлення ініціюється вузлами USNET, мережа може бути відновлена до нормального робочого стану за час T, який визначається як:

$$T = T_{poll} + T_{move}^{max} + 2k * RTO + RTT + \xi + \tau, \quad (3.26)$$

Коли USNET зазнає серйозного пошкодження, час, необхідний для відновлення мережі, включає час, необхідний $T_{identify}$ для виявлення пошкодження та $T_{aggregate}$ для агрегування підмереж.

$$T = T_{identify}^{max} + T_{aggregate}^{max} = T_{poll} + 2k * RTO + RTT + \xi + \tau, \quad (3.27)$$

що узгоджується з (3.27). Тобто мережа відновлюється до нормального робочого стану протягом часу T після пошкодження.

3.4 Висновок до розділу

У цьому розділі було розглянуто 3 метода (AuR, RTN, SIDR), які слугують для забезпечення відновленню зв’язку в високомобільному рою БПЛА при винекнені надзвичайної ситуації.

Метод AuR дозволяє мережі відновлювати з’єднання лише за допомогою локальної координації між вузлами в окремих сегментах. Ідея полягає в саморозповсюдженні вузлів і переміщенні їх до центру зони розгортання. AuR моделює рух як електростатичні сили притягання та відштовхування та залишає прийняття рішень і координацію в руках вузла та його сусідів з 1-стрибком.

Метод RTN полягає в тому, що коли виникає надзвичайна ситуація виконується процес самовиявлення окремих кластерів. Учасники переговорів від кожного кластера передають інформацію про кластер і переходять до круглого столу, де відбувається обмін інформацією, після чого йдуть

переговори щодо вирішення шляхів повторного підключення та вузлів, які використовуються в процесі повторного підключення. Після повернення учасників переговорів до своїх відповідних кластерів виконується процес повторного підключення.

У наступному розділі буде моделюватись метод SIDR з огляду на його унікальні можливості та переваги над іншими методами. SIDR використовує ройовий інтелект, де кожен БПЛА діє як агент та самостійно коригує свою поведінку, враховуючи власні можливості зберігання, зв'язку, обчислень, позиціонування та мобільності. Цей метод сприяє самоорганізації вцілих БПЛА та може допомогти відновити зв'язок та подолати розриви у мережі, що спричинені аварійними ситуаціями, забезпечуючи більш ефективні та швидкі результати.

Обрання методу SIDR для моделювання в наступному розділі обумовлене його спроможністю пристосовуватись до змінних умов, забезпечуючи більшу стабільність та швидкість відновлення зв'язку у надзвичайних ситуаціях порівняно з іншими методами. Його здатність до автономності та ефективного управління ресурсами може значно покращити відновлення мережі в умовах непередбачуваних ситуацій.

4 МОДЕЛЮВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ

Середовище для тестування та моделювання методу SIDR виконується в симуляторі OMNet++. Вузли рівномірно розподіляються з визначеною щільністю D на певній площі квадрата з однаковою висотою. Кругла область з початковою маршрутною точкою як центром і $\frac{R}{2}$ як радіусом може охоплювати геометричний центр USNET і принаймні один вузол. Коли всі вузли досягають консенсусу щодо інформації про траєкторію польоту та топології, вузли починають пошкоджуватися з певною ймовірністю.

Таблиця 4.1 – Оцінка факторів виробничого середовища і трудового процесу

Параметр	Значення
Simulation Area ($lengh \times width \times height$)	$10km \times 10km \times 2\ km$
Routing Algorithm	OLSR
Data rate of link	$54\ Mbps$
Transmission range (R)	$200\ m$
Maximum flight speed (V max)	$20\ m/s$
Node density (D)	$45\ nodes/km^2$
Poll interval (T poll)	$5\ s$
Retransmission Timeout (RTO)	$1\ s$
Round Trip Time (RTT)	$0.1\ s$
Merging Time (τ)	$5\ s$
Retransmission times (k)	3
Distributed election algorithm	Bully

Шляхом моделювання перевіряється продуктивність методу SIDR за різних рівнів пошкодження (від випадково пошкоджених вузлів до всіх

вузлів). Сценарій динамічного тестування рою БПЛА встановлюється відповідно до моделі пошкодження, а метод SIDR перевіряється з точки зору часу конвергенції, відповідності між попередньо визначеною траєкторією польоту та траєкторією вузлів, а також витрати на зв'язок. Так званий динамічний сценарій означає, що зона покриття мережі динамічно змінюється через необхідність відстеження траєкторії польоту. Статичний сценарій означає, що зона покриття мережі залишається незмінною. Методи AuR і RTN в основному зосереджена на статичних сценаріях, тому метод SIDR порівнюється з методами AuR і RTN в статичному сценарії з точки зору часу конвергенції процесу відновлення та кількості надісланих повідомлень.

4.1 Динаміний сценарій

Перший сценарій використовується для перевірки того, чи може запропонований механізм SIDR відновити пошкоджену мережу USNET до нормального робочого стану протягом часу T. По-перше, ми перевіряємо, чи задовольняє механізм SIDR T-обмежене зв'язне обмеження. Траєкторія польоту під час випробування задана як пряма лінія і $\| v_f \| = 10$ м/с. Кількість вузлів $n \in \{49,64,81,100\}$. Рівень шкоди коливається від 20% до 90%. Час ідентифікації пошкодження при різній кількості вузлів і рівнях пошкодження майже не змінюється (рисунок 4.1). Загалом, коли кількість вузлів однакова, час агрегації збільшується зі збільшенням шкоди. Це пояснюється тим, що кількість вузлів, що вижили, зменшується зі збільшенням рівня пошкодження, що призводить до того, що вцілілі вузли мають меншу ймовірність зустрітися з іншими вузлами під час процесу агрегації, що призводить до збільшення відстані переміщення. Коли рівень пошкодження однаковий, час агрегації збільшується зі збільшенням кількості вузлів, особливо коли рівень пошкодження перевищує 30%. Це пояснюється тим, що охоплення мережі розширюється зі збільшенням кількості вузлів, що

призводить до збільшення відстані від вузлів до точки агрегації, таким чином споживаючи довший час переміщення. Однак, коли рівень пошкодження низький, вузлам потрібно лише переміститися на невелику відстань, щоб зустріти інші вузли. У цьому випадку більша частина часу агрегації використовується для відновлення маршрутизації між підмережами.

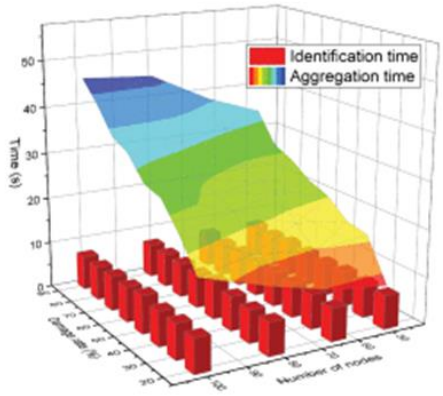


Рисунок 4.1 – Час ідентифікації та агрегації з різною кількістю вузлів і рівнем пошкоджень

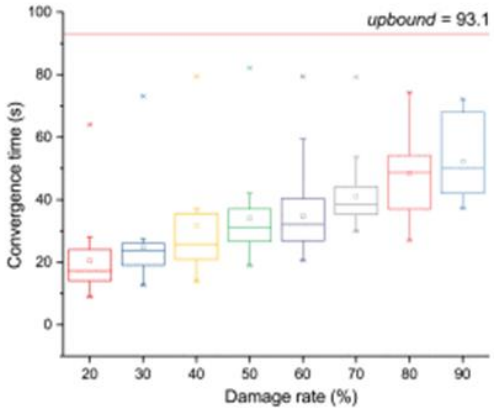


Рисунок 4.2 – Розсіювання часу конвергенції з різними рівнями пошкодження, коли кількість вузлів становить 81

Дисперсію часу конвергенції при змінних рівнях пошкодження при кількості вузлів 81 (рисунок 4.2). Можна побачити, що зі збільшенням рівня

пошкодження середній час сходження процесу відновлення збільшується, але в гіршому випадку , час збіжності спочатку збільшується, а потім зменшується. Час конвергенції в найгіршому випадку досягає максимуму, коли рівень пошкодження становить 50%. Це пов'язано з тим, що коли коефіцієнт пошкодження становить не більше 50%, максимальна відстань між підмережею та точкою шляху збільшується зі збільшенням рівня пошкодження, що призводить до збільшення часу T_{move}^{max} для переміщення підмережі в кругову область навколо шляхова точка. Коли рівень пошкодження перевищує 50%, максимальна відстань між підмережею та точкою маршруту в гіршому випадку залишається незмінною, але час злиття підмереж T_{merge} дещо зменшується зі зменшенням кількості вцілілих вузлів. Час конвергенції при змінних рівнях пошкодження не перевищуватиме теоретичну верхню межу (рисунок 4.2). Іншими словами, механізм SIDR задовольняє T-обмежене пов'язане обмеження.

На рисунку 4.3 показано стан USNET після серйозного пошкодження. У момент $t = 20s$ USNET розділяється на 6 роз'єднаних підмереж. На рисунку 4.4 показано процес агрегації USNET, а його ордината представляє відстань між підмережею та маршрутною точкою. Коли $t \approx 28s$, підмережі ідентифікують пошкодження та ініціюють процес агрегування мережі. Коли $t \approx 39s$, підмережі S_1 і S_3 зливаються в нову підмережу $S_{1,3}$. Підмережі S_2 і $S_{1,3}$ стають новою підмережею $S_{1,2,3}$, коли $t \approx 42s$. У $t \approx 43s$ підмережі S_4 і S_5 зливаються в нову підмережу $S_{4,5}$. Коли $t \approx 47s$, підмережі $S_{1,2,3}$ і $S_{4,5}$ зливаються в нову підмережу $S_{1,2,3,4,5}$. Нарешті, в $t \approx 55s$ підмережі S_0 і $S_{1,2,3,4,5}$ зливаються в остаточну підмережу $S_{0,1,2,3,4,5}$. Відстань між підмережами та маршрутною точкою зменшується лінійно, а відстань між остаточною об'єднаною підмережею та шляховою точкою не перевищує $\frac{R}{2}$, що задовольняє визначення добре працюючої USNET (рисунок 4.4).

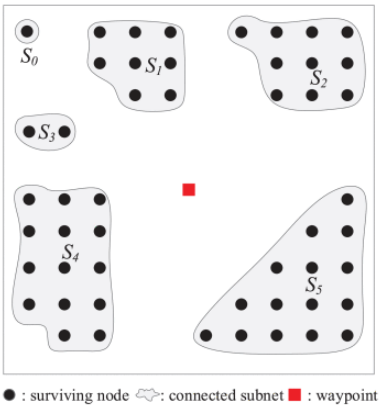


Рисунок 4.3 – Пошкоджений USNET

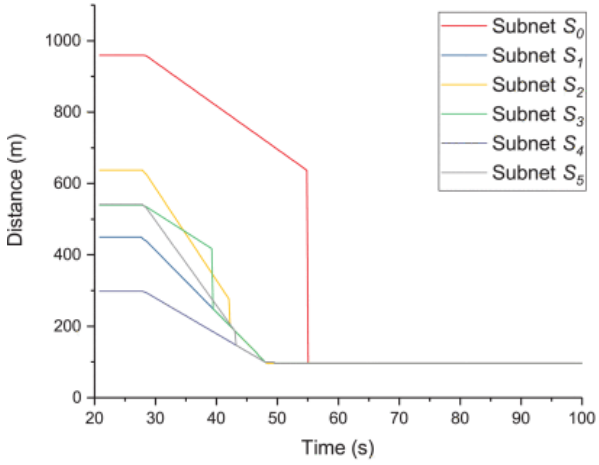


Рисунок 4.4 – Процес агрегації

Другий сценарій використовується для перевірки того, чи задовольняє механізм SIDR обмеження Track, де кількість вузлів становить 81, а початкова точка маршруту розташована в геометричному центрі USNET. Щоб чітко продемонструвати траєкторію вцілілих вузлів, усі вузли, крім чотирьох вузлів у куті, пошкоджені, коли $t = 20\ s$. Коли відбувається пошкодження, підмережі відстежуватимуть траєкторію польоту під час агрегування до точки маршруту, яка демонструє хорошу відповідність між попередньо визначеною траєкторією польоту та траєкторією вузлів і задовольняє track обмеження (рисунок 4.5).

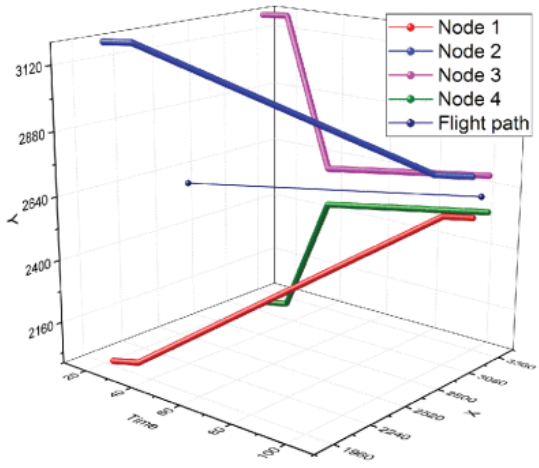


Рисунок 4.5 – Відстеження track обмеження

Третій сценарій використовується для перевірки накладних витрат зв’язку механізму SIDR. На рисунку 4.6 наведені накладні витрати на зв’язок для відновлення мережі з різною швидкістю пошкодження, коли кількість вузлів становить 81, а налаштування її параметрів такі ж, як на рисунку 4.2. Накладні витрати на зв’язок на етапі ідентифікації пошкоджень зменшуються зі збільшенням рівня пошкоджень (рисунок 4.6). Це пов’язано з тим, що накладні витрати на етапі ідентифікації пошкодження в основному включають накладні витрати на вибір нового головного вузла, а накладні витрати на вибори зменшуються разом із кількістю вузлів, що вижили. Зв'язок фази агрегації мережі в основному включає накладні витрати на опитування та злиття підмереж. Накладні витрати на опитування позитивно корелюють із часом відновлення та кількістю збережених вузлів, а накладні витрати на злиття підмереж позитивно корелюють із кількістю розділів і збережених вузлів. Зі збільшенням рівня пошкодження час відновлення зростає (рисунок 4.2), кількість розділів спочатку збільшується, а потім зменшується, і досягає максимуму, коли рівень пошкодження становить 70% (рисунок 4.6). Завдяки вищезазначеним факторам накладні витрати на зв’язок фази агрегації спочатку зростають, а потім зменшуються зі збільшенням рівня пошкодження, і досягають максимуму, коли рівень пошкодження

становить 60% (рисунок 4.6). Загальні накладні витрати на зв'язок включає накладні витрати на етапі ідентифікації пошкодження та етапі агрегування мережі. Зі збільшенням рівня пошкодження загальні накладні витрати на зв'язок зазвичай демонструють тенденцію до зниження (рисунок 4.5).

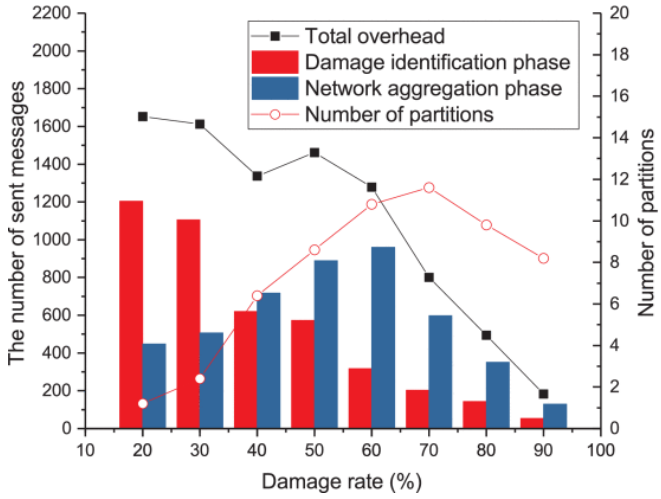


Рисунок 4.6 – Накладні витрати на зв'язок і кількість розділів із різним рівнем пошкодження, коли кількість вузлів становить 81

Результати випробувань показують, що метод SIDR може задовольнити вимоги стійкості до пошкоджень у динамічних сценаріях і сходитися за певний час, і це підтверджує теоретичний аналіз.

4.2 Статичний сценарій

Продуктивність механізму SIDR порівнюється з AuR і RTN з точки зору часу конвергенції та кількості надісланих повідомлень у статичному сценарії. Кількість вузлів n=100, а точка зустрічі розташована в геометричному центрі мережі. Вузли пошкоджуються з ймовірністю 20%-90%.

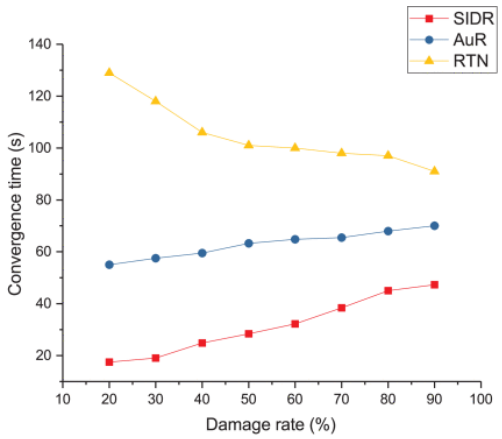


Рисунок 4.7 – Час конвергенції

Можна побачити, що час конвергенції методу SIDR є найкоротшим, за ним слідує механізм AuR, а метод RTN є найдовшим (рисунок 4.7). Це пояснюється тим, що вузли узгоджувача в методі RTN повинні переміститися до точки зустрічі, щоб узгодити стратегію відновлення, а потім повернутися до своїх відповідних розділів, що призводить до довшого часу відновлення.

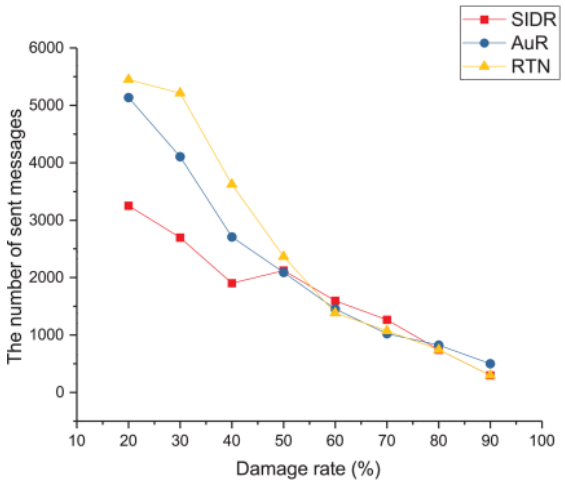


Рисунок 4.8 – Рівень пошкодження

Можна побачити, що коли рівень пошкодження становить менше 50%, метод SIDR надсилає найменшу кількість повідомлень, за ним слідує AuR, а метод RTN надсилає найбільшу кількість повідомлень (рисунок 4.8). Коли

рівень пошкодження перевищує 50%, кількість повідомлень, надісланих трьома механізмами, приблизно однакова.

ВИСНОВКИ

Метою кваліфікаційної роботи було дослідження методу забезпечення надійності полінгової мережі у поставарійному стані високомобільного вузла рою БПЛА.

Зв'язок відіграє важливу роль у контролі та координації зграї БПЛА. Комунікаційна архітектура визначає спосіб обміну інформацією між БПЛА або між БПЛА та центральним центром управління. Протоколи маршрутизації допомагають забезпечити надійну наскрізну передачу даних. Тому особливо важливо розробити архітектуру зв’язку роя БПЛА та протоколи маршрутизації з високою продуктивністю та стабільністю.

Рій БПЛА був представлені як багатообіцяюча парадигма для проведення моніторингу та взаємозв’язку завдань у неконтрольованих або навіть ворожих середовищах. Однак суворий сценарій розгортання може зробити БПЛА чутливими до великомасштабних пошкоджень й таким чином, погіршити підключення та продуктивність мережі.

Для досягнення мети курсового проекту було досліджено комунікаційну архітектуру рою БПЛА, визначено їх плюси та мінуси, досліджено протоколи маршрутизації, оглянуто існуючі стратегії забезпечення надійності мережі, а також проаналізувати основні аспекти, які можуть зробити БПЛА чутливими до пошкоджень. Після чого було розглянуто декілька існуючих методів, які спроможні забезпечити надійність мережі й відновити з’єднання у поставарійному стані. Після дослідження було обрано один метод з огляду на його унікальні можливості та переваги над іншими методами, який був розглянутий в моделюючій частині й порівняний з іншими методами.

Подальшим розвитком даної роботи може бути адаптація розглянутого методу до нових умов. Це включатиме в себе аналіз та адаптацію методу до змін у технологіях передачі даних, нових вимог щодо стійкості мережі та

впровадження нових технологій у сфері безпеки й зв'язку. Також оптимізація методу. Адаптація та налаштування параметрів обраних стратегій до специфіки високомобільних умов може підвищити їхню ефективність та стабільність у поставарійному стані.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Review of Unmanned Aerial Vehicle Swarm Communication Architectures and Routing Protocols. *MDPI*. URL: https://www.mdpi.com/2076-3417/10/10/3661 (дата звернення: 30.11.2023).

2. Search Strategies for Multiple UAV Search and Destroy Missions - Journal of Intelligent & Robotic Systems. SpringerLink. URL: https://doi.org/10.1007/s10846-010-9486-8 (дата звернення: 29.11.2023).

3. Erdelj, M.; Chowdhury, K.R.; Akyildiz, I.F.; Natalizio, E. Help from the Sky: Leveraging UAVs for Disaster Management. IEEE Pervasive Comput. 2017, 16, 24–32.

4. Beard, R.W.; McLain, T.W.; Nelson, D.B.; Kingston, D.; Johanson, D. Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs. Proc. IEEE 2006, 94, 1306–1324.

5. Barrado, C.; Meseguer-Pallarés, R.; Lopez, J.; Pastor, E.; Santamaria, E.; Royo, P. Wildfire monitoring using a mixed air-ground mobile network. IEEE Pervasive Comput. 2010, 9, 24–32.

6. Search Strategies for Multiple UAV Search and Destroy Missions - Journal of Intelligent & Robotic Systems. SpringerLink. URL: https://doi.org/10.1007/s10846-010-9486-8 (дата звернення: 30.11.2023).

7. Cho, A.; Kim, J.; Lee, S.; Kee, C. Wind Estimation and Airspeed Calibration using a UAV with a Single-Antenna GPS Receiver and Pitot Tube. IEEE Trans. Aerosp. Electron. Syst. 2011, 47, 109–117.

8. Li, J.; Zhang, X.L.; Bao, J.H.; Geng, G.L. A Novel DSR-Based Protocol for Signal Intensive UAV Network. Appl. Mech. Mater. 2012, 241, 2284–2289.

9. Singh, K.; Verma, A.K. Experimental analysis of AODV, DSDV and OLSR routing protocol for flying adhoc networks (FANETs). In Proceedings of the 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Tamilnadu, India, 5–7 March 2015;

IEEE: New York, NY, USA, 2015; pp. 1–4.

10. Alshbatat, A.I.; Dong, L. Cross layer design for mobile ad-hoc unmanned aerial vehicle communication networks. In Proceedings of the 2010 International Conference on Networking, Sensing and Control (ICNSC), Chicago, IL, USA, 10–12 April 2010; IEEE: New York, NY, USA, 2010; pp. 331–336.

11. Alshabtat, A.I.; Dong, L.; Li, J.; Yang, F. Low latency routing algorithm for unmanned aerial vehicles ad-hoc networks. Int. J. Electr. Comput. Eng. 2010, 6, 48–54.