

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки
(повне найменування інституту, факультету)

Кафедра обчислювальної техніки
(повна назва кафедри)

До захисту допущено:

Завідувач кафедри

_____ Сергій СТИРЕНКО
(підпис) (ім'я, прізвище)

«__» _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою “Комп'ютерні системи та мережі”

спеціальності 123 “Комп'ютерна інженерія”

на тему “Спосіб кластеризації в мобільних SDN мережах VANET на основі fuzzy logic алгоритмів”

Виконала: студентка VI курсу, групи ІО-01мп

Осієвська Валентина Сергіївна

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник ст. викл., к.т.н., Куц Володимир Юрійович

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

_____ (підпис)

Консультант н. контроль доц. д. т. н. Кулаков Ю. О.

(назва розділу) (вчене звання та звання, прізвище, ініціали)

_____ (підпис)

Рецензент доц. каф. СПіСКС к.т.н., доц. Орлова М. М.

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

_____ (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.

Студентка _____
(підпис)

Київ - 2021 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки
(повне найменування інституту, факультету)

Кафедра обчислювальної техніки
(повна назва кафедри)

Рівень вищої – освіти – другий (магістерський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма – «Комп'ютерні системи та мережі»

До захисту допущено:

Завідувач кафедри

_____ Сергій СТИРЕНКО
(підпис) (ім'я, прізвище)

«__» _____ 2020 р.

ЗАВДАННЯ

на дипломний проект студентці

Осієвській Валентині Сергіївні

1. Тема проекту “Спосіб кластеризації в мобільних SDN мережах VANET на основі fuzzy logic алгоритмів”, керівник дисертації Куц Володимир Юрійович, старший викладач, кандидат технічних наук, затверджені наказом по університету від «26» жовтня 2021р. № 3132-с
2. Термін подання студентом проекту 24.11.2021р.
3. Вихідні дані до проекту: технічне завдання, теоретичні дані, наукові публікації.
4. Зміст пояснювальної записки: огляд методів кластеризації мобільних мереж, конструювання трафіку в програмно визначених транспортних мережах, проектування та розробка програмного забезпечення, тестування програмного модулю та аналіз отриманих результатів.

5. Перелік завдань, які потрібно розробити: огляд існуючих рушень для кластеризації інтелектуальних мереж, спосіб кластеризації в мобільних SDN мережах VANET на основі алгоритмів нечіткої логіки, побудова програмної моделі згідно розробленого алгоритму кластеризації, тестування та аналіз розробленої моделі

6. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1.	ст.викл., Куц В.Ю.		
2.	ст.викл., Куц В.Ю.		
3.	ст.викл., Куц В.Ю.		
3.	ст.викл., Куц В.Ю.		

7. Дата видачі завдання 01.09.2021 року

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів дипломного проекту (роботи)	Строк виконання етапів проекту(роботи)	Примітки
1	Обговорення теми дисертації. Визначення предмету дослідження	1.09.21-7.09.21	
2	Дослідження існуючих рішень та проблем кластеризації у досліджуваній області	22.09.21-28.09.21	
3	Побудова архітектурного рішення	29.09.21-05.10.21	
4	Програмна реалізація створеного алгоритму	06.10.21-10.10.21	
5	Розборка функціоналу по самоорганізації кластеру за принципами нечіткої логіки	10.10.21-19.10.21	
6	Розробка процесу обробки, агрегації та акумуляції мережесих даних	10.11.21-13.11.21	
7	Розробка стартам-проекту	14.11.21-18.11.21	
8	Оформлення магістерської дисертації.	19.11.21-24.11.21	

Студентка Валентина ОСІЄВСЬКА _____ (підпис)

Керівник Володимир КУЦ _____ (підпис)

Анотація

Магістерська дисертація присвячена вирішенню проблем пов'язаних із високою динамічністю топології та масштабованістю мережі. Застосування ефективного алгоритму кластеризації в мобільних мережах VANET дозволяю вирішити ці проблеми, а також сприяють покращенню процесів маршрутизації, безпеки та керування системи. У цій роботі запропоновано універсальний гібридний алгоритм кластеризації для мобільний мереж, що базується на поєднанні технологій машинного навчання та нечіткої логіки. Згідно аналізу отриманих результатів експериментального дослідження, запропонований спосіб продемонстрував найбільш релевантні показники по формуванню кластерів, що може достатньо зменшити загальну вартість транспортної мережі. Більш того, це також сприяє зниженню кількості затримок та кількості стрибків повідомлення у межах одного кластеру.

Annotation

The master's dissertation is devoted to solving problems related to the high dynamics of topology and scalability of the network. An efficient clustering algorithm usage in VANET mobile networks allows us to solve these problems and contribute to the improvement of routing, security and system management processes. This paper proposes a universal hybrid-clustering algorithm for mobile networks based on a combination of machine learning technologies and fuzzy logic. According to the analysis of the results of the experimental study, the proposed method demonstrated the most relevant indicators for the formation of clusters, which can significantly reduce the total cost of the transport network. Moreover, it also helps to reduce the number of delays and the number of message jumps within a single cluster.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	7
ВСТУП	8
РОЗДІЛ 1	11
ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ДЛЯ КЛАСТЕРИЗЦІЇ ІНТЕЛЕКТУАЛЬНИХ МЕРЕЖ.....	11
1.1 Інтелектуальні транспортні мережі.....	11
1.1.1 Характеристика VANETs.....	12
1.1.2 Основні компоненти мережі VANET	14
1.1.3 Архітектура спеціальних транспортних мереж.....	15
1.1.4 Програмне забезпечення VANET	19
1.1.5 Дослідницькі проблеми та відкриті питання	20
1.2 Система VANET на основі SDN.....	22
1.2.1 Програмно визначена архітектура VANET.....	24
1.2.2 Проблеми, пов'язані з Cross layer	27
1.3 Огляд кластеризації у VANET.....	28
1.3.1 Базова концепція кластеризації VANET	28
1.3.2 Таксономія кластеризації транспортної мережі	31
1.3.3 Конкуруючі рішення по кластеризації вузлів.....	32
ВИСНОВКИ ДО РОЗДІЛУ 1	34
РОЗДІЛ 2	36
СПОСІБ КЛАСТЕРИЗЦІЇ В МОБІЛЬНИХ SDN МЕРЕЖАХ VANET НА ОСНОВІ АЛГОРИТМІВ НЕЧІТКОЇ ЛОГІКИ.....	36
2.1 Постановка задачі та опис архітектури системи	36
2.1.1 Формування кластеру та вибір СН.....	38
2.1.2 Ініціювання вибору нового СН та передача керування	39
2.1.3 Злиття кластеру та додавання нових вузлів	40
2.1.4 Вихід транспортного засобу за межі кластеру	42
2.2 Обумовленість вибору алгоритму кластеризації.....	42
2.2.1 Алгоритми машинного навчання	43
2.2.2 Алгоритми нечіткої логіки.....	44
2.2.3 Гібридні алгоритми кластеризації.....	46

2.3 Математична модель процесу кластеризації.....	48
2.3.1 Метрики для вибору СН.....	48
2.3.2 FLS для формування кластерів.....	51
ВИСНОВКИ ДО РОЗДІЛУ 2	53
РОЗДІЛ 3	55
ПОБУДОВА ПРОГРАМНОЇ МОДЕЛІ ЗГІДНО РОЗРОБЛЕНОГО АЛГОРИТМУ КЛАСТЕРИЗАЦІЇ	55
3.1 Розробка системи генерування мережі згідно заданих характеристик ..	55
3.2 Візуалізація мережі VANET	58
3.3 Реалізація алгоритму DBSCAN	59
3.4 Реалізація алгоритму вибору СН.....	62
3.5 Аналіз роботи запропонованого способу кластеризації	65
3.5.1 Перевірка роботи алгоритму на експериментальній моделі	65
3.5.2 Порівняння гібридного алгоритму із існуючими рішеннями	71
ВИСНОВКИ ДО РОЗДІЛУ 3	74
РОЗДІЛ 4	76
РОЗРОБКА СТАРТАП ПРОЕКТУ	76
4.1. Загальна характеристика стартап-проекту	76
4.2. Технологічний аудит ідеї проекту.....	80
4.3. Аналіз ринкових можливостей запуску стартап-проекту.....	81
4.4. Розробка ринкової стратегії проекту	89
4.5 Розроблення маркетингової програми стартап-проекту.....	92
ВИСНОВКИ ДО РОЗДІЛУ 4	94
ВИСНОВКИ.....	95
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	98
ДОДАТОК 1	105
Частина програмного коду	105

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ITS	(Intelligent Transport Systems) Інтелектуальні транспортні системи
VANET	(Vehicular ad-hoc networks) Спеціальні транспортні мережі
RSU	(Road Side Unit) Розподілена система управління
CH	(Cluster Head) Голова кластеру
CM	(Cluster Member) Учасник кластеру
V2V	(Vehicle-to-vehicle) Зв'язок транспортний засіб - транспортний засіб
V2I	(Vehicle-to-Infrastructure) Зв'язок транспортний засіб – інфраструктура
SDN	(Software-defined networking) Програмно-конфігурована мережа
ZOR	(Zone of Relevance Computation) Обчислення зони релевантності
СaaS	(Cooperation as a service) Співфункціонування як сервіс

ВСТУП

Згідно останніх досліджень наукової області, автомобільної промисловості та державного транспортного сектору, інтелектуальні транспортні системи (ITS) наразі є ключовою технологією для вирішення проблем і задач, які можуть виникнути під час дорожнього руху. Бездротові самоорганізуючі мережі VANET можуть підтримувати широкий вибір програмної логіки, від базової передачі пакетів даних між сусідніми вузлами до розповсюдження повідомлень на великі відстані, проходячи велику кількість вершин. Така здатність VANET стає причиною підвищення рівня безпеки під час дорожнього руху, зниження трафіку, підвищення ефективності та якості водіння, тощо.

Інтелектуальні транспортні мережі являють собою чудовий ресурс для реалізації та застосування будь-якого бездротового програмного забезпечення, зорієнтованого на підвищення якості процесів, що виникають під час руху автомобілів. Особливістю такої мережі є здатність транспортних засобів встановлювати зв'язок у режимі реального часу з іншими транспортними засобами, що знаходяться на допустимій відстані, та з наявною мережевою інфраструктурою. Застосування цієї здатності надає можливість водіям отримувати актуальну інформацію про надзвичайні ситуації, які можуть виникнути на дорогах, уникати аварійних випадків, надавати допомогу у керуванні автомобілем або надавати звіти про погодні та дорожні умови.

Завдяки своїй мобільності та частоті зміни положення транспортних засобів у часі, VANET має динамічну топологію. Для такої топології є характерними часті виключення між вузлами та періодична втрата повідомлень, особливо при збільшенні кількості транспортних засобів [1]. Саме тому основним завданням протоколу зв'язку є забезпечення надійності комунікації транспортних вузлів та масштабування системи. Одним із методів вирішення проблем забезпечення стабільності, масштабованості та

підвищення продуктивності автомобільних мереж є застосування методів кластеризації транспортних вузлів.

У проєкції на інтелектуальну мережу, кластер являє собою логічну підмножину вершин, об'єднаних подібними характеристиками. Кожен кластер має визначену структуру відповідно до особливостей самої мережі. Динамічний рух автомобілів зумовлює приєднання або виїзд транспортного засобу за межі кластеру відповідно до заданих показників швидкості та відстані до ідентифікованої голови кластера (CH). Серед обов'язків CH також є забезпечення внутрішньо кластерної, так і міжкластерної комунікації.

Окрім безпосереднього спілкування вузла з CH, у автомобілів є можливість спілкування з іншими членами кластеру (CM) на базі зв'язку транспортний засіб – транспортний засіб (V2V) [2]. Зв'язок V2V є гнучким і незалежним по відношенню до дорожніх умов, що робить його вдалим вибором для районів, де придорожня інфраструктура (RSU) не обов'язково наявною [3].

Метою даної роботи є вирішення однієї з ключових проблем формування кластерів та пошук оптимальної кількості необхідних кластерів для стабільного функціонування системи. Ця проблема є ваговою, оскільки занадто велика кількість кластерів може призвести до більших затримок, тоді як менша кількість кластерів може призвести до переривчастого з'єднання. Функціональне наповнення середовища кластеру також має бути сформовано таким чином, щоб забезпечити стабільну роботу всього кластеру та попередити втрату зв'язку та інформації під час взаємодії.

У цій роботі запропоновано спосіб кластеризації в мобільних SDN мережах на основі алгоритмів нечіткої логіки. Дане рішення представлено у вигляді гібридного алгоритму для попереднього визначення оптимальної кількості кластерів, базуючись на щільності розподілення транспортних вузлів на дорогах, та подальшого використання отриманого значення для формування логічних груп та оптимального вибору CH.

У межах даної роботи, об'єктом дослідження виступають інтелектуальні транспортні мережі VANET.

Предметом дослідження є реалізація способу кластеризації в мобільних SDN мережах на основі алгоритмів нечіткої логіки.

РОЗДІЛ 1

ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ДЛЯ КЛАСТЕРИЗЦІЇ ІНТЕЛЕКТУАЛЬНИХ МЕРЕЖ

1.1 Інтелектуальні транспортні мережі.

З моменту появи інформаційно-комунікаційних технологій (ІКТ) та бездротових запрограмованих датчиків, вбудованих у транспортні засоби, Інтелектуальна транспортна система (ITS) набуло широкого розвитку та стала незамінним інструментом для побудови розумних міст.

Основною метою ITS постає вирішення проблем пов'язаних з безпекою дорожнього руху та підвищенням його ефективності. Крім того є цілком можливий функціонал, присвячений забезпеченню надання інформаційних даних про стан та прогнозування мережі. Безумовно, наявність своєчасного попередження водіїв автомобілів про непередбачувані ситуації, що можуть виникнути на дорогах, та надання аналізу та рекомендацій на базі аналізу транспортної мережі, може сприяти підвищенню рівня безпеки людей та зменшенню кількості заторів.

У основі інтелектуальної транспортної системи лежать самоорганізовані бездротові мережі. Спеціальні транспортні мережі VANET можна визначити як бездротову мережу без попередньо визначеної топології, де вузли постійно змінюють своє положення відносно один одного та інфраструктури у реальному часі. У мережах з такою динамічною топологією, кожен вузол можна вважати кінцевим або маршрутизатором. Згідно такої умови, кожний транспортний засіб має обмінюватись власними ресурсами з іншими, сприяючи побудові транспортної мережі.

У якості базової характеристики описаного транспортного середовища, використовуються такі поняття як: мобільність та швидкість пересування.

Мобільність спеціальної мережі VANET пояснюється відсутністю хаотичного руху автомобілів у просторі. Транспортні засоби, як правило, рухаються уздовж доріг, вулиць або шосе. Це говорить про те, що вузли

мережі, у більшій або меншій мірі, будуть рухатись по заздалегідь визначеному маршруту і їх рух не є випадковим, а слідує заданому напрямку, відповідно до існуючої карти місцевості, дорожніх обмежень та знаків.

Також цілком можливим є прогнозування високої концентрації авто в залежності від місцевості (перехрестя, автомагістралі) та часу. Під час таких ситуацій, може виникати нестабільне з'єднання, що призведе до погіршення продуктивності роботи мережі, спричиненого зіткненнями у бездротовому середовищі у дуже щільних регіонах, і частою фрагментацією мережі в розріджених зонах [4].

Беручи до уваги високу швидкість пересування транспортних засобів, необхідно також враховувати частоту зміни сусідства вузлів у силу різниці швидкості між ними та можливу протилежність напрямків руху. Таким чином, за короткий період часу суміжні до заданої вершини можуть кілька разів опинитись за межами заданого діапазону, що є причиною короткочасності зв'язку між вершинами та відсутності обслуговування підлеглого маршруту. Таку особливість динамічної топології необхідно враховувати при конструюванні трафіку всередині мережі.

1.1.1 Характеристика VANETs

Базуючись на попередніх дослідженнях спеціальних транспортних мереж [5] та [6], можна виділити наступні характеристики VANET, що відрізняються.

Першою та найбільш очевидною характеристикою мережі є мобільність, оскільки транспортні мережі переважно складаються з фіксованих RSU та транспортних вузлів, що перебувають у постійному русі. Відповідно, під час аналізу такої мережі варто враховувати не лише положення одного вузла відносно іншого, а й швидкість руху транспортного засобу (яка, до того ж, періодично коливається) та напрямок його руху. Такі аспекти самої мережі постають складним ускладненням для комунікації всередині мережі.

У місцевостях, для яких характерні часті затори, повільний рух автомобілей та їх часті зупинки суттєво покращують надійність передачі

даних. Проте, за таких обставин може виникнути ряд наступних ускладнень, пов'язаних з високою щільністю транспортних вузлів, таких як зіткнення даних, згасання каналів, відкидання повідомлень та інші проблеми з перешкодами. З іншого боку, у місцевостях низьким, наприклад, шосе, транспортні засоби навпаки рухаються на дуже високій швидкості, що призводить до інших проблем із зв'язком. Серед них дуже мале вікно зв'язку, часті збої зв'язку, велика наскрізна затримка (ETE) тощо.

Суттєвою відмінністю мобільної мережі VANET від MANET є чітка визначеність руху. Як відомо, вузли однорангової мережі MANET можуть вільно переміщуватись у будь-який момент часу по довільній траєкторії. [7] У той самий час вузли транспортної мережі пересуваються по завчасно географічно-визначеному шаблону у вигляді доріг, автомагістралей та шосе. Визначеність руху транспорту також залежить від місцевості. Таким чином міська територія має складнішу мережу доріг та більш високу зв'язність за кількістю транспортних засобів, ніж сільська місцевість. Крім того, пересування таким середовищем супроводжуються більшою кількістю перешкод, таких як багатосмуговість, перехрестя, сигнали світлофора та RSU. Просторові атрибути дорожньої мережі впливають на ефективність комунікацій.

З попереднього абзацу випливає, що щільність руху вузлів виступає окремою характеристикою мережі. Вона коливається залежно від географічного регіону, (тобто висока щільність руху притаманна міській місцевості, а низька щільність руху - сільській та на автомагістралі) та часового періоду (тобто висока щільність руху та численні затори у пікові години). Основні проблеми, що спричиняє висока щільність пов'язані з розробкою ефективних протоколів зв'язку VANET. Наприклад, у сільській місцевості протоколи передачі даних мають бути здатні ефективно вирішувати проблеми з відключенням мережі.

Спеціальні мережі VANET також характеризуються неоднорідністю самих вузлів. Транспортні засоби та їх програмного забезпечення мають свої

власні діапазони зв'язку, потужність передавача, чутливість бездротового інтерфейсу автомобіля, у той час як RSU-це стаціонарні вузли, розміщені у фіксованих місцях та мають стандартне обладнання зі спеціальними функціями, визначеними протоколом.

1.1.2 Основні компоненти мережі VANET

Відповідно до стандартів IEEE 1471-2000 [8] та ISO/IEC 42010 [8], компоненти мережі VANET класифікуються в наступних трьох областях:

- Мобільний домен. Включає в себе домени будь-якого транспортного засобу, у тому числі автомобілі, автобуси, мотоцикли поїзди, тощо. Та мобільного пристрою, наприклад, смартфони, розумні годинники.
- Інфраструктурний домен. Включає придорожні інфраструктурні домени, такі як світлофор, дорожні камери, а також центральний дорожній інфраструктурний домен. Найбільш популярним його прикладом є RSU центр управління дорожнім рухом або центр управління транспортними засобами.
- Універсальний домен. Включає в себе Інтернет та приватну архітектуру.

VANET може бути реалізовані різними типами зв'язків, наприклад, бездротовий зв'язок між транспортними засобами (V2V). Зв'язок «інфраструктура-інфраструктура» (I2I) між RSU дозволяє розширити покриття мережі. Транспортний засіб-інфраструктура (V2I) відноситься до двонаправленого бездротового зв'язку між транспортними засобами та RSU, підключеними до інфраструктури. Зв'язок від транспортних засобів до широкопasmової хмари (V2B) між транспортними засобами та широкопasmовою хмарою за допомогою бездротових широкопasmових технологій, таких як 3G/4G. [9]. На рисунку 1.1 зображено приклад взаємодії транспортної мережі з використанням різних типів зв'язку.

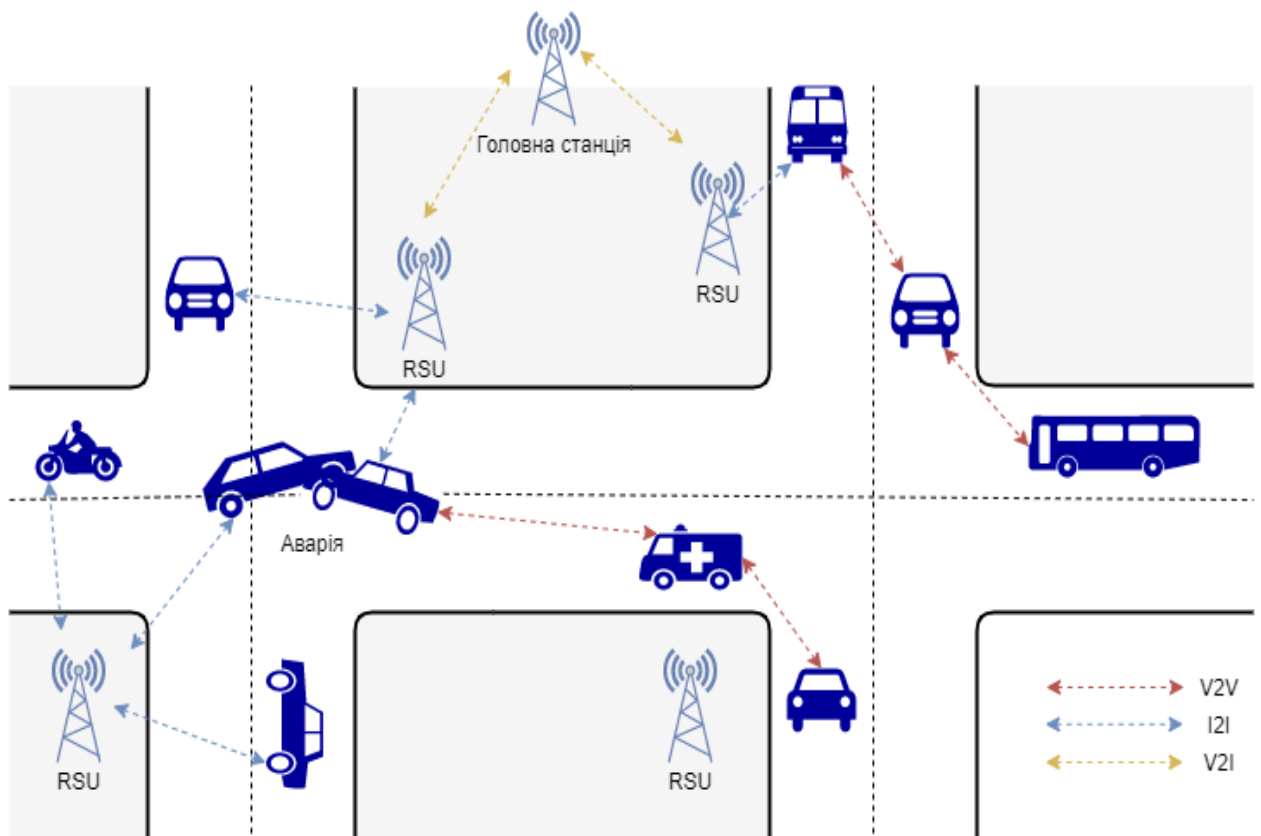


Рис. 1.2. Структура комунікації мережі VANET

1.1.3 Архітектура спеціальних транспортних мереж

В інтелектуальних транспортних мережах можна виділити дві основних будови архітектури [10].

Перший вид архітектури характеризується відсутністю інфраструктурного рівня. У таких мережах VANET кожен вузол оснащений бездротовим інтерфейсом [11], що використовується для встановлення та підтримки зв'язку між транспортними засобами. Першочерговою проблемою відсутності інфраструктури є неможливість комунікації через інтернет за допомогою VANET та, відповідно, не відсутності додаткового програмного забезпечення.

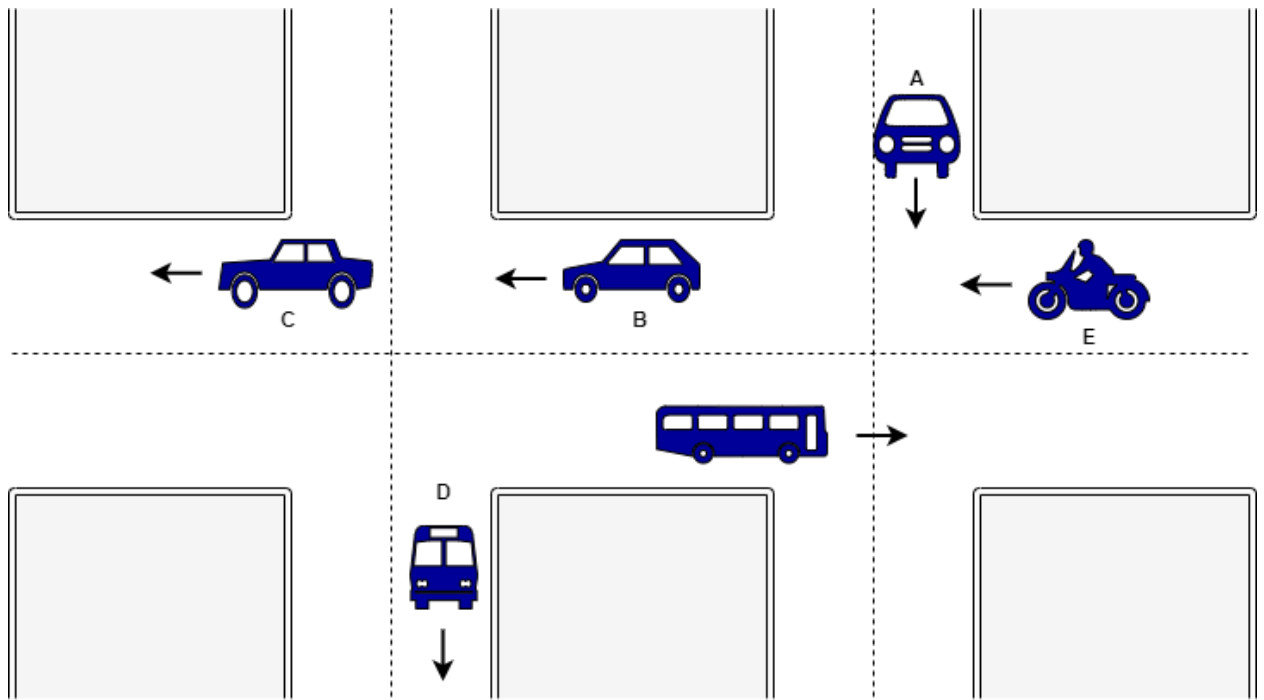


Рис. 1.2. Модель транспортної мережі без використання інфраструктурного рівня

На рисунку 1.2. зображено приклад транспортної мережі без інфраструктури. Згідно даної моделі видно, що при передачі повідомлення між двома далеко віддаленими вузлами використовуються проміжні транспортні засоби. Але така модель поведінки не є досконалою, оскільки для побудови надійного маршруту для передачі даних може просто не існувати. Подібна ситуація виникає для автомобілів E та C. Відстані від вихідного вузла E до проміжного вузла B надто велика, що може призвести до втрати пакетів даних.

Подібна ситуація може виникнути і для вузла D. Поряд із даним автомобілем може бути достатня кількість транспортних засобів, але беручи до уваги напрямок руху та, припустимо, високий показник швидкості руху, побудувати надійний шлях для обміну даними стане окремою проблемою для такої архітектури. Рішенням таких багаточисленних ситуацій є використання фіксованих бездротових маршрутизаторів. Вони розташовуються у певних місцях досліджуваного середовища та допомагають вирішувати проблеми маршрутизації мережі та проблемним підключенням вузлів у місцях з недостатньою кількістю транспортних засобів.

На рисунку 1.3. зображено мобільну мережу VANET з використанням інфраструктурного рівня.

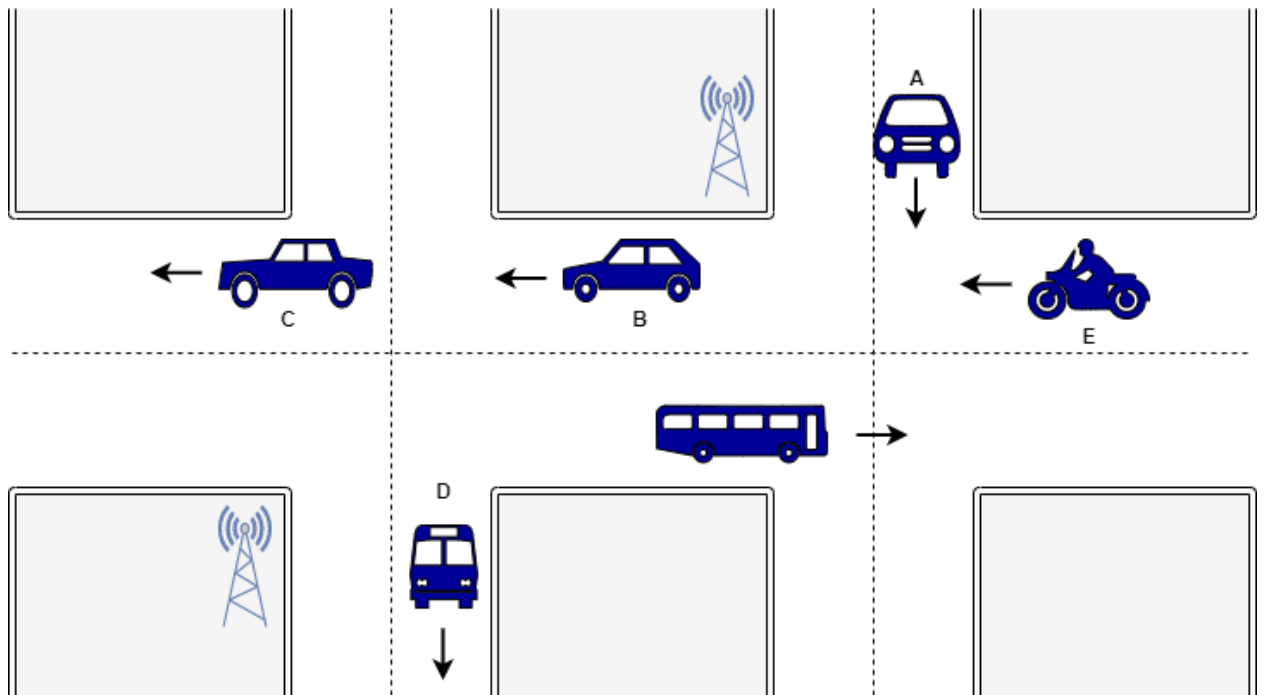


Рис. 1.3. Модель транспортної мережі з застосуванням інфраструктурного рівня

У такій інфраструктурній мережі, фіксовані бездротові маршрутизатори слугують як допоміжні проміжні засоби для маршрутизації повідомлення від вихідної до кінцевої точки.

Якщо повторно розглянути виключні ситуації попередньої архітектури, з вдосконаленням інфраструктурним рівнем, маршрут від точки Е до точки призначення буде містити додатковий вузол у вигляді RSU. Подібний підхід застосовується і для точки D. Ці фіксовані бездротові маршрутизатори можуть бути підключені або не підключені до дротової магістралі.

У разі використання дротової магістралі, при відправці повідомлення воно зберігається у VANET до тих пір, поки не воно не досягне першого фіксованого маршрутизатора мережі. Лише при наявності близько розташованого транспортного засобу в межах такої RSU, повідомлення буде передано до наступного вузла, або коли цільовий автомобіль не досягне маршрутизатору [12].

Таким чином, зв'язність інфраструктурної мережі VANET значно зменшується, адже доволі велика частина трафіку передається у відповідальність автомагістралі. Додатковим плюсом у використанні такого архітектурного рішення є встановлення проміжного програмного забезпечення та послуг для водіїв, наприклад, gps або надання доступу в інтернет.

Недоліком такого рішення є висока вартість його використання, оскільки воно вимагає встановлення великої кількості таких бездротових маршрутизаторів вздовж магістралей та доріг у різних місцевостях. Також, якщо рівень трафіку буде не високим, то повідомлення, що знаходиться на RSU буде передаватись від лише від одного автомобіля до іншого.

У [13] запропоновано альтернативний підхід шляхом поділу фізичної карти на кластери, де фіксовані бездротові маршрутизатори використовуються для міжкластерної комунікації вузлів. Таке рішення має значну перевагу над попередніми. Висока якість зв'язку забезпечується відсутністю на дротову інфраструктуру. Крім того, якщо фіксовані бездротові маршрутизатори розміщені на перехрестях, проблема перешкод у будівництві усувається, використовуючи їх як реле, коли потрібно перетнути дорогу не по прямій [14].

При застосування інфраструктурної архітектури VANET з магістральною мережею, або без неї, є цілком можливим використання зв'язку I2V, або ми можемо використовувати різні технології в кожному випадку. Зв'язок V2V з технологією 802.11 використовується більшістю авторів та проектів [15] [16]. Така статистика пов'язана з нижчою вартістю, та вищим діапазоном бездротової передачі в міському середовищі, де лінія видимості скорочується з кривими траєкторіями руху та будівлями. Як альтернативну технологію для зв'язку I2V можна використати 802.16. Недоліком є те, що обладнання буде значно дорожчим, а на самих транспортних вузлах буде інша технологія зв'язку. Приклад такого зв'язку наведено у [17]. Більш того, переваги 802.16 не матимуть високої вартості ми більшу частину часу роботи мережі, оскільки, як було наведено вище, лінія

видимості більшості вулиць не достатньо довга, тому радіусу 802.11 на 300 метрів достатньо. У зв'язку з цим я вважаю, що найкращим рішенням є використання 802.11 для зв'язку V2V і I2V.

1.1.4 Програмне забезпечення VANET

Використання спеціальних транспортних мереж відкриває широкий простір для застосування додаткового програмного забезпечення зорієнтованого на забезпечення безпеки руху, покращення умов водіння, попередження аварійних ситуацій та інформаційно розважального аспекту.

У опитувальному дослідженні [19] автори представили основні характеристики транспортних мереж. У роботі також наведено опис основних характеристик VANET та наведено огляд програмних застосунків, що можуть використовуватись разом із ним, їх вимоги, переваги та недоліки.

На рисунку 1.4 наведено таксономію додатків VANET.

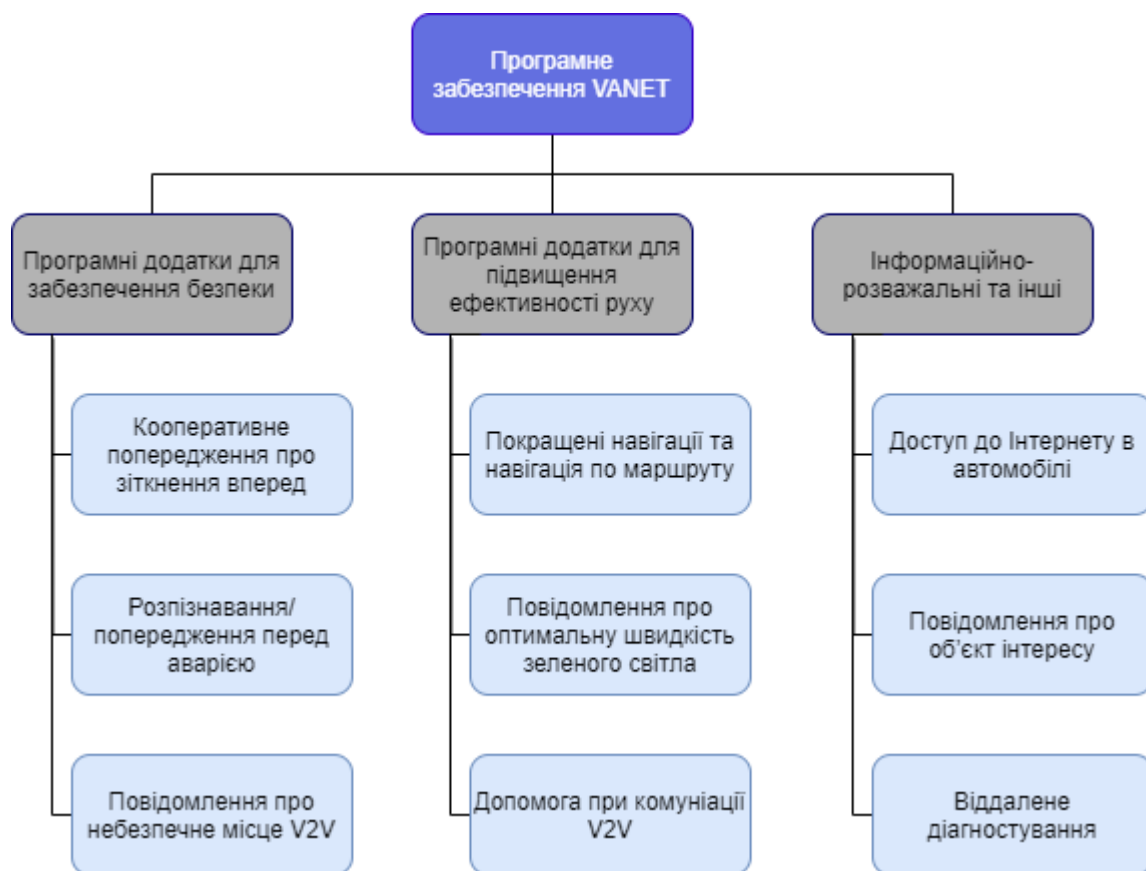


Рис. 1.4 Таксономія додатків VANET

Перша категорія програмного забезпечення спрямованна підвищення рівня безпеки водіїв на дорогах. Попередження володарів автомобілів про небезпечні ситуації, зіткнення, пошкодження на дорогах, складність водіння спричинена складними погодними умовами (обмерзання доріг, тумани, тощо) підпадають під цю категорію.

Програмні додатки по підвищенню ефективності дорожнього руху спрямовані на надання актуальної інформації водіям або операторам доріг про, безпосередньо, умови руху. Даний функціонал реалізовується за рахунок обмінну інформації через трафік VANET. Таким чином учасники дорожньо-транспортного руху отримують переваги по відношенню до часу дорожнього руху та зниженню витрат на будівництво та утримання доріг та магістралей.

До третьої категорії програмного забезпечення належать виключно додатки інтелектуально-розважального характеру. Вони використовуються у випадках, що не стосуються безпеки або ефективності руху. Певна їх частина використовується водієм, а інша для покращення функціональності окремого транспортного засобу. RSU виступають у ролі інтернет шлюзів та надають водіям та пасажиром вихід у Інтернет через VANET. Подібний концепт також використовується рекламними компаніями та трейдерами для поширення бізнес-реклами. Отримані рекламні оголошення будуть відфільтровані кожним транспортним засобом відповідно до профілю водія та контексту, після чого відповідні оголошення будуть представлені водієві.

1.1.5 Дослідницькі проблеми та відкриті питання

На даний момент існує безліч невирішених питань пов'язаних з розробкою ефективних та економічно вигідних протоколів та додатків для VANET. Наразі, однією з найактуальніших проблем є розповсюдження даних в межах інтелектуальної транспортної мережі. Задача цього функціоналу полягає у своєчасному інформуванні водіїв про події, пов'язані з дорожнім рухом, уникаючи проблеми переривання зв'язку, затримки, малий час

з'єднання, тощо. На рисунку 1.5 таксономію існуючих протоколів трансляції, що запропоновані у [20] та [21].

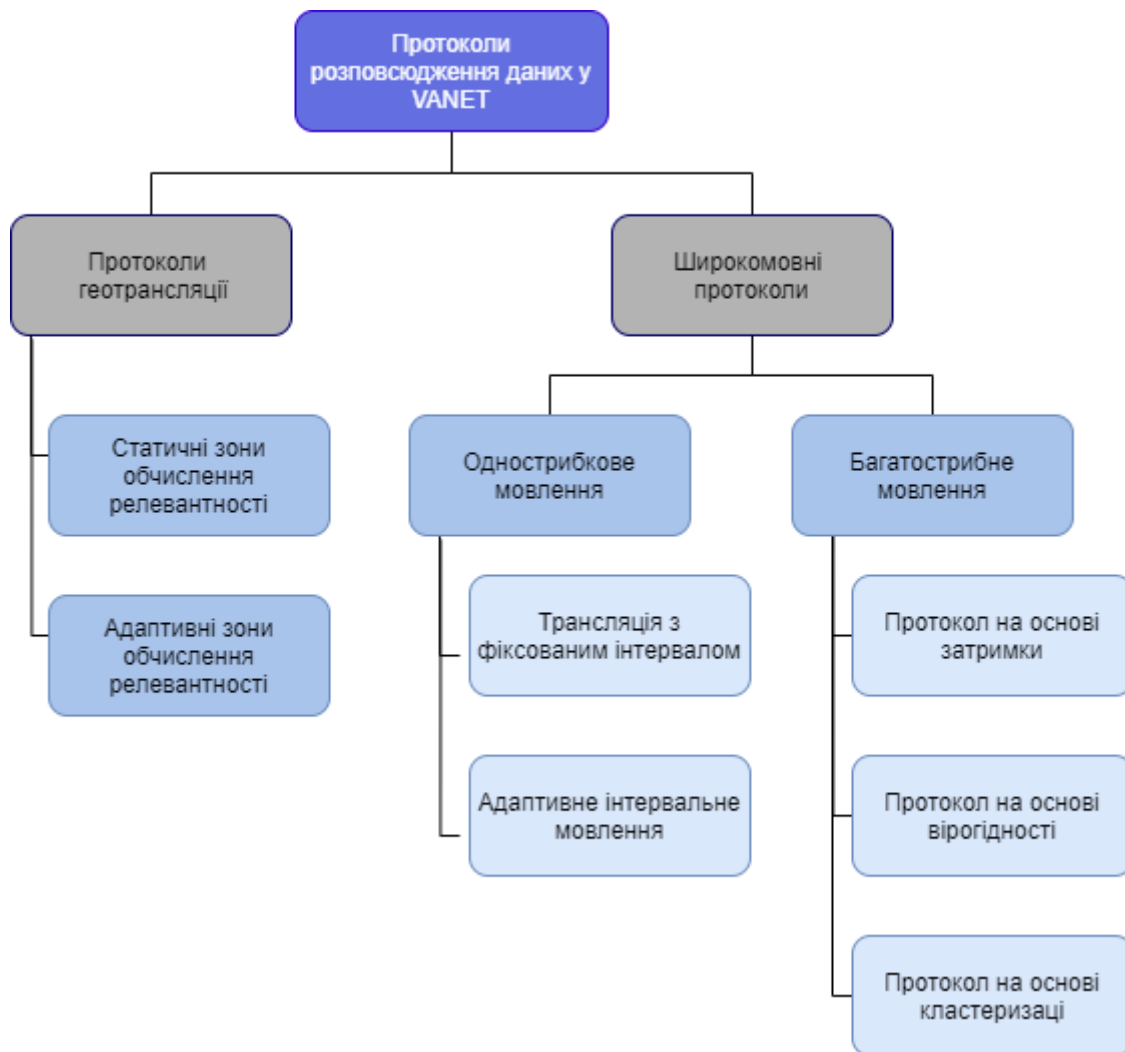


Рис. 1.5 Таксономія протоколів поширення даних в VANETs

Основою класифікації є широкомовні або геотрансляційні протоколи. Їх метою виступає поширення повідомлень про подію, що відбулась усім вузлам мережі без виключень. Такий принцип характерний для широкомовної трансляції, у той час як для вузлів, що рухаються у географічній зоні, яка називається зоною релевантності (ZOR), розповсюдження повідомлень спрямовано на цільовий набір транспортних вузлів [7].

Обчислення зони релевантності також є однією із головних проблем поширення інформації про події. Ефективну розсилку даних для підвищення рівня безпеки безпеки можна здійснити за допомогою підходу геотрансляції.

Такий варіант вимагає доставки даних транспортним засобам всередині ZOR. Відповідно, першим кроком реалізації протоколу є саме обчислення ZOR для описуваної події.

Ще однією проблемою широкомовного розповсюдження даних є управління часом життя події. У мережах VANET дані для розповсюдження мають зберігатися в ZOR з певною затримкою, щоб відправити повідомлення якомога більшій кількості транспортних засобів, що прибувають. Така характерна риса мережі вимагає впровадження нових механізмів для підтримки відповідних подій всередині ZOR, вилучаючи необхідність проведення повторної трансляції повідомлень про події, як це наразі працює у існуючих підходах.

Агрегація даних у VANET наразі, однією із найбільш обширних та важливих проблем у функціонуванні мобільних мереж. Оскільки VANET, за визначенням, є дуже динамічною мережею через високий рівень мобільності вузлів та неоднорідність руху, інформація має обмінюватися між транспортними засобами згідно протоколу, що забезпечує ефективність та надійність передачі даних. У цьому відношенні агрегація даних виглядає цікавим підходом, який дозволяє об'єднати кілька наборів даних про відносно схожі події. У результаті такої методики буде створено підсумок (або агрегат), що буде ключем до зменшення мережевого трафіку. Отже, розробка результативного підходу до агрегації даних, що поєднує в собі корельовану інформацію про трафік, є складною, досі не вирішеною проблемою [22].

1.2 Система VANET на основі SDN

При виборі протоколів для застосування у транспортній мережі VANET, слід звертати увагу на її властивості по забезпеченню безпеки, ефективності та надійності системи. Для вдосконалення існуючих рішень спеціальних мереж було введено у використання технологію програмно визначених мереж (SDN), що, з часом, отримала назву SDVN.

Такий підхід дозволяє вирішити ряд проблем, що виникають у неінфраструктурних транспортних мережах. Керуючи загальною мережею за допомогою одного пульта дистанційного керування можна суттєво зменшити навантаження на мережу. У цьому розділі дослідження, надано характеристику мережі VANET на основі SDN, її роботу, програмні додатки, послуги, проблеми та недоліки, загрози безпеці та переваги порівняно з попередніми методами.

Основними компонентами цієї системи є контролери SDN, бездротовий вузол SDN і SDN RSU, які показані на малюнку 1.6.

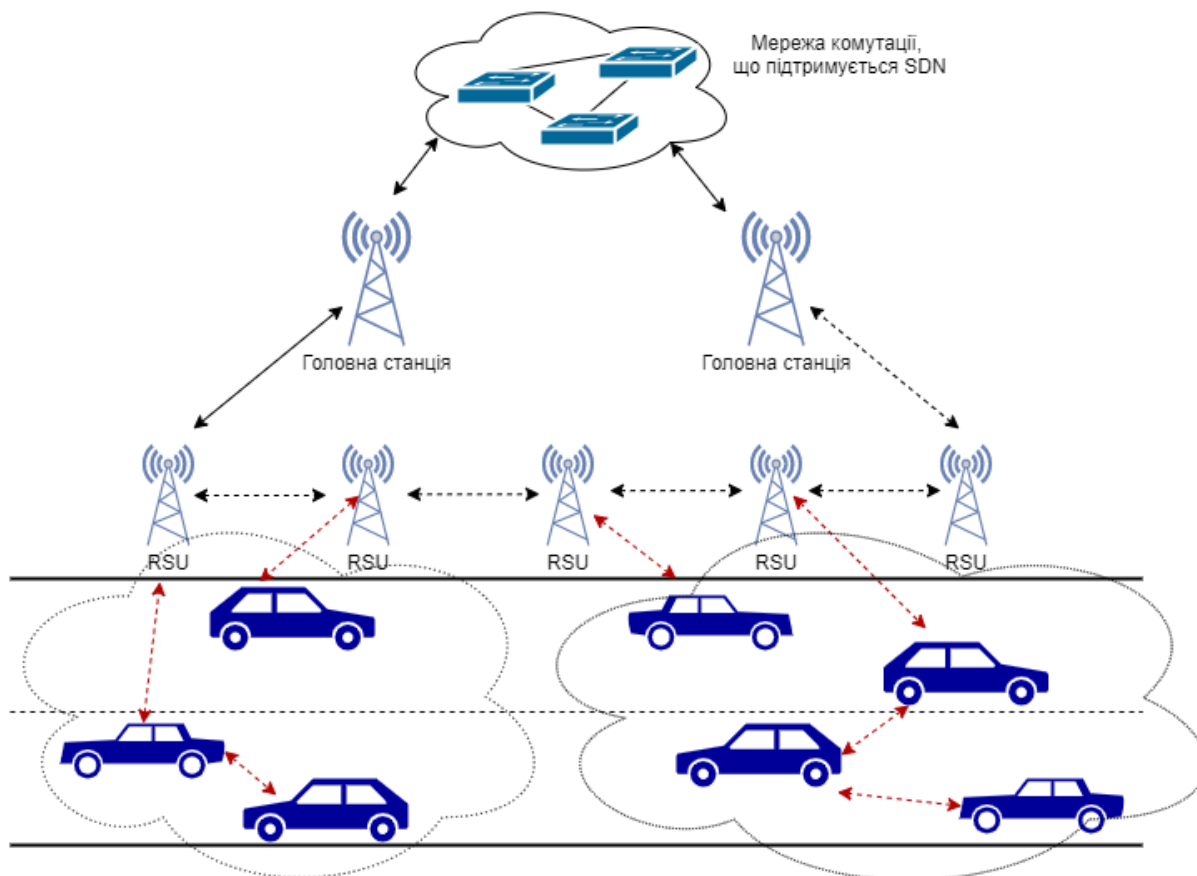


Рис. 1.6. Архітектура технології SDVN

Мережа комунікації являє собою контролер SDN. Його головне завдання - управління мережевою поведінкою всієї системи. Як правило, він підключається до RSU. Транспортні засоби у мережі SDVN також відомі як вузли системи, оснащені OBU та AU та аналогічно приймають участь у

спілкуванні. AU можна використовувати як персональний пристрій або персональний цифровий помічник, що спілкується через мережу за допомогою пристрою OBU і відповідає за всі види зв'язку. Пристрій RSU має фіксовану позицію (на узбіччі дороги або на паркувальній зоні) та підключається до мережі для забезпечення зв'язку між транспортними засобами та з контролером SDN. Такий бездротовий маршрутизатор використовується для надання послуг, необхідних для розміщення програми, а OBU використовує ці послуги. Додатковим бонусом є те, що RSU має можливість підключатися до Інтернету, що дозволяє AU в кількох автомобілях підключатися до Інтернету.

1.2.1 Програмно визначена архітектура VANET

Обмін даними між транспортними засобами та між RSU проходить через бездротовий простір. Така концепція спрямована на надання водіям та пасажиром актуальної інформації про умови дорожнього руху та можливі надзвичайні ситуації.

Особливістю системи VANET на основі SDN є те, що зв'язок здійснюється між площиною даних і площиною керування. Площина даних містить пристрої пересилання трафіку, тоді як площина управління керує потоком зв'язку. Площина керування, у свою чергу, має різні рівні, що також працюють у системі постійної комунікації, щоб забезпечити функціональність контролера в мережі. На рисунку 1.7 схематично зображено архітектуру такої системи.

У базовій архітектурі технології SDN можна виділити три основні рівні, такі як: прикладний рівень, керуючий рівень та інфраструктурний рівень. Її графічне відображення ви можете побачити на рисунку 1.4.

Інфраструктурний рівень містить пристрої пересилання та обробки даних, такі як RSU. Пристрої, підключені до RSU або транспортного засобу, використовуються для виконання різних дій на основі політик, визначених для кожного пристрою контролером. Зв'язок між контролером SDN та пристроями здійснюється через стандартний протокол OpenFlow [23], який

використовується для зв'язку між контролером і RSU або вузлами, які безпосередньо підключені до контролера.

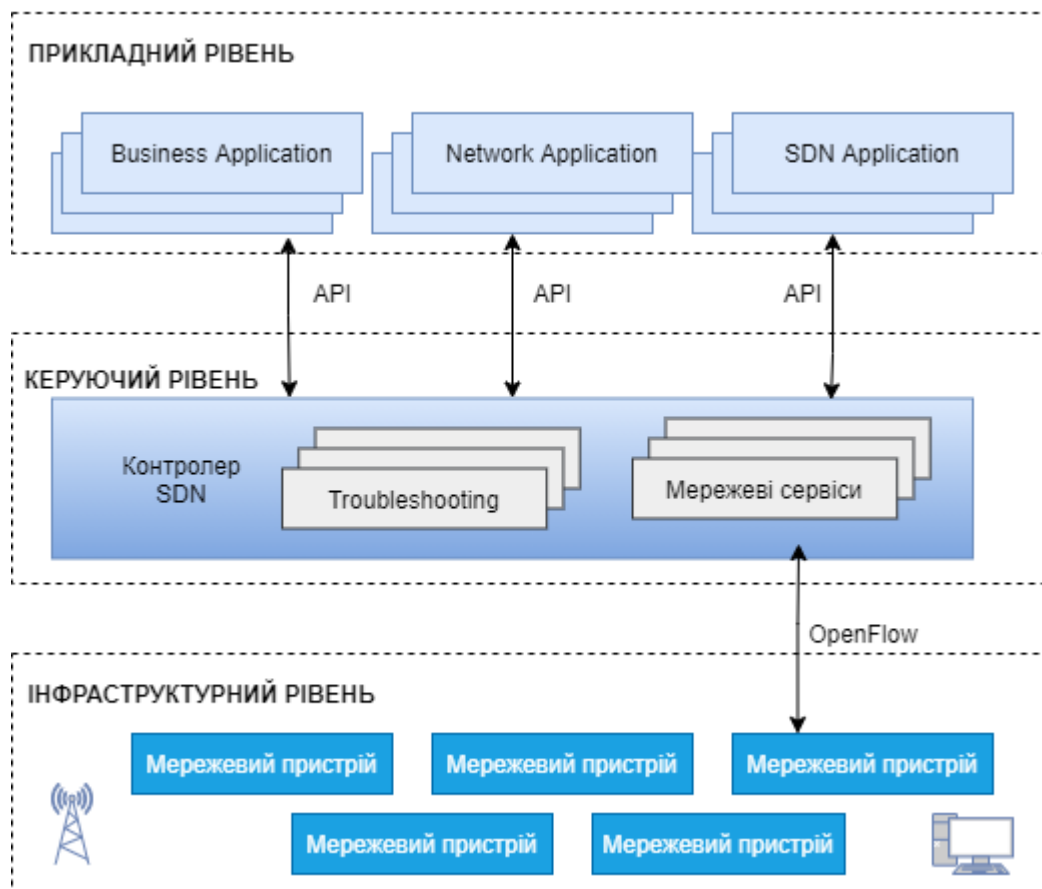


Рис. 1.7. Архітектура програмно-визначених мереж

Протокол OpenFlow, має свою власну структуру і розділяється на чотири поля. Перше з них є полем пріоритету, яке використовується для визначення порядку, у якому дані відповідають визначеним правилам. Правила призначають рівень пріоритету. Дані, які відповідають правилу вищого пріоритету, обробляються першими. Друге поле - це умова відповідності, яка визначається відповідно до точки зору IP. Коли дані отримуються з іншого IP-вузла, це поле відповідає умовам відповідно до IP-адреси, яка зазвичай є IP-адресою джерела та IP-адреси призначення. Третє поле — це поле дії, яке використовується для визначення дії, яку слід виконати при отриманні пакету даних. Четверте поле — лічильник, який використовується для підрахунку кількості пакетів даних, отриманих щодо конкретного завдання. Це допомагає системі перевірити, чи повністю

отримані дані чи ні. З часом було випущено багато версій системи OpenFlow, які відповідають новим умовам пересилання даних.

У VANET на основі SDN, на відміну від традиційної комп'ютерної системи, де ОС несе відповідальність за управління всіма ресурсами системи, цю роль виконує контролер SDN. Основною структурою керуючого рівня системи є мережевий контролер. Його умовно можна назвати «мозком мережі». Головний функціонал такого контролеру полягає у встановленні, оновленні та видаленні правил на кожному пристрої, що є безпосередньо підключеним до мережевого контролера. При цьому, контролер SDN не використовує стандартного правила по встановленню послуг, вони визначаються як певні засоби, що надаються контролером SDN. Відповідно, ця властивість контролеру SDN виявляється важкою в реалізації та у впровадженні нових політик і методів для підвищення ефективності роботи системи [24].

Окрім складності у реалізації, застосування централізованої системи також може супроводжуватись такими проблемами, як низька надійність системи, загроза безпеці мережі, нестабільна робота системи.

Як нам відомо, однією з головних особливостей мережі VANET є висока мобільність транспортних вузлів. Відповідно, на кожен транспортний встановлено власну політику зв'язку з отриманими даними. У разі ініціювання передачі нового потоку даних вузлам або транспортним засобам, для зниження кількості трафіку, до роботи системи долучається RSU або інший пристрій, що підключається безпосередньо до контролера SDN. Найбільш вигідно розмістити контролер поблизу зони з щільним рухом, таким чином створюється додаткова можливість уникнути затримок або очікування відповіді від проміжного вузла, а також полегшити контакт кожного вузла з контролером. Крім того, використання фіксованих бездротових маршрутизаторів у місцях зі щільним трафіком дозволяє зменшити загальну вартість зв'язку.

На прикладному рівні знаходяться мережеві програми розроблені з використанням мов програмування на основі SDN та пов'язаних з ними компіляторів. Як правило, такого роду програми використовуються для регулювання функціональності контролера. Програми для SDN розробляються безпосередньо в API контролера, але цей рівень полегшує переносимість програми, можливість повторного використання коду та високорівневу мережеву абстракцію. Тому розробник повинен розробляти програму з дотриманням політики високого рівня. Компілятор автоматично згенерує правила, пов'язані з OpenFlow, які потім встановлюються на кожному пристрої, підключеному безпосередньо до контролера.

1.2.2 Проблеми, пов'язані з Cross layer

Перехресний рівень площини керування використовується для вирішення проблем, пов'язаних з цією площиною, таких як тестування, налагодження програмного забезпечення SDN та вирішення проблемних ситуацій. Для тестування функціональності системи OpenFlow доступно кілька інструментів. Наприклад, система Mininet [25] використовується для створення прототипів VANET на основі SDN за допомогою однієї комп'ютерної системи. Надана можливість віртуалізації системи спрощує виявлення проблем у системі. fs-sdn також використовується для виконання моделювання з використанням потоку let замість пакета [26]. Порівняно з Mininet він може моделювати велику мережу з більш точними результатами.

Для усунення несправностей у мережі використовуються ping або traceroute для виявлення несправностей у мережі. Однак іноді дуже важко визначити проблему, якщо вона згенерована параметрами конфігурації. Програма налагодження програмного забезпечення NICE використовується для налагодження застосування системи SDN, яка допомагає у вирішенні помилок, пов'язаних з налагодженням.

1.3 Огляд кластеризації у VANET

У даній роботі багато уваги присвячено проблемам пов'язаними з мобільністю мережі. Оскільки відносна швидкість транспортних засобів відносно один одного безперервно змінюється, навіть незважаючи на обмеження дорожнього руху, що домомагають його регулювати, проблема надійності та масштабованості протоколу маршрутизації у великих середовищах у VANET досі є дуже актуальним. Одним з перспективних методів керування топологією VANET, що дозволяє зменшити кількість витрат на маршрутизацію, є розділення мережі на логічні групи, які називаються кластерами.

Такий процес об'єднання сусідніх вузлів у визначені групи називається кластеризацією в домені VANET. Сам кластер має у своєму складі один вузол, як голову кластеру СН, а решта визначається як члени кластеру СМ. З метою зменшення частоти змін у глобальній топології мережі, кластери формуються шляхом групування транспортних засобів, що рухаються, на основі відносної швидкості та співвіднесеного просторового розподілу.

Найперші дослідження кластеризації розпочалися з пакетної радіомережі DARPA [27], а пізніше були реалізовані в MANETs [28], які зосередилися на стабільності кластера. Відтоді в літературі було запропоновано багато методів кластеризації, які задовольнили вимоги широкого кола застосувань.

1.3.1 Базова концепція кластеризації VANET

Формування кластеру мережі VANET базується на показниках відносної швидкості та відстані між транспортними засобами. У більш глобальному розумінні така система, об'єднана в кластери є менш динамічною, незважаючи на безперервно рухомі вузли. У дослідженні [29] було продемонстровано, що застосування способів кластеризації є валідним рішенням проблеми надійності та масштабованості VANET. Таке твердження зумовлено меншою

кількістю комунікацій, викликаних обслуговуванням і виявленням маршрутів, порівняно з іншими методами динамічної маршрутизації мережі.

Кластеризація значно зменшує витрати контрольних пакетів завдяки ефективному заповненню. Наприклад, у мережі без застосування кластеризації кожен вузол пересилає повідомлення шляхом багатоадресної передачі його своїм сусідам. Це також призводить до того, що кожен вузол повторно отримує повідомлення від кожного зі своїх сусідів. З транспортними засобами, організованими в кластери, тільки голова кластера багатоадресно передає повідомлення своїм сусідам по кластеру. Такий спосіб передачі даних також називається вибіркоким пересиланням. Члени кластеру не пересилають повідомлення, якщо воно не є частиною шлюзу. Вузли шлюзу створюють шляхи комунікації між головками кластеру. Схематичне зображення такого спілкування зображено на рисунку 1.8.

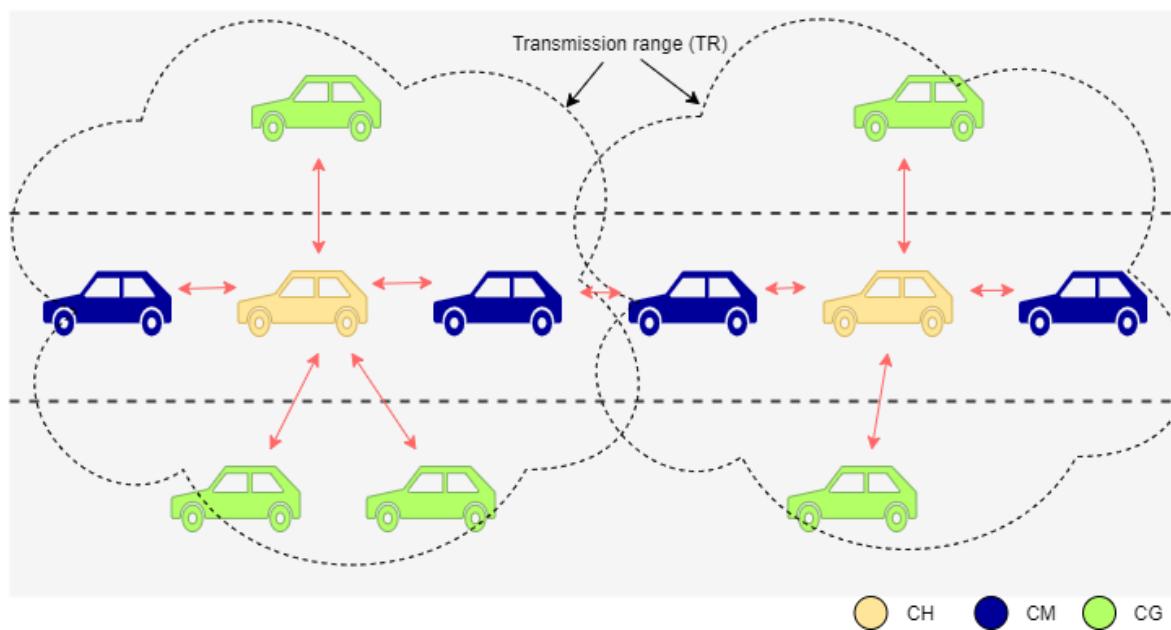


Рис. 1.8 Комунікація в мережі VANET із застосуванням кластеризації вузлів.

У кластеризованій транспортній системі головка кластера (CH) відіграє ключову роль у процесі формування кластеру. Спосіб формування кластеру безпосередньо залежить від вхідних метрик. Також відомі алгоритми кластеризації з використанням двох або більше CH [30]. Для міжкластерного зв'язку від імені CH, використовуються вузли, що називаються кластерними

шлюзами (CGs). Якщо жоден з вузлів не визначається як CG, всі члени кластера називаються членами кластеру (CM). У кластері можуть бути присутніми один або декілька СН, декілька CG (або навіть жодного) та будь-яка кількість CM, залежно від вхідних обмежень алгоритму кластеризації.

У кластеризації VANET СН грає роль мобільного маршрутизатора, у той час як CM працює як штатний мобільний вузол. Кластер формується на основі таких показників, як середня відносна швидкість транспортного засобу, напрямок руху, прискорення, географічне положення вузлів відносно один одного, ступінь руху транспортного засобу, щільність розташування транспортних засобів та інші метрики, що можуть характеризувати дорожній рух у визначеній місцевості.

Процес вибору СН базується на аналізі локальних характеристик транспортних засобів, що порівняно із іншими членами кластеру, мають найбільш стабільні показники. Решта транспортних засобів приєднуються до кластеру як CM. Таким чином, вибір СН є частиною процесу формування кластеру, а вибір CM базується на критеріях відбору відносно СН.

Голова та члени кластеру підтримують власну таблицю маршрутизації, що містить інформацію про СН і CM кластеру для внутрішньокластерного зв'язку. Відповідно CM не підтримує таблицю маршрутизації для сусідніх кластерів мережі. СН може володіти такою функціональністю, якщо це обумовлено алгоритмом. Отже, велика мережа розглядається як група деяких малих мереж або кластерів.

Оскільки покриття кластера обмежено діапазоном передачі (TR) каналу, як показано на рис 1.8, транспортний засіб, розташований на межі кластеру, має високу ймовірність втрати зв'язку з головою кластеру. Відносна швидкість двох транспортних засобів може постійно змінюватися в залежності від швидкості двох транспортних засобів, і, відповідно, коли позиція транспортного засобу знаходиться на межі, транспортний засіб може часто в'їжджати та знову повертатись у діапазон каналу. В результаті вірогідність втрати даних буде стабільно високою, коли транспортний засіб залишається

на краю кластера. З цієї причини деякі алгоритми віддають перевагу транспортному засобу з географічним центром як СН для надійності.

1.3.2 Таксономія кластеризації транспортної мережі

Серед алгоритмів кластеризації також відомі одно- та кількострибкові алгоритми. Для однострибкових алгоритмів можна розділити на дві основні стратегії: стратегії на основі інтелекту та стратегії на основі мобільності. Отже, схеми кластеризації VANET можна розділити на три основні алгоритми: алгоритми на основі інтелекту, алгоритми на основі мобільності та алгоритми на основі кількох стрибків [31], як зображено на рисунку 1.9.

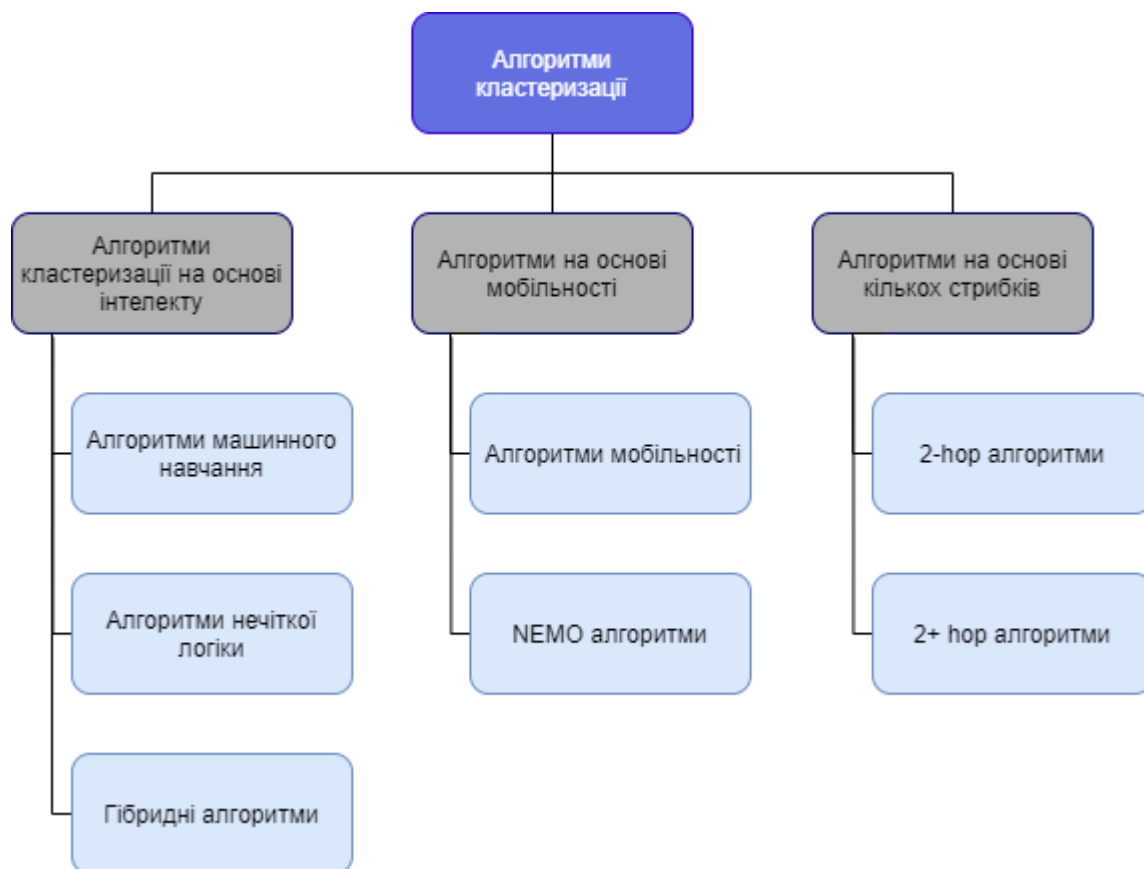


Рис. 1.9 Таксономія схем кластеризації VANET.

Алгоритми кластеризації на основі інтелекту надалі можуть класифікуватись як алгоритми машинного навчання, нечіткої логіки та гібридні алгоритми, що поєднують у собі декілька стратегій. Алгоритми, засновані на мобільності, поділяються на алгоритми мобільності

транспортних засобів і мережевої мобільності, тоді як стратегії на основі кількох стрибків поділяються на алгоритми з 2-х і 2+ стрибками на основі кількості стрибків.

1.3.3 Конкуруючі рішення по кластеризації вузлів

Маршрутизація — це інтелектуальний процес прийняття рішень, що визначає найоптимальніший маршрут передачі даних від вихідного до результуючого вузла. Маршрутизація в VANET стикається з різноманітними проблемами, пов'язаними із функціональністю мережі. Наприклад, забезпечення якості обслуговування кількох служб керування мережею та керування дорожнім рухом. Висока мобільність автомобілів сприяє швидкій зміні топології та відхиленню показників швидкості та щільності розташування транспортних засобів у межах доріг. Відсутність щільного розподілення транспортних засобів у географічних регіонах погіршує роботу системи та спричиняє нестабільність показників пропускної здатності, з більшою затримкою передачі та збільшенням накладних витрат маршрутизації. Додатковим ускладненням є те, що зона VANET включає будівлі, колії та дороги, по яких транспортні засоби рухаються в певному напрямку.

Наразі, багато авторів приділяють увагу питанню оцінки протоколів маршрутизації з QoS, що класифікуються на основі протоколу топологічної маршрутизації, стратегій передачі та протоколів маршрутизації, орієнтованої на позицію.

AODV - це реактивний протокол маршрутизації, метою якого завжди є пошук шляху між джерелом і стоком. Він виконує процедуру визначення шляху з повідомленнями RREQ, RREP і RERR, що обмінюються між вузлом джерелом і результуючим вузлом. AODV забезпечує одноадресну, багатоадресну і широкомовну комунікацію. Такий підхід дозволяє зменшити маршрутизацію та накладні витрати. Тим не менш, застосування такого рішення призводить до більшої величини затримки [32]. Розширенням протоколу маршрутизації AODV є AOMDV, який також надає можливість

багатопрореневого зв'язку, а також використовується для обробки шляхів без циклів і неперетинання. У результаті ми отримуємо менше навантаження на маршрутизацію під час пошуку маршруту [33].

OLSR - протокол активної маршрутизації, відомий як чистий стан зв'язку. Використання повідомлень Topology Control (TC) і Hello визначає та поширює інформацію про стан зв'язку по всій мережі. Це дозволяє уникнути надмірної передачі пакетів стану зв'язку (LSP). Головною характеристикою OLSR є процес вибору багатоточкового реле (MPR). Вузол, мережі з однокроковою передачею інформації обирає MPR і повторно надсилає всі ширококомвні повідомлення, отримані цим вузлом [34] [35].

ВИСНОВКИ ДО РОЗДІЛУ 1

Беручи до уваги характеристику функціонування транспортної мережі, зі збільшенням кількості транспортних засобів, масштабованість мережі VANET стає дедалі важливою проблемою. За відсутності будь-якої центральної інфраструктури VANET зазнає великих втрат пакетів через великий обсяг поширення повідомлень між засобами зв'язку V2V. VANET також страждає від таких проблем, як проблема прихованого терміналу, висока затримка передачі повідомлень безпеки, безпека повідомлень, проблема ширококомовного шторму, якість обслуговування (QoS), маршрутизація пакетів, контроль перевантажень та управління ресурсами. Для вирішення цих питань у літературі досліджено ієрархічну структуру. В ієрархічній структурі два або більше сусідніх транспортних засобів, які мають деякі спільні ознаки, об'єднуються в групу, яка називається кластеризацією.

Концепція кластеризації широко використовується в аналізі даних і машинному навчанні. Вона також використовується в MANET, який є попередником VANET, для кластерування мобільних вузлів. У кластерному середовищі транспортних засобів велика мережа транспортних засобів розглядається як мережа деяких невеликих мереж або кластерів.

Тим не менш, застосування алгоритмів кластеризації в спеціальних мережах не ідеальним рішенням, оскільки не існує однозначно найкращого критерію якості кластеризації. Кількість кластерів, як правило, невідомо заздалегідь, вона встановлюється відповідно до певного суб'єктивного критерію. Результат кластеризації істотно залежить від метрики, вибір якої, зазвичай, також суб'єктивний і визначається експертом.

Оскільки оптимальне рішення по формуванню кластеру залежить від мережевого середовища, будь-яка математична модель задачі вибору головки кластера складна для отримання, і рішення, засноване на ній, було б негнучким. Таким чином, основним завданням даного дослідження є проектування та реалізація гібридного алгоритму кластеризації для VANET, що ґрунтується на нечіткій логіці. Прогнозується, що такий підхід сприятиме

зниженню енергоспоживання всієї мережі та збільшенню терміну служби мережі. Використання метрик, пов'язаних із зв'язністю вузлів та RSU, показниками якості передачі повідомлення між вузлами та характеристиками наскільки добре маршрутизатор може прослуховувати та відфільтровувати перешкоди та шум, можна уникнути виникнення гарячої зони та збалансувати споживання вузлів у кластерах.

РОЗДІЛ 2

СПОСІБ КЛАСТЕРИЗАЦІЇ В МОБІЛЬНИХ SDN МЕРЕЖАХ VANET НА ОСНОВІ АЛГОРИТМІВ НЕЧІТКОЇ ЛОГІКИ

2.1 Постановка задачі та опис архітектури системи

Об'єктом даного дослідження є інтелектуальні транспортні мережі VANET, де набір транспортних засобів рухається по дорозі та використовує програми, обумовлені протоколом маршрутизації. Будемо вважати, що використовуватимуться саме програми які включають велику пропускну здатність, сховище та обчислювальні ресурси. Такими властивостями володіють більшість нових програм. Крім того, ми припускаємо, що всі транспортні засоби обладнані системою позиціонування (наприклад, GPS), за допомогою якої він може отримувати інформацію про своє поточне місцезнаходження та напрямок, а також радіоприймачем, сумісним із VANET, за допомогою якого він може спілкуватися з іншими транспортними засобами.

Окрім множини транспортних засобів, що виступатимуть вузлами мережі, існує набір RSU, які розташовані у фіксованих, заздалегідь визначених точках, таких як перехрестя, світлофори, затори, точки виїзду з міста, тощо. Усі RSU підключені до мережі комутації, що підтримується SDN і надають деякі послуги для транспортних засобів. Слід також враховувати, що кожен RSU доступний лише для тих вузлів системи, які знаходяться в межах його діапазону передачі даних (TR).

Оскільки метод кластеризації є хорошим підходом для вирішення проблем високої мобільності мережі та частих змін топології [36], ми формуємо транспортну хмару на основі структури кластера як один із хмарних сервісів. Варто враховувати, що через високу мобільність транспортних засобів реалізація транспортної хмари (VC) сильно відрізняється від хмари в традиційній комп'ютерній мережі. Атрибут мобільності VC означає, що довгострокове об'єднання ресурсів цих транспортних засобів майже

неможливе. Ця функція не існує в звичайних хмарах і виявляється важливою визначальною характеристикою VC.

Динамічні кластери утворюються шляхом об'єднання транспортних засобів, а ті вузли, як відрізняються високими показниками стабільності, стають головою кластера (CH). Динамічні кластери самі по собі мобільні, рухаються разом із швидкісними транспортними засобами, а транспортні засоби комунікують з RSU через вузол CH. Таким чином, навіть для високошвидкісних транспортних засобів архітектура рухомого кластера залишається зі стабільною топологією, до тих пір, поки швидкість транспортних засобів залишається більш-менш рівномірною в межах того самого кластеру. На рисунку 2.1 показано схематичне зображення описаної архітектури.

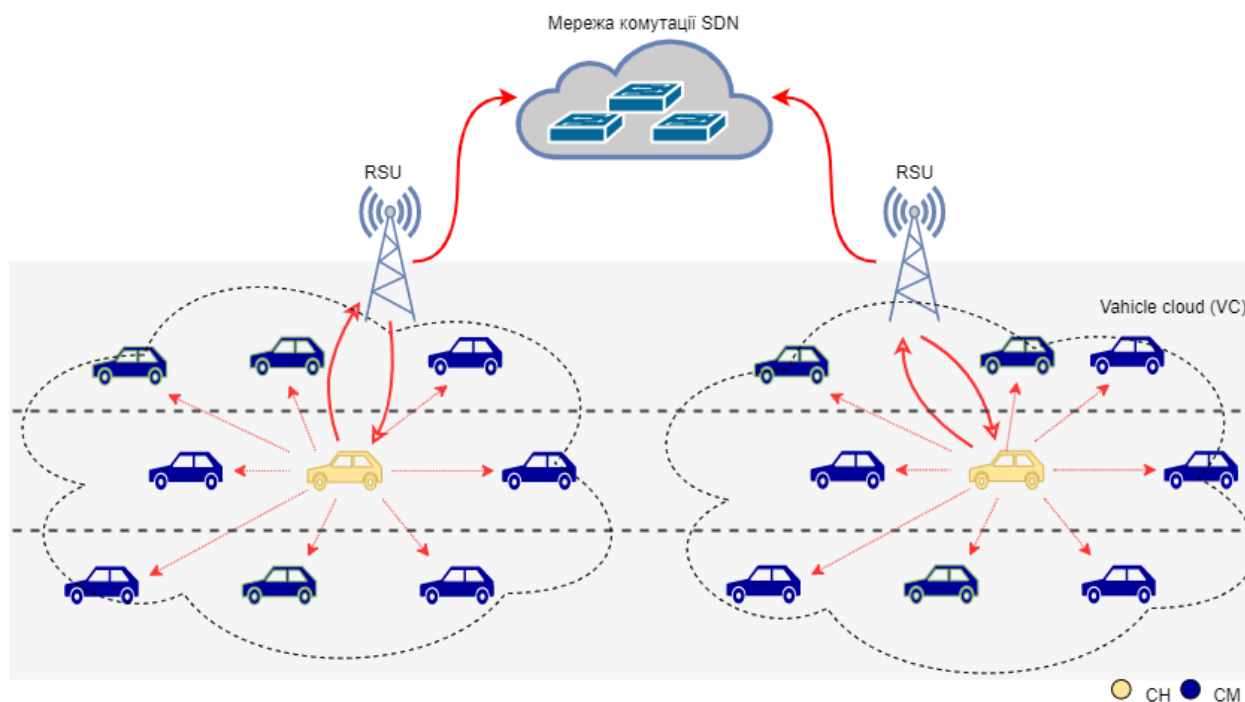


Рис. 2.1 Архітектура кластеризованої транспортної SDN мережі

За такої архітектури, кожен кластер можна розглядати, як окрему транспортну хмару, де CH фактично є контролером хмари, який відповідає за створення, обслуговування та видалення транспортної хмари. Усі транспортні засоби або CM віртуалізують свої фізичні ресурси та реєструють віртуальні

ресурси в голові кластеру, де він планує всі віртуальні ресурси транспортної хмари.

Якщо транспортному засобу потрібні деякі ресурси транспортної хмари, вони повинні надіслати запит до голови кластеру. Транспортна хмара на базі кластеру (VC) схожа на звичайну стратегію розгортання хмари, в якій хмарні ресурси плануються контролером. З точки зору використання ресурсів, голова кластеру може глобально планувати та розподіляти всі ресурси транспортної хмари. Як результат, нам потрібна схема кластеризації та метод вибору СН.

2.1.1 Формування кластеру та вибір СН

Описана вище схема кластеризації об'єднує транспортні засоби в кластери відповідно до їх розташування, напрямку та швидкості руху. У запропонованій архітектурі не використовуються CG, а міжкластерна комунікація проходить за участі RSU. Після формування кластеру та вибору СН, голова кластеру передає свої поточні дані про швидкість та місцезнаходження, вбудовані в повідомлення HELLO, іншим транспортним засобам у межах їхнього діапазону зв'язку та найближчому фіксованому маршрутизатору. Учасниками кластеру є транспортні засоби, що рухаються в одному напрямку та поблизу один від одного.

У цьому дослідженні пропонується гібридний спосіб кластеризації мережі, що використовує за основу методи машинного навчання. Умовно, процес кластеризації можна розділити на кілька етапів: визначення оптимальної кількості кластерів для варіанту існуючої мережі, визначення найбільш вигідного положення голови кластеру, проведення голосування всередині сформованого кластеру про вибір СН; надсилання повідомлення про HELLO до найближчої RSU.

Автори існуючих досліджень у даній області часто наголошують на проблемі кластеризації, пов'язаній з відсутністю чітких даних про кількість кластерів усередині транспортної мережі. У своїй роботі я пропоную додати до процесу кластеризації проміжний етап по визначенню кількості кластерів для фіксованого зрізу динамічної мережі. Базуючись на щільності

розташування вузлів є цілком можливим застосування алгоритму формування кластеру без вибору СН. Виділяючи основні групи транспортних засобів, опираючись на географічне положення вузлів, та заданий діапазон передачі повідомлень, можна визначити можливу кількість кластерів рівномірно розподілених по мережі.

Наступним етапом є вибір голови кластеру на основі fuzzy logic алгоритмів. Особливістю застосування такого принципу є відсутність жорсткого підходу по вибору найкращого кандидата. У процесі роботи алгоритму, для кожної новоствореної групи транспортних засобів буде визначено уявний СН, із найбільш вигідними метриками положення, швидкості та надійності зв'язку. Інформація про вибір голови кластера для будь-якого вузла обмежена вузлами, які знаходяться в межах його діапазону зв'язку. Пріоритет вузла стати СН визначається його коефіцієнтом підгонки відносно уявного СН.

Отже, кожен вузол, щоб стати СН, обчислює свій коефіцієнт підгонки та передає його. Потім усі вузли голосують за свого сусіда, який має локальне максимальне значення коефіцієнта підгонки. Вузол також може голосувати за себе, якщо він має максимальне значення. Після завершення процедури виборів обраний вузол підтверджує обрання головою кластера, змінюючи свій стан на СН і надсилаючи повідомлення Ask. Згодом сусідні транспортні засоби змінюють свій ідентифікатор кластера на ID нового СН, і таким чином формується динамічний кластер.

2.1.2 Ініціювання вибору нового СН та передача керування

Цілком можливим є виникнення ситуацій, коли транспортний засіб, визначений, як голова кластеру, більше не може виконувати свої функції. Наприклад, коли транспортний засіб може бути змушений зупинитись або уповільнити швидкість через деякі випадки, які трапляються на дорозі, наприклад, затор через аварію або затор на дорозі. Покидаючи свій кластер, СН надсилає повідомлення про вихід із кластеру RSU, і той, у свою чергу, бере на себе керівну роль у кластері замість реконфігурації кластеру. Відповідно,

після проведення такої процедури, статус основного СН буде змінено на СМ. Ця зміна не впливає на структуру кластера, лише роль СН буде передана на RSU.

Після цього, за заданою архітектурою, RSU має ініціювати вибори нової голови кластеру. Зауважте, що транспортний засіб, який збирається зупинитися, досягнувши місця призначення, або хоче виїхати, повинен сповільнитися. Тому ми вважаємо швидкість одним з основних критеріїв для визначення того, чи підходить досліджуваний транспортний вузол на позицію ведення лідерства. Крім того, СН повинен періодично підтримувати інформацію про статус свого програмного налаштування та метрик, щоб гарантувати, що він бере на себе відповідальність, коли виникає необхідність.

Отже, після вибору нового СН, RSU має оновити інформацію про стан кластеру та поширити цю інформацію серед всіх інших кластерів.

2.1.3 Злиття кластеру та додавання нових вузлів

У транспортних мережах VANET досить часто виникають ситуації, коли транспортний засіб не має прямої приналежності до якогось із вже визначених кластерів. Два найпоширеніших варіанти можна спостерігати на рисунку 2.2.

Розглянемо ситуацію для транспортного засобу А. Припустимо, що вузол А у певний період часу набрав швидкість та вийшов за межі другого кластеру. Наразі можливі два варіанти: автомобіль А продовжуватиме набирати швидкість, і з часом зможе бути частиною першого кластеру, або він буде рухатись зі сталою швидкістю. Прийняття рішення по приєднанню вузла до існуючого кластеру займається RSU. Згідно свого внутрішнього протоколу, бездротовий маршрутизатор розсилає інформацію про А, до всіх СН транспортної мережі та очікує на результат аналізу з боку кластеру. Голова кластеру аналізує відділений вузол А та перевіряє, чи задовольняє розглядаємий вузол всім вимогам його кластеру, наприклад, показникам відносної швидкості кластеру, приналежність до TR, тощо. Отримавши

відгуки від всіх СН мережі, RSU повторно надсилає повідомлення до СН, що найбільш оптимально може бути застосований кластером.

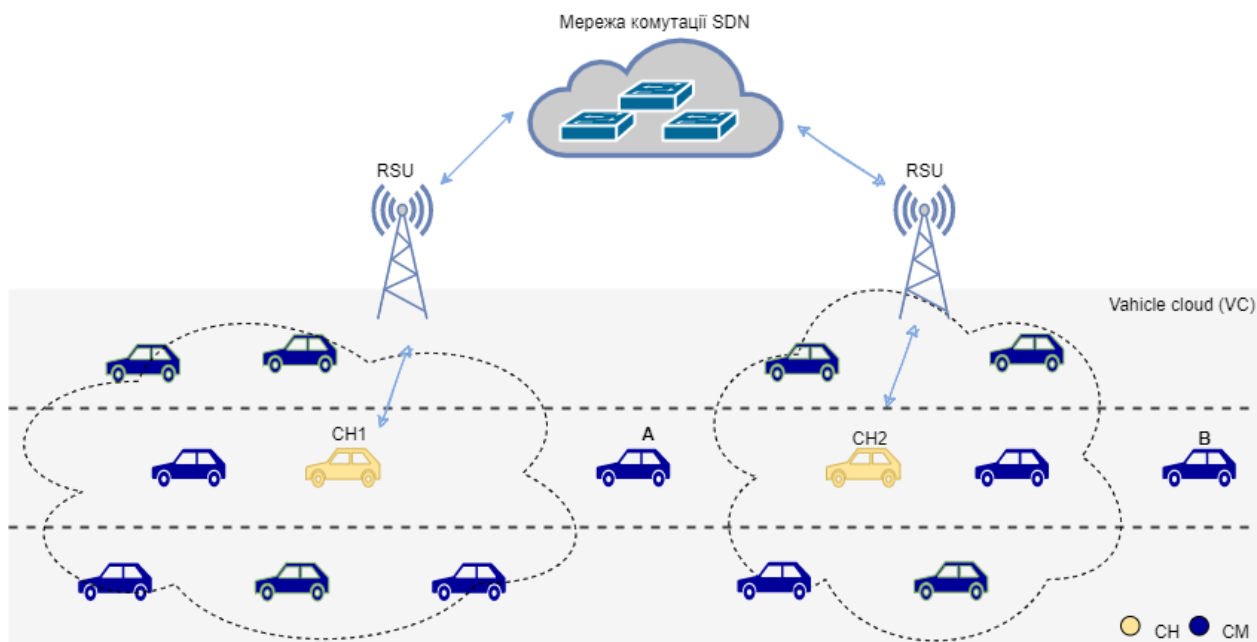


Рис. 2.2. Процес злиття кластеру мережі VANET

Також можливою є ситуація, коли транспортний засіб знаходить на межі одного чи декількох кластерів, але так і не може бути приєднаним до жодного із них. Уявімо, що вузол А продовжує рухатись зі сталою швидкістю та не потрапляє до діапазону жодного із кластерів. Злиття у такому випадку не є вигідним рішенням, оскільки таким чином зростає ризик втрати пакетів даних.

Аналогічну ситуацію можна спостерігати і з вузлом В, що тільки що приєднався до досліджуваної зони, умовно, його можна вважати новим вузлом системи, Процес злиття, у такому випадку буде виглядати таким самим чином, як і для вузла А. Проте, якщо транспортний засіб В, не знаходитиметься у зоні досяжності СН2, то RSU буде вимушеним створити новий кластер, де вузол В стане СН. У подальшому, також можливе розростання цього кластеру шляхом приєднання нових вузлів.

2.1.4 Вихід транспортного засобу за межі кластеру

Алгоритм покидання транспортним засобом визначеного кластеру є подібним до злиття. Основна різниця полягає у обов'язках виконання даної процедури, якщо під час злиття кластерів, RSU ініціює з'єднання, а СН лише оновлює свої дані, то ініціатором розпадання кластеру є СН .

Повернемося до ситуації автомобіля А на рисунку 2.2. Як ми вже визначились, даний транспортний засіб, підвищив власну швидкість до показника більше середньої відносної швидкості всередині другого кластеру, та з часом виїхав за його межі. Оскільки адміністративна роль транспортної хмари лежить на СН, саме він має прослідити зміни всередині кластеру. Першим кроком голова кластеру має проапдейтити власну табличку маршрутизації та надіслати повідомлення про внутрішні зміни до найближчого RSU. Після отримання відповіді, СН широкомовно розсилає повідомлення про внутрішні зміни всім членам кластеру.

Подальша доля транспортного вузла або вузлів, що покидають кластер лежить на відповідальності RSU, саме він має або запустити процедуру приєднання вилучених вузлів до іншого кластеру, або сформувати новий кластер та визначити нового СН.

2.2 Обумовленість вибору алгоритму кластеризації

Протягом багатьох років було розроблено безліч алгоритмів кластеризації, серед яких є k-середніх та ієрархічна кластеризація. Алгоритми кластеризації з машинного навчання використовуються у VANET для кластеризації транспортних засобів. Алгоритми нечіткої логіки також досить часто використовується для кластеризації VANET. Кероване навчання, таке як Q-навчання, та інші алгоритми кластеризації машинного навчання використовуються разом із нечіткою логікою для створення гібридної стратегії для VANET. У цій роботі запропоновано саме гібридний алгоритм кластеризації, що має за основу обидві стратегії застосування машинного

навчання та нечіткої логіки. У цьому розділі наведено аналіз існуючих методів та обумовленість вибору їх конкретної версії.

2.2.1 Алгоритми машинного навчання

Найбільш часто використовуваним алгоритмом машинного навчання для кластеризації подібних типів об'єктів у VANET є алгоритм К-середніх [37]. Під час роботи цього алгоритму створюється k кількість кластерів, що об'єднують транспортні засоби. Також допускається визначення початкових центроїдів, а в якості вхідних даних надаються координати транспортного засобу. На наступному кроці обчислюються евклідові відстані, щоб визначити новий центроїд, і центроїди обираються як СН. Щоразу, коли СН зливається з кластером або залишає кластер, середнє значення кластера може змінюватися в алгоритмі k -середніх, і йому потрібно повторно обчислити нове середнє кластеру, щоб відобразити зміну, яка може призвести до обрання нового транспортний засіб як СН.

Щоб подолати обмеження алгоритму k -середніх, у [38] використовується підхід агломеративної ієрархічної кластеризації, де напрямок і швидкість транспортних засобів розглядаються як кластер разом з деякими параметрами якості обслуговування (QoS). При ієрархічній кластеризації обчислюються евклідові відстані всіх транспортних засобів і транспортні засоби з'єднуються один з одним послідовно, починаючи з мінімальної відстані, і не вимагає кількості кластерів як вхідних даних на відміну від k -середніх [38]. Однак, хоча реалізація k -середніх є простою, ієрархічний підхід вимагає обчислення матриці близькості, що призводить до $O(n^2)$ складності простору і $O(N^2 \log(n))$ до $O(N^3)$ часової складності. Більше того, подію приєднання СН до кластеру не можна скасувати, але топологія може змінитися в будь-який час у VANET і також вимагає зміни в кластері.

Існує також відмінний підхід від [37-38]. Наукова робота [39] демонструє використання природного алгоритму, заснованого на оптимізації Grasshoppers для VANET (GOA) для оптимального вибору голови кластера. Запропонований алгоритм зменшує накладні витрати мережі в

непередбачуваних сценаріях щільності вузлів. Для цього були проведені різні експерименти для порівняльного аналізу GOA з іншими найсучаснішими методами, такими як алгоритм бабки, оптимізатор сірого вовка (GWO) та оптимізація колоній мурах (ACO). У різних експериментах використовувалися численні параметри, такі як кількість кластерів, площа мережі, щільність вузлів і діапазон передачі. Нарешті, застосування GOA у домені Flight Ad-hoc мережі (FANET) також було запропоновано для мереж наступного покоління.

2.2.2 Алгоритми нечіткої логіки

Нечітка логіка — це процес прийняття рішень, заснований на вхідних функціях належності та групі нечітких правил. Це схоже на те, як працює людський мозок, який моделює інтерпретацію невизначеної сенсорної інформації [40]. Крім того, було продемонстровано, що підходи нечіткої логіки в системах реального часу є дуже ефективними [40].

Багато стратегій кластеризації у VANET засновані на алгоритмах нечіткої логіки. Замість фіксованого значення істини чи хибності в плані приналежності вузла до кластеру, ступінь визначеності розглядається в системі нечіткої логіки (FLS). Можна виділити п'ять основних етапів FLS з точки зору VANET. На першому кроці визначаються вхідні параметри, такі як відносна швидкість, відстань транспортного засобу, напрямок руху та прискорення. На другому кроці виконується фазифікація, коли фазифікатор перетворює вхідні параметри у нечітку множину. Третій крок виконується механізмом висновку, де нечіткі правила визначаються на основі бази знань і застосовуються до нечіткої множини для отримання вихідних нечітких наборів. Процес дефазифікації виконується дефазифікатором на наступній фазі для створення чітких вихідних значень із вихідних нечітких наборів. На останньому кроці виконується настройка системи, переглядаючи діапазон входів і виходів, переглядаючи нечіткі набори та налаштовуючи правила. На рисунку 2.3 зображено FLS для формування кластерів із демонстрацією вищеписаних етапів [41].

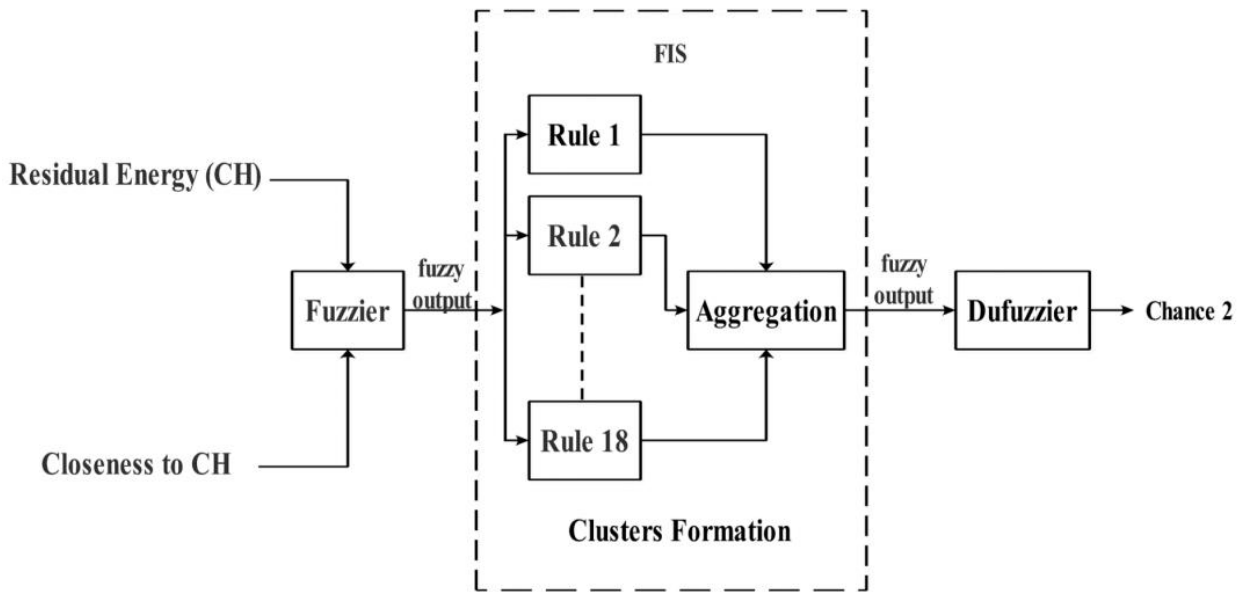


Рис. 2.3 FLS для формування кластерів

Як згадувалось у попередньому розділі, для вибору голови кластеру, кожен транспортний засіб повинен визначити свій Fit Factor. Оскільки прийняття оптимального рішення залежить від мережевого середовища, будь-яка математична модель задачі вибору СН є досить складною для формування, тому і рішення, засноване на ній, буде відповідно негнучким. З метою подолання такої перешкоди у моєму дослідженні використовується саме нечітка логіка для вирішення цієї задачі без виведення математичної моделі.

Оскільки нечіткі функції належності та нечіткі правила можуть бути модифіковані, щоб задовольнити конкретне середовище, система на основі нечіткої логіки є гнучкою [42]. Таким чином, як показано на рис. 2.3, ми пропонуємо новий контролер нечіткої логіки (FLC) для розрахунку значення коефіцієнта підгонки, який, як передбачається, буде розгорнутий у кожному транспортному засобі. Транспортні засоби оцінюють свою придатність стати СН, використовуючи нечітку логіку з точки зору середньої швидкості, ступеня сусідства та якості зв'язку RSU. Процедура розрахунку коефіцієнта підгонки на основі нечіткої логіки складається з чотирьох кроків, описаних нижче.

2.2.3 Гібридні алгоритми кластеризації

Щоб зробити процес формування кластерів і процес вибору СН більш ефективними у гібридному порядку використовуються комбінації алгоритмів машинного навчання інтегрованих із системами нечіткої логіки.

У цій роботі запропоновано застосування двох алгоритмів машинного навчання у зв'язці з нечіткою логікою. Умовно цей алгоритм можна розділити на три визначні етапи: визначення кількості кластерів, формування уявних кластерів та СН, фінальне формування кластерів на основі FLS.

Для реалізації першого етапу кластеризації було прийнято рішення по використанню алгоритму DBSCAN для визначення оптимальної кількості кластерів у транспортній мережі VANET. Алгоритм DBSCAN являє собою просторову кластеризацію вузлів із формуванням так званого шуму на основі щільності. Точки даних на площині згруповані у довільні фігури або містять шум (викиди). Алгоритми кластеризації на основі щільності дуже ефективні при пошуку областей з високою щільністю та викидів. Така модель є досить вигідною для використання у спеціальних транспортних мережах, оскільки розташування транспортних засобів вздовж доріг або автомагістралей у щільному порядку, сприятиме утворенню більш надійного зв'язку та безпечнішої комунікації між вузлами. І навпаки, у рідкозаповнених місцевостях більш вірогідним буде використання більшої кількості кластерів для попередження втрати інформації.

На відміну від Density Based алгоритму кластеризації, k-середніх та ієрархічна кластеризація не надає можливості створення кластерів довільної форми. Таки стратегії не здатні утворювати групування на основі різної щільності. На рисунку 2.4 зображено результат роботи кількох алгоритмів кластеризації даних, щільно розташованих у вигляді концентричних кіл.

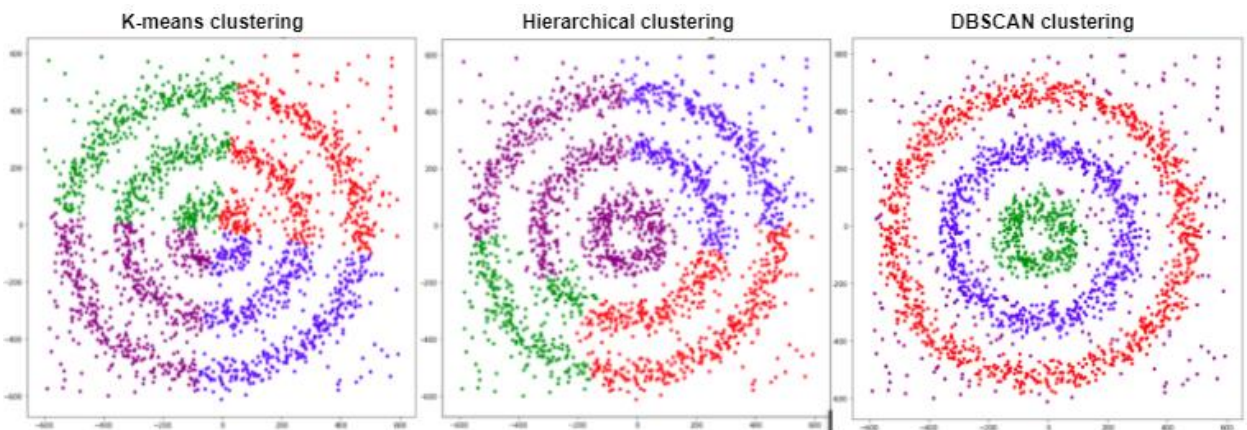


Рис. 2.4 Застосування різних алгоритмів кластеризації на датасеті вузлів розташованих у вигляді концентричних кіл

У результаті ми бачимо три різні щільні скупчення у вигляді концентричних кіл, уздовж яких розташовано певний відсоток шуму. Тепер давайте запусимо алгоритми К-середніх та ієрархічних кластерів і подивимося, як вони групують ці точки даних. Виходячи зі специфіки транспортної системи VANET можна зробити висновок, що третій варіант формування логічних груп підходить нам більше, оскільки метою застосування даного алгоритму є усунення проблем втрати даних при передачі повідомлення. Зверніть увагу на просторові проміжки між колами. У двох перших (секторних) способах кластеризації, при низькому TR ефективність маршрутизації буде низькою. Тому використання алгоритму на основі щільності розташування вузлів є чудовим вибором для спеціальних мереж.

Для наступного етапу запропонованого алгоритму формування кластеру з метою визначення явних кластерів із явними СН, що матимуть найвигідніші показники стабільності. Для реалізації цього функціоналу було обрано алгоритм нечіткої логіки fuzzy c-means (FCM). Давайте порівняємо два потужних алгоритми, щоб отримати чітке уявлення про те, наскільки підходить нечіткий алгоритм c-середніх до запропонованої ситами.

Нечіткі методи дозволяють одному і тому же об'єкту одночасно належати декільком, або навіть усім кластерам, але з різними ступенями приналежності, у той час як у традиційних k-середніх, кожна точка має

повністю належати лише одному кластеру. У кластеризації fuzzy c-means кожна точка має вагу, пов'язану з певним кластером, тому більш коректно буде говорити не про чітку приналежність вузла до кластеру, а про ймовірність входу до кластеру. На рисунку 2.5 схематично зображено різницю між hard soft кластеризацією.

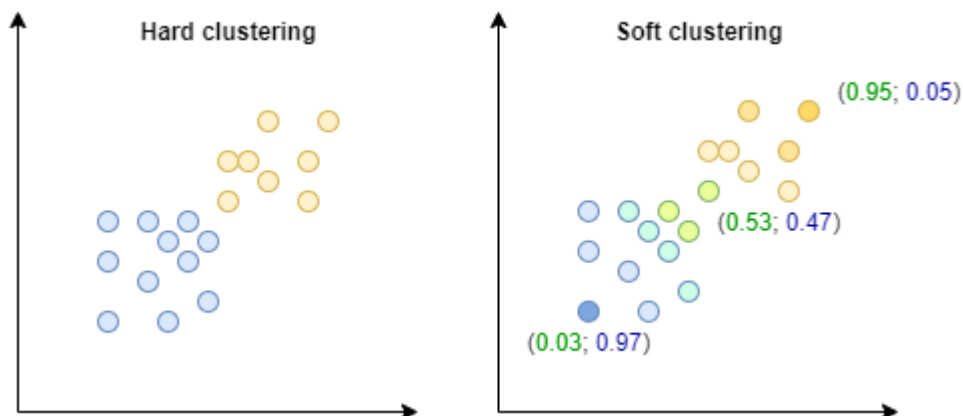


Рис. 2.5 Порівняння методів застосування чіткої та нечіткої логіки

Згідно досліджень [45], FCM, як правило, працюватиме повільніше, ніж стандартний алгоритм k-means. Причиною таких характеристик є більша кількість виконуваної роботи. Кожен вузол мережі оцінюється окремо для кожного кластеру, і, відповідно, для кожного процесу оцінювання здійснюється більше операцій, у той час як k-means потрібно лише обчислити відстань між заданими вузлами.

Отже алгоритм нечіткої логіки FCM постає як більш ефективний та заощадливий в плані енергоспоживання всієї мережі та сприяє збільшенню терміну служби мережі.

2.3 Математична модель процесу кластеризації

2.3.1 Метрики для вибору СН

У якості вхідних даних для дослідження мережі буде використано показники: швидкість руху транспортних засобів, вихідна потужність передавача, чутливість бездротового інтерфейсу автомобіля, загальне посилення антени передавача та приймача. У якості метрик будуть

використовуватись середня швидкість транспортного засобу, ступінь сусідства між вузлами, якість зв'язку з RSU та значення евклідової відстані між СН та іншими вузлами для відповідності діапазону TR. Тому при оцінці Fit Factor ці показники слід розглядати в комплексі. Задані метрики використовуються для визначення найоптимальнішої голови кластеру серед існуючих.

Як вже було зазначено у попередньому розділі, кожен вузол має самостійно підрахувати свій Fit Factor, а отже показувати кожен із метрик окремо. Першим кроком, кожен транспортний засіб визначає, наскільки близька його швидкість до середньої швидкості всіх його сусідів. Транспортні засоби обчислюють свою відносну середню швидкість ψ за наступною формулою (2.1).

$$\psi_v = \frac{S_{avg} - \mu_s}{\sigma_s} \quad (2.1)$$

У даній формулі S_{avg} – середнє значення швидкості конкретного транспортного засобу v , n -ї вибірки за різний інтервал часу; а μ_s – середня швидкість усіх його сусідніх вузлів. Також σ_s є стандартним відхиленням швидкості всіх сусідніх вузлів. Слід також звернути увагу, що ψ_v являє собою абсолютне значення відстані між вихідним балом середньої швидкості транспортного засобу v і середньою швидкістю в одиницях стандартного відхилення. Таким чином отримане значення відносної середньої швидкості потребує нормалізації. Додатковим плюсом використання заданої метрики є надання гарантій, щодо стабільності руху транспортного засобу у різні інтервали часу в межах кластеру, по відношенню до суміжних вузлів.

Результат обчислення ступеню сусідства між вузлами, сприятиме встановленню стабільності кластерної структури. Відповідно новообраний СН має володіти найвищим ступенем сусідства D_n . У якості метрики, ми використовуємо різницю швидкостей транспортних засобів для ідентифікації реальних сусідів транспортного засобу v , де v – поточний досліджуваний

вузол. В результаті ступінь сусідства D_n — це кількість відповідних вузлів-сусідів по кластеру, де різниця між їх швидкістю та швидкістю v є меншою ніж S_{thr} . З метою уникнення великої варіації кількості сусідів, поріг входу можна визначити як функцію стандартного відхилення [2]:

$$D_n = count\{(S_{avg} - v_{neig.speed}) < S_{thr}\} \quad (2.2)$$

$$S_{thr} = b \cdot \sigma_s, \text{ де } b = const \quad (2.3)$$

У свою чергу, поріг розглядатиметься як динамічний параметр, що залежить від швидкісних характеристик вузлів поблизу.

Третім важливим аспектом при виборі стабільного СН є якість зв'язку з RSU, а саме транспортний засіб з максимальною надійністю зв'язку має право бути претендентом на роль голови кластеру. З цієї причини ми віддаємо перевагу транспортному засобу v як динамічному каналу, який підключений до RSU, та забезпечує найвище значення ρ . RSU Link Quality ~ 1 , то транспортний засіб v і RSU будуть спілкуватися з максимальною швидкістю, і транспортний засіб v може зберегти зв'язок каналу з RSU і впоратися з впливом мобільності транспортного засобу. RLQ визначається наступним чином:

$$RLQ = \min\left\{\frac{\rho}{\rho_{thr}}, 1\right\} \quad (2.4)$$

У даній формулі параметр ρ представляє собою якість з'єднання між транспортним засобом v і RSU і обчислюється транспортним засобом v за наступною формулою:

$$\rho = P_{rx}^v - S_v, \quad (2.5)$$

де P_{rx}^v — це потужність повідомлення *HELLO* (в децибелах, дБ), отриманого від RSU, і обчислюється на основі бюджету каналу; S_v — це чутливість бездротового інтерфейсу автомобіля (дБ). А Бюджет канал — це врахування

усіх потужностей, що прибули та були втрачені у телекомунікаційній системі від передатчика до приймача. Він пов'язує потужність TX, потужність RX, втрати на шляху, шум RX, додаткові втрати та об'єднує їх в єдине рівняння. Просте рівняння бюджетного каналу виглядає чином:

$$P_{rx}^v = P_{tx} + G - L, \quad (2.6)$$

де P_{tx} – це вихідна потужність передавача; G – загальне посилення антени передавача та приймача; а L – загальні втрати передавача, приймача та інші великого та малого масштабу загасання, затінення, перешкоди тощо.

Якщо повернутись до початкової формули (2.4), Крім того, ρ_{thr} визначається як різниця між ранговими значеннями потужності отриманого повідомлення мінімальної (P_{rxMin}) та максимальної (P_{rxMax}) швидкості передачі, що підтримується бездротовим інтерфейсом автомобіля v . Формулу обчислення ρ_{thr} наведено далі:

$$\rho_{thr} = P_{rxMax} - P_{rxMin} \quad (2.7)$$

$$P_{rxMax} = \max\{P_{neigh.rx}\} \quad (2.8)$$

$$P_{rxMin} = \min\{P_{neigh.rx}\} \quad (2.9)$$

Фінальної метрикою аналізу є обчислення евклідової відстані між досліджуваним вузлом та центром кластеру відносно всіх його. Обчислення Best Agent Location здійснюється за наступною формулою:

$$BAL = \sqrt{(CC_x - v_x)^2 + (CC_y - v_y)^2} \quad (2.10)$$

2.3.2 FLS для формування кластерів

Повернемось до рисунку 2.3. Першим етапом системи кластеризації на базі алгоритмів нечіткої логіки є Fuzzier. Даний блок приймає чіткі значення показників вибору СН як вхідні дані та надає, як вихідні дані, відповідні

нечіткі значення належності вузла на шукану позицію на основі попередньо визначених лінгвістичних змінних і функцій.

Щоб забезпечити високу стабільність функціонування голови кластеру та запобігти створенню малих за наповненістю кластерів, ми повинні вибрати вузол, який має майже однакову швидкість з сусідами, а поблизу знаходиться більше транспортних засобів. Як нам вже відомо, щоб стати СН, кожен вузол обчислює свій Fit Factor та надсилає його іншим вузлам кластеру. Потім усі вузли голосують за більш підходящий вузол, який має локальне максимальне значення коефіцієнта підгонки. На цьому кроці Dufuzzier приймає вихідне значення нечітких даних і перетворює його на чітке значення, використовуючи попередньо визначену формулу належності вихідних даних (2.11):

$$Fit\ Factor = a \cdot \psi_v^2 + b \cdot (1 - D_n)^2 + c \cdot (1 - RLQ)^2 + d \cdot BAL^2, \quad (2.11)$$

де a , b , c , d – по визначенню пріоритетності кожної з характеристик. Отже, функція відносної середньої швидкості ψ_v і функція приналежності ступеня сусідства D_n для певної місцевості можуть мати більшу вагу ніж географічне розташування відносно центру кластеру. Крім того таких підхід є більш придатним для масштабування та врегулювання ніж стандартний набір правил по визначенню придатності вузла.

ВИСНОВКИ ДО РОЗДІЛУ 2

У другому розділі було сформовано постановку задачі, розглянуто існуючі, найбільш поширені способи кластеризації для заданої моделі спеціальної транспортної мережі VANET, а також запропоновано новий гібридний спосіб формування кластеру та вибору СН.

У якості досліджуваної моделі розглядається множина вузлів, що володіє заданими характеристиками швидкості руху, географічного положення у просторі, вихідна потужність передавача, чутливість бездротового інтерфейсу автомобіля, загальне посилення антени передавача та приймача. На основі цих характеристик, самоорганізуюча мережа може формувати кластери довільної форми та підтримувати внутрішню та міжкластерну комунікацію.

Згідно запропонованої схеми було прийнято рішення про поєднання кількох підходів. Як нам вже відомо велика кількість існуючих алгоритмів вимагають показник кількості кластерів мережі як вхідні дані. Оскільки, досліджувана система є динамічною, то визначення цього показника постає відкритою задачею. У запропонованому способі кластеризації для вирішення даної проблеми на першому етапі запускається алгоритм формування кластерів, що базується на щільності розташування вузлів. Додатковим плюсом застосування такого алгоритму є забезпечення надійності зв'язку між сусідніми вузлами для передачі даних. Саме, базуючись на цій характеристиці, буде визначено поточну кількість кластерів. Під час другого етапу відбувається формування умовних логічних груп та так званих best agent вузлів.

Для визначення голови кластеру було запропоновано архітектуру на основі локального підрахунку коефіцієнту відповідності кандидату з подальшим голосуванням вузлів за найбільш стабільний транспортний засіб відносно існуючого кластеру. Крім того, у нашому підході ми визначили основні метрики для визначення оптимального кандидату, серед яких: відносна середня швидкість руху транспортного засобу, ступінь сусідства із

суміжними вузлами, якість зв'язку із RSU та відношення географічного положення вузла відносно центру кластеру.

Таким чином у запропонованому процесі кластеризації, спочатку здійснюється об'єднання вузлів у логічні групи за вищеописаними характеристиками, а потім вибір формального лідера. Такий підхід дозволяє підвищити гнучкість схеми формування кластерів за різних умов транспортного середовища, використовуючи нечітку логіку в процедурі вибору СН.

РОЗДІЛ 3

ПОБУДОВА ПРОГРАМНОЇ МОДЕЛІ ЗГІДНО РОЗРОБЛЕНОГО АЛГОРИТМУ КЛАСТЕРИЗАЦІЇ

В даному розділі описано процес проектування та розробки програмного забезпечення для реалізації нового алгоритму кластеризації для інтелектуальної транспортної мережі VANET.

3.1 Розробка системи генерування мережі згідно заданих характеристик

Першим етапом дослідження є генерування моделі транспортної мережі для подальшого застосування реалізації запропонованого алгоритму кластеризації на її основі. Базовим об'єктом програмованої моделі є імплементація інтерфейсу *Clusterable.class* – *Vehicle.class* який має наступну структуру:

```
1. package com.vosiiivska.dbscanclustering.entity;
2.
3. import ...
4.
5. @Getter
6. @Setter
7. @AllArgsConstructor
8. @Builder
9. public class Vehicle implements Clusterable {
10.
11.     private static final long serialVersionUID = 3946024775784901369L;
12.
13.     private int id;
14.     private double x;
15.     private double y;
16.     private double speed; // [22 - 30 m/s] could be negative due to movement direction
17.     private double sensitivity; // [0-100 dB] sensitivity of the vehicle
18.     private double p_th; // [0-100 dB] transmitter output power
19.     private double g; // [0-100 dB] total transmitter and receiver antenna gain
20.     private double l; // [0-100 dB] total transmitter;
21.     private VehicleFunction function;
22.     private Set<Vehicle> neighbors;
23.
24.     public void setNeighbors(Set<Vehicle> neighbors) {...}
25.
26.     @Override
27.     public double[] getPoint() {... }
28.
29.     @Override
30.     public boolean equals(Object o) {...}
31.
32.     @Override
33.     public int hashCode() {...}
34.
35.     @Override
36.     public String toString() {...}
37. }
```

Вихідний код 3.1. Програмний об'єкт *Vehicle.class*

Інтерфейс `Clusterable.class` використовується для визначення базової поведінки для всіх його нащадків. Таким чином імплементациєю батьківського методу `double[] getPoint() {...}` є повернення координат вузла x та y . Крім того, для правильної роботи програми необхідно також перевизначити методи `toString()`, `equals()` і `hashCode()` класу `Object`. Перший використовується для зручного виводу даних про поточний екземпляр класу та здебільшо буде використовуватись у процесі виводу `DEBUG` логів у консолі. Перевизначення двох наступних методів вимагає батьківська класу є необхідним для валідного порівняння об'єктів.

Таким чином, всі приватні поля окрім `function` та `neighbors` визначаються на етапі формування даних. Для запуску генерації вузлів програмної моделі мережі VANET, використовується імплементация інтерфейсу `NetworkGraphService.class`.

Сигнатура методу генерації даних має наступний вигляд:

```
1. public List<Vehicle> createRandomData(int vehiclesAmount, int dimension, int minRange, int maxRange) {
```

Вихідний код 3.2. Сигнатура методу генерації даних

Вхідними даними даного методу є приблизна кількість вузлів для даної мережі, мінімальна та максимальна межа по наповненості мережі та розмірність. Відповідно до отриманих даних, на початку генерації випадковим чином визначається кількість кластерів та їх наповненість по кількості вузлів. Таким чином у проміжному варіанті роботи алгоритму ми будемо мати структуру по типу ключ – значення, де ключ виступатиме ідентифікатором кластеру, а його значення – прогнозованою кількістю вузлів, що до нього входять. Для кожної такої пари буде сформовано екземпляр класу `Vehicle.class` із встановленням приватних полів. Створення такого елемента показано на наступному елементі вихідного коду:


```

1.  for (int i = 0; i < clusterAmount; i++) {
2.      for (int j = 0; j < numberOfDataInEachArea[i]; j++) {
3.          Vehicle vehicle = Vehicle.builder()
4.              .id(idCounter++)
5.              .x((float) (centroids.get(0).get(i) + fRandom.nextGaussian() * variance))
6.              .y((float) (centroids.get(1).get(i) + fRandom.nextGaussian() * variance))
7.              .speed(new Random().nextInt(MAX_VEHICLE_SPEED_RANGE -
                MIN_VEHICLE_SPEED_RANGE) + MIN_VEHICLE_SPEED_RANGE)
8.              .sensitivity(new Random().nextInt(MAX_POWER_RANGE - MIN_POWER_RANGE) +
                MIN_POWER_RANGE)
9.              .p_th(new Random().nextInt(MAX_POWER_RANGE - MIN_POWER_RANGE) +
                MIN_POWER_RANGE)
10.             .g(new Random().nextInt(MAX_POWER_RANGE - MIN_POWER_RANGE) +
                MIN_POWER_RANGE)
11.             .l(new Random().nextInt(MAX_POWER_RANGE - MIN_POWER_RANGE) +
                MIN_POWER_RANGE)
12.             .build();
13.         vehicles.add(vehicle);
14.     }
15. }

```

Вихідний код 3.3. Процес створення екземпляру класу *Vehicle.class* для генерування даних моделі мережі VANET

Залежно від отриманої кількості кластерів за щільністю, визначаються центроїда для кожного кластеру. Як ми бачимо, визначення координат, базується на стандартному відхиленню кожного вузла від фіксованого значення центроїди кожного кластеру за гауссовським розподілом. Аналогічна ситуація спостерігається і для інших вхідних величин найменшої базової одиниці досліджуваної моделі. Основною відмінністю заповнення всіх приватних полів є визначення нижньої та верхньої межа показнику. Таким чином для швидкості руху транспортного засобу буде визначатись величина з межею від 22 - 30 m/s, а для чутливості бездротового інтерфейсу (*sensitivity*), вихідної потужності передавача (*p_th*) автомобіля, загального посилення антени передавача (*g*) та приймача (*l*), генеруються випадковим чином, в межах від -100 дБ до 0 дБ. У результаті ми отримуємо список об'єктів *Vehicle.class* із усіма заданими вхідними даними, що у подальшому будуть використовуватись для визначення СН.

3.2 Візуалізація мережі VANET

Для візуалізації досліджуваної моделі мережі було використано бібліотеку GraphStream.

GraphStream — це бібліотека Java для обробки графіків, що базується на аспектах формування та підтримки динамічних графіків. Така його характеристика робить використання цього сервісу хорошим варіантом для моделювання динамічної транспортної мережі. Основна увага даної бібліотеки зосереджена на моделюванні мереж динамічних взаємодій різного розміру.

Метою застосування описуваної бібліотеки GraphStream є надання способу представлення графіків і роботи з ними. Для її реалізації, програмний компонент пропонує кілька класів графів, які дозволяють моделювати орієнтовані та неорієнтовані графи, 1-графи або мультиграфи, тобто графи, які можуть мати кілька ребер між двома вузлами.

Бібліотека GraphStream також дозволяє зберігати будь-які атрибути даних на елементах графіка: числа, рядки або будь-який об'єкт. Такий аспект досить зручним при візуалізації графу, оскільки дозволяє зберігати проміжну інформації про кожну вершину та відображати (або не відображати її на графі). Це також сприяє створенню зручного процесу конвертації даних у рамках програмної моделі. Таким чином, GraphStream надає спосіб обробляти еволюцію графіка в часі, а саме: обробку способів додавання та видалення вузлів і ребер, а також способів появи, зникнення та розвитку атрибутів даних. У процесі створення графу йому будуть задані атрибути стилістики вузлів та ребер, а таж атрибути координат розташування. Результат роботи використання даної бібліотеки для множини об'єктів Vehicle.class зображено на рисунку 3.1.

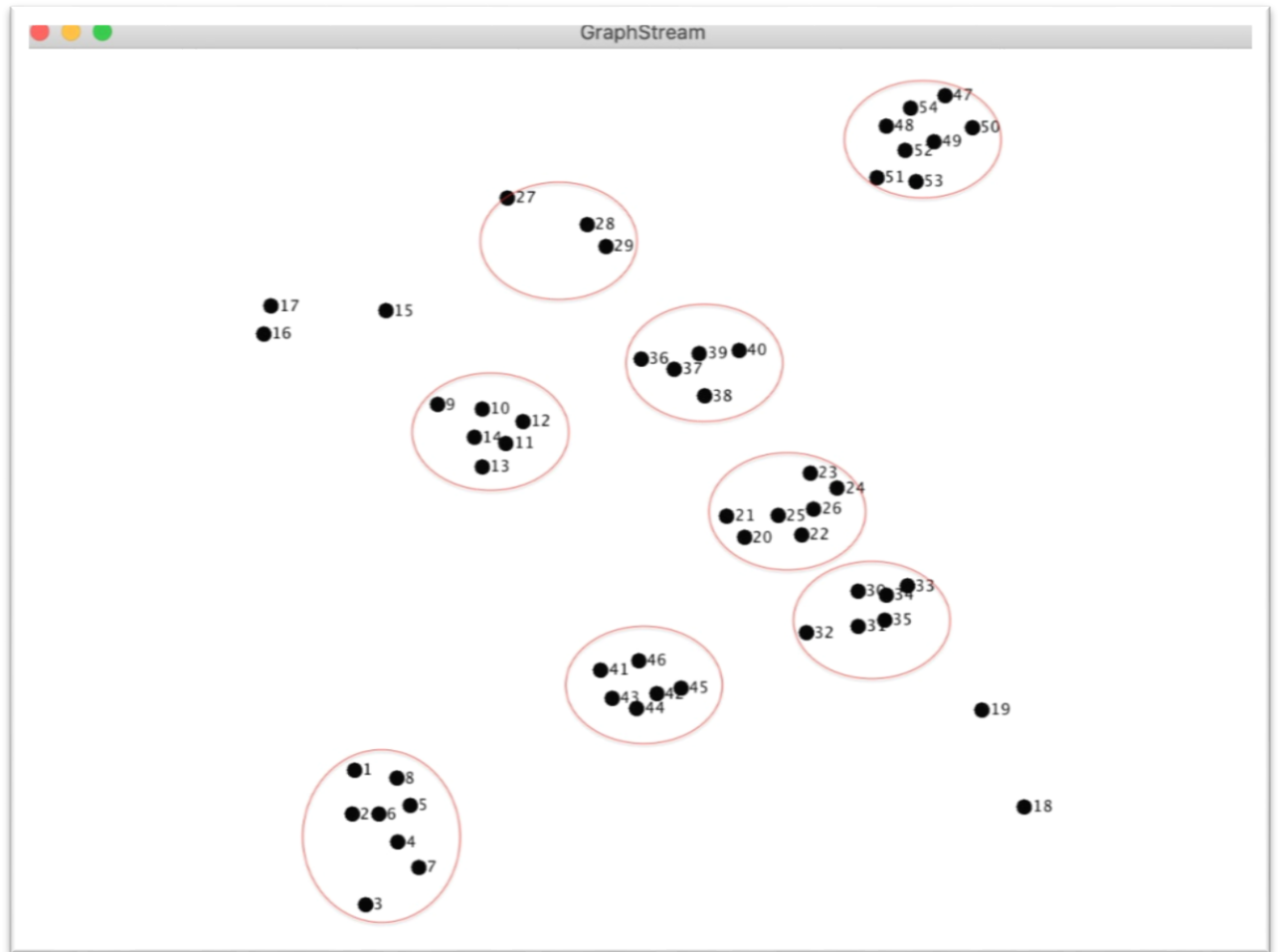


Рис. 3.1 Згенерована система вузлів спеціальної мережі

Як ми бачимо на заданому рисунку, новостворені вузли вже мають розташування, що дозволяє умовно об'єднати їх у логічні групи. Але таке логічне угруповання із більшою імовірністю не буде відповідати стандартам мережі. Уявімо, що множину вузлів $\{1, 2, 3, 4, 5, 6, 7, 8\}$ можна об'єднати в один кластер без попереднього дослідження щільності їх розташування. У такому випадку ми можемо отримати кластер із нестійкими показниками надійності зв'язку. Отже такий спосіб генерування даних використовується, у тому числі, для полегшення надання оцінки правильності роботи системи.

3.3 Реалізація алгоритму DBSCAN

Процес кластеризації можна також описати як метод навчання без нагляду. Кластерний аналіз розбиває множину вузлів на логічні групи або ряд об'єднаних точок площини із схожими властивостями та характеристиками.

Таким чином вузли одної групи, тим чи іншим чином мають відрізнятися від вузлів іншої групи. Такий підхід також має безліч різних способів, заснованих на диференційній еволюції, таких як вже відомий нам, алгоритм k-means, що базується на значенні відстані між точками, алгоритм поширення спорідненості, алгоритм середнього зміщення, dbscan, спектральна кластеризація, тощо.

Більшість із відомих алгоритмів використовують спільну стратегію дій: в першу чергу під час аналізу множини точок, визначаються елементи їх схожості, які після того використовуються у групування точок даних у групи або пакети. Нам вже відомо, що більшість алгоритмів кластеризації у транспортних мережах VANET, що використовуються наразі, стикається з проблемою відсутності вхідної інформації про кількість кластерів системи. Для вирішення цієї проблеми, у запропонованому способі кластеризації вузлів, було прийнято рішення про використання алгоритму машинного навчання DBSCAN, що базується на щільності розташування вузлів.

Алгоритм DBSCAN слід використовувати для пошуку асоціацій та структур у даних, які важко знайти вручну, але які можуть бути релевантними та корисними для пошуку закономірностей та прогнозування тенденцій. Особливістю його роботи є формування так званого шуму (noise). Це окремий транспортний засіб, який, згідно базової ідеї, не належить до жодного кластеру, а розглядається як окремий вузол. Для транспортної мережі VANET такий принцип не є актуальним, оскільки кожен транспортний вузол має підтримувати комунікацію з RSU на пряму або проміжним чином. Тому, кожен вузол викиду буде визначено як окремий кластер із подальшим прогнозованим приєднанням до нього сусідніх вузлів.

Описуваний алгоритм ґрунтується на групування точок, що знаходяться близько одна до одної, на основі підрахованої евклідової відстані та мінімальної кількості точок, що утворюють кластер. Пропоную розглянути вхідні параметри цього способу більш детально. Параметр *eps* визначає, наскільки щільно повинні бути точки одна до одної, щоб вважатися частиною

кластеру. Це означає, що якщо відстань між двома точками менша або дорівнює цьому значенню (*eps*), ці точки вважаються сусідами. У той самий час, якщо у якості *eps* буде встановлене занадто мале значення, то велика частина даних не буде угруповано та стане викидами системи, оскільки не задовольняє кількість точок для створення щільної області. З іншого боку, якщо вибране значення занадто високе, сформується великі за розміром кластери. Недоліком такого розкладу є складність вибору голови кластеру та ненадійність зв'язку із сусідніми вузлами. Тому варто вибирати невеликі значення *eps*, у нашому випадку, це значення має бути меншим за TR.

Наступний параметр *minPoints*: означає мінімальну кількість точок для формування щільної області. Вона може бути отримана з ряду вимірів (D) у наборі даних, оскільки $minPoints \geq D + 1$. Використання більшого значення цього параметру зазвичай більш вигідне для наборів даних із створенням шуму, оскільки формують більш значущі кластери.

У власній програмній реалізації, алгоритм DBSCAN реалізовано у імплементації інтерфейсу *VANETClusterer.class*, а саме – *DBSCANClustererImpl.class*. При створенні ініціалізації екземпляру класу *DBSCANClustererImpl*, задаються вхідні параметри *eps* та *minPoints*, що надалі будуть застосовані у процесу кластеризації. Метод *cluster* приймає на вхід множину вузлів *Vehicle.class* та повертає кількість кластерів які можуть бути сформовані.

Для тестування та налагодження роботи програми було також додано проміжний крок, по відображенню результату dbscan кластеризації із вхідними параметрами $eps = 5$ та $minPoints = 1$. Це означає, що кластер може утворювати множина із двох транспортних засобів та евклідова відстань між ними має бути не більше п'яти. Графічне відображення роботи алгоритму зображено на рисунку 3.2.

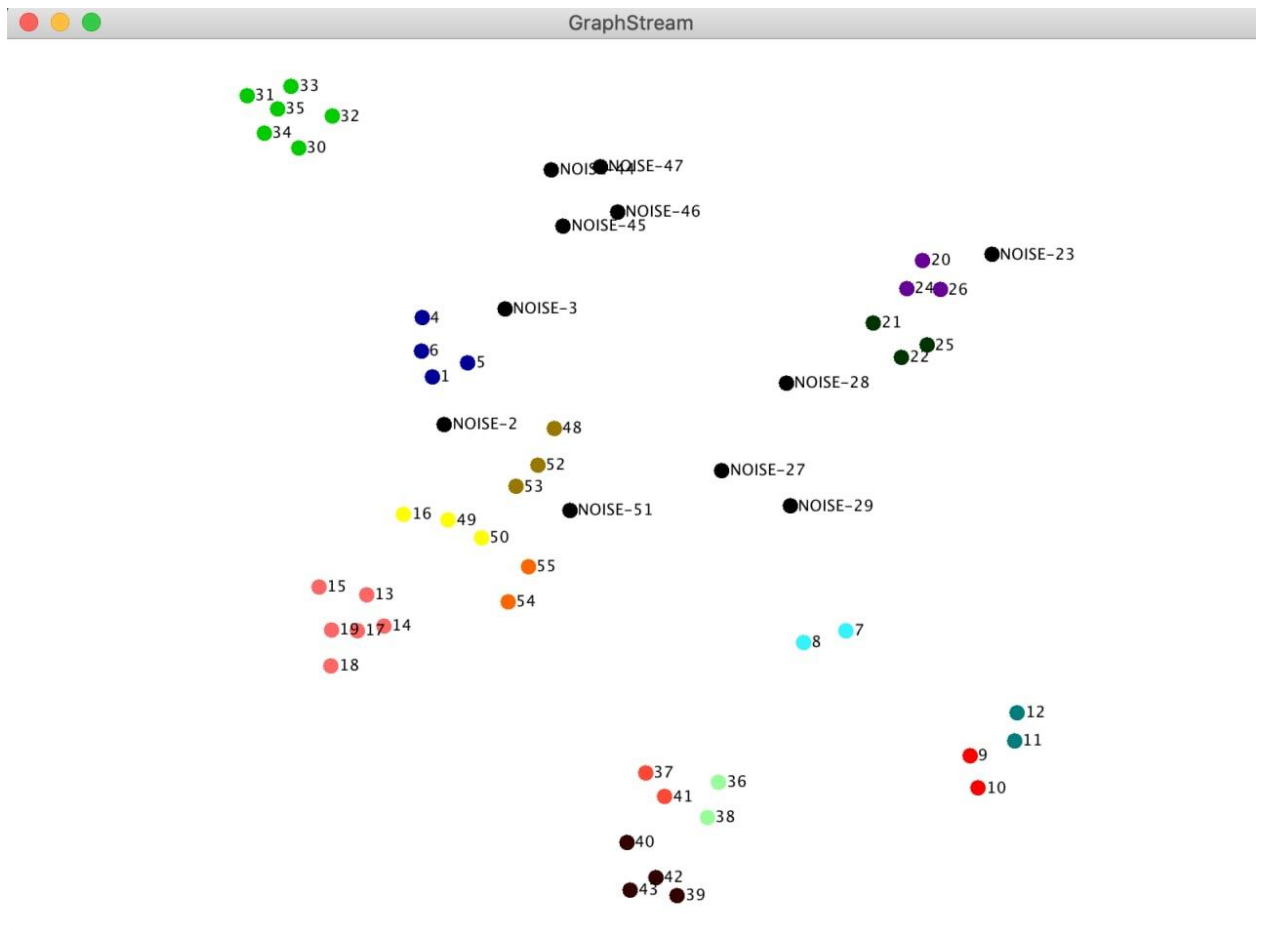


Рис. 3.2 Графічне відображення роботи алгоритму DBSCAN

Таким чином, ми бачимо, що алгоритм сформував 14 кластерів а також визначив побічний шум що не може бути доданим у існуючий кластер. У подальшому кожен такий викид буде ідентифіковано як окремий кластер.

3.4 Реалізація алгоритму вибору СН

Алгоритм кластеризації нечітких с-середніх - це метод кластеризації, який дозволяє одній частині даних належати до двох або більше кластерів.. Таким чином, вірогідність приналежності вузла до кластеру залежить від близькості до кластерного центру, чим ближче, тим більше його приналежність до конкретного кластерного центру (див. рисунок 2.5). У такому випадку сума приналежності кожної точки даних до всіх кластерних центрів має дорівнювати одиниці.

Цей метод був розроблений Данном у 1973 році та вдосконалений Бездеком у 1981 році. Він також доволі часто використовується для

розпізнавання образів і заснований на мінімізації наступної цільової функції (3.1):

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, \quad 1 \leq m < \infty \quad (3.1)$$

де m – це будь-яке дійсне число, більше одиниці, u_{ij}^m – ступінь належності x_i до кластеру j , x_i - i -й з d -вимірних даних, c_j - центр d -виміру кластера, і $\|\dots\|$ будь-яка норма, що виражає подібність між будь-якими вимірними даними та центром. Нечітке розбиття здійснюється за допомогою ітеративної оптимізації цільової функції, показаної вище, з оновленням членства u_{ij}^m і логічних центрів кластеру c_j за допомогою наступної формули (3.2):

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}, \text{ де} \quad (3.2)$$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m} \quad (3.4)$$

Дану ітерацію буде припинено коли $\max\{|u_{ij}^{k+1} - u_{ij}^k|\} < \varepsilon$, де ε — критерій завершення між 0 та 1, а k — крок ітерації. Ця процедура сходиться до локального мінімуму або точки J_m .

Математична модель алгоритму Fuzzy c-means має наступну послідовність кроків:

1. Ініціювання матриці $U = [u_{ij}], U^{(0)}$.
2. На k -тому кроці здійснюється процес підрахунку умовних центрів кластеру $C^{(k)} = [c_j], U^{(k)}$ за формулою (3.4).
3. Процес оновлення членства u_{ij} та логічних центрів кластеру c_j за допомогою формули (3.3).

4. Якщо $\|U^{k+1} - U^k\| < \varepsilon$, то припиняється робота алгоритму, інакше перехід до кроку 2.

Результат роботи Fuzzy c-means зображено на рисунку 3.3.

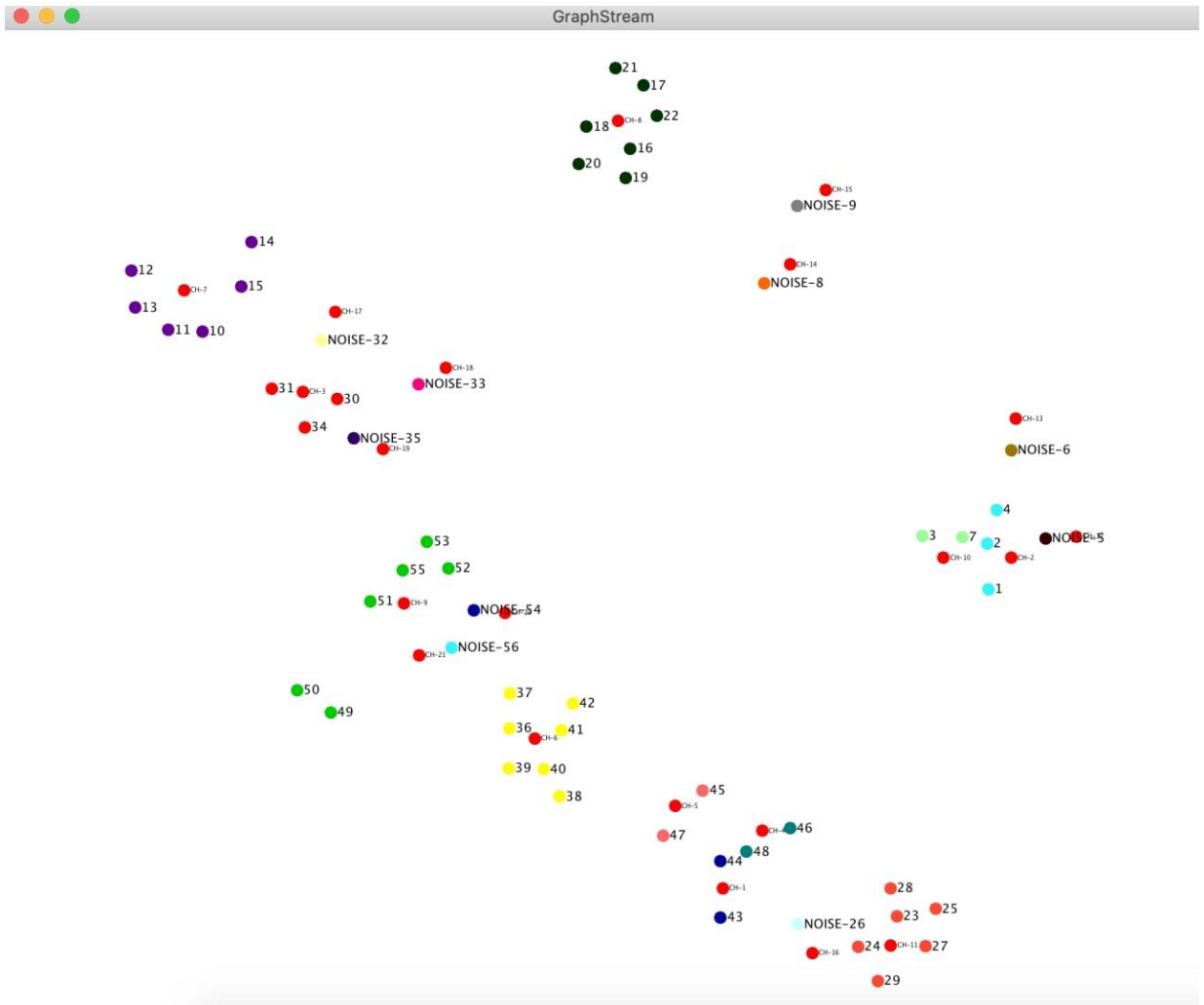


Рис. 3.3. Результат роботи алгоритму FCM у вигляді графу.

Як ми бачимо, кожному новоствореному кластеру було присвоєно свій колір. Окрім заданих вершин, на графі також зображено червоні вузли із префіксом “CH-”, що відповідають умовним центрам кластеру по відношенню до всіх сусідніх вузлів-транспортних засобів. Просторова позиція цього центру буде також використовуватись як один із вхідних параметрів для підрахунку Fit Factor для кожного вузла. Саме ця процедура є наступним кроком запропонованого способу кластеризації. Кожен член окремого кластеру має визначити свій коефіцієнт придатності за формулою 2.11 і

широкомовно розіслати його своїм колегам по кластеру. Аналіз процесу голосування та його результативність наведено у наступному підрозділі.

3.5 Аналіз роботи запропонованого способу кластеризації

3.5.1 Перевірка роботи алгоритму на експериментальній моделі

Для тестування запропонованого алгоритму формування кластерів було згенеровано систему розмірністю у 300 вузлів. Під час першого етапу роботи алгоритму, а саме - функціонування DBSCAN, було визначено кількість кластерів мережі та кількість викидів. На рисунку 3.4 зображено проміжний результат кластеризації мережі.

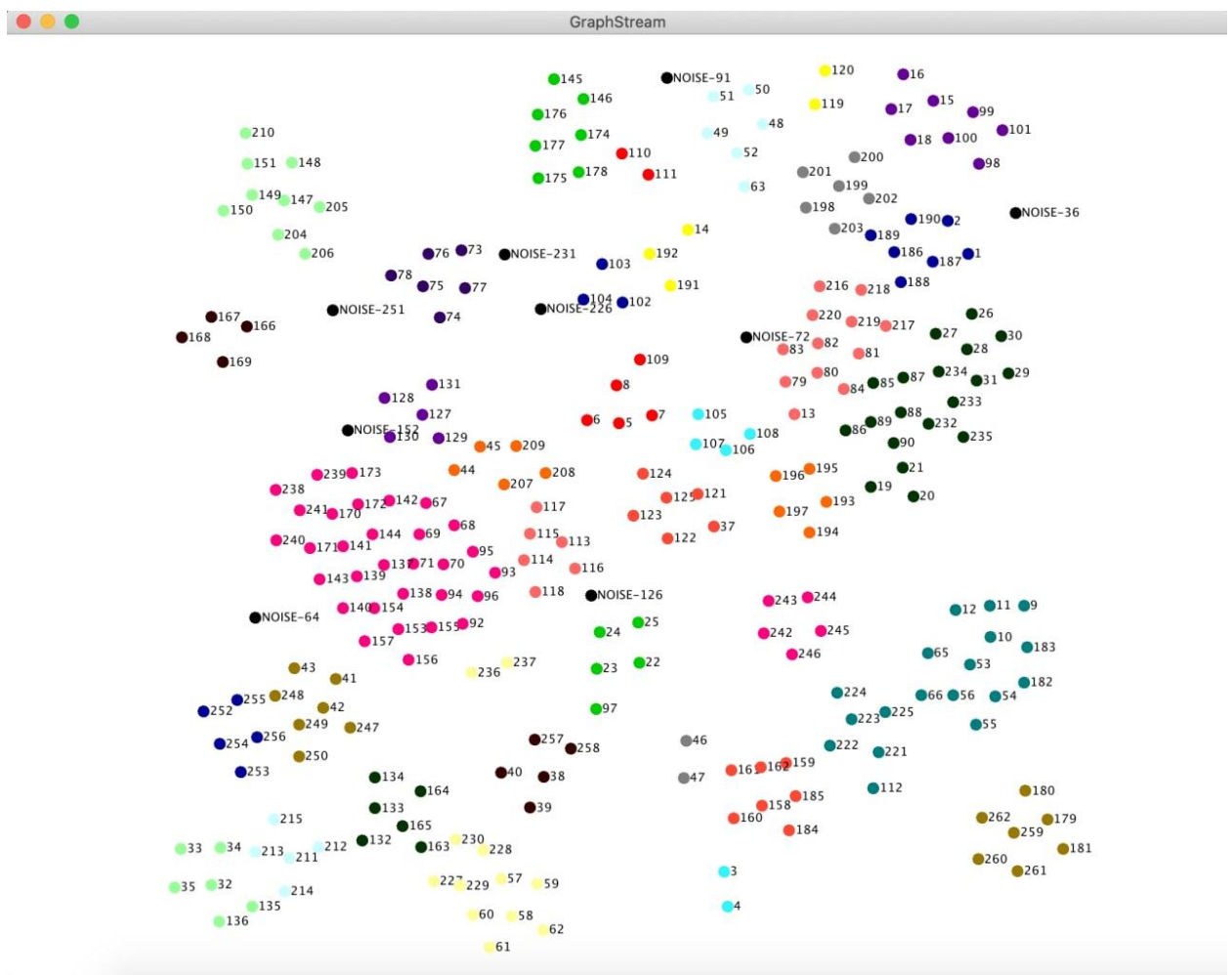


Рис. 3.4 Застосування алгоритму DBSCAN

Як і очікувалось, результат такої кластеризації не є ідеальним, оскільки створенні кластери виходять досить великими за наповненістю, а це може

негативно вплинути на надійність передачі даних та стабільність роботи системи. Тим не менш, основною метою застосування такого способу планувалось визначення хот-спотів мережі та отримання прогнозованої кількості кластерів. Отже, ми отримали 38 кластерів та 9 вузлів-викидів (36, 64, 72, 91, 126, 152, 226, 25). У таблиці 3.1 наведено дані про статистику на даному етапі дослідження.

Таблиця 3.1. Результат роботи DBSCAN

id кластеру	Множина вузлів	id кластеру	Множина вузлів
1	1, 2,186, 187, 188, 189, 190	20	102, 103, 104
2	3, 4	21	105, 106,107, 108
3	5, 6, 7, 8, 109	22	110, 111
4	9, 10, 11, 12, 182,183, 53, 54, 56, 55, 65, 66	23	112, 221, 222, 223, 225, 224
5	13, 79, 80, 81, 82, 83, 84, 220, 216, 219, 217, 218	24	113, 114, 115,116, 117, 118
6	14, 191, 192	25	119, 120
7	15, 16, 17, 18, 100, 98, 99, 101	26	127, 128, 129,130, 131
8	19, 20, 21, 86, 89, 90, 85, 87, 88, 232, 233, 234, 235, 31, 26, 27, 28, 29, 30	27	132, 133, 134,163, 164, 165
9	22, 23, 24, 25, 97	28	145, 146, 174, 175, 176, 177, 178
10	32, 33, 34, 35, 135, 136	29	147, 148, 149, 150, 151, 204, 205, 206, 210
11	37, 121, 122, 123, 124, 125	30	158,159, 160, 161, 162,184,185

12	38, 39, 40, 257, 258	31	166, 167, 168, 169
13	41, 42, 43, 247, 248, 249, 250	32	179, 180, 181, 259, 260, 261, 262
14	44, 45, 207, 209, 208	33	193, 194, 195, 196, 197
15	46, 47	34	198, 199, 200, 201, 202, 203
16	48, 49, 50, 51, 52, 63	35	211, 212, 213, 214, 215
17	57, 58, 59, 60, 61, 62, 227, 228, 229, 230	36	236, 237
18	67, 68, 69, 70, 71, 92, 93, 94, 95, 96, 137, 138, 139, 140, 154, 153, 155, 156, 157, 141, 142, 143, 144, 170, 171, 172, 173, 238, 239, 240, 241	37	242, 243, 244, 245, 246
19	73, 74, 75, 76, 77, 78	38	252, 253, 254, 255, 256

Наступним етапом виконання алгоритму слідує визначення уявних СН, тобто центрів кластеру. Метод кластеризації Fuzzy c-means приймає у якості вхідних даних кількість кластерів, яку ми визначили на початку роботи.

Виходячи з попередньої таблиці 3.1, ми маємо в сумі 47 кластерів, цей параметр передається на вхід алгоритму FCM разом із згенерованою множиною точок. На виході ми отримуємо список об'єктів типу Cluster.class, із визначеними полями вузлів, що до нього належать та центру кластеру. Візуальне зображення результату роботи другого етапу алгоритму знаходиться на рисунку 3.5.

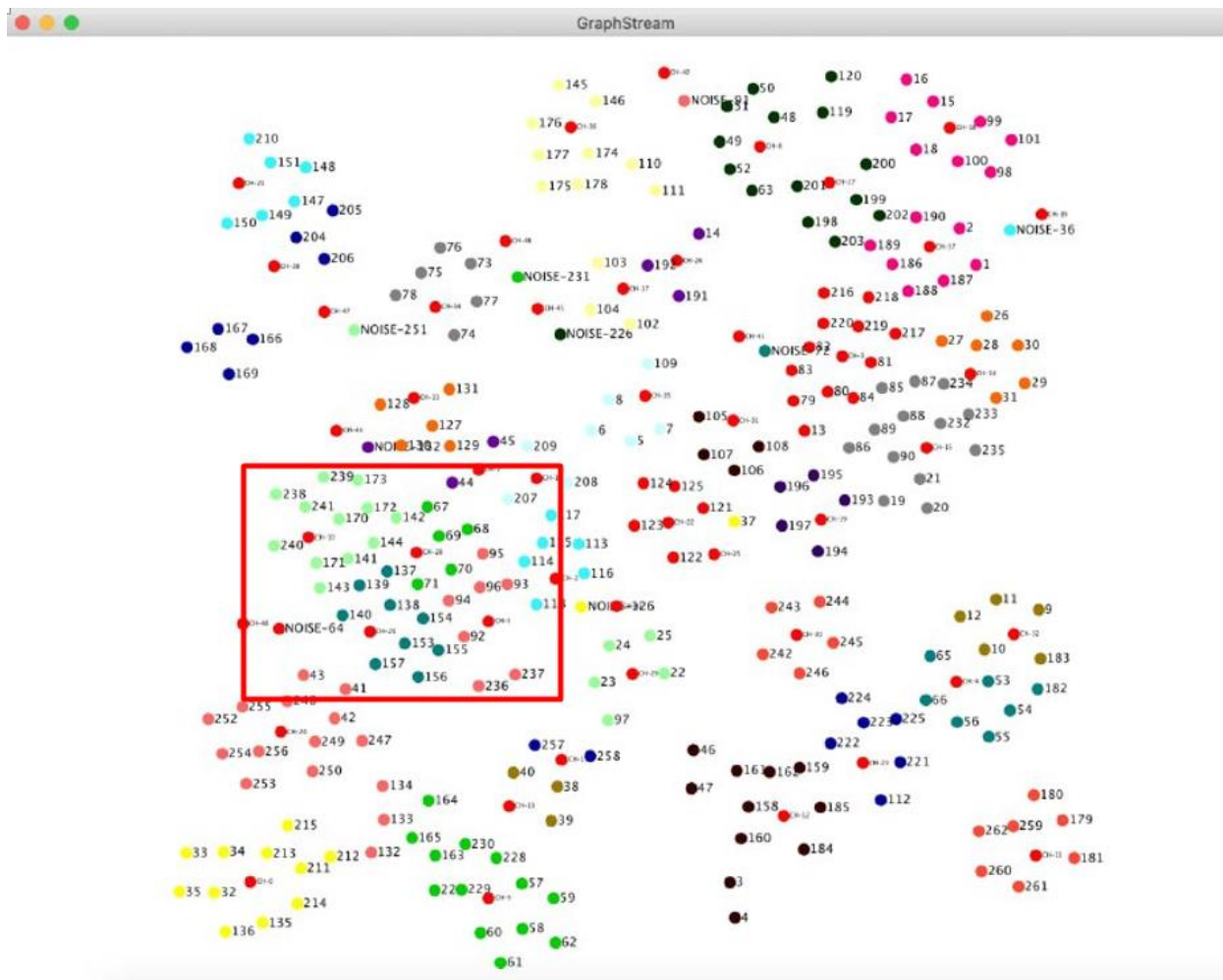


Рис. 3.5 Результат роботи FCM

Додатковий ряд вузлів з префіксом “СН-“ позначає просторове положення центру кластеру. Про них можна сказати, що вони мають позицію best agent node. Це означає, що подібне розташування вузла буде найбільш вигідним для підтримки належного рівня взаємодії транспортних засобів.

Також слід звернути увагу на поточний результат розбиття загальної мережі на кластери. Порівнюючи два графіки, видно, що на другому етапі було проведено більш якісний процес кластеризації, основуючись на заданих властивостях метрик. Наприклад, порівняємо набір вузлів із таблиці 3.1 для кластеру із ідентифікатором 18, на рисунку 3.5 цей кластери виділено червоним квадратом. Після застосування алгоритму FCM даний кластер був розділений на три логічні частини, що виступає більш коректним рішенням

для заданої системи вершин. Аналогічну ситуацію можна спостерігати і для інших груп вузлів мережі, наприклад, для кластерів 4, 5, 29, 8 і т. д.

Заключним етапом роботи даного алгоритму є вибір голови кластеру. У якості прикладу, розглянемо кластер 29, що виділений на рисунку 3.6.

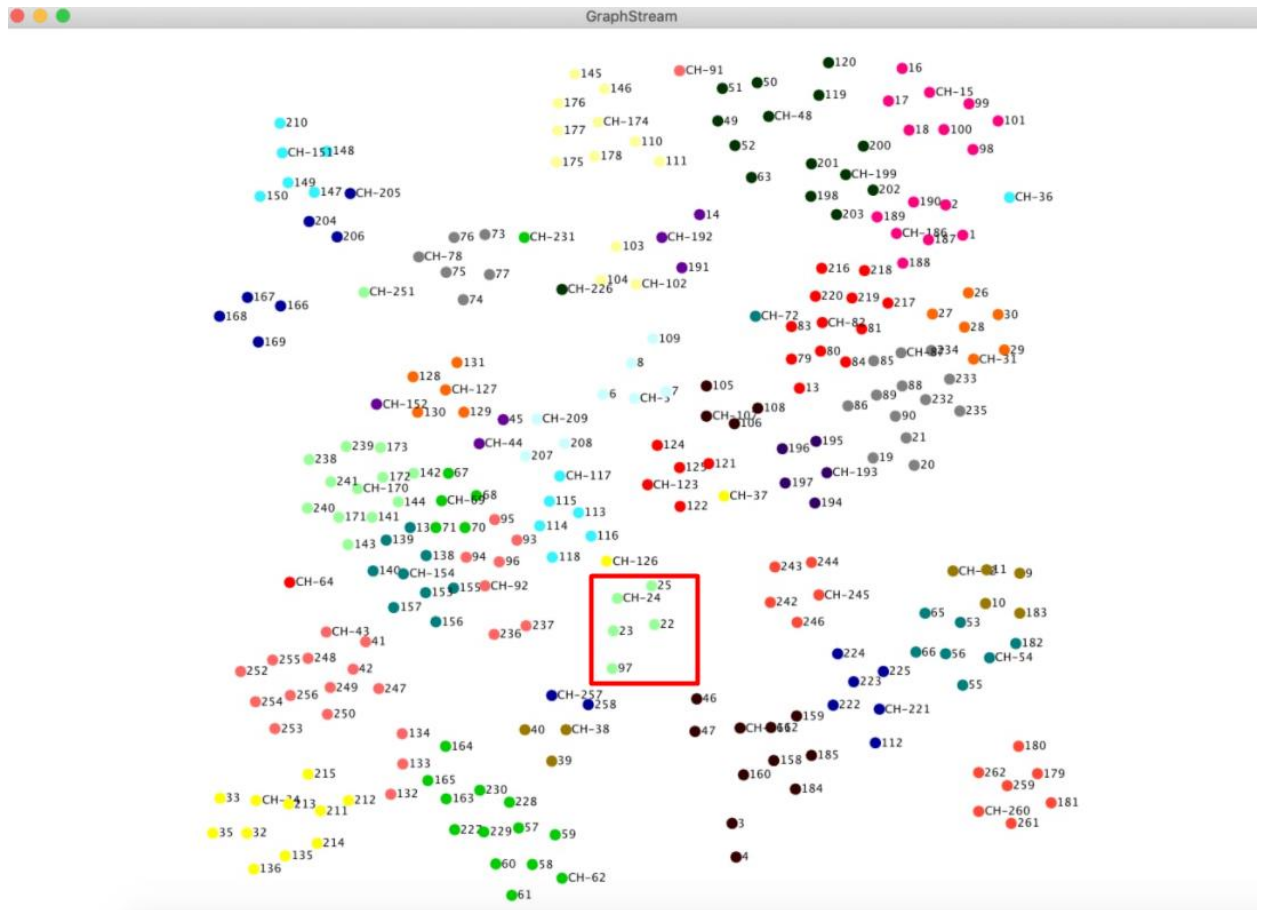


Рис. 3.6. Результат роботи гібридного алгоритму кластеризації для заданої кількості вузлів

На попередньому етапі ми сформували логічні угруповання вузлів згідно заданих метрик і наразі нас цікавить саме вибір голови кластеру. Отже, у кластері 29 наявно 5 вузлів, і всі вони, за протоколом, стають претендентами на позицію СН. У таблиці 3.2 наведено вхідні дані для кожного вузла кластеру, що були задані на початку роботи програми, серед них: просторові координати вузла x та y , середня швидкість руху, чутливість бездротового інтерфейсу автомобіля, вихідна потужність передавача, загальне посилення антени передавача та приймача відповідно. Назви колонок таблиці відповідають

назвам приватних полів об'єкту *Vehicle.class*. Саме на основі цих даних, здійснюється підрахунок метрик для визначення голови кластеру за представленими формулами (2.1) - (2.10). Функціонал по підрахунку даних метрик реалізовано у відповідній імплементації інтерфейсу *VehicleService.class*.

Таблиця 3.2. Характеристика вузлів кластеру 29

Id	x	y	v, (м/с)	S, (дБ)	Pth, (дБ)	G, (дБ)	L, (дБ)
22	41,513	26,638	23	53	46	82	82
23	40,046	26,135	27	37	52	99	26
24	40,71	27,376	26	9	55	11	27
25	41,302	27.77	22	58	49	27	28
97	38.852	21.544	29	9	21	3	95

Для підрахунку коефіцієнту належності кожного вузла, використовується об'єкт *VehicleFunction.class*. При створенні екземпляру цього класу, йому визначаються статичні параметри коефіцієнтів, що використовуються у формулі (2.11), а у якості параметрів конструктору класу, передаються метрики, приклад яких наведено у таблиці 3.3.

У нашому експерименті вхідні параметри мали наступні значення: $a = 0.31$; $b = 0.31$; $c = 0.31$; $d = 0.07$.

Таблиця 3.3. Метрики кожного вузла кластеру 29 для визначення СН

Vehicle	Relative average speed	Neighborhood degree	RSU link quality	Best agent location	Fit Factor
22	-0.931	0.4	0.235	0.703	0.597
23	0.621	0.4	0.638	0.991	0.34
24	0.233	0.6	0.199	0.625	0.293
25	-1.319	0.2	0.245	1.122	1.003
97	1.397	0.4	-0.362	5.573	3.466

Згідно формули обчислення Fit Factor (2.11), першим в черзі на присвоєння статусу СН буде вузол із найменшим значенням. Оскільки показники середньої відносної швидкості та найбільш вигідної просторової

позиції прямують до нуля, немає потреби у проведенні додаткових маніпуляцій з ними. У той час як показники ступеня сусідства вузлів кластеру та якість зв'язку із RSU мають бути конвертовані.

Таким чином, ми бачимо, що вузол 24 рухається зі найбільш стабільною швидкістю, просторово знаходиться найближче до центру кластеру та має високий ступінь зв'язності із іншими вузлами групи. Тим не менш, якість зв'язку із RSU є досить низькою. Якщо на думку менеджера мережі цей показник є пріоритетнішим, йому можна присвоїти відповідний коефіцієнт і тоді позицію СН може зайняти, наприклад, вузол 23, оскільки він має найбільш стійкий зв'язок із фіксованим маршрутизатором, але просторово має менш вигідне положення і як наслідок низький ступінь зв'язності іншими членами кластеру.

3.5.2 Порівняння гібридного алгоритму із існуючими рішеннями

У даному підрозділі надано аналіз експериментальної установки на базі порівняння отриманих результатів даного алгоритму із вже існуючими популярними рішеннями кластеризації, що мають схожий підхід. Для порівняння було обрано способи MOPSO [45] та CLPSO [46].

Отримані результати свідчать про те, що запропонована методика кластеризації функціонує ефективніше та є більш адаптивною в порівнянні з іншими методами в середовищі VANET. Також продемонстровано, що досліджуваний алгоритм дозволяє здійснювати адресацію всередині мережі з меншою кількістю кластерів, що може знизити вартість маршрутизації мережі. Крім того, це дозволяє зменшити кількість стрибків і затримок пакетів у маршрутизації у межах одного кластеру. Але також слід звернути увагу, що регулювання кількості кластерів може проводитись за рахунок встановлення відповідного діапазону передачі повідомлення між вузлами. Для пошуку оптимального рішення проводиться аналіз вхідних характеристик кожного транспортного засобу. Параметри експериментальної моделі наведені у таблиці 3.4.

Таблиця 3.4. Вхідні параметри експериментальної мережі

Назва параметру	Значення
Розмірність мережі	300
Кількість ітерацій	150
Площина	1000 м × 1000 м
Діапазон передачі повідомлення, ϵ	5 (~500 м)
Діапазон швидкості	22 - 30 м/с
Діапазон потужності передавача	0 – 100 дБ
Діапазон потужності приймача	0 – 100 дБ
Коефіцієнт метрики ψ_v	0.31
Коефіцієнт метрики D_n	0.31
Коефіцієнт метрики RLQ	0.31
Коефіцієнт метрики BAL	0.07

Експериментальне дослідження проводиться на машині з 8 ГБ оперативної пам'яті та процесором Intel(R) Core(TM) i5-8250U CPU з тактовою частотою 1.60GHz - 1.80 GHz. Змінним параметром заданої мережі є кількість вузлів, що коливається від 30 до 300. Напрямок руху вузлів є двонаправленим, це можна відслідкувати по знаку швидкості руху транспортних засобів. При цьому, середня швидкість рівномірно змінюється від 22 м/с (80 км/год) до 30 м/с (120 км/год). Діапазон передачі повідомлення для кожного вузла коливається від 100 м до 600 м. Інші параметри стосовно швидкості, потужності, площини, тощо, залишаються статичними протягом усього експерименту. Кожен алгоритм виконується у п'ять симуляцій, а їх середні значення, у свою чергу, представлено у вигляді графіків на рисунку 3.7.

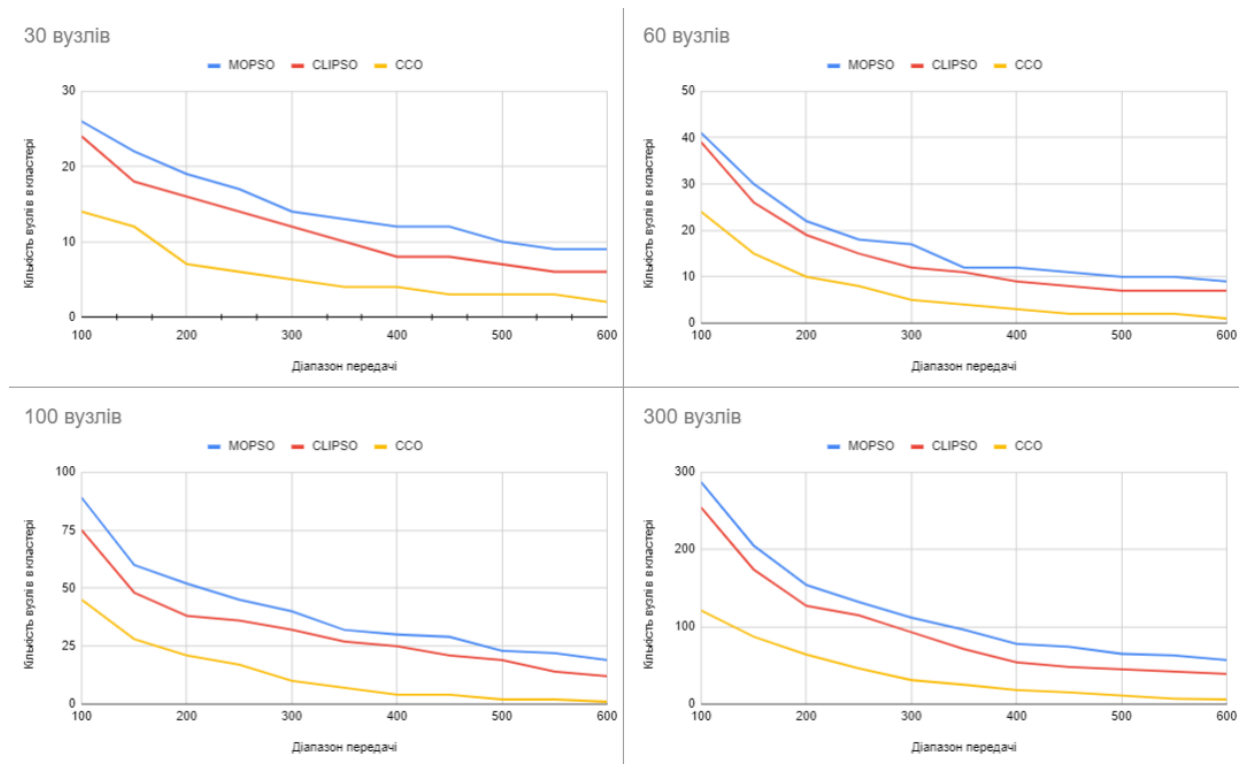


Рис. 3.7. Графік залежності кількості вузлів у кластері від діапазону передачі повідомлення для VANET із застосуванням різних алгоритмів кластеризації.

Середня кількість кластерів використовується як показник продуктивності и зображається по осі Y на рис. 3.7. Як видно з графіків, для заданої розмірності, запропонований алгоритм (CCO) створює меншу кількість кластерів для кожного діапазону передачі, щоб охопити всю мережу, порівняно із досліджуваними алгоритмами CLPSO і MOPSO. Згідно отриманих результатів, ми можемо сформулювати залежність, що зі збільшенням діапазону передачі повідомлення, буде зменшуватись кількість кластерів. Це пов'язано із тим, що зі збільшенням TR, кількість вузлів кластеру також стає більшим. Для нашого гібридного алгоритму, показник меншої кількості вузлів буде визначено вже на 1-му етапі.

Більш детальну інформацію про процес кластеризації та проміжну інформацію про вузли та підрахунок метрик, можна переглянути у додатках.

ВИСНОВКИ ДО РОЗДІЛУ 3

У даному розділі було проведено детальний опис програмної реалізації запропонованого алгоритму кластеризації для спеціальної мережі VANET та проведено ряд експериментів для дослідження отриманого рішення.

Першим етапом дослідження є генерування транспортної системи, що виступатиме об'єктом дослідження. Формування характеристик кожного вузла здійснюється за нормальним розподілом. Слід також зауважити, що при створенні тестової системи мережі, як вхідний параметр також задається приблизна або прогнозована кількість кластерів. Це зроблено для того, щоб під час роботи алгоритму для рандомно визначених центрів кластерів, так само, довільним чином генерувались вузли і розташовувались за гауссівським розподілом відносно заданих центрів. Такий принцип полегшує процес аналізу та дослідження мережі.

Крім того було надано детальну інформацію про реалізації алгоритмів DBSCAN та FCM. Відносно 2-го етапу по визначенню кількості кластерів, варто звернути увагу на те, що за своєю природою, алгоритм DBSCAN окрім визначених кластерів також створює так званий шум (одиночні вузли, що не можуть бути приєднані до кластеру). У нашому способі, такі вузли розглядатимуться як окремий кластер, із можливістю приєднання до нього інших вузлів. Крім того, на подальших етапах виконання методу кластеризації не буде потреби в обробці викидів, що може підвищити швидкість виконання алгоритму.

Під час перевірки роботи алгоритму було розглянуто кілька кластерів та новообрану голову кластеру. Згідно отриманих результатів, видно, що СН визначається правильно згідно заданих коефіцієнтів по пріоритетності застосування метрик кластеризації. Також було проведено безліч тестувань із різними комбінаціями вхідних параметрів, і кожного разу результати відповідали дійсності. Це свідчить про здатність системи до масштабування, тобто зміна пріоритетності метрик регулюється набагато простіше, порівняно

із уже існуючими рішеннями по застосуванню алгоритмів нечіткої логіки для кластеризації VANET.

Також було проведено дослідницький експеримент по порівнянню запропонованого алгоритму кластеризації та уже існуючих способів, таких як CLPSO і MOPSO. У результаті, було продемонстровано, що розроблений спосіб сприяє формуванню меншої кількості кластерів, що може достатньо зменшити загальну вартість транспортної мережі. Більш того, це також сприяє зниженню кількості затримок та кількості стрибків повідомлення у межах одного кластеру. І, відповідно, зі збільшенням діапазону передачі повідомлення для статичної вихідної кількості вузлів. Кількість утворених кластерів також зменшується.

РОЗДІЛ 4

РОЗРОБКА СТАРТАП ПРОЕКТУ

4.1. Загальна характеристика стартап-проекту

Основною ідеєю даного стартап проекту реалізація можливості застосування кластеризації вузлів в інтелектуальних транспортних мережах для забезпечення функцій моніторингу, підвищення безпеки руху та комунікації між транспортними засобами, використання побічних технологій для покращення та оптимізації якості роботи мережі. Більш детальна інформація по особливостям та характеристиці проекту наведена у таблицях 4.1 – 4.3.

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Кластеризація в мобільних SDN мережах VANET на основі fuzzy logic алгоритмів	1. Повідомлення про небезпеку на дорозі	Організована комунікація між автомобілями дозволяє оперативно поширювати інформацію аварійні ситуації, особливості стану дороги, різного виду сповіщення, тощо
	2. Віддалена персоналізація /діагностика автомобіля	Застосування персоналізованих даних та налаштувань автомобіля для його діагностики з/до інфраструктури
	3. Завантаження цифрової карти	Транспортні засоби можуть отримати доступ до Інтернету через RSU, у випадку, що RSU працює як маршрутизатор

	4. Доступ до Інтернету	Водії можуть завантажити карту для подорожей. Крім того, використання порталу для отримання інформації з мобільних точок доступ
	5. Завантаження цифрової карти	Включає в себе як трансляцію відео для пасажирів транспорту, так і передача відео з налаштованих відеореєстраторів
	6. Відеотрансляція в режимі реального часу	Для постачальників послуг, які хочуть залучити клієнтів до своїх магазинів
	7. Реклама з доданою вартістю	

Таблиця 4.2. Опис унікальних властивостей продукту

№ п/п	Назва властивості	Пояснення
1	Високодинамічна топологія	Висока швидкість транспортних засобів разом з можливістю вибору кількох шляхів визначає динамічну топологію VANET.
2	Часте відключення мережі	Висока швидкість транспортних засобів з одного боку визначає динамічну топологію, тоді як, з іншого боку, необхідні часті потреби придорожного блоку, відсутність якого призводить до частих відключень.

3	Моделювання та прогнозування мобільності	Передбачити положення транспортних засобів та їх рух дуже важко. Ці особливості моделювання та прогнозування мобільності в VANET засновані на наявності попередньо визначених моделей дорожніх карт. Швидкість транспортних засобів знову є важливою для ефективного роз'єднання мережі.
4	Комунікаційне середовище	Коли ми маємо модель мобільності, ми ще не закінчили. Оскільки модель мобільності може мати різні особливості в залежності від архітектури доріг, автомагістралей або міського середовища. Спілкування в таких ситуаціях має бути обережним.
5	Жорсткі обмеження затримки	Під час надзвичайної ситуації вчасна доставка повідомлень є критичною проблемою. Тому обробляйте такі ситуації, а говорити лише про високу швидкість передачі даних недостатньо.
6	Взаємодія з бортовими датчиками	Датчики - це спосіб зв'язку. Датчики можуть зчитувати дані, пов'язані зі швидкістю транспортного засобу, напрямком і можуть передавати дані в центр обробки даних. Таким чином, датчики можуть використовуватися при формуванні каналів і в протоколах маршрутизації.
7	Необмежена потужність батареї та зберігання	Вузли в VANET не мають обмежень по потужності та пам'яті, як у сенсорних мережах; тому оптимізація робочого циклу не настільки актуальна, як у сенсорних мережах.

Таблиця 4.3. Визначення сильних, слабких та нейтральних характеристик

ідеї проекту

№ п/ п	Технікоєконо мічні характеристик и ідеї	(потенційні)			W (слабк а сторо на)	N (нейтрал ь-на сторона)	S (сильн а сторо на)
		товари/концепції конкурентів	Мій проект	Конку- рент1			
1	Бюджетне фінансування	Розроб ка за класни й рахуно к	Викори стання інвести цій	Розроб ка за класни й рахуно к	Відсу тність фінан суван ня	Часткове фінансув ання	Повне фінанс уванн я проект у
2	Використання сучасної техніки та якісної матеріальної бази	Викори стання сучасн их технол огій	Викори стання сучасн их технол огій	Викори стання сучасн их технол огій	Застос уванн я застар ілих техно логій	Часткова технічна комплект ація	Застос уванн я новітн іх техно логій
3	Наявність маркетингової стратегії	Достат не викори стання реклам и	Достат не викори стання реклам и	Недост атне викори стання реклам и	Відсу тність рекла муван ня	Часткова трансляц ія	Актив на трансл яція
4	Високий професійний	Алгори тм розроб	Розроб лено групою	Потреб ує допрац	Ненал ежни м	Необхідн ість у системат	Належ ним чином

	рівень розробки	лений згідно стандартів	професіоналів	юваннн я	чином реалізовані й продукт	ичній підтримці продукту	розроблений продукт
5	Використання бізнес-послуг	Наявна	Наявна	Відсутня	Відсутність консультавання	Часткове консультування	Повне ведення проекту

4.2. Технологічний аудит ідеї проекту

Таблиця 4.4. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технологій
1	Кластеризація	JDK 11	Наявна	У відкритому доступі
2	в мобільних SDN мережах VANET на основі fuzzy logic	IntelliJ IDEA Ultimate / Community	Наявна	US \$499.00 на рік / Безкоштовна версія, на базі відкритого вихідного коду
3	алгоритмів	Бібліотеки java	Наявні	У відкритому доступі

Застосування спеціальної мережі транспортних засобів (VANET) створює додаткові можливості для транспортних засобів. Виділений зв'язок на короткій відстані (DSRC) і бездротовий доступ IEEE 802.11p для транспортного середовища (WAVE) були затверджені як стандарти для рівнів PHY і MAC для автомобільних мереж. Процес стандартизації IEEE 802.11p

WAVE походить від виділення смуги спектру DSRC 5,9 ГГц із смугою пропускання 75 МГц і приблизним діапазоном 1000 м. Основною метою DSRC і стандарту IEEE 802.11p WAVE є визначення правил для низької затримки встановлення з'єднання, швидкого розпізнавання мережі та диференціації додатків для нормального та аварійного використання. Вони забезпечують високу пропускну здатність зв'язку з малою затримкою між транспортними засобами. Це забезпечує ефективний аварійний зв'язок. Наприклад, у разі аварій повідомлення про сповіщення, яке передається між транспортними засобами, може бути швидшим і, таким чином, своєчасним, ніж зв'язок, що надсилається через інфраструктурну мережу (наприклад, системи стільникового зв'язку).

4.3. Аналіз ринкових можливостей запуску стартап-проекту

Відносно загальних характеристик ринку відносно даної сфери, можна сказати, він знаходиться, по більшій мірі, на стадії активного розвитку. Існує дуже велика кількість досліджень та удосконалень існуючих рішень по застосуванню інтелектуальних мереж VANET, що є достатньо вигідним часом по впровадженню стартап рішень. У таблиці 4.4 наведено попередню характеристику потенційного ринку стартап-проекту.

Таблиця 4.5. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	5
2	Загальний обсяг продаж, usd/ум.од	760000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Дослідження випереджують реальне впровадження

5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	87
7	Попит	На стадії зростання
8	Технологія	На стадії активного розвитку
9	Зміни в міжнародній торгівлі і прямих інвестиціях	На стадії зростання
10	Інтенсивність конкуренції	На стадії зростання
11	Швидкість зміни умов ринку	Ринкові умови змінюються достатньо швидко, що потребує додаткових витрат на пристосування.
12	Доступність інвестицій	Необмежено

Таблиця 4.6. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності поведінки потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Ефективний моніторинг та управління дорожньо-транспортним рухом	Державні та міські адміністрації	Забезпечення ефективного та своєчасного поширення інформації	Надання різного виду сповіщень про аварійні ситуації, особливості стану дороги до адміністрації мережі

2	Збір та обробка інформації транспортної мережі	Інформаційні служби та приватні підприємства	Надання персоналізованих даних та користувацьких налаштувань автомобіля	Діагностування трафіку та окремих транспортних засобів
3	Підвищення рівня безпеки	Державні та міські адміністрації	Надання даних про існуючі та можливі непередбачувані ситуації	Допомога при уникненні утворення непередбачуваних ситуацій
4	Поширення рекламної інформації	Рекламні агентства, власники підприємств, служби по наданню товарів та послуг	Надання можливості по поширенню рекламних сповіщень трансляції відео	Забезпечення розповсюдження рекламної інформації

Таблиця 4.7. Аналіз загроз запуску та роботи стартап-проекту

№ п/п	Фактор загрози	Опис загрози	Планове реагування компанії
1	Недостатній рівень попиту	Прийняття невдалих маркетингових рішень може призвести до низького рівня попиту на продукт	Перегляд маркетингової стратегії
2	Не належний рівень	Велика кількість досліджень даної	Застосування виваженої цінової політики за

	конкурентної спроможності	сфери створює серйозні конкурентні відносини	забезпечення якісного надання послуг
3	Агресивність конкурентного впливу	Здійснює вплив на стратегію поширення продукту	Використання професійних стратегій продажу
4	Нестабільність політичної ситуації	Балансування курсу	Отримання кваліфікованої юридичної допомоги
5	Складна економічна ситуація	Відсутність фінансування	Пошук додаткового фінансування та заощадження витрат

Таблиця 4.8. Аналіз можливостей по реалізації стартап-проекту

№ п/п	Фактор можливості	Зміст можливості	Можлива реакція компанії
1	Отримання лідерської позиції на ринку	Підвищення показників попиту на продукт	Збільшення кількості продажів та надання якісних послуг
2	Колаборація із існуючими технологіями та рішеннями на ринку	Збільшення об'ємів закупівлі продукту	Збільшення кількості продажів та надання якісних послуг
3	Вихід на міжнародний ринок	Вдосконалення продукту та збільшення рівнів попиту	Збільшення потужностей виробництва

Таблиця 4.9. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	Прояв характеристики конкурентного середовища	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Тип конкуренції	Чиста конкуренція	Реалізації в сфері алгоритмів маршрутизації вузлів VANET та управління мережею
Рівень конкурентної боротьби	Національний	Створення стратегій для охоплення усього ринку
Галузева ознака	Внутрішньогалузева	Створення акценту на перевагах даного продукту, порівняно із конкурентними рішеннями
Конкуренція за видами товарів	Товарно-родова	Вплив на методи організації мережі
Характер конкурентних переваг	Нецінова	Ефективність методу полягає у оптимальній кластеризації транспортної мережі
Інтенсивність конкуренції	Марочна	Видозміна способів формування мережі згідно запитів користувача

Таблиця 4.10. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товаризамінники
	Продуктові компанії	Потреба у додаткових інвестиціях	Залучення постачальників в послуг	Орієнтованість на клієнтські потреби	Прийняття рішень в процесі детального дослідження
Висновки	Основною загрозою є поява нових конкурентів у доменній області. Конкуренція на міжнародному ринку, наразі, є вкрай складною та вимагає професійного підходу.	Недостатня кількість або повна відсутність інвестицій може стати великим бар'єром для виходу на великий ринок	Показник залучення постачальників є низьким	Для даного рішення маємо більш стабільну ситуацію. Більш можливими є варіанти запровадження додаткового програмного забезпечення у якості нових фічей.	Ускладненням може слугувати виключно відсутність кваліфікованої експертизи

Таблиця 4.11. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
-------	-------------------------------	---

1	Цінова політика	Встановлення ціни на продукт має бути вигідним для обох сторін, постачальника продукту та клієнтів
2	Репутація виробника	Маючи бездоганну репутацію, довіра до компанії з боку клієнта зростає. Також вона може сприяти вдалому виходу на міжнародний рівень
4	Унікальність сервісних послуг	Надання якісного та ефективного сервісного рішення як продукту є важливим аспектом для користувача

Таблиця 4.12. Порівняльний аналіз сильних та слабких сторін стартап-проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Порівняння рейтингу товарів-конкурентів						
			-3	-2	-1	0	+1	+2	+3
1	Лідерські позиції на ринку	20					+		
2	Цінова політика компанії	18						+	
3	Торговий маркетинг	18				+			
4	Репутація компанії	19					+		
5	Лояльність до бренду	16				+			
6	Інвестиційний бюджет	19			+				

На основі попереднього аналізу різних факторів загроз та можливостей стартап-проекту по застосуванню кластеризації вузлів в мобільних інтелектуальних транспортних мереж VANET на основі SDN, у таблиці 4.13

наведено матрицю SWOT. Дана схема відображає загальну картину про розвитку проекту на ринку.

Таблиця 4.13. SWOT-аналіз стартап-проекту

<p>Сильні сторони:</p> <ol style="list-style-type: none"> 1. Застосування сучасних технологій та підходів 2. Використання відтестованих рішень 3. Правильно обрана стратегія продажу 4. Забезпечення сервісної підтримки 	<p>Слабкі сторони:</p> <ol style="list-style-type: none"> 1. Потреба у постійному фінансуванні 2. Відсутність спеціалізованого маркетингового досвіду 3. Складна політична та економічна ситуація
<p>Можливості:</p> <ol style="list-style-type: none"> 1. Використання запропонованого методу на ринку 2. Створення нового програмного забезпечення 3. Ріст обсягів продажу продукту 	<p>Загрози:</p> <ol style="list-style-type: none"> 1. Відсутність попиту на продукт 2. Неконкурентоспроможність

Таблиця 4.14. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Активне опанування ринку та укладення нових договорів із підприємцями	Висока	Короткі

2	Слідування стратегії по забезпеченню конкурентоспроможності	Середня	Короткі
---	---	---------	---------

4.4. Розробка ринкової стратегії проекту

При планування ринкової стратегії проекту варто зосередити увагу на цільових групах потенційних споживачів.

Таблиця 4.15. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Державні та міські адміністрації	Готовий	Високий	Середня	Проста
2	Інформаційні служби та приватні підприємства	Готовий	Високий	Висока	Проста
3	Рекламні агенства, власники підприємств, служби по наданню	Готовий	Високий	Низька	Проста

	товарів та послуг				
Обрані цільові групи: стратегія диференційованого маркетингу					

Згідно аналізу потенційних комерційних груп споживачів на запропонований продукт, було обрано стратегію диференційованого маркетингу, оскільки компанія працює водночас із кількома сегментами, з розробкою окремих програм для ринкового впливу.

Таблиця 4.16. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Стратегія зростання	Прагнення компаній до активних темпів економічного зростання	Стратегія концентрованого зростання, внутрішнє розширення продукту, а також виробництво нових продуктів	Стратегія диференційованого маркетингу

Таблиця 4.17. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першо-прохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або	Чи буде компанія копіювати основні характеристики	Стратегія конкурентної поведінки*

		забирати існуючих у конкурентів?	товару конкурента, і які?	
1	Так	Стратегія поєднання нових та старих споживачів продукту	Відсутність потреби	Стратегія наслідування лідера

Таблиця 4.18. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Забезпечення високої якості продукту	Стратегія диференційованого маркетингу	Професіональний підхід до розробки продукту згідно стандартів	Якість, надійність, безпека
2	Вигідна ціна на продукт	Стратегія наслідування лідера	Широкий спектр можливостей по застосуванню додаткового ПЗ	Доступність, унікальність, новизна

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.19 Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Кластеризація мережі	Товар забезпечує ефективне формування груп вузлів	Якісний процес кластеризації. новизна
2	Скорочення витрат на інфраструктуру	Особливість даного методу сприяє скороченню витрат	Ефективність обробки даних

Таблиця 4.20 Формування системи збуту стартап-проекту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Зорієнтованість на високоякісний продукт	Пропозиція на надійність функціональності продукту	Значна	Влаштування тендорних торгів, сильна рекламна програма

Таблиця 4.20 Концепція маркетингових комунікацій стартап-проекту

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Адаптивність у використанні	Мережа Інтернет	Гарантування надійності	Транслявання товару	Ефективна транспортна мережа

ВИСНОВКИ ДО РОЗДІЛУ 4

Спеціальні мережі VANET на сьогоднішній день вже стали частиною державних проектів. Національне управління автомобільних доріг Індії (NHAI) планує замінити стягнення плати вручну на площах на електронні системи збору (ETC) по всій країні. Система ETC буде заснована на радіочастотній ідентифікації (RFID), яка буде доповнена бездротовим бортовим блоком (OBU) на транспортному засобі, а також канцелярським придорожнім блоком (RSU) на платіжній платформі.

Австралійська поліція в Новому Південному Уельсі (Новий Південний Уельс) і Вікторії розглядає можливість впровадження нового типу лазерної камери контролю швидкості, яка зможе ловити водіїв за допомогою мобільних телефонів, а також водіїв, які перевищують швидкість на відстані півмилі [13].

Згідно дослідження, проведеного у даному розділі було виявлено можливість комерціалізації проекту згідно маркетингових умов ринку. Попит на даний проект створює особливість продукту по зниженню витрат на інфраструктуру, що значно виграє у порівнянні із існуючими конкуруючими рішеннями.

З огляду на потенційні групи клієнтів, стартап-проект по кластеризації транспортних вузлів SDN мережі VANET із використанням алгоритмів нечіткої логіки має перспективи розвитку зважаючи на фактори загроз та можливостей. Таким чином, можна зробити висновок, що подальший розвиток проекту та його імплементація в інтелектуальні мережі є доцільною.

ВИСНОВКИ

Дана робота зорієнтована на вирішення проблем VANET, спричинені особливостями мережі, а саме висока динамічність топології, складність комунікаційного середовища, жорсткі обмеження по затримці, нестабільна швидкість зміни положення вузлів мережі. У своїй роботі, мною було запропоновано альтернативний підхід по кластеризації мережі цієї доменної області. На початку роботи було проведено дослідження інтелектуальних мереж загалом та існуючі способи подолання проблем конструювання та управління транспортної мережі. Було виявлено велику варіативність у підходах та технологіях, пристосованих до вирішення проблем мобільних мереж, у тому числі і MANET. На даний момент, представлено кілька доволі перспективних стратегій, більшість яких базується на складних та дорогих техніках. Крім того, вони досить часто створюють додаткові накладні витрати.

У результаті попереднього дослідження було прийнято рішення по реалізації способу кластеризації із поєднанням технологій машинного навчання та алгоритмів нечіткої логіки, оскільки специфіка машинного навчання дозволяє ефективно формувати кластери, а використання нечіткої логіки підвищує рівень стабільності та простоту у функціонуванні та життєвому циклі кластеру. Згідно досліджень, імплементація саме гібридних алгоритмів забезпечує:

- 1) Високу адаптивність до мережевого середовища
- 2) Зниження енергоспоживання мережі
- 3) Стійкість до утворення гарячих зон
- 4) Балансування споживання вузлів окремих кластерів
- 5) Зниження інфраструктурних витрат

Запропонований гібридний алгоритм дозволяє формувати кластери для існуючої мережі та обирати голову кластеру. Він дозволяє вирішувати певні проблеми, пов'язані з рішеннями по кластеризації та забезпечує високу стабільність та надійність роботи системи. Наприклад, для багатьох методів кластеризації, основним із масиву вхідних даних є кількість кластерів для

заданої системи. Відповідно до особливостей динамічної мережі, використання константного попередньо заданого значення може нашкодити при формуванні логічної групи вузлів, і як наслідок – вибору голови кластеру. Мій гібридний алгоритму включає в себе етап по визначенню цього показника виключно для діючого формату мережі за допомогою машинного навчання. Такий підхід також є привабливим тим, що він дозволяє створити логічні групи динамічного розміру, попереджуючи утворення дуже великих кластерів.

Нечітка логіка застосовується для вибору голови кластеру та визначення наповнення кластеру в залежності від вірогідності приналежності кожного вузла до кластерного центру. В запропонованій архітектурі передбачається здатність кожного вузла до підрахунку коефіцієнту відповідності кандидату на позицію СН. Вхідними метриками, при цьому, слугують відносна середня швидкість руху, ступінь сусідства із іншими вузлами формованого кластеру, потужність зв'язку транспортного засобу з RSU та відношення географічного положення вузла відносно центру кластеру.

Такий підхід по поєднанню технологій забезпечує гнучкість системи формування логічних груп транспортних одиниць. Дане твердження було підкріплено експериментальними дослідженнями, описаними у третьому розділі роботи. У якості кандидатів для порівняння було обрано такі способи кластеризації як CLPSO і MOPSO. Отримані результати демонструють. Що розроблений гібридний алгоритм дозволяє скоротити кількість кластерів мережі, що призводить до скорочення витрат на загальну вартість транспортної мережі та зменшенню кількості затримок передачі повідомлення. Кращих результатів можна очікувати при збільшенні діапазону передачі повідомлення.

У четвертому розділі було проведено розробку стартап-проекту по просуванню продукту програмного забезпечення по кластеризації вузлів в транспортній SDN мережі VANET. Було виявлено способи по комерціалізації проекту із такими напрямками застосування, як урегулювання дорожнього

руху з метою попередження аварійних ситуацій, віддалена діагностика окремого транспортного засобу та надання допомоги або підказок водіям, у тому числі і карт для орієнтування на місцевості. Додатковим слотом для ринкових можливостей проекту є створення та використання додаткового програмного забезпечення, зорієнтованого на користувача та заповнення рекламної ніші.

Отже, дана наукова робота вносить свій вклад в дослідження VANET та демонструє перспективне рішення проблеми масштабування мережі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. S. A. A. Shah, E. Ahmed, J. J. P. C. Rodrigues, I. Ali and R. Md Noor, "Shapely Value Perspective on Adapting Transmit Power for Periodic Vehicular Communications," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 977-986, March 2018, doi: 10.1109/TITS.2017.2775965.
2. Rawashdeh, Z.Y., Mahmud, S.M. A novel algorithm to form stable clusters in vehicular ad hoc networks on highways. *J Wireless Com Network* 2012, 15 (2012). <https://doi.org/10.1186/1687-1499-2012-15>
3. Venkataraman, H., Delcelier, R., & Muntean, G. (2013). A moving cluster architecture and an intelligent resource reuse protocol for vehicular networks. *Wireless Networks*, 19, 1881-1900.
4. M. S. Talib, A. Hassan, B. Hussin, and A. A.-h. Hassan, "Vehicular ad-hoc networks: Current challenges and future direction of research," *Jour Adv Res. Dyn. Control Syst*, vol. 10, no. 2, pp. 2065–2074, 2020.
5. "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages," in *IEEE Std 1609.2-2019*, vol., no., pp.0_1-105, 2019, doi: 10.1109/IEEESTD.2006.243731.
6. G. Karagiannis et al., "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions," in *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 584-616, Fourth Quarter 2011, doi: 10.1109/SURV.2011.061411.00019.
7. Toor, Vikas and N. S. Ambedkar. "Comparison of MANETs and VANETs." (2021).
8. S. A. A. Shah, E. Ahmed, J. J. P. C. Rodrigues, I. Ali and R. Md Noor, "Shapely Value Perspective on Adapting Transmit Power for Periodic Vehicular Communications," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 977-986, March 2018, doi: 10.1109/TITS.2017.2775965.

9. Rawashdeh, Z.Y., Mahmud, S.M. A novel algorithm to form stable clusters in vehicular ad hoc networks on highways. *J Wireless Com Network* 2012, 15 (2012). <https://doi.org/10.1186/1687-1499-2012-15>
10. Venkataraman, H., Delcelier, R., & Muntean, G. (2013). A moving cluster architecture and an intelligent resource reuse protocol for vehicular networks. *Wireless Networks*, 19, 1881-1900.
11. M. S. Talib, A. Hassan, B. Hussin, and A. A.-h. Hassan, "Vehicular ad-hoc networks: Current challenges and future direction of research," *Jour Adv Res. Dyn. Control Syst*, vol. 10, no. 2, pp. 2065–2074, 2018.
12. "IEEE Recommended Practice for Architectural Description for Software-Intensive Systems," in *IEEE Std 1471-2000* , vol., no., pp.1-30, 9 Oct. 2000, doi: 10.1109/IEEESTD.2000.91944.
13. "ISO/IEC/IEEE Systems and software engineering -- Architecture description," in *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)* , vol., no., pp.1-46, 1 Dec. 2011, doi: 10.1109/IEEESTD.2011.6129467.
14. Yeferny, Taoufik and Sofian Hamad. "Vehicular Ad-hoc Networks: Architecture, Applications and Challenges." *ArXiv* abs/2101.04539 (2021): n. pag.
15. Zhang Guoqing, Mu Dejun, Xu Zhong, Yang Weili and Cai Xiaoyan, "A survey on the routing schemes of urban Vehicular Ad Hoc Networks," 2008 27th Chinese Control Conference, 2008, pp. 338-343, doi: 10.1109/CHICC.2008.4605717.
16. F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," in *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12-22, June 2007, doi: 10.1109/MVT.2007.912927.
17. W. Wang, F. Xie and M. Chatterjee, "TOPO: Routing in Large Scale Vehicular Networks," 2007 IEEE 66th Vehicular Technology Conference, 2007, pp. 2106-2110, doi: 10.1109/VETECONF.2007.442.

18. M. Mabiala, A. Busson and V. Veque, "Inside VANET: Hybrid Network Dimensioning and Routing Protocol Comparison," 2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring, 2007, pp. 227-232, doi: 10.1109/VETECS.2007.59.
19. T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato and Y. Nemoto, "A Stable Routing Protocol to Support ITS Services in VANET Networks," in IEEE Transactions on Vehicular Technology, vol. 56, no. 6, pp. 3337-3347, Nov. 2007, doi: 10.1109/TVT.2007.906873.
20. O. K. Tonguz, N. Wisitpongphan, J. S. Parikh, F. Bai, P. Mudalige and V. K. Sadekar, "On the Broadcast Storm Problem in Ad hoc Wireless Networks," 2006 3rd International Conference on Broadband Communications, Networks and Systems, 2006, pp. 1-11, doi: 10.1109/BROADNETS.2006.4374403.
21. M. Umar Farooq, Junaid Ahmed Shadab and Khaleel Ur Rahman Khan, "An effective way of broadcasting alert messages in vehicular Ad hoc Network," 2010 International Conference on Communication and Computational Intelligence (INCOCCI), 2010, pp. 132-137.
22. "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages," in IEEE Std 1609.2-2006 , vol., no., pp.0_1-105, 2006, doi: 10.1109/IEEESTD.2006.243731.
23. G. Karagiannis et al., "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions," in IEEE Communications Surveys & Tutorials, vol. 13, no. 4, pp. 584-616, Fourth Quarter 2011, doi: 10.1109/SURV.2011.061411.00019.
24. E. Schoch, F. Kargl, M. Weber and T. Leinmuller, "Communication patterns in VANETs," in IEEE Communications Magazine, vol. 46, no. 11, pp. 119-125, November 2008, doi: 10.1109/MCOM.2008.4689254.
25. Toor, Vikas and N. S. Ambedkar. "Comparison of MANETs and VANETs." (2012).

26. G. Karagiannis et al., "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions," in *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 584-616, Fourth Quarter 2011, doi: 10.1109/SURV.2011.061411.00019.
27. S. Allani, R. Chbeir, T. Yeferny and S. Ben Yahia, "Smart Directional Data Aggregation in VANETs," 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), 2018, pp. 63-70, doi: 10.1109/AINA.2018.00022.
28. Saraswathi, R. J. Vijaya and A. Subramani. "Adaptive Broadcast Routing Protocol using Fuzzy Logic System for MANET." (2019).
29. S. Allani, R. Chbeir, T. Yeferny and S. Ben Yahia, "Smart Directional Data Aggregation in VANETs," 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), 2018, pp. 63-70, doi: 10.1109/AINA.2018.00022.
30. F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: from concept to implementation," *Abstract—Software-Defined IEEE Communication Surveys & Tutorials*, vol. 16, no. 4, 2014.
31. M. Mousa, A. M. Bahaa-Eldin and M. Sobh, "Software Defined Networking concepts and challenges," 2016 11th International Conference on Computer Engineering & Systems (ICCES), 2016, pp. 79-90, doi: 10.1109/ICCES.2016.7821979.
32. Lantz, Bob, et al. "A Network in a Laptop: Rapid Prototyping for Software-Defined Networks." *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010, p. 19.
33. Pupatwibul, Pakawat, Ameen Banjar, Abdallah Al Sabbagh and Robin Braun. "A Comparative Review: Accurate OpenFlow Simulation Tools for Prototyping." *J. Networks* 10 (2015): 322-327.
34. A. Ephremides, J. E. Wieselthier and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," in

- Proceedings of the IEEE, vol. 75, no. 1, pp. 56-73, Jan. 1987, doi: 10.1109/PROC.1987.13705.
35. Gerla, M. and Tsai, J.T.C. (1995) Multicluster Mobile Multimedia Radio Network. *Wireless Networks*, 1, 255-265.
36. Z. Y. Rawshdeh and S. M. Mahmud, "Toward Strongly Connected Clustering Structure in Vehicular Ad Hoc Networks," 2009 IEEE 70th Vehicular Technology Conference Fall, 2009, pp. 1-5, doi: 10.1109/VETECONF.2009.5378799.
37. Ghada H. Alshli, Ahmed Khattab, Yasmine A. Fahmy, and Nathalie Mitton. 2019. Double-Head Clustering for Resilient VANETs. *Wirel. Commun. Mob. Comput.* 2019 (2019). DOI:<https://doi.org/10.1155/2019/2917238>
38. Madhulatha, T. Soni. "An Overview on Clustering Methods." *ArXiv abs/1205.1117* (2012): n. pag.
39. Z. Ismail and R. Hassan, "Effects of Packet Size on AODV Routing Protocol Implementation in Homogeneous and Heterogeneous MANET," 2011 Third International Conference on Computational Intelligence, Modelling & Simulation, 2011, pp. 351-356, doi: 10.1109/CIMSim.2011.70.
40. S. G. Kawale and V. Bondre, "Performance evaluation of different density AOMDV routing protocol for VANET," 2015 International Conference on Communications and Signal Processing (ICCSP), 2015, pp. 1123-1127, doi: 10.1109/ICCSP.2015.7322678.
41. J. Toutouh, J. Garcia-Nieto and E. Alba, "Intelligent OLSR Routing Protocol Optimization for VANETs," in *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1884-1894, May 2012, doi: 10.1109/TVT.2012.2188552.
42. A. Awang, K. Husain, N. Kamel and S. Aïssa, "Routing in Vehicular Ad-hoc Networks: A Survey on Single- and Cross-Layer Design Techniques, and Perspectives," in *IEEE Access*, vol. 5, pp. 9497-9517, 2017, doi: 10.1109/ACCESS.2017.2692240.
43. S. Vodopivec, J. Bešter and A. Kos, "A survey on clustering algorithms for vehicular ad-hoc networks," 2012 35th International Conference on

- Telecommunications and Signal Processing (TSP), 2012, pp. 52-56, doi: 10.1109/TSP.2012.6256251.
44. R. Chai, Xianlei Ge and Q. Chen, "Adaptive K-Harmonic Means clustering algorithm for VANETs," 2014 14th International Symposium on Communications and Information Technologies (ISCIT), 2014, pp. 233-237, doi: 10.1109/ISCIT.2014.7011907.
45. P. Bhosale and A. Vidhate, "An agglomerative approach to elect the cluster head in VANET," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPE), 2016, pp. 1340-1344, doi: 10.1109/SCOPE.2016.7955658.
46. M. Fahad, F. Aadil and S. Khan, "Optimization of vehicular node clustering process using evolutionary algorithms," 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI), 2017, pp. 1-5, doi: 10.1109/UIC-ATC.2017.8397432.
47. Mamdani, "Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis," in IEEE Transactions on Computers, vol. C-26, no. 12, pp. 1182-1191, Dec. 1977, doi: 10.1109/TC.1977.1674779.
48. Q. Wang, D. Lin, P. Yang and Z. Zhang, "A Fuzzy-logic Based Energy-efficient Clustering Algorithm for the Wireless Sensor Networks," 2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2018, pp. 1-6, doi: 10.23919/SOFTCOM.2018.8555848.
49. C. Wu, S. Ohzahata and T. Kato, "Flexible, Portable, and Practicable Solution for Routing in VANETs: A Fuzzy Constraint Q-Learning Approach," in IEEE Transactions on Vehicular Technology, vol. 62, no. 9, pp. 4251-4263, Nov. 2013, doi: 10.1109/TVT.2013.2273945.

50. C. Cooper, D. Franklin, M. Ros, F. Safaei and M. Abolhasan, "A Comparative Survey of VANET Clustering Techniques," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 657-681, Firstquarter 2017, doi: 10.1109/COMST.2016.2611524.
51. A. Hassan, M. H. Ahmed and M. A. Rahman, "Adaptive beaconing system based on fuzzy logic approach for vehicular network," 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2013, pp. 2581-2585, doi: 10.1109/PIMRC.2013.6666582.
52. Hamid Ali, Waseem Shahzad, and Farrukh Aslam Khan. 2012. Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization. *Appl. Soft Comput.* 12, 7 (July, 2012), 1913–1928. DOI:<https://doi.org/10.1016/j.asoc.2011.05.036>
53. Ji, Chunlin, Yangyang Zhang, Shixing Gao, Ping Yuan and Zhe Li. "Particle swarm optimization for mobile ad hoc networks clustering." *IEEE International Conference on Networking, Sensing and Control*, 2004 1 (2004): 372-375 Vol.1.

ДОДАТОК 1

Частина програмного коду

```
package com.vosiievska.dbscanclustering.service.impl;

import com.vosiievska.dbscanclustering.entity.Cluster;
import com.vosiievska.dbscanclustering.entity.Vehicle;
import com.vosiievska.dbscanclustering.service.NetworkGraphService;
import com.vosiievska.dbscanclustering.service.fuzzy.c.means.clustering.FuzzyClustering;
import com.vosiievska.dbscanclustering.service.response.DBSCANResponse;
import com.vosiievska.dbscanclustering.utils.ClusterColorsUtils;
import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;
import lombok.extern.slf4j.Slf4j;
import org.apache.commons.lang3.StringUtils;
import org.graphstream.graph.Node;
import org.graphstream.graph.implementations.DefaultGraph;

@Slf4j

public class NetworkGraphServiceImpl implements NetworkGraphService {

    @Override
    public List<Vehicle> generateVehicles(int clustersAmount, int vehiclesNumber) {
        log.info("Generate network. Number of clusters: {}; Data set size: {}.", clustersAmount,
vehiclesNumber);

        FuzzyClustering clusterer = new FuzzyClustering();

        List<Vehicle> vehicles = clusterer.createRandomData(vehiclesNumber, 2, 1, 100,
clustersAmount);

        log.info("Finish network generation.");
        log.info("Display generated network graph.");
        DefaultGraph graph = new DefaultGraph("Fuzzy C-means");
        graph.display();
    }
}
```

```

for (int i = 0; i < vehicles.size(); i++) {
    Node node = graph.addNode(String.valueOf(vehicles.get(i).getId()));
    node.addAttribute("ui.frozen");
    node.addAttribute("ui.style", "text-size: 20px;");
    node.addAttribute("x", vehicles.get(i).getX());
    node.addAttribute("y", vehicles.get(i).getY());
    node.addAttribute("ui.label", i + 1);
}
return vehicles;
}

@Override
public DBSCANResponse computeClusters(List<Vehicle> vehicles, double eps, int minPts) {
    log.info("Start DBSCAN clustering. Vehicles number: {}; eps: {}; minPts: {}.",
vehicles.size(), eps, minPts);

    DBSCANClustererImpl<Vehicle> dbscanClusterer = new DBSCANClustererImpl<>(eps,
minPts);

    List<Cluster<Vehicle>> clusters = dbscanClusterer.cluster(vehicles);

    // get noise
    List<Vehicle> partOfCluster = clusters.stream()
        .flatMap(c -> c.getVehicles().stream())
        .collect(Collectors.toList());

    List<Vehicle> noise = vehicles.stream()
        .filter(v -> !partOfCluster.contains(v))
        .collect(Collectors.toList());

    log.info("Finish DBSCAN clustering. Clusters number: {}; noise vehicles: {}; total
clusters: {}.",
        clusters.size(), noise.size(), clusters.size() + noise.size());
}

```

```

    log.info("Clusters: {}",
clusters.stream().map(Cluster::toString).collect(Collectors.joining()));

    log.info("Noise: {}", noise.stream().map(Vehicle::toString).collect(Collectors.joining(""));

log.info("Display DBSCAN clustering result graph.");
DefaultGraph graph = new DefaultGraph("DBSCAN");
graph.display();

for (Cluster<Vehicle> cluster : clusters) {
    for (Vehicle vehicle : cluster.getVehicles()) {
        Node node = graph.addNode(Integer.toString(vehicle.getId()));
        node.addAttribute("ui.frozen");
        node.setAttribute("ui.class", "cluster " + cluster.getId());
        node.setAttribute("ui.style", cluster.getNodeStylistic());
        node.setAttribute("x", vehicle.getX());
        node.setAttribute("y", vehicle.getY());
        node.setAttribute("ui.label", vehicle.getId());
    }
}

for (int i = 0; i < noise.size(); i++) {
    Node node = graph.addNode(Integer.toString(noise.get(i).getId()));
    node.addAttribute("ui.frozen");
    node.setAttribute("ui.style", "text-size: 20px;");
    node.setAttribute("x", noise.get(i).getX());
    node.setAttribute("y", noise.get(i).getY());
    node.setAttribute("ui.label", "NOISE-" + noise.get(i).getId());
}

return DBSCANResponse.builder()
    .clustersNumber(clusters.size())
    .partOfCluster(partOfCluster)
    .noise(noise)
    .build();

```

```

}

@Override
public List<Cluster<Vehicle>> computeClusterHeads(DBSCANResponse dbscanResponse,
int iterations) {
    log.info("Start Fuzzy clustering. Iterations { }.", iterations);

    FuzzyClustering clusterer = new FuzzyClustering();
    clusterer.run(dbscanResponse.getClustersNumber(), iterations,
dbscanResponse.getPartOfCluster());
    ArrayList<ArrayList<Float>> clusterCenters = clusterer.getClusterCenters();

    log.info("ClusterHeads: { }.", clusterCentersToString(clusterCenters));

    log.info("Form clusters according to the membership point.");
    List<Cluster<Vehicle>> clusters = new ArrayList<>();
    ClusterColorsUtils colorsUtils = new ClusterColorsUtils();
    for (int i = 0; i < clusterCenters.size(); i++) {
        Cluster<Vehicle> cluster = new Cluster<>(i + 1, new
Double[] { Double.valueOf(clusterCenters.get(i).get(0)),
        Double.valueOf(clusterCenters.get(i).get(1)) }, colorsUtils.getNextColor());
        clusters.add(cluster);
    }

    for (int i = 0; i < dbscanResponse.getPartOfCluster().size(); i++) {
        log.debug("Get max probability for vehicle.");
        int max = 0;
        for (int j = 0; j < clusterer.getU()[i].length; j++) {
            max = clusterer.getU()[i][j] > clusterer.getU()[i][max] ? j : max;
        }

        log.debug("Set vehicle to cluster with max probability.");
        clusters.get(max).addPoint(dbscanResponse.getPartOfCluster().get(i));
    }
}

```

```

log.debug("Create separate cluster for noise.");
for (int i = 0; i < dbscanResponse.getNoise().size(); i++) {
    Cluster<Vehicle> cluster = new Cluster<>(clusters.size() + i,
        new Double[]{dbscanResponse.getNoise().get(i).getX(),
dbscanResponse.getNoise().get(i).getY()},
        colorsUtils.getNextColor());
    cluster.addPoint(dbscanResponse.getNoise().get(i));
    clusters.add(cluster);
}

log.info("Finish fuzzy clustering. Expected clusters number: {} ; cluster centers number: "
    + "{} ; noise clusters: {}, total clusters: {}", dbscanResponse.getClustersNumber(),
clusterCenters.size(),
    dbscanResponse.getNoise().size(), clusters.size());

log.info("Clusters: {} ",
clusters.stream().map(Cluster::toString).collect(Collectors.joining()));

log.info("Display Fuzzy c-means clustering result graph.");
DefaultGraph graph = new DefaultGraph("Fuzzy C-means");
graph.display();

log.debug("Print cluster heads for parts of cluster.");
for (int i = 0; i < clusterCenters.size(); i++) {
    Node node = graph.addNode(Integer.toString(i + 100));
    node.addAttribute("ui.frozen");
    node.setAttribute("ui.style", "fill-color: rgb(255, 0, 0);");
    node.setAttribute("x", clusterCenters.get(i).get(0));
    node.setAttribute("y", clusterCenters.get(i).get(1));
    node.setAttribute("ui.label", "CH-" + (i + 1));
}

log.debug("Print cluster heads for noise.");

```

```

for (int i = 0; i < dbscanResponse.getNoise().size(); i++) {
    Node node = graph.addNode(Integer.toString(clusterCenters.size() + i + 100));
    node.addAttribute("ui.frozen");
    node.setAttribute("ui.style", "fill-color: rgb(255, 0, 0);");
    node.setAttribute("x", dbscanResponse.getNoise().get(i).getX());
    node.setAttribute("y", dbscanResponse.getNoise().get(i).getY());
    node.setAttribute("ui.label", "CH-" + (clusterCenters.size() + i + 1));
}

```

```

log.debug("Print cluster members.");

```

```

for (Cluster cluster : clusters) {
    for (Vehicle vehicle1 : (List<Vehicle>) cluster.getVehicles()) {
        Node node = graph.addNode(Integer.toString(vehicle1.getId()));
        node.addAttribute("ui.frozen");
        node.setAttribute("ui.style", cluster.getNodeStylistic());
        node.setAttribute("x", vehicle1.getX());
        node.setAttribute("y", vehicle1.getY());

        if (dbscanResponse.getNoise().contains(vehicle1)) {
            node.setAttribute("ui.label", "NOISE-" + vehicle1.getId());
        } else {
            node.setAttribute("ui.label", vehicle1.getId());
        }
    }
}

```

```

log.debug("Set vehicle neighbors.");

```

```

for (int i = 0; i < clusters.size(); i++) {
    for (int j = 0; j < clusters.get(i).getVehicles().size(); j++) {

        int finalI = i;
        int finalJ = j;
        clusters.get(i).getVehicles().get(j).setNeighbors(

```

```

        clusters.get(i).getVehicles().stream()
            .filter(v -> v.getId() != clusters.get(finalI).getVehicles().get(finalJ).getId())
            .collect(Collectors.toSet())
    );
}
}
return clusters;
}

public String clusterCentersToString(ArrayList<ArrayList<Float>> inpData) {
    String result = "";
    for (int i = 0; i < inpData.size(); i++) {
        String line = StringUtils.center("\n" + i + ": ", 3);
        for (int j = 0; j < inpData.get(i).size(); j++) {
            if (j == inpData.get(i).size() - 1) {
                line += StringUtils.center(Float.toString(inpData.get(i).get(j)), 20);
            } else {
                line += StringUtils.center(inpData.get(i).get(j) + ", ", 20);
            }
        }
        result += line;
    }
    return result;
}

package com.vosiievska.dbscanclustering.service.impl;

import static com.vosiievska.dbscanclustering.utils.Utils.getDecimalFormater;

import com.vosiievska.dbscanclustering.entity.Cluster;
import com.vosiievska.dbscanclustering.entity.Vehicle;
import com.vosiievska.dbscanclustering.entity.VehicleFunction;
import com.vosiievska.dbscanclustering.service.VehicleService;

```

```

import java.text.DecimalFormat;
import lombok.extern.slf4j.Slf4j;
import org.apache.commons.lang3.StringUtils;
import org.springframework.stereotype.Service;

@Slf4j
@Service
public class VehicleServiceImpl implements VehicleService {

    private static double THRESHOLD_CONSTANT = 1.5;

    @Override
    public double getRelativeAverageSpeed(Vehicle vehicle, Cluster<Vehicle> cluster) {
        log.debug("Vehicle {}. Calculate relative average speed.", vehicle.getId());
        return cluster.getStandardDeviation() == 0 ? 0
            : (vehicle.getSpeed() - cluster.getMean()) / cluster.getStandardDeviation();
    }

    @Override
    public double getNeighborhoodDegree(Vehicle vehicle, Cluster<Vehicle> cluster) {
        log.debug("Vehicle {}. Calculate neighborhood degree.", vehicle.getId());
        double threshold = THRESHOLD_CONSTANT * cluster.getStandardDeviation();
        long count = vehicle.getNeighbors().stream()
            .filter(n -> Math.abs(vehicle.getSpeed() - n.getSpeed()) < threshold)
            .count();
        return count / (double) cluster.getVehicles().size();
    }

    @Override
    public double getLinkQuality(Vehicle vehicle, Cluster<Vehicle> cluster) {
        log.debug("Vehicle {}. Calculate RSU link quality.", vehicle.getId());
        double vehicleMessagePower = vehicle.getP_th() + vehicle.getG() - vehicle.getL();
        double rlq =

```



```

        vehicleMessagePower / (cluster.getMaxReceivedMessagePower() -
cluster.getMinReceivedMessagePower());
        return Math.min(rlq, 1);
    }

    @Override
    public double getBestAgentLocation(Vehicle vehicle, Cluster<Vehicle> cluster) {
        log.debug("Vehicle { }. Calculate best agent location.", vehicle.getId());
        return Math.sqrt(Math.pow(cluster.getClusterCenter()[1] - vehicle.getY(), 2) + Math
            .pow(cluster.getClusterCenter()[0] - vehicle.getX(), 2));
    }

    @Override
    public void getFitFactor(Vehicle vehicle, Cluster<Vehicle> cluster) {
        double ras = getRelativeAverageSpeed(vehicle, cluster);
        double nd = getNeighborhoodDegree(vehicle, cluster);
        double rlq = getLinkQuality(vehicle, cluster);
        double bal = getBestAgentLocation(vehicle, cluster);

        vehicle.setFunction(new VehicleFunction(ras, nd, rlq, bal));

        DecimalFormat df = getDecimalFormater();
        String fitFactor = String.format("%s%s%s%s%s", StringUtils.center("Vehicle " +
vehicle.getId() + "->", 20),
            StringUtils.center("relative average speed = " + df.format(ras) + ";", 40),
            StringUtils.center("neighborhood degree = " + df.format(nd) + ";", 30),
            StringUtils.center("RSU link quality = " + df.format(rlq) + ";", 40),
            StringUtils.center("best agent location = " + df.format(bal) + ";", 30),
            StringUtils.center("Fit Function = " + df.format(vehicle.getFunction().getFitFactor()) +
";", 30));

        log.info(fitFactor);
    }
}

```

```

package com.vosiievska.dbscanclustering.service.impl;

import static com.vosiievska.dbscanclustering.utils.Utils.getDecimalFormatter;

import com.vosiievska.dbscanclustering.entity.Cluster;
import com.vosiievska.dbscanclustering.entity.Vehicle;
import com.vosiievska.dbscanclustering.entity.VehicleFunction;
import com.vosiievska.dbscanclustering.service.VehicleService;
import java.text.DecimalFormat;
import lombok.extern.slf4j.Slf4j;
import org.apache.commons.lang3.StringUtils;
import org.springframework.stereotype.Service;

@Slf4j
@Service
public class VehicleServiceImpl implements VehicleService {

    private static double THRESHOLD_CONSTANT = 1.5;

    @Override
    public double getRelativeAverageSpeed(Vehicle vehicle, Cluster<Vehicle> cluster) {
        log.debug("Vehicle {}. Calculate relative average speed.", vehicle.getId());
        return cluster.getStandardDeviation() == 0 ? 0
            : (vehicle.getSpeed() - cluster.getMean()) / cluster.getStandardDeviation();
    }

    @Override
    public double getNeighborhoodDegree(Vehicle vehicle, Cluster<Vehicle> cluster) {
        log.debug("Vehicle {}. Calculate neighborhood degree.", vehicle.getId());
        double threshold = THRESHOLD_CONSTANT * cluster.getStandardDeviation();
        long count = vehicle.getNeighbors().stream()

```

```

        .filter(n -> Math.abs(vehicle.getSpeed() - n.getSpeed()) < threshold)
        .count();
return count / (double) cluster.getVehicles().size();
}

@Override
public double getLinkQuality(Vehicle vehicle, Cluster<Vehicle> cluster) {
    log.debug("Vehicle {}. Calculate RSU link quality.", vehicle.getId());
    double vehicleMessagePower = vehicle.getP_th() + vehicle.getG() - vehicle.getL();
    double rlq =
        vehicleMessagePower / (cluster.getMaxReceivedMessagePower() -
cluster.getMinReceivedMessagePower());
    return Math.min(rlq, 1);
}

@Override
public double getBestAgentLocation(Vehicle vehicle, Cluster<Vehicle> cluster) {
    log.debug("Vehicle {}. Calculate best agent location.", vehicle.getId());
    return Math.sqrt(Math.pow(cluster.getClusterCenter()[1] - vehicle.getY(), 2) + Math
        .pow(cluster.getClusterCenter()[0] - vehicle.getX(), 2));
}

@Override
public void getFitFactor(Vehicle vehicle, Cluster<Vehicle> cluster) {
    double ras = getRelativeAverageSpeed(vehicle, cluster);
    double nd = getNeighborhoodDegree(vehicle, cluster);
    double rlq = getLinkQuality(vehicle, cluster);
    double bal = getBestAgentLocation(vehicle, cluster);

    vehicle.setFunction(new VehicleFunction(ras, nd, rlq, bal));

    DecimalFormat df = getDecimalFormater();

```

```
String fitFactor = String.format("%s%s%s%s%s%s", StringUtils.center("Vehicle " +
vehicle.getId() + "->", 20),
    StringUtils.center("relative average speed = " + df.format(ras) + ";", 40),
    StringUtils.center("neighborhood degree = " + df.format(nd) + ";", 30),
    StringUtils.center("RSU link quality = " + df.format(rlq) + ";", 40),
    StringUtils.center("best agent location = " + df.format(bal) + ";", 30),
    StringUtils.center("Fit Function = " + df.format(vehicle.getFunction().getFitFactor()) +
";", 30));

    log.info(fitFactor);
}
}
```