

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерна інженерія»

Група: 2БКС-29

# **КВАЛІФІКАЦІЙНА РОБОТА**

**здобувача освіти денної форми навчання**  
**БКС.29.04.000.КРБ**

***ВАСИЛЮКА***  
***ОЛЕКСАНДРА***  
***ВОЛОДИМИРОВИЧА***

**м. Одеса**  
**2025 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерна інженерія»

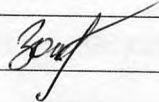
Група: 2БКС-29

## ПОЯСНЮВАЛЬНА ЗАПИСКА

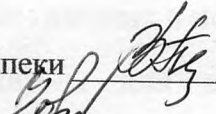
До кваліфікаційної роботи бакалавра на тему: Аналіз ефективності алгоритмів розподілу навантаження до контролерів SDN

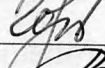
Проектний матеріал складається з пояснювальної записки на 69 сторінках та графічного (презентаційного) матеріалу на 18 аркушах (слайдах)

Виконавець  (Василюк О.В.)

Керівник проекту  (Закроєв Ю.М.)


### Консультанти:

з розділу охорони праці та техніки безпеки  (Чорновол Н.І.)

з нормоконтролю  (Петрашова В.І.)

старший консультант  (Кривченко Ю.В.)

### До захисту допущений

Завідувач кафедри  (Іванова Л.В.)

Завідувач відділення  (Краснокутська К.Г.)

Захист «15» 06 2025 р. Протокол ЕК № 1

Оцінка ЕК 4 (добре) / 75

Секретар ЕК 

# АНОТАЦІЯ

У кваліфікаційній роботі розглянуто та адаптовано алгоритм розподілу навантаження для контролерів програмно-визначуваних мереж (SDN) із застосуванням сучасних методів кластеризації мережевих вузлів. Основою дослідження є математична модель, що враховує локальну щільність зв'язків між маршрутизаторами та мінімальні відстані до вузлів з вищими значеннями щільності, що дозволяє визначити оптимальні центри кластеризації для розміщення контролерів.

В роботі детально розглянуто існуючі підходи до розподілу навантаження (метод  $k$ -середніх, алгоритм РОСО) та проведено їх порівняльний аналіз за показниками швидкості обчислень, затримок розподілу сервісного трафіку та насичення топології контролерами при обмеженнях. Для цілей моделювання створено програмний застосунок на мові Java із використанням бібліотеки JavaFX, що забезпечує візуальне представлення мережевої топології, редагування графів, а також інтерактивну симуляцію роботи обраного алгоритму.

Експериментальні результати показали, що запропонований модернізований алгоритм забезпечує зниження затримок, швидше розгортає контролери та оптимально розподіляє навантаження в порівнянні із традиційними методами, що робить його перспективним для впровадження у сучасні SDN-системи. Отримані результати дозволяють зробити висновок про високу ефективність і практичну доцільність застосування запропонованої методики аналізу ефективності алгоритмів розподілу навантаження для оптимізації роботи мережевої інфраструктури, а також відкривають можливості для подальших досліджень у сфері управління мережами та розвитку технологій SDN.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення Комп'ютерних систем Кафедра Комп'ютерної інженерії  
Спеціальність 123 «Комп'ютерна інженерія»  
Освітньо-професійна програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР

Беркань І.В.

« 28 » 05 20 25р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

здобувачеві освіти Василюку Олександрову Володимировичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Аналіз ефективності алгоритмів розподілу навантаження до контролерів SDN

затверджена наказом по коледжу від 24 11 20 24 р. № 246

2. Термін здачі студентом кваліфікаційної роботи 20.06.25

3. Вихідні дані до роботи 1. Варіанти розташування контролерів у SDN; 2. Модифікований алгоритм має забезпечувати високу ефективність службового трафіку та мінімальні затримки; 3. Провести моделювання SDN-мережі та кластеризації, розподілу навантаження, виконати аналіз результатів моделювання

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

Аналіз архітектури SDN-мереж; Аналіз алгоритмів розподілу навантаження до контролерів SDN; Огляд методів розподілу навантаження до контролерів SDN; Адаптація математичної моделі розподілу навантаження; Реалізація модифікованого алгоритму розподілу навантаження; Розробка діаграми класів та блок-схеми алгоритму застосунку для моделювання роботи SDN; Реалізація застосунку для моделювання роботи SDN

5. Перелік графічного матеріалу (слайдів мультимедійної презентації) Архітектура SDN-мережі; Розподілена модель управління з локальними контролерами SDN; Кластеризація SDN-мережі; Пошкодження окремих ланок у мережі SDN; Кластеризація мережі за алгоритмом на основі щільності; Діаграма класів застосунку для моделювання роботи SDN; БСА застосунку для моделювання роботи SDN; Результати створення графу мережі; Результати розподілу навантаження у імітованій мережі; Ефективність алгоритмів розподілу навантаження в залежності від кількості вузлів мережі SDN

6. Консультанти по кваліфікаційній роботі, із зазначенням розділів, що їх стосуються

Розділ	Консультант	ПІДПИС	
		Завдання видав	Завдання прийняв
Основний розділ	Закросв Ю.М.		
Розділ охорони праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко Ю.В.		

7. Дата видачі завдання 20.04.25

Керівник роботи Закросв Ю.М.

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Вступ. Аналіз технічного завдання	07.06.25	Викон
2.	Огляд методів розподілу навантаження	08.06.25	Викон
3.	Способи вирішення проблеми розміщення контролерів у програмно-визначуваних мережах	05.06.25	Викон
4.	Аналіз оптимального розгортання контролерів	10.06.25	Викон
5.	Розробка математичної моделі процесу кластеризації програмно-визначуваних мереж	11.06.25 12.06.25	Викон
6.	Оцінка обчислювальної складності алгоритму	13.06.25	Викон
7.	Моделювання SDN-мережі	14.06.25	Викон
8.	Вибір і аналіз засобів розробки застосунку	15.06.25	Викон
9.	Розробка програмного забезпечення для моделювання	16.06.25	Викон
10.	Тестування методів розподілу навантаження	17.06.25	Викон
11.	Аналіз результатів тестування	18.06.25	Викон
12.	Розробка питань з охорони праці та техніки безпеки	19.06.25	Викон
13.	Підготовка матеріалів мультимедійної презентації	20.06.25	Викон

Здобувач освіти

(підпис)

Керівник роботи

(підпис)

Формат	Зона	Поз.	Позначення	Назва	Кіл.	Примітка
				<u>Документація</u>		
			БКС 29. 04 000. 00 КРБ	Випускна робота		
A4			БКС 29. 04 000. 00 КРБ ПЗ	Пояснювальна записка	1	

				<b>БКС 29. 04 000. 00 КРБ</b>				
Зм.	Арк.	№ докум.	Підпис	Дата	<b>Аналіз ефективності алгоритмів розподілу навантаження до контролерів SDN</b>	Літ.	Аркуш	Аркушів
Розробив		Василюк О.В.	<i>[Handwritten signature]</i>	20.08		Н Д П	4	69
Перевірив		Закроев Ю.М.	<i>[Handwritten signature]</i>	20.08				
Н. Контр.		Петрашова В.І.	<i>[Handwritten signature]</i>	20.08				
Затверд.		Іванова Л.В.	<i>[Handwritten signature]</i>			<b>ВСП "ОТФК ОНТУ"</b> <b>гр.2БКС-29</b>		

# ЗМІСТ

Вступ.....	7
1 Основний розділ.....	8
1.1 Аналіз архітектури SDN-мереж.....	9
1.2 Аналіз алгоритмів розподілу навантаження до контролерів SDN.....	18
1.2.1 Рішення задачі оптимізації розміщення контролерів у SDN.....	20
1.2.2 Рішення задачі компоновки контролерів у мережі SDN.....	23
1.2.3 Рішення задачі врахування відмов контролерів у мережі SDN.....	25
1.2.4 Рішення задачі врахування пошкодження цілісності мережі SDN..	26
1.2.5 Рішення задачі врахування затримок у мережі SDN.....	27
1.3 Огляд методів розподілу навантаження до контролерів SDN.....	29
1.3.1 Метод розподілу навантаження на основі k-середніх.....	29
1.3.2 Метод розподілу навантаження на основі щільності.....	31
1.3.3 Метод розподілу навантаження за критерієм стійкості до відмов...33	
1.4 Адаптація математичної моделі розподілу навантаження до контролерів SDN.....	34
1.5 Аналіз вимог до реалізації модифікованого алгоритму розподілу навантаження до контролерів SDN.....	36
1.6 Реалізація модифікованого алгоритму розподілу навантаження до контролерів SDN.....	37
1.7 Розробка діаграми класів та блок-схеми алгоритму застосунку для моделювання роботи SDN.....	40
1.8 Реалізація застосунку для моделювання роботи SDN.....	44
1.9 Аналіз результатів моделювання та оцінка ефективності алгоритмів розподілу навантаження.....	47
2 Розділ охорони праці та техніки безпеки.....	52
2.1 Аналіз небезпечних і шкідливих факторів, що впливають на програміста.....	52
2.2 Гігієнічні вимоги до виробничого середовища.....	53
2.2.1 Вимоги до приміщення.....	53

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

2.2.2 Освітлення.....	54
2.2.3 Шум.....	54
2.2.4 Мікроклімат.....	55
2.2.5 Вимоги до організації робочого місця працівника.....	55
2.6 Електробезпека.....	56
2.7 Пожежна безпека.....	56
Висновки.....	57
Перелік використаних інформаційних джерел.....	59
Додаток А. Фрагмент коду мовою Java застосунку для моделювання роботи SDN.....	60
Додаток Б. Слайди мультимедійної презентації.....	64

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

## ВСТУП

Потреба в ефективних та гнучких мережевих рішеннях набуває першорядного значення. Програмно-визначувані мережі (SDN) стали однією з найпопулярніших парадигм у сучасних телекомунікаціях, адже вони дозволяють розділити функціонал мережевого обладнання на керуючий (control plane) та передавальний (data plane) рівні. Завдяки цьому архітектурному підходу зростає можливість централізованого управління, що веде до більшої гнучкості, адаптивності та оптимізації ресурсів мережі. Водночас зростання мережевих потоків, збільшення кількості пристроїв і зростаючі вимоги до безперебійної роботи обумовлюють необхідність удосконалення механізмів розподілу навантаження між контролерами.

Основною проблемою сучасних SDN-мереж є не тільки централізоване управління, але й ефективне забезпечення розподілу мережевого навантаження. Керуючі елементи, або контролери, виконують функції маршрутизації, забезпечення безпеки, моніторингу та аналізу мережевого трафіку. У зв'язку з цим виникає потреба у вирішенні питання рівномірного розподілу обчислювального навантаження між цими контролерами, що є критично важливим для забезпечення високої продуктивності та надійності роботи всієї мережі. Недостатньо ефективний розподіл навантаження може призвести до перевантаження окремих контролерів, зниження швидкості реакції мережі та навіть до виникнення збоїв у роботі мережевої інфраструктури.

Метою даної кваліфікаційної роботи є проведення системного аналізу існуючих алгоритмів розподілу навантаження між контролерами SDN, а також розробка модифікованого алгоритму, який дозволить досягти більш рівномірного й оптимізованого розподілу ресурсів у мережі. Для досягнення цієї мети необхідно спочатку ознайомитись з теоретичними основами програмно-визначуваних мереж, вивчити історію їх розвитку, а також проаналізувати сучасні підходи до керування мережею та розподілу навантаження. Особливу увагу приділено механізмам балансування навантаження, що використовуються у глобальному масштабі, та їх впливу на продуктивність мережі.

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

# 1 ОСНОВНИЙ РОЗДІЛ

У підрозділі 1.1 планується розглянути як класичні, так і сучасні алгоритми розподілу навантаження. Аналіз здійснюватиметься за допомогою порівняльних характеристик, таких як ефективність використання ресурсів, надійність, масштабованість та здатність до адаптації в умовах змінної мережевої ситуації. Проведення експериментальних досліджень із застосуванням симуляторів дозволить не лише порівняти різні алгоритми з теоретичної точки зору, але й отримати практичні дані щодо їх виробничої ефективності. Отримані результати слугуватимуть базою для розробки і модифікації обраного алгоритму, що враховуватиме специфіку конкретних програмно-визначуваних мереж з високою динамікою трафіку.

Актуальність дослідження зумовлена високою важливістю оптимізації мережевих процесів у сучасних умовах. Велика кількість підключених пристроїв, зростання об'єму даних та збільшення вимог до швидкості обробки інформації ставлять перед розробниками завдання створення таких алгоритмів розподілу навантаження, які зможуть адаптуватися до змін у мережевому середовищі практично в режимі реального часу. Очевидна необхідність вирішення даної проблематики відображається і в наукових дослідженнях, де питання балансування навантаження набуває особливої актуальності в контексті забезпечення надійності та високої доступності мережевої інфраструктури.

У роботі будуть розглянуті наступні завдання:

- Вивчення основних понять і принципів функціонування програмно-визначуваних мереж.
- Аналіз існуючих алгоритмів розподілу навантаження, що використовуються в SDN.
- Порівняльна характеристика алгоритмів з точки зору використання обчислювальних ресурсів, швидкості обробки запитів, надійності та масштабованості.
- Побудова експериментальної моделі для перевірки ефективності алгоритмів.

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

- Розробка та модифікація обраного алгоритму для досягнення більш рівномірного розподілу навантаження.

- Оцінка отриманих результатів та формулювання рекомендацій щодо подальшого розвитку алгоритмів розподілу навантаження в SDN.

Емпірична складова дослідження базуватиметься на моделях симуляції мережевих середовищ, які дозволять оцінити поведінку алгоритмів у різних сценаріях. Для цього буде використано як стандартні бібліотеки моделювання, так і спеціалізовані програмні засоби, що забезпечують застосування сучасних методів аналізу даних. Результати експериментальних досліджень допоможуть не лише підтвердити або спростувати ефективність аналізованих алгоритмів, а й визначити потенціал їх модифікації для поліпшення якості управління мережевим трафіком.

Новизна даного дослідження пов'язана з комплексним підходом до аналізу існуючих алгоритмів та розробкою модифікованої схеми розподілу навантаження, що враховує специфіку сучасних SDN-середовищ. У результаті роботи планується запропонувати рішення, яке зможе забезпечити більш оперативну реакцію мережі на змінні умови, що в свою чергу сприятиме оптимізації роботи всієї інфраструктури, підвищенню її стійкості та зниженню витрат на обслуговування. Впровадження такого алгоритму має потенціал для практичного застосування у великих мережевих системах, що використовують програмно-визначувану архітектуру, а також для подальших досліджень у даній галузі.

## 1.1 Аналіз архітектури SDN-мереж

Програмно визначувані мережі (SDN) радикально змінюють традиційні підходи до розгортання та управління мережевими системами. У центрі цієї парадигми стоїть принцип розділення інтелектуального управління і функцій пересилання даних. Сутність такої архітектури полягає в тому, що логіка прийняття рішень винесена в окремий рівень, який керує роботою мережевих пристроїв, що відповідають виключно за обробку мережевого трафіку. Завдяки цьому поділу відбувається не лише оптимізація маршрутизації пакетів, але й

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

відкриваються нові можливості для централізованого адміністрування мережевих ресурсів. Контрольний рівень здійснює управлінські функції, приймаючи рішення з розподілу ресурсів, організації безпеки та адаптації до змін у мережевій ситуації, в той час як пристрої, що складають виконавчий рівень, забезпечують передачу даних відповідно до встановлених політик. Ця модель сприяє більшій гнучкості та дозволяє швидко реагувати на непередбачувані навантаження.

Особливу увагу заслуговує розмежування взаємодії між різними рівнями через стандартизовані інтерфейси. З одного боку, інтерфейси верхнього рівня (northbound API) створюють міст між контролером та застосунковими платформами, що забезпечує можливість розробки комплексних мережевих сервісів. З іншого – нижнього рівня (southbound API) дозволяють безпосередньо управляти апаратними засобами, отримувати інформацію про поточний стан мережі та впроваджувати оперативні зміни. Така модульність надає системі SDN здатність до швидкого масштабування і налаштування політик, відповідно до вимог сучасного динамічного середовища.

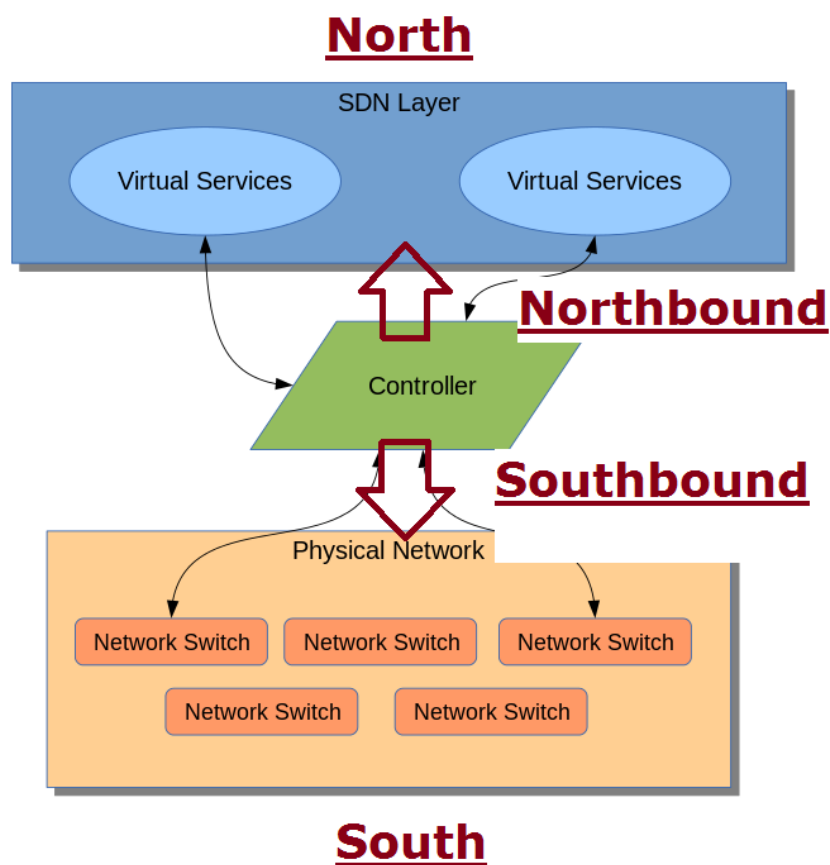


Рисунок 1.1. Схема зв'язків через northbound та southbound API у SDN

Рис.1.1 демонструє, як дані надходять від програмних застосунків до управлінського рівня і, у свою чергу, передаються на мережеві пристрої для виконання заданих операцій.

Застосування централізованого контролю супроводжується певними викликами, зокрема, пов'язаними із забезпеченням високої надійності та зменшенням ризику виникнення "точок відмов". Через концентрацію обчислювальних ресурсів в одному або невеликій групі вузлів можуть з'явитися потенційні вразливості, які негативно впливають на стабільність роботи всієї мережі. Саме тому сучасні рішення спрямовані на розробку ієрархічних або розподілених моделей управління, які дозволяють створювати резервні канали зв'язку та розподіляти навантаження між декількома контролерами. Це забезпечує безперервну роботу мережі навіть за умов виникнення збоїв або перевантажень.

Ще одним аспектом, який варто виділити під час аналізу архітектури SDN, є можливість інтеграції з іншими сучасними технологіями, зокрема з віртуалізацією мережевих ресурсів та хмарними сервісами. Відкриття стандартних API стимулює розробників до створення нових рішень, здатних адаптуватися до мінливих умов експлуатації. Такий підхід дозволяє не лише підвищити оперативність управління, але й забезпечити стандартизацію інтерфейсів, що сприяє легшій взаємодії між пристроями різних виробників.

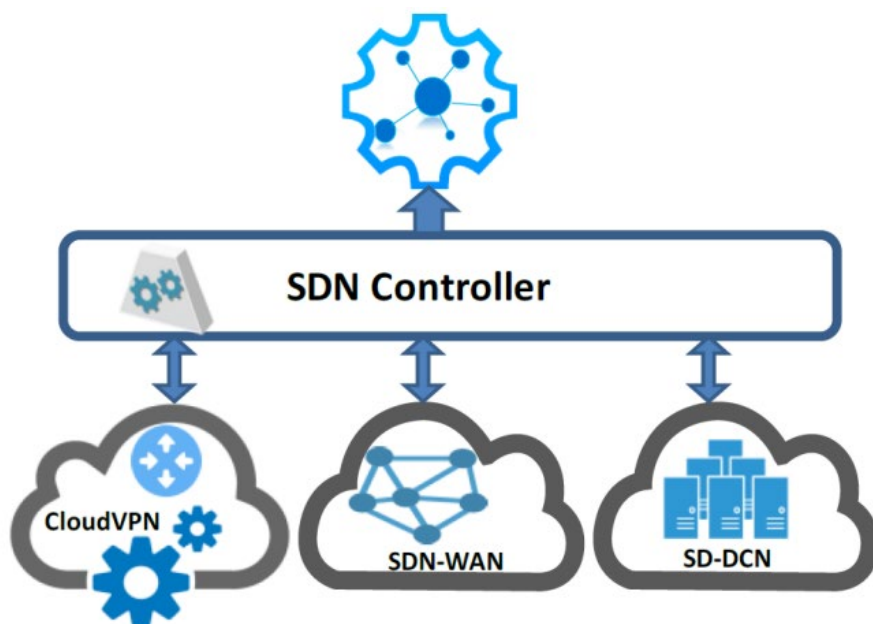


Рисунок 1.2. Інтеграція SDN з додатковими мережевими технологіями

Рис.1.2 демонструє, як SDN-архітектура може бути розширена за рахунок інтеграції з віртуалізованими мережевими платформами та хмарними сервісами, створюючи єдину екосистему управління.

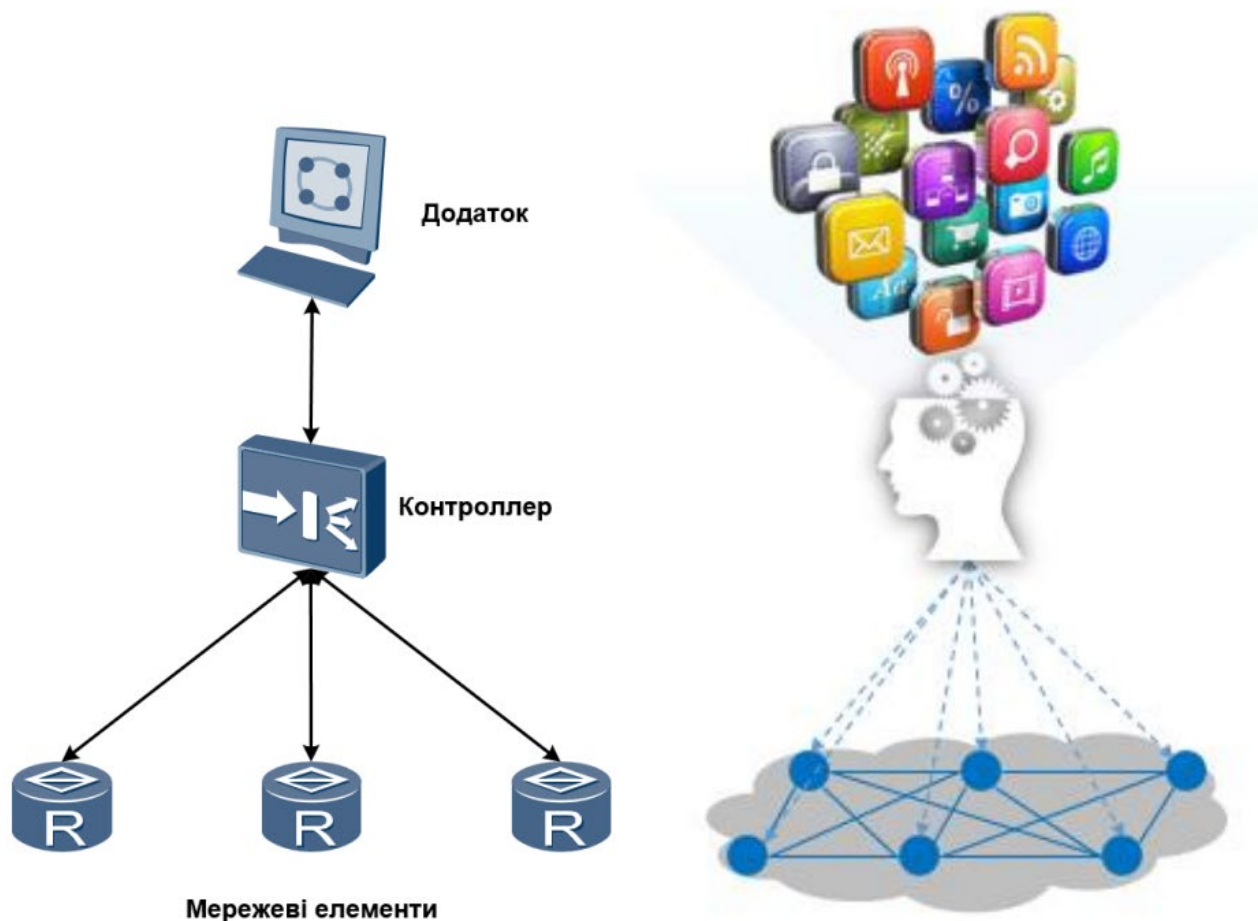


Рисунок 1.3. Архітектура багаторівневої структури SDN-мережі

Діаграма на рис.1.3 демонструє розмежування на три основні рівні: інфраструктурний (апаратне забезпечення та канали комунікації), рівень управління (мережеве ядро з централізованим контролером і API) та прикладний рівень (програмні засоби віртуалізації, безпеки та управлінські додатки).

На першому, інфраструктурному рівні розміщується мережеве обладнання, яке включає пристрої для мінімальної обробки даних та засоби передачі інформації. Це можуть бути комутатори, маршрутизатори, сервери та канали зв'язку, які фізично забезпечують пересилання пакетів. Основна функція цього шару – оперативна обробка пакетів згідно з попередньо запрограмованими інструкціями. Завдяки такій конфігурації мережевий транспортний рівень стає максимально оптимізованим для виконання операцій без інтервенції додаткового

програмного забезпечення, що сприяє високій швидкості передачі даних та мінімальним затримкам.

Другий рівень – це ядро управління мережі. Сюди входить мережеве ядро, побудоване на операційній системі, яке забезпечує взаємодію контролера з апаратною частиною мережі за допомогою стандартизованих API, найпоширенішим з яких є OpenFlow. Контролери отримують дані про стан мережі, аналізують інформацію та направляють команди пристроям для зміни конфігурації або перенаправлення потоку інформації. Централізоване управління дозволяє адміністратору задавати глобальну політику, яку потім реалізують всі мережеві елементи. Особливістю такого підходу є відсутність залежності від конкретного виробника мережевого обладнання, адже API розроблено таким чином, щоб бути універсальним незалежно від апаратури.

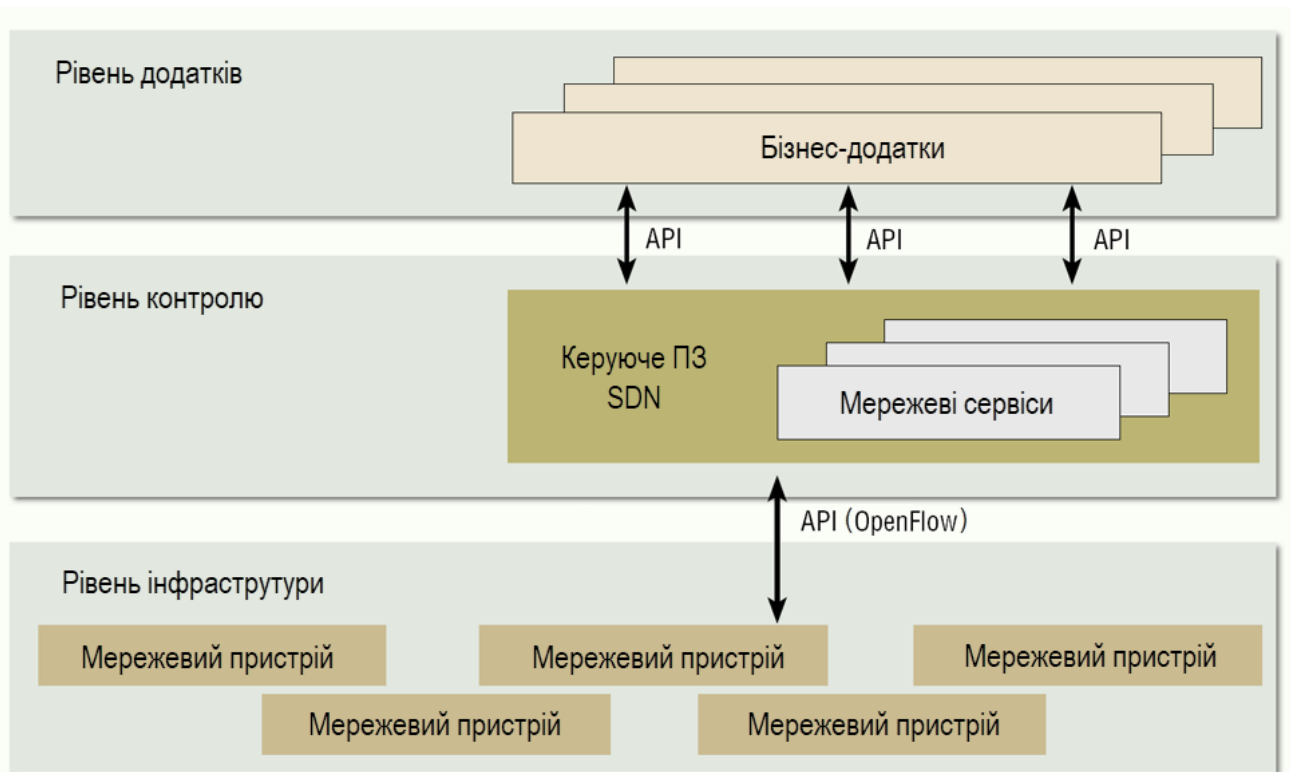


Рисунок 1.4. Схема взаємодії контролера з мережевими пристроями через API

Рис. 1.4. відображає основні потоки управлінських повідомлень, що проходять через уніфікований інтерфейс, з розподілом на команди, оповіщення та синхронізаційні запити. Обмін інформацією відбувається за допомогою трьох типів повідомлень:

- Асиметричні повідомлення: відправляються пристроями до контролера для сповіщення про події, такі як отримання нового даного пакету, зміна стану або видалення записів через закінчення часу життя;
- Синхронізаційні повідомлення: використовуються для контролю якості зв'язку між контролером та мережевими пристроями, встановлюючи параметри з'єднання, вимірюючи затримки та перевіряючи активність;
- Повідомлення контролер-пристрій: дозволяють централізовано змінювати конфігурацію, вносити корективи до таблиць маршрутизації й регулювати процес передачі даних.

За допомогою такої схеми управління створюється сучасна мережа, здатна адаптуватися до змін у трафіку та умовах експлуатації, забезпечуючи сталий рівень якості послуг.

На прикладному рівні розміщують додатки та сервіси, які забезпечують інтеграцію та оптимізацію роботи всієї мережевої системи. Цей шар охоплює служби віртуалізації, системи безпеки, аналітичне та управлінське програмне забезпечення, які створюють гнучке середовище для реалізації індивідуальних політик та маршрутів передачі даних. Віртуалізація дозволяє поділити єдину фізичну мережу на кілька логічних сегментів (віртуальних мереж), кожна з яких має власні правила маршрутизації, політики якості обслуговування і адмініструється окремо. Це сприяє ізоляції різних потоків даних, а також полегшує тестування нових конфігурацій без впливу на основне оперативне середовище.

Спочатку технологія SDN базувалася на єдиному централізованому контролері. Проте з розширенням масштабів мережі виникли складнощі, пов'язані із зростанням затримок та підвищеним навантаженням на окремі вузли. Для вирішення цих проблем було розроблено розподілені моделі управління, де мережеве середовище розподіляється між декількома контролерами. Сучасні архітектури, такі як HyperFlow або ієрархічна модель Kandoo, дозволяють здійснювати локальне управління з одночасною синхронізацією глобального стану мережі. В таких системах кожен регіональний контролер підтримує власну

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

частину мережі, взаємодіючи з іншими через механізми публікації та підписки. Це забезпечує швидке реагування на локальні збої та мінімізує ризик створення «вузьких місць».

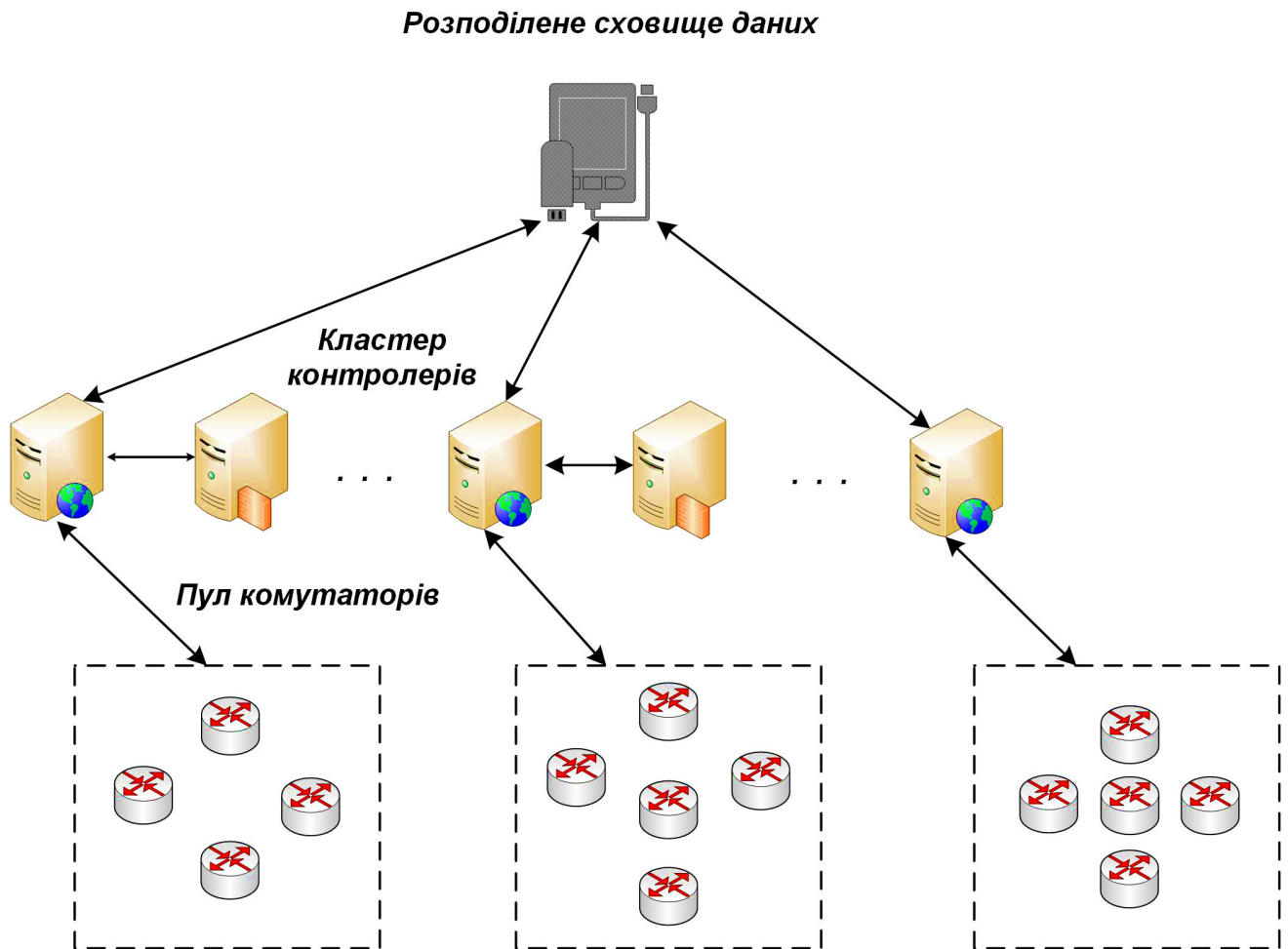


Рисунок 1.5. Розподілена модель управління з локальними контролерами SDN

Схематичне подання на рис.1.5 ілюструє взаємодію між контролерами в різних регіонах мережі, які обмінюються оперативною інформацією для формування єдиного глобального зображення мережевого стану.

Ключові переваги розподіленої архітектури включають зменшення затримок при передачі управлінських інструкцій, підвищення надійності мережі завдяки резервуванню контролерів та загальне покращення масштабованості. У випадку відмови одного вузла, відповідні пристрої можуть швидко переналаштуватися на іншого активного контролера без значних втрат продуктивності. Основою взаємодії між контролером і мережевими пристроями у SDN є протокол OpenFlow. Він встановлює стандартизований механізм обміну

інформацією, що дозволяє відокремити процес прийняття рішень від фактичної передачі даних. OpenFlow підтримує як асинхронну передачу повідомлень, що фіксують події, так і синхронізацію параметрів з'єднання, що дозволяє створити динамічну і адаптивну систему управління. Крім того, можливість керувати таблицями маршрутизації в режимі реального часу дає змогу адміністраторам ефективно реагувати на різкі зміни в мережевому трафіку.

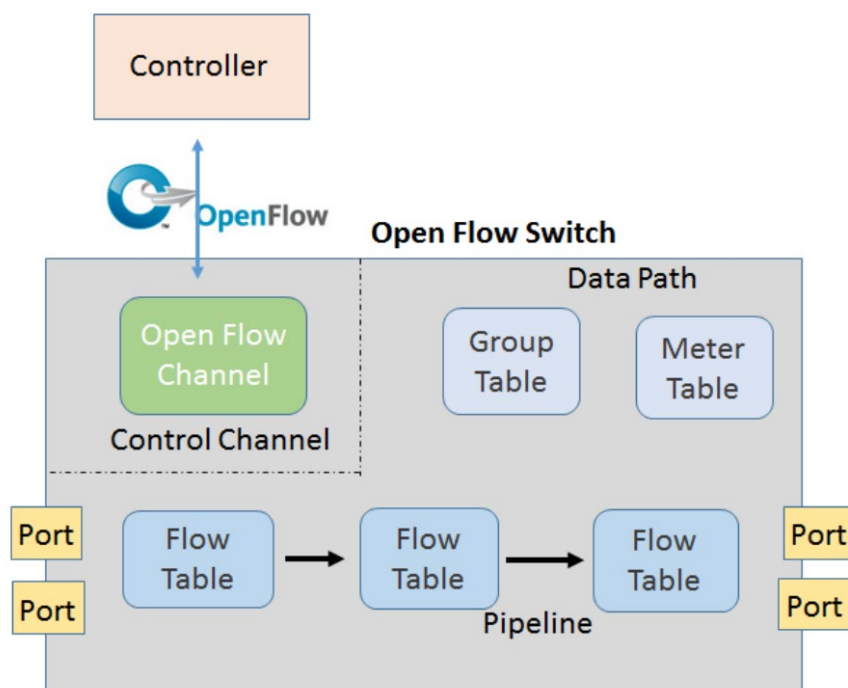


Рисунок 1.6. Схема синхронізації даних при використанні OpenFlow у SDN

Діаграма на рис.1.6 демонструє, як різні типи повідомлень організовано у протоколі OpenFlow дозволяють досягти балансу між локальною швидкістю реагування та централізованим контролем мережі.

Також варто зазначити, що реалізація функціоналу OpenFlow у сучасних мережах забезпечує сумісність з усіма основними виробниками обладнання, що дозволяє інтегрувати різнопланові рішення у єдину систему управління незалежно від бренду чи моделі пристроїв. Це суттєво розширює можливості модернізації існуючих мережевих інфраструктур та підтримує технологічний розвиток у напрямку більш динамічного розподілу ресурсів.

Перші реалізації SDN базувалися на єдиному централізованому контролері, але з ростом масштабів мереж з'явилися проблеми збільшення затримок та навантаження на один вузол. Сучасні підходи, такі як архітектура HyperFlow

(рис.1.7) або ієрархічні моделі на зразок Kandoo, вирішують ці завдання за допомогою розподіленого управління. Розподілені контролери, що обмінюються інформацією через механізми публікації-підписки, формують єдине логічне представлення мережі, забезпечують оперативну синхронізацію та мінімізують ризик виникнення «вузьких місць».

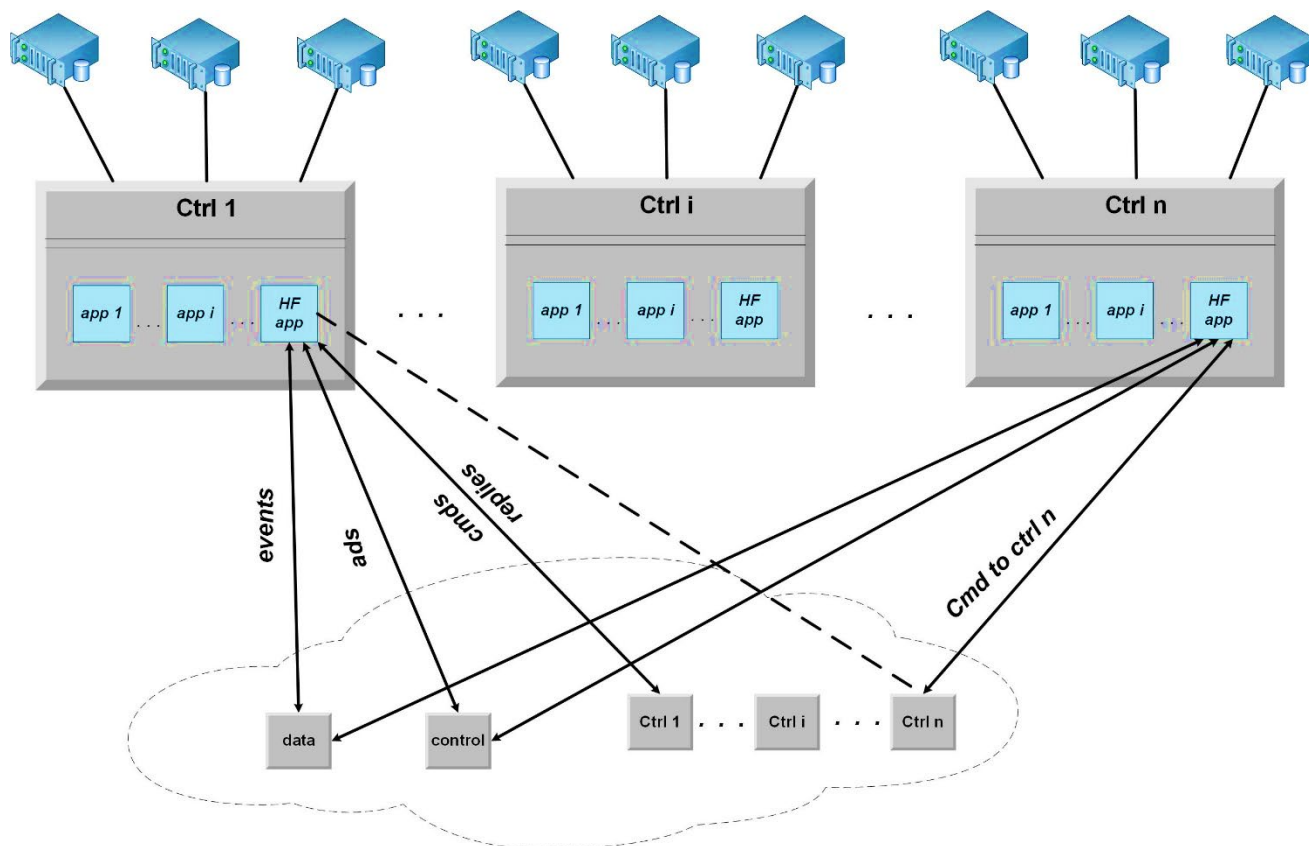


Рисунок 1.7. Архітектура технології HyperFlow

Впровадження технології SDN на практиці володіє численними перевагами, зокрема, можливістю адаптувати конфігурацію мережі до поточних умов експлуатації без фізичних змін в апаратурі. Централізоване управління дозволяє адміністраторам швидко впроваджувати необхідні зміни, використовуючи оновлення програмного забезпечення чи зміну політик керування. Завдяки цьому мережа стає надзвичайно гнучкою та здатною до масштабування від невеликих локальних систем до глобальних інфраструктур зв'язку.

Окрім цього, застосування віртуалізації відкриває нові можливості для розподілу ресурсів. Розподілення фізичних мережевих засобів на кілька логічних сегментів дозволяє зменшити вплив надмірного навантаження на окремі частини

мережі та забезпечити ізоляцію трафіку, що критично важливо для підтримання високих стандартів безпеки та якості обслуговування. В умовах стрімкого зростання обсягу інформації й різноманітних вимог до мережевих сервісів, така модульність дозволяє ефективно управляти ресурсами та зменшувати витрати на їх утримання.

Незважаючи на очевидні переваги, впровадження SDN супроводжується певними викликами. Основною проблемою залишається забезпечення надійності центрального елемента управління в умовах великих мереж. Ризик виникнення «точок відмови» спонукає інженерів розробляти розподілені моделі управління, де кожен компонент несе частину функцій централізованого контролера. Сучасні рішення охоплюють ієрархічне розподілення завдань: локальні контролери взаємодіють один з одним, забезпечуючи загальне стабільне керування, що дозволяє системі швидко відновлювати роботу при виникненні збоїв.

Дослідження у сфері SDN спрямовані на розробку адаптивних алгоритмів балансування навантаження, що враховують як локальні, так і глобальні показники роботи мережі. Використання технологій машинного навчання та аналізу даних дозволяє прогнозувати збої, оптимізувати ресурси та забезпечити високий рівень продуктивності в режимі реального часу. Завдяки цим підходам SDN-інфраструктура може автоматично коригувати свою конфігурацію, пристосовуючись до змін у мережевих потоках та умовах експлуатації.

## **1.2 Аналіз алгоритмів розподілу навантаження до контролерів SDN**

У сучасних SDN-мережах відділення рівня управління від рівня даних породжує комплекс завдань, пов'язаних з оптимальним розміщенням контролерів. Ефективність мережевого управління залежить від правильної розбивки інфраструктури на підмережі, де кожна зона забезпечується окремим контролером, що дозволяє мінімізувати затримки та знизити ризик перевантаження одного вузла.

Першим кроком у розробці алгоритмів розподілу навантаження є визначення оптимальних точок розташування контролерів відповідно до

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

мережевої топології. Рішення базуються на ключових показниках, серед яких латентність відіграє вирішальну роль. Завдяки розбиттю мережі на логічні сегменти, кожен контролер може бути розміщений в тій підмережі, де затримка між ним і кінцевими пристроями є мінімальною. При цьому розглядаються не тільки географічні особливості, а й параметри каналів зв'язку, що дозволяє досягти оптимального балансу між швидкістю обробки управлінських повідомлень і витратами на комунікацію.

Алгоритми компонування контролерів орієнтовані на забезпечення взаємодії між різними регіональними вузлами управління. Сучасні стратегії включають створення кластерів, де контроль над локальними сегментами здійснюється регіональними контролерами, які, у свою чергу, синхронізуються з іншими за допомогою механізмів публікації-підписки. Такий підхід дозволяє підтримувати єдине глобальне уявлення про топологію мережі, зберігаючи при цьому можливість оперативного реагування на локальні зміни та навантаження.

Ще одним важливим аспектом є забезпечення відмовостійкості. Алгоритми розподілу навантаження повинні включати стратегії резервування, що допускають автоматичне перемикання на резервного контролера у випадку збою або перевантаження основного вузла. Також важливо враховувати можливі зміни або пошкодження топології мережі (наприклад, через аварійні ситуації або ремонтні роботи), що впливають на маршрути передачі даних та тимчасово змінюють показники затримки. Адаптивні алгоритми моніторингу дозволяють відстежувати стан мережі та оперативно виконувати перерозподіл навантаження, зберігаючи єдиний глобальний стан системи.

Перевантаження контролерів виникає при зростанні кількості запитів або аномальній інтенсивності мережевого трафіку. Для протидії цій проблемі розробляються динамічні алгоритми розподілу навантаження, які передбачають рівномірний розподіл обчислювальних ресурсів між активними контролерами. Одночасно, модель враховує затримки передачі даних між вузлами через взаємодію контролерів. Зменшення латентності є пріоритетним критерієм, оскільки забезпечує більш оперативну реакцію на зміни умов мережі та дозволяє

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

швидко коригувати маршрутизацію пакетів.

Підхід до розташування контролерів у SDN повинен вирішувати мультифакторну задачу: забезпечення локальної оптимізації через розбиття мережі на підмережі, мінімізацію затримок завдяки оптимальному розміщенню, а також гарантувати відмовостійкість за рахунок резервування та динамічного перерозподілу навантаження. Алгоритми, що враховують ці параметри, дозволяють побудувати систему, здатну оперативно реагувати на зміни в мережевих потоках, підтримувати високий рівень продуктивності та забезпечувати стабільність роботи навіть у випадку аварійних ситуацій.

### 1.2.1 Рішення задачі оптимізації розміщення контролерів у SDN

Однією з ключових проблем сучасних SDN-систем є визначення оптимальних точок розташування мережевих контролерів. Проблема Вебера (або задача про розташування об'єкта) спрямована на мінімізацію сумарної відстані між обраним розташуванням і заданими точками – клієнтами чи споживачами послуг. Такі підходи використовуються для зниження затрат на транспортування, оптимізації логістики або, як у випадку SDN, для мінімізації затримок між контролерами та кінцевими вузлами мережі.

У базовому формулюванні задача оптимального розташування об'єктів містить множину потенційних точок розташування, яка складається з  $L$  можливих позицій, та множину  $D$  точок (або клієнтів), які необхідно обслуговувати. Мета полягає у виборі певної підмножини  $F$  точок із  $L$  для відкриття об'єктів таким чином, щоб мінімізувати загальну суму відстаней від кожної точки обслуговування до найближчого розташованого об'єкта, а також мінімізувати сумарні витрати на утримання або експлуатацію цих об'єктів. Формально, завдання можна сформулювати як мінімізацію наступної цільової функції:

$$\text{minimize } \sum_{(p \in D)} \min_{(q \in S)} d(p, q) + (\text{вартість розміщення}) \quad (1.1)$$

де  $S \subseteq L$  та  $|S| = F$ , а  $d(p, q)$  – відстань між точкою  $p$  і потенційним розташуванням  $q$ .

Також існують варіанти задачі, де метою є мінімізація максимального

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

значення відстані між будь-якою точкою  $p \in D$  та її найближчим розташуванням  $q \in S$  (тобто, задача мінімізації функції  $\max_{p \in D} \{\min_{q \in S} \{d(p, q)\}\}$ ). Обидва варіанти є NP-повними, що вказує на їхню складність і потребу у пошуку апроксимативних або еволюційних рішень для великих мережевих сценаріїв.

Класичні задачі розташування об'єктів, зокрема проблема Вебера, демонструють, як навіть відносно прості задачі оптимізації можуть бути складними з обчислювальної точки зору. Без додаткових припущень щодо властивостей відстаней (тобто, якщо не виконуються такі вимоги, як нерівність трикутника), проблема набуває вигляду неметричної і може бути апроксимована з фактором  $O(\log n)$ . Якщо ж припустити метричність відстаней, тобто, що відстані задовольняють властивості симетрії та трикуткової нерівності, задача набуває більш структурованого вигляду, що дозволяє застосовувати спеціалізовані алгоритми для її вирішення.

Для вирішення задачі оптимального розташування в SDN використовують як точні, так і наближенні методи:

- Цілочисельне лінійне програмування (ЦЛП) дозволяє отримати глобально оптимальне рішення для невеликих мереж, але його застосування часто обмежене через експоненційну складність при великій кількості потенційних точок;
- Методи динамічного програмування розбивають загальну задачу на менші підзадачі та поступово будують оптимальне рішення;
- Еволюційні алгоритми (зокрема, генетичні алгоритми) та методи мурашиних колоній дозволяють знаходити прийнятні, хоча й не завжди глобально оптимальні, рішення для дуже великих задач з численними параметрами.

Інтеграція адаптивних модулів моніторингу в алгоритмічну базу дозволяє враховувати динамічний характер мережевого трафіку, змінювати маршрутизацію та оптимізувати розташування контролерів у режимі реального часу. Таким чином, підхід до оптимізації в SDN ґрунтується не лише на початковій постановці задачі, але й на безперервній адаптації до змін у топології

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

мережі, завантаженні мережевих вузлів та поточних вимогах до якості обслуговування.

У контексті програмно-визначуваних мереж оптимальне розміщення контролерів стає вирішальним фактором для забезпечення мінімальної затримки між командами управління та пристроями передачі даних. Розбивши мережу на підмережі, де кожна з них обслуговується окремим контролером, можна значно знизити затримки зв'язку, покращити відмовостійкість за рахунок резервування, а також зменшити витрати на передачу даних. Завдяки використанню гібридних алгоритмічних рішень, які комбінують точні підходи з наближеними методами, дослідники досягають успішного балансу між оптимізацією задовільних значень цільових функцій та обчислювальною ефективністю. Діаграма на рис.1.8 показує, як розбиття мережі на підмережі з мінімальними середніми затримками дозволяє вибрати розташування контролерів таким чином, щоб зменшити як витрати на комунікацію, так і ризик перевантаження окремих вузлів.

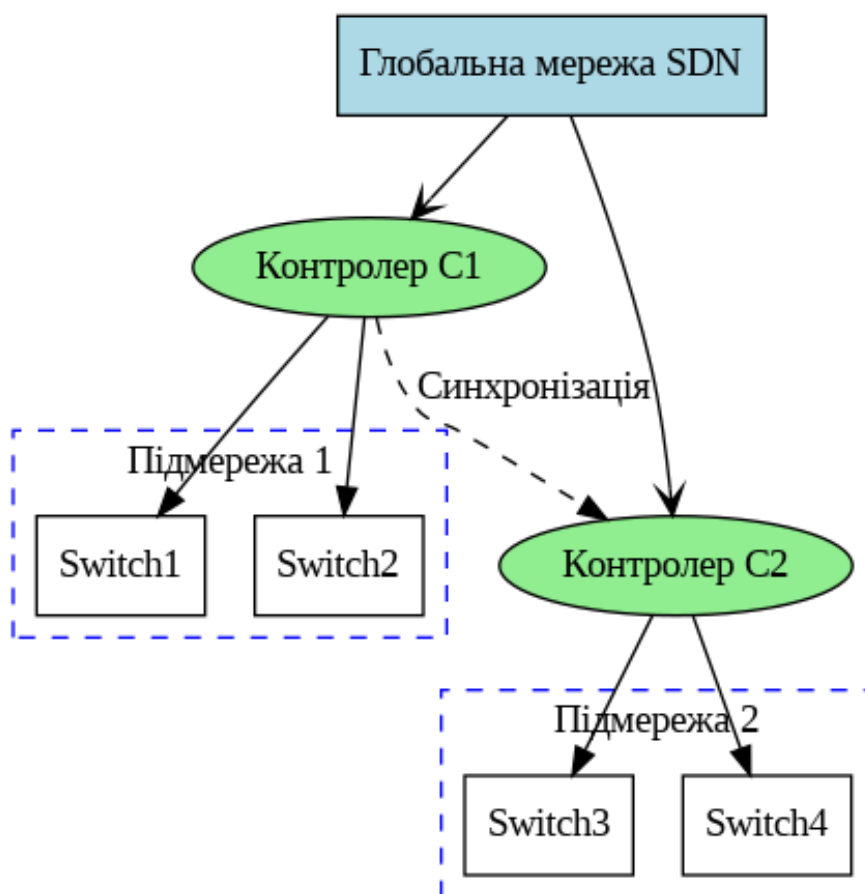


Рисунок 1.8. Принцип оптимального розташування контролерів у SDN

Зм.	Арк.	№ докум.	Підпис	Дата

БКС 29. 04 000. 00 КРБ ПЗ

Арк.

22

## 1.2.2 Рішення задачі компоновки контролерів у мережі SDN

У сучасних SDN системах, окрім оптимізації розташування окремих контролерів, важливо ефективно організувати їх компонування в єдину систему управління. Компонування (кластеризація) контролерів дозволяє узгодити дії локальних контролерів, що обслуговують окремі підмережі, і забезпечує їх синхронізацію через обмін інформацією для підтримки глобальної цілісності мережі.

Основна ідея цього підходу полягає у поділі мережі на логічні сегменти, кожен з яких має свій локальний контролер. Розбиття на підмережі є обґрунтованим через відокремлення рівня управління та рівня передачі даних. Для кожного сегмента вибирається оптимальне розташування контролера, яке дозволяє зменшити затримку між ним та кінцевими пристроями, забезпечуючи оперативну реакцію на локальні події. Компонування контролерів здійснюється із застосуванням наступних підходів:

- Локальна обслуговуюча одиниця: Кожна підмережа (кластер) містить власний контролер, який відповідає за швидку обробку запитів і управління внутрішнім трафіком. Локальні контролери забезпечують адаптивність до змін у внутрішній топології та знижують загальну затримку;
- Глобальна координація: Для синхронізації станів між підмережами місцевих контролерів встановлюється зв'язок між ними. Обмін інформацією (часто через механізми публікації/підписки або через спеціалізовані протоколи синхронізації) дозволяє отримати єдину глобальну картину мережі, що необхідно для прийняття рішень у випадках відмов чи перевантажень;
- Резервування і відмовостійкість: Компонування контролерів сприяє побудові системи з надлишковістю – у разі відмови одного з контролерів відповідні підмережі можуть швидко переключитися на резервні вузли, що мінімізує вплив на загальну продуктивність мережі.

Приклад кластеризації мережі в SDN ілюструє, як мережа розбивається на окремі підмережі (позначено пунктирними лініями), де кожна підмережа має свій

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

локальний контролер. Ці контролери, у свою чергу, координуються через глобальний управляючий шар, що дозволяє забезпечити балансування навантаження, синхронізацію оновлень і відновлення роботи у разі виникнення збоїв. На рис. 1.9 показано розподілення мережі на три підмережі з власними контролерами, які синхронізуються між собою через глобальний управляючий шар.

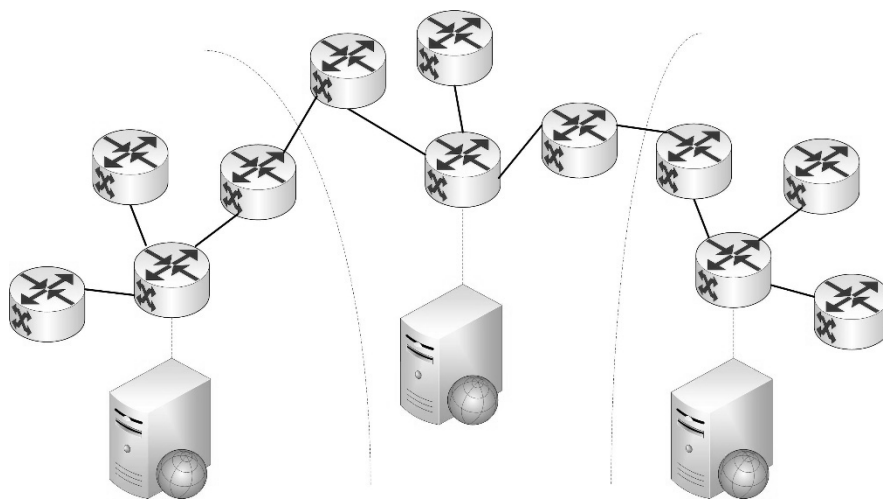


Рисунок 1.9. Кластеризація SDN-мережі на три підмережі

### 1.2.3 Рішення задачі врахування відмов контролерів у мережі SDN

У сучасних SDN-системах відмова контролерів є критичним фактором, оскільки керування мережею концентрується в невеликій кількості центральних елементів. Якщо у мережі, що складається з п'яти контролерів, чотири виходять з ладу, то управління повністю покладається на останній функціонуючий контролер. Такий сценарій може призвести до значного зростання затримок між мережевими вузлами та центральним управляючим елементом, оскільки вузли, розташовані далі, повинні чекати довше на надходження керівних повідомлень.

Для ілюстрації цього явища розглядається приклад (рис.1.10), де з п'яти контролерів лише один (Контролер С1) функціонує, а інші (С2, С3, С4, С5) позначені як “зупинені” – їх стан відображається за допомогою блакитних хрестиків. Мережа складається приблизно з 40 вузлів, що розташовані у вигляді сітки. Відстані та відповідні затримки між вузлами та останнім працездатним контролером візуалізовано за допомогою кольорового кодування:

- Рожевий колір позначає затримку, що дорівнює нулю (вузли, які знаходяться безпосередньо біля активного контролера);
- Синій – визначає затримку, що складає п'ятдесят відсотків від діаметра мережі (помірна затримка);
- Блакитний (skyblue) – відображає максимальну затримку, рівну сто відсотків діаметра сітки (вузли, розташовані на межах мережі).

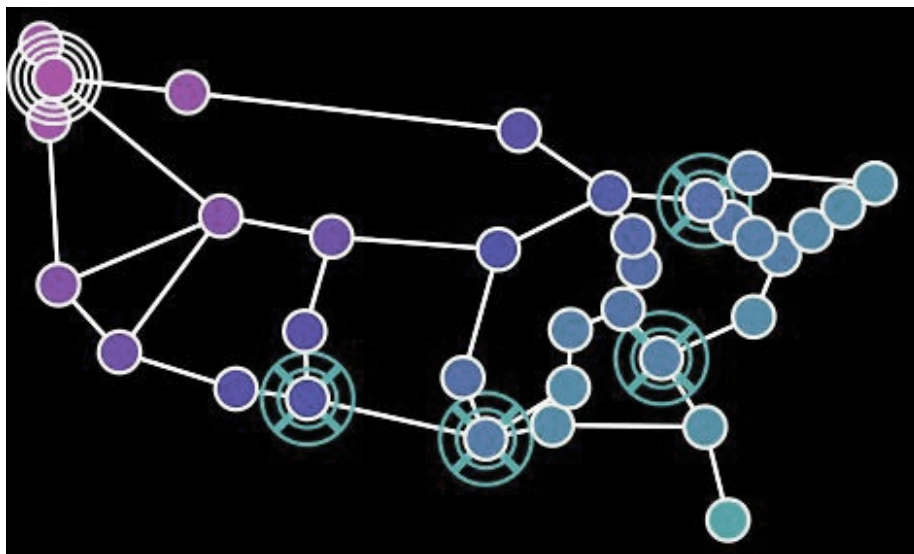


Рисунок 1.10. Зупинка чотирьох з п'яти з контролерів мережі SDN

Такий підхід дозволяє виявити «гарячі точки» – ділянки, де збільшені затримки можуть негативно впливати на оперативність управління. Результати такого аналізу слугують основою для розробки адаптивних алгоритмів перемикавання на резервні контролери, переналаштування топології чи інших заходів щодо відновлення цілісності роботи мережевої системи.

#### **1.2.4 Рішення задачі врахування пошкодження цілісності мережі SDN**

У ситуації, коли відбувається фізичне пошкодження мережевих компонентів, таких як вузли або з'єднання, це зумовлює зміну топології мережі, що має набагато серйозніший вплив на її стабільність, ніж відмова окремих контролерів. Фізичне пошкодження призводить до зміни найкоротших маршрутів між вузлами, що може вплинути на затримки передачі даних та спричинити перепризначення вузлів до інших контролерів. Крім того, під час аварійних ситуацій існує ризик того, що певна група вузлів повністю ізолюється від управлінського впливу, тобто ці вузли, хоч і залишаються працездатними для передачі даних, більше не отримують керівних інструкцій від контролерів. Такий сценарій може мати критичні наслідки для роботи мережі, адже функціонування мережевих сервісів залежить від постійного контролю за станом системи. При цьому, навіть якщо об'єкти взаємопов'язані між собою, відсутність доступу до управління позначає, що мережа з певних ділянок опиняється поза зоною дії централізованої логіки, що у найгіршому випадку може призвести до повного зупинення відповідних сервісів.

Для вирішення проблеми врахування пошкодження цілісності використовуються алгоритмічні підходи, що дозволяють автоматично виявляти втрату зв'язності в мережі та ініціювати відповідні заходи з відновлення з'єднань або переналаштування топологічної структури. Такі алгоритми спираються на аналіз зміни найкоротших маршрутів, що виникають внаслідок видалення певних вузлів або з'єднань, і дозволяють виявити «гарячі точки», тобто ділянки мережі, де підвищені затримки або повна втрата зв'язку з керуючим контролером є критичними. Оцінка впливу пошкоджень здійснюється за допомогою моделювання затримок, де для вузлів, що знаходяться безпосередньо біля

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

останнього функціонуючого контролера, встановлюється нульова затримка, для вузлів середнього розташування – середній рівень затримок, а для вузлів на периферії «діаметру» мережі – максимальна затримка (рис.1.11). Такий підхід дозволяє точно визначати, які ділянки мережі потребують негайного втручання, а також сприяє розробці стратегій забезпечення резервування зв'язків або активації додаткових ресурсів для відновлення керування.

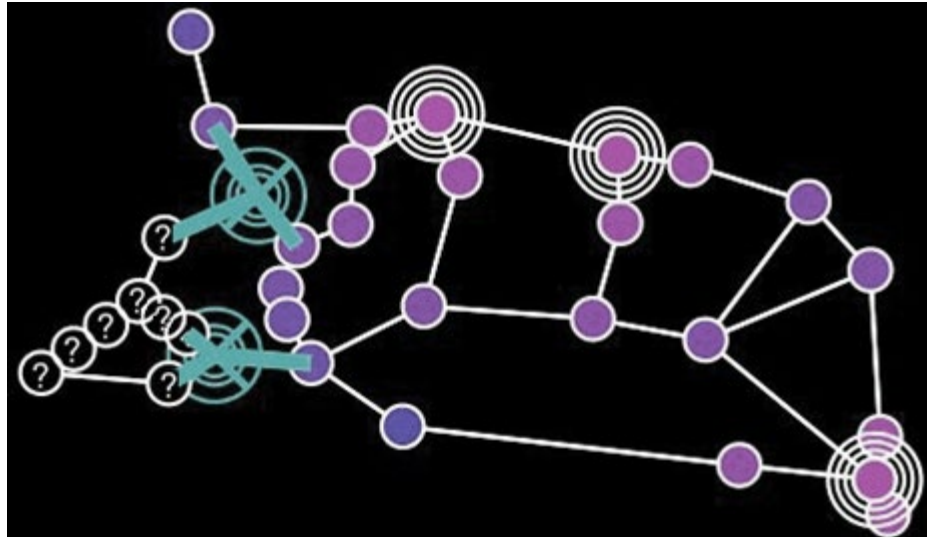


Рисунок 1.11. Пошкодження окремих ланок у мережі SDN

### 1.2.5 Рішення задачі врахування затримок у мережі SDN

Будь-яке збільшення затримок може негативно вплинути на загальну продуктивність мережі. Для вирішення цієї проблеми використовують систему постійного моніторингу параметрів зв'язку, яка здійснює вимірювання затримок у режимі реального часу та аналізує історичні дані для виявлення тенденцій і аномалій. Отримані дані використовуються для адаптивного переналаштування маршрутів передачі даних, зміни алгоритмів балансування навантаження та оптимізації розподілу ресурсів між контролерами. Завдяки інтеграції таких механізмів SDN може миттєво реагувати на зміну умов мережевого середовища, автоматично перенаправляючи управлінські потоки через менш завантажені канали або змінюючи критерії маршрутизації для мінімізації затримок.

Підхід до врахування затримок ґрунтується на багаторівневому аналізі, який охоплює як мережевий рівень, так і рівень управління. Основна ідея полягає у тому, що кожен мережевий вузол постійно проводить вимірювання часу відгуку

від контролера, а результати цих вимірювань використовуються для побудови інтерактивної моделі топології мережі. Модель дозволяє прогнозувати виникнення затримок і заздалегідь коригувати параметри мережевих з'єднань або переналаштовувати логіку управління для зниження негативного впливу затримок. При цьому враховуються як затримки, які виникають на коротких відстанях (де час передачі практично дорівнює нулю), так і ті, що характерні для вузлів, розташованих на периферії мережі, де затримка може досягати крайніх значень. У результаті інтегрованої системи моніторингу і адаптації мережі SDN формується усталена модель, здатна стабільно працювати навіть в умовах мінливих мережевих умов, гарантуючи своєчасну доставку команд управління.

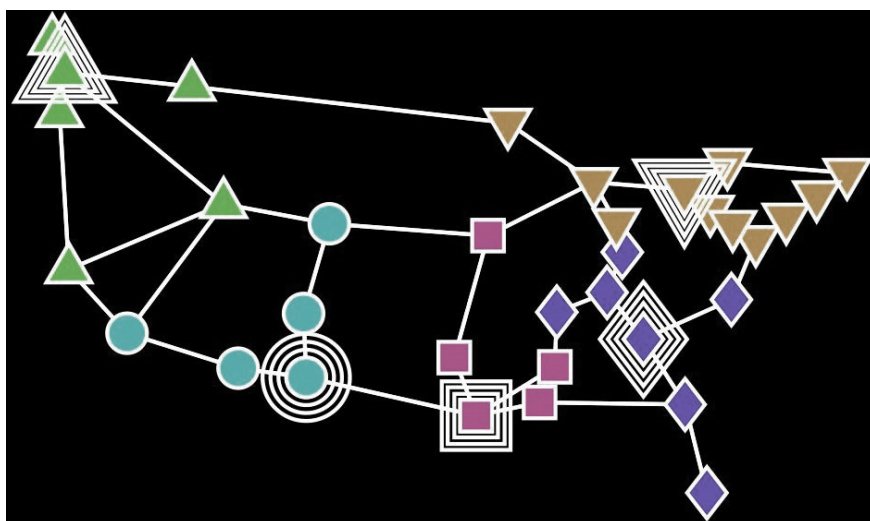


Рисунок 1.12. Організація мережі без урахування навантаження контролерів

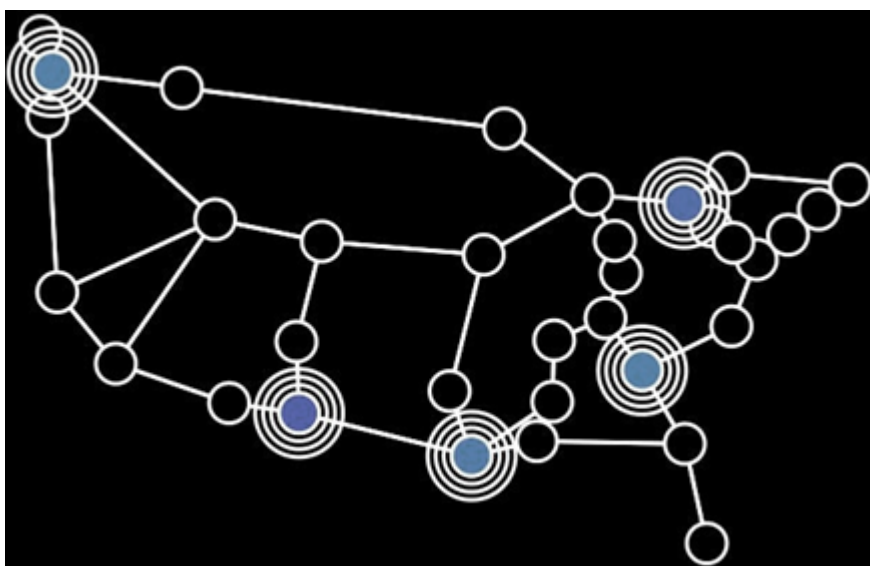


Рисунок 1.13. Організація мережі з урахуванням затримки між контролерами

Для демонстрації механізму врахування затримок наведено приклади візуальної моделі у вигляді схем, де мережа представлена у вигляді сітки вузлів, а зв'язки між останнім функціонуючим контролером та мережевими пристроями позначені різними кольорами (рис.1.12, рис.1.13). Рожевий колір вказує на нульову затримку (вузли, що знаходяться у безпосередній близькості), синій – на середню затримку (приблизно 50% від максимальної), а блакитний (skyblue) – на максимальну затримку (100% від діаметра сітки). У цій моделі кожен вузол передає дані про затримку до центру, що дозволяє в режимі реального часу відстежувати, які ділянки мережі потребують оптимізації.

### **1.3 Огляд методів розподілу навантаження до контролерів SDN**

Ефективність роботи SDN безпосередньо залежить від здатності системи управляти великою кількістю запитів та забезпечувати оперативну координацію між мережевими елементами, що зумовлює необхідність застосування оптимізаційних методів для розподілу навантаження.

Сучасні алгоритмічні рішення дозволяють адаптувати розподіл ресурсів до динамічних умов мережі. Серед них можна виділити кілька ключових підходів, які спираються на різні математичні та статистичні принципи. Зокрема, буде розглянуто класичний метод k-середніх як основу для кластерних рішень, що дозволяє ефективно розподілити контролери за принципом мінімізації різниці між внутрішньо-кластерними відстанями. Окрім того, аналіз методів, заснованих на щільності, дозволяє врахувати локальні зміни в інтенсивності мережевого трафіку та, відповідно, адаптувати структуру розподілу навантаження до ділянок з високим навантаженням. Нарешті, буде розглянуто підхід, орієнтований на критерій стійкості до відмов, що забезпечує безперервність роботи мережі навіть у разі відмов деяких контролерів або збоїв у зв'язності між ними.

#### **1.3.1 Метод розподілу навантаження на основі k-середніх**

Метод k-середніх являє собою один із найпоширеніших підходів до кластеризації в задачах розташування мережевих контролерів у SDN. Його основна ідея полягає у розбитті всієї мережі на окремі кластери, кожний з яких

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

оптимізовано з точки зору затримки між мережевими вузлами та центральним елементом кластеру. За допомогою цього алгоритму визначаються центроїди (або медіани, залежно від формулювання задачі), що відповідають оптимальним точкам розміщення контролерів, і шляхом мінімізації середнього значення затримок отримують як рішення, що забезпечують найменшу сукупну затримку, так і варіанти, спрямовані на зменшення максимального часу відповіді.

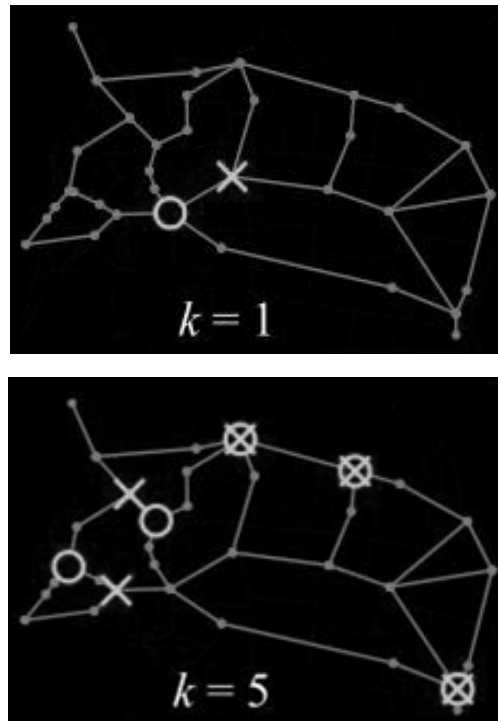


Рисунок 1.14. Розташування одного і п'яти контролерів для оптимальної середньої та максимальної затримок

При використанні методу  $k$ -середніх увага зосереджується виключно на швидкості залучення мережі до управління, тому питання аварійних поломок або перевантаження контролерів ігноруються. Цей підхід формалізується із застосуванням модифікації, що відповідає мінімізації середньої затримки (за принципом  $k$ -медіан) або мінімізації найбільшої затримки (як у задачі  $k$ -центрів). В результаті отримуються два типи оптимальних розташувань: розташування з оптимізацією середньої величини затримки, що покращує загальну ефективність мережевого управління, та розташування, яке мінімізує максимальну затримку, що гарантує, що навіть найбільш віддалені вузли отримують команди у прийнятному часовому проміжку. На рис. 1.14 результати роботи алгоритмів

зображуються символами: розташування, позначене знаком «о», демонструє конфігурацію для мінімізації середньої затримки, а розташування, відзначене знаком «х», – для зменшення максимальної затримки.

Крім того, існує варіант проблеми, який часто називають «проблемою максимального покриття». У ньому критерієм оптимізації виступає максимальна кількість вузлів, що розташовані в межах заданої величини затримки до обраного контролера. Такий підхід дозволяє гарантувати, що найбільша кількість мережевих елементів буде обслуговуватися в режимі, коли час реагування не перевищує заданих обмежень, що особливо важливо для забезпечення стабільності роботи та високої якості обслуговування у великих SDN-системах.

### 1.3.2 Метод розподілу навантаження на основі щільності

У даному підході використовується алгоритм кластеризації на основі щільності для розбиття мережі на підмережі із високо згрупованими вузлами, що мають інтенсивну внутрішню взаємодію та менш щільні зв'язки із елементами інших груп. Цей метод, відомий як DVCP, дозволяє не лише визначити оптимальну кількість підмереж (а отже, і контролерів), але й знайти їхні центри, які задають найкраще розташування для забезпечення мінімальних затримок і високої надійності управління.

При розгляді кожного вузла  $s_i$  в мережі обчислюються дві основні характеристики: локальна щільність  $\rho_i$  та мінімальна відстань  $\delta_i$  до вузлів з більшою високою щільністю. Локальна щільність  $\rho_i$  визначається кількістю сусідніх вузлів, що знаходяться на відстані не більше певного параметра  $d_c$ , який для масштабних мереж традиційно обирають як 30% від діаметра графу. Значення  $\delta_i$  — це мінімальна відстань від вузла  $s_i$  до будь-якого вузла з більшою щільністю; для вузла, який має найбільшу щільність,  $\delta_i$  прирівнюється до максимальної відстані між ним та іншими вузлами. Саме ці два параметри дозволяють визначити кандидатів у ролі центрів кластерів – саме там, де величини  $\delta_i$  є високими, що свідчить про їх відносну ізолюваність від інших високощільних вузлів, та де внутрішньокластерна щільність максимальна.

Після вибору центрів кожен з інших вузлів, опираючись на правило найближчого сусіда серед чинників вищої щільності, приписується до відповідного кластеру. Завдяки цьому мережа ділиться на групи, кожна з яких оптимально охоплюється окремим контролером. При цьому розмір кожної підмережі враховується з урахуванням потужності обраного контролера, а оптимальна кількість контролерів автоматично визначається як результат процесу кластеризації.

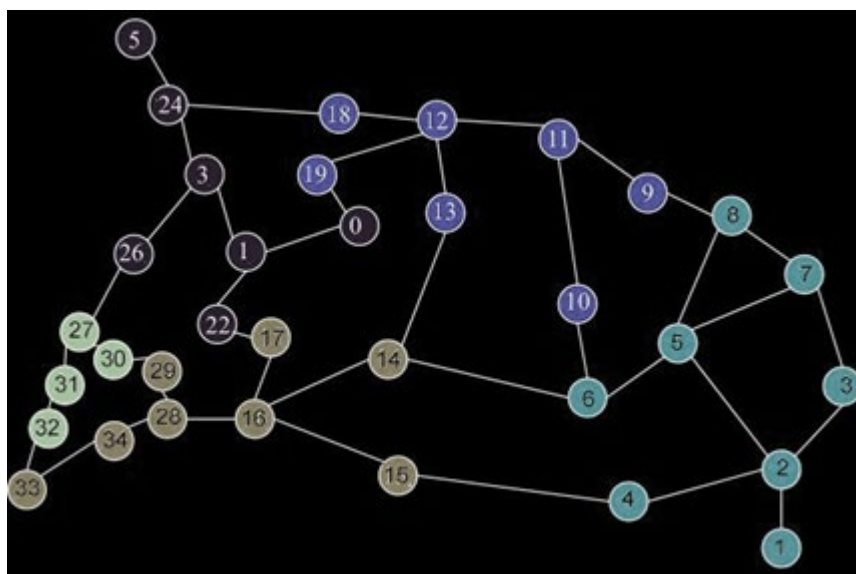


Рисунок 1.15. Кластеризація мережі за алгоритмом на основі щільності

З точки зору управління затримками, після розбиття мережі на кластери розташування контролерів для кожної підмережі оптимізується за такими критеріями, як надійність зв'язку і мінімізація затримки сигналу. При цьому для кожного кластеру обчислюється середня затримка передачі сигналу ( $\pi_{avglatency}$ ) між контролером і вузлами, які до нього відносяться, а також максимальна затримка ( $\pi_{maxlatency}$ ) для визначення граничної величини затримки в кластері. Окрім цього, для зниження витрат на обмін інформацією між контролерами важливо, щоб їх позиції були якомога ближчими одна до одної. Оскільки вибір окремих розташувань контролерів проводиться незалежно в кожному кластері, заздалегідь точно визначити відстані між усіма контролерами не завжди можливо, однак аналіз затримок від контролера до вузлів в інших підмережах дозволяє вибрати такі оптимальні позиції, які мінімізують латентність.

На рис. 1.15 наведено приклад кластеризації мережі за алгоритмом на основі щільності, де мережа розбита на п'ять груп, кожна з яких має свій контролер. Різні групи позначені окремими кольорами, що дозволяє візуально оцінити якість кластеризації та потенціал для оптимізації розподілу навантаження. Такий підхід дозволяє врахувати як внутрішньо-кластерні затримки, так і зв'язок між різними підмережами, що є ключовим для забезпечення надійного функціонування SDN-систем у складних і динамічно змінних умовах мережі.

### 1.3.3 Метод розподілу навантаження за критерієм стійкості до відмов

Деякі топології мереж дозволяють забезпечити ефективне управління навіть за умов відмови окремих контролерів, проте вимоги до стійкості часто потребують більш комплексного підходу. Один із таких методів – це алгоритм РОСО (Placement Optimized for Controller Resilience), який спеціально орієнтований на вибір стратегічних положень керуючих елементів із врахуванням ризику їх відмови та впливу цього на затримки системи.

Суть алгоритму РОСО полягає у використанні двох основних критеріїв для оцінки розташування контролерів. По-перше, розраховується максимальний час відгуку ( $\pi_{0max}$ ) від мережевого вузла до контролера у нормальних умовах; це значення позначимо як  $\pi_{0max}$ . По-друге, алгоритм враховує потенційне збільшення затримки ( $\pi_c^{max}$ ) у випадках, коли деякі контролери виходять з ладу, і їх функції перекладаються на інші. Для кожного можливого сценарію відмов (від 0 до  $k - 1$ , якщо планується розмістити  $k$  контролерів) формується набір результатів, і максимальна затримка у цьому наборі потім визначається як ключовий показник.

З практичної точки зору, якщо метою є мінімізація затримок у режимі безвідмовної роботи, то вигідніше рівномірно розподіляти контролери по мережі. Але якщо пріоритетом є стійкість до відмов, то оптимальне розміщення може зажадати більш централізованої позиції, щоб у разі збою одного з контролерів решта змогла оперативно захопити підопічні вузли та звести до мінімуму збільшення затримок. Алгоритм РОСО гарантує, що обидва ці параметри – базова затримка при нормальному режимі ( $\pi_{0max}$ ) та максимальна затримка при відмовах

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

( $\pi_c^{\max}$ ) – знаходяться на оптимальному рівні за принципом Парето, що дозволяє знайти баланс між продуктивністю та стійкістю мережі.

Окрім аналізу часу відгуку, алгоритм враховує залишкове навантаження на контролери, а також відстані між ними, оскільки зменшення інтерконтрольних затримок соціально важливо для швидкої синхронізації. В реальних умовах, навіть коли окремих контролерів може мати достатню пропускну здатність для обслуговування мережі, підвищені вимоги до надійності стимулюють розгортання додаткових контролерів з метою резервування. Внаслідок цього, навіть при непередбачуваних збоях система має можливість автоматично перенавантажувати управляючі функції, що дозволяє зберегти прийнятну оперативність мережевих сервісів.

#### **1.4 Адаптація математичної моделі розподілу навантаження до контролерів SDN**

У сучасній архітектурі програмно-визначуваних мереж критично важливо правильно розподілити навантаження між окремими контролерами для забезпечення високої продуктивності та надійності системи. Для цього розроблено математичну модель, яка дозволяє врахувати як топологічну структуру мережі, так і її експлуатаційні параметри, зокрема затримки, пропускну здатність каналів та ймовірність відмови зв'язків.

Нехай основною структурою мережі є граф  $G = (V, E, W, C, P)$ , де множина ( $V$ ) містить вузли мережі (як-от маршрутизатори), ( $E \subset V \times V$ ) описує з'єднання між цими вузлами, а функції ( $W$ ), ( $C$ ) і ( $P$ ) задають відповідно вагу кожного вузла (наприклад, час затримки, що асоціюється з кожним маршрутизатором), характеристику кожного ребра (пропускну здатність або затримку каналу) та надійність цих з'єднань (ймовірність їх нормальної роботи).

Для цілей оптимізації розподілу навантаження граф ( $G$ ) подається як сукупність неперетинних блоків або підграфів. Зокрема, множина ( $V$ ) розбивається на ( $r$ ) диз'юнктивних підмножин ( $V_1, V_2, \dots, V_r$ ) таким чином, що  $\bigcup_{k=1}^r V_k = V$  і  $V_k \cap V_l = \emptyset \quad \forall k \neq l$ . Кожен підграф ( $G_k = (V_k, E_k)$ ) формується шляхом вибору з ( $E$ ) всіх ребер, що з'єднують пару вузлів із ( $V_k$ ).

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

Ребра, які зв'язують вузли з різних підмножин (тобто, між різними  $(V_k)$  та  $(V_l)$ ), утворюють так званий край розділу. Оцінка якості такого розбиття може базуватися як на характеристиках окремих підграфів (визначених функціями  $(g(V_k))$  чи  $(h(G_k))$ ), так і на величинах, що відображають зв'язність між ними (наприклад, сумарна вартість краєвих ребер, яку можна представити як  $(\sum_{k<l} f(E_{kl}))$ ), де  $(E_{kl})$  – підмножина ребер між  $(V_k)$  та  $(V_l)$ ).

Для забезпечення рівномірності поділу важливо ввести обмеження щодо балансу розмірів доменів. Нехай середній розмір частини дорівнює  $(\frac{|V|}{r})$ . Тоді допускається, щоб абсолютне відхилення розміру будь-якого  $(V_k)$  не перевищувало  $(\varepsilon_v (\frac{|V|}{r}))$ , де  $(0 < \varepsilon_v < 1)$  є параметром, що обмежує дисбаланс (1.1).

$$\sum_{kl} f(E_{kl}) = \sum_{kl} \sum_{(i,j) \in E_{kl}} c_{ij} \rightarrow \min, \quad (1.1)$$

Додатково, параметр, який характеризує максимальний діаметр кожного підграфа  $(G_k)$ , не повинен перевищувати заздалегідь визначене значення  $(\Delta_c)$  (1.2).

$$|V_k| \geq (1 - \varepsilon_v) \lfloor \frac{|V|}{r} \rfloor, k = 1..r. \quad (1.2)$$

При поділі мережі на зони маршрутизації також мають бути виконані умови, що мінімізують ймовірність збоїв у зв'язках між різними зонами, а діаметр кожного підграфа повинен бути обмежений — це дозволяє забезпечити стабільне управління навіть при непередбачених подіях (1.3 та 1.4).

Якщо мережа розділена на зони, ймовірність того, що перебоїв не буде, є мінімальною, а діаметр підграфів не повинен перевищувати задане значення  $\Delta_c$ . Тоді модель буде виглядати так:

$$\sum_{kl/|E_{kl}| \neq 0} f(E_{kl}) = \sum_{kl/|E_{kl}| \neq 0} \prod_{(i,j) \in E_{kl}} (1 - p_{ij}) \rightarrow \min, \quad (1.3)$$

$$d(G_k) \leq \Delta_c, k = 1..r. \quad (1.4)$$

Цільова функція математичної моделі визначається як мінімізація

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

сумарного часу затримки на краях розділу між підграфами. Іншими словами, потрібно звести до мінімуму суму затримок, які виникають при передачі даних між вузлами, що належать до різних зон, що записується як:  $\min \pi_{\text{total}}$  (1.5), де ( $\pi_{\text{total}}$ ) — сумарна затримка на краях розділу.

$$\sum_{kl/|E_{kl}| \neq 0} \sum_{i \in V_{kl}} \omega_i \rightarrow \min \quad (1.5)$$

Оптимальний поділ мережі за даною моделлю дозволяє збалансувати навантаження між зонами, зменшити витрати на обмін інформацією між різними регіонами мережі та мінімізувати ризик відмов маршрутизаторів, особливо на кордонах зон. Завдяки цьому, система розподілу навантаження забезпечує ефективну маршрутизацію керуючих повідомлень, що критично важливо для роботи сучасних SDN-систем.

### **1.5 Аналіз вимог до реалізації модифікованого алгоритму розподілу навантаження до контролерів SDN**

Програмна реалізація модифікованого алгоритму розподілу навантаження до контролерів SDN має враховувати особливості топології мережі, заданої графом  $G(S, L)$ , де (S) є множиною маршрутизаторів, а (L) — набір багатонаправлених з'єднань між ними. У цій моделі, якщо між вузлами мережі встановлено прямий зв'язок, довжина з'єднання приймається сталою, а у випадках відсутності прямого зв'язку — значення визначається як нульове. Така специфікація дозволяє спростити аналіз і подальшу кластеризацію, адже вузли, що активно взаємопов'язані, виділяються як єдине ціле.

Модифікований алгоритм, розроблений у даній роботі, аналізує топологію мережі шляхом розподілу маршрутизаторів на окремі підмережі. При цьому використовується поняття щільності з'єднань: вузли, які мають високий показник внутрішніх зв'язків і слабе зв'язування з іншими блоками, формують окремі кластеризовані зони. Поділ мережі на такі підмережі дозволяє розглядати кожну з них як автономну зону маршрутизації, у якій знижується ймовірність відмов завдяки більш тонкій кластеризації, що, у свою чергу, сприяє зниженню затримок у передачі сигналів.

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

Програмна реалізація алгоритму проходить три основні етапи. Спочатку здійснюється аналіз мережевої топології з метою визначення щільності з'єднань між маршрутизаторами. На цьому етапі обчислюються показники, що характеризують інтенсивність взаємодії вузлів, що дозволяє виявити групи з високою кількістю внутрішніх зв'язків. Далі проводиться кластеризація мережевих електронних вузлів з використанням відстаней від кожного вузла до найближчого вузла з більш високим показником щільності. Цей крок дозволяє сформувати межі підмереж з логічно відібраними групами вузлів, які мають як внутрішню зв'язність, так і гарантовано менший вплив зовнішніх з'єднань. Нарешті, на основі отриманих кластерів вирішується задача вибору оптимального розташування контролера у кожній підмережі. При цьому враховуються критерії, орієнтовані на мінімізацію затримок у даній зоні та забезпечення максимальної надійності через зниження ймовірності відмов.

## 1.6 Реалізація модифікованого алгоритму розподілу навантаження до контролерів SDN

Реальна мережа складається із великої кількості маршрутизаторів, що з'єднані між собою багатонаправленими зв'язками. Для оптимізації розподілу навантаження між контролерами SDN запропоновано алгоритм, заснований на аналізі топологічної структури мережі шляхом поділу множини маршрутизаторів на окремі підмережі (кластеризовані зони). Основною ідеєю даного підходу є визначення двох ключових характеристик для кожного маршрутизатора ( $s_i$ ):

- Локальна щільність  $\rho_i$  – показник, що відображає інтенсивність зв'язків вузла з іншими маршрутизаторами, обчислюється за допомогою формули, у якій враховується відстань між вузлами  $d_{ij}$  та коефіцієнт підключення  $k_{cl}$ . У виразі 1.6 фіксується, що вузол враховується у підрахунку, якщо відстань до нього не перевищує максимально допустиму величину  $d_c$ , яка зазвичай дорівнює 0,3 діаметра мережі для великих розподілених систем;
- Відстань до вузла з більш високою щільністю  $\delta_i$  – це мінімальна відстань від маршрутизатора  $s_i$  до будь-якого іншого вузла, що має більшу локальну

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

щільність. Співвідношення 1.7 задає, як саме обчислюється  $\delta_i$ : для вузла з максимальною локальною щільністю ця величина прирівнюється до найбільшої відстані в мережі, а для решти вузлів – як мінімум відстань до сусіднього вузла з вищим показником  $\rho$ .

$$\rho_i = \sum_j k_{cl}(d_{ij} - d_c) \quad (1.6)$$

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}) \quad (1.7)$$

Алгоритм спершу проводить аналіз мережевої топології: для кожного маршрутизатора обчислюються  $\rho_i$  та  $\delta_i$ . На основі отриманих значень формується черга кандидатів для вибору центрів підмереж. При цьому, якщо фактична кількість потенційних центрів менша за рекомендоване значення, відбір ведеться за критерієм максимального  $\delta_i$ . У випадку перевищення рекомендованого ліміту спершу обираються вузли з високим «пріоритетом» — тобто, де значення  $\delta_i$  є найбільшим серед тих, що мають однаковий рівень локальної щільності. Після визначення центрів для кожної з підмереж, решта вузлів приписуються до того кластеру, центр якого знаходиться найближче до них за виміряною відстанню.

Застосування такого підходу дозволяє розбити мережу на незалежні зони з високою внутрішньою зв'язністю, що призводить до зниження загального рівня затримок та підвищення відмовостійкості системи. Завдяки тонкій кластеризації вузлів з домінуючими зв'язками всередині підмереж, кожна окрема зона відзначається як більш стабільна та менш схильна до аварійних відмов, при цьому її внутрішня латентність суттєво зменшується. Алгоритм також дає змогу обрати рекомендовану кількість центрів як потенційні точки розташування контролерів, що значно спрощує подальше розміщення управлінських елементів.

Для ілюстрації роботи алгоритму розподілу навантаження використовується псевдокод, який складається з двох основних функцій. Перша функція виконує аналіз щільності зв'язків у мережі. Для кожного маршрутизатора обчислюється локальна щільність, заснована на кількості вузлів, що знаходяться на відстані не більші за  $d_c$ , та коефіцієнті підключення  $k_{cl}$ . Друга функція на

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

основі отриманих показників формується кожен кластер: вузли, які не були визначені як самостійні центри, приєднуються до найближчого вузла з більш високою щільністю, що формує певну групу з центром кластеризації.

Модифікований алгоритм об'єднує аналіз топології мережі за параметрами інтенсивності зв'язків та відстаней між вузлами, що дозволяє відокремити мережеві домени для розподілу навантаження. Реалізація даного підходу покликана забезпечити оптимальну кількість та розташування контролерів у SDN, зменшуючи ймовірність відмов та затримок, і сприяє підвищенню загальної стабільності системи. Псевдокод реалізації модифікованого алгоритму розподілу навантаження до контролерів SDN:

```
// Функція аналізу щільності для кожного маршрутизатора
int analyzeDensity(Graph G, double dc) {
    for (router s in G.S) {
        // Обчислити локальну щільність  $\rho(s)$ 
         $\rho[s] = 0$ ;
        for (router t in G.S) {
            if (distance(s, t) <= dc && (dc - distance(s, t)) > 0) {
                 $\rho[s] += 1$ ; // або kcl (де kcl=1 при позитивній різниці)
            }
        }
    }
    // Обчислити  $\delta(s)$  для кожного маршрутизатора
    // як мінімальну відстань до вузла з більш високою щільністю
    for (router s in G.S) {
         $\delta[s] = +\infty$ ;
        for (router t in G.S) {
            if ( $\rho[t] > \rho[s]$  && distance(s, t) <  $\delta[s]$ ) {
                 $\delta[s] = \text{distance}(s, t)$ ;
            }
        }
    }
    // Якщо s має максимальну щільність,  $\delta[s]$  = максимальна відстань в мережі
    if ( $\delta[s] == +\infty$ ) {
         $\delta[s] = \max(\text{distance}(s, t) \text{ for all } t \text{ in } G.S)$ ;
    }
    // Повернути кількість кандидатів для центрів кластерів
    return countCandidates( $\delta$ );
}

// Функція кластеризації – формування кластерів,
```

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

```

// де кожен вузол приєднується до найближчого центру
void clustering(Graph G, double dc) {
    int candidateCount = analyzeDensity(G, dc);
    centerQueue = selectCenters(candidateCount);
    for (router s in G.S) {
        if (centerQueue.contains(s)) {
            createNewCluster(s);
        } else {
            nearest = findNearestHigherDensityNode(s);
            assignToCluster(s, nearest.cluster);
        }
    }
}
}
}

```

У представленому вище псевдокоді показано основні етапи реалізації: обчислення локальної щільності маршрутизаторів, визначення відстані до вузла з вищою щільністю, відбір кандидатів для центрів кластерів та їх агрегацію у групи. Алгоритм не враховує навантаження трафіку безпосередньо, проте фокусується на структурних зв'язках між вузлами, що є ключовим для формування стабільних зон управління мережевими контролерами.

## 1.7 Розробка діаграми класів та блок-схеми алгоритму застосунку для моделювання роботи SDN

Для моделювання роботи програмно-визначуваних мереж (SDN) було створено застосунок, реалізацію якого виконано на об'єктно-орієнтованій мові Java із використанням бібліотеки JavaFX для графічної візуалізації топології мережі. Архітектура застосунку побудована за принципом багаторівневості, що включає три основні шари: реалізацію алгоритму розподілу навантаження до контролерів, візуалізацію моделі мережевого графу та збереження (постійне зберігання) самої моделі.

Основна логіка роботи застосунку розділена на декілька незалежних модулів, кожен з яких виконує свою роль. Рівень моделі містить класи, які формують представлення мережі у вигляді графа. Клас Node інкапсулює інформацію про окремий маршрутизатор мережі: він містить ідентифікатор, координати для розміщення на екрані, а також показники, що використовуються

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

для візуального відображення (колір, стан тощо). Клас Link представляє з'єднання між вузлами та містить інформацію про ваги цих зв'язків, що визначають затримки або пропускну здатність каналів. Клас Model об'єднує всі екземпляри класів Node і Link, забезпечуючи методи для додавання, видалення та модифікації елементів мережевого графу (такі як addNode, removeNode, moveNode, addLink, removeLink, changeLinkWeight). Окрім того, модель має функціонал для отримання окремих вузлів або набору зв'язків, що спрощує операції з обчисленням параметрів топології.

Для представлення моделі мережі на екрані використано клас View, який, за допомогою JavaFX, відповідає за перемальовування вузлів і з'єднань (методи repaintNodes і repaintLinks) відповідно до змін у моделі. Інтерфейс застосунку організовано таким чином, що користувач може в режимі реального часу створювати, редагувати та зберігати граф мережі. Додаткові класи, що підтримують роботу з файлами, забезпечують функції відкриття, збереження і завантаження мережевої моделі.

Клас Algorithm зосереджує у собі логіку моделювання алгоритму розподілу навантаження до контролерів SDN. Сам алгоритм побудований на підходах аналізу топологічних параметрів мережі – зокрема, на визначенні щільності зв'язків між маршрутизаторами та відстаней між ними, що дозволяє визначити оптимальні зони для розміщення контролерів. Для генерації початкової топології мережі використовується клас GraphGenerator, який отримує параметри (кількість вузлів, мінімальна та максимальна вага, показники зв'язності) через конструктор. Крім того, клас має методи для випадкового «змішування» груп вузлів (метод mixClusterPointers) для подальшої кластеризації. Інші методи, такі як CheckParameters, makeClusterLinks, MakeInterClusterLinks і GenerateTrafficQuantity, відповідають за створення зв'язків між вузлами, розрахунок загальних ваг, а також генерування значень навантаження для з'єднань, що дозволяє моделювати різні режими роботи мережі.

Структура застосунку організована у вигляді трьох окремих, але взаємопов'язаних блоків:

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

- Модуль моделювання алгоритму: забезпечує розподіл навантаження шляхом аналізу топології мережі, побудови кластерів та вибору оптимальних положень для розміщення контролерів.
- Візуальна частина: відповідає за відображення мережевої моделі, організацію інтерактивного інтерфейсу для користувача та підтримку операцій з перемальовуванням графічного представлення (реалізовано за допомогою JavaFX).
- Модуль зберігання графу: дозволяє зберігати та завантажувати модель мережі у файл, забезпечуючи постійність даних між сесіями роботи.

Розроблена діаграма класів (рис. 1.16) демонструє взаємозв'язок основних класів: Model, Node, Link, View, Algorithm, а також допоміжних класів для роботи з файлами і генерації топології (GraphGenerator, DensityAlgorithm тощо). Діаграма відображає, як дані переходять від логіки розрахунків у моделі до візуального інтерфейсу, а також забезпечує можливість розширення функціональності шляхом додавання нових алгоритмів розподілу навантаження.

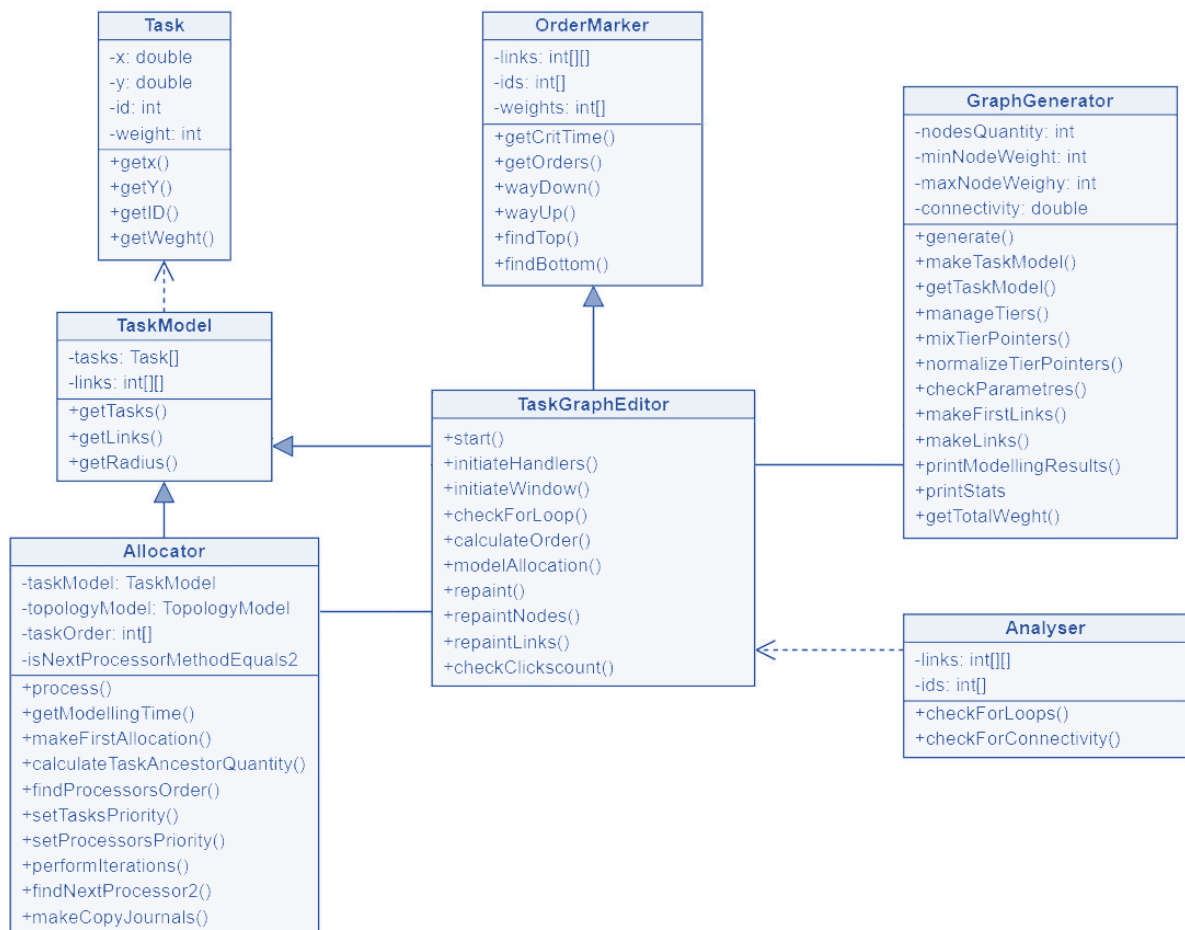


Рисунок 1.16. Діаграма класів застосунку для моделювання роботи SDN

Блок-схема застосунку (Рис. 1.17) наочно демонструє послідовність операцій: від завантаження або генерації початкової топології, через розподіл вузлів на кластери та розрахунок параметрів, до візуалізації результатів моделювання і збереження моделі. Алгоритм починається із завантаження або генерації початкового графа, який є моделлю мережі з вузлами та з'єднаннями між ними. На цьому етапі система отримує повну топологічну інформацію про мережу, що дає змогу надалі виконувати аналіз структурних властивостей. Після цього для кожного вузла графа проводиться обчислення локальної щільності. Ця характеристика визначається кількістю сусідніх вузлів, що знаходяться у межах заданої відстані, що є пороговим значенням. Таким чином, вузли з великою кількістю «близьких» сусідів отримують високий показник щільності. Результати цього етапу використовуються для подальшого сортування всіх вузлів за спаданням локальної щільності. Наступний крок полягає у виборі кандидатів у ролі центрів кластерів – вузлів з найвищою щільністю. При цьому алгоритм перевіряє, чи відповідає кількість таких кандидатів рекомендованим величинам. Якщо число кандидатів виявилось недостатнім, алгоритм повертається до попередніх кроків для уточнення розрахунків, а якщо – відповідає вимогам, продовжується обчислення наступної характеристики. Далі для кожного вузла визначається мінімальна відстань до вузла, який має більшу локальну щільність. Цим значенням позначається  $\delta$ . Воно є ваговою характеристикою, що свідчить про те, наскільки ізольовано певний вузол у контексті мережевої топології. Вузли, для яких це значення є особливо великим, потенційно можуть бути обрані як центри кластерів.

На основі отриманих показників локальної щільності та відстані до вузла з вищою щільністю формується остаточний набір вузлів, які будуть виступати центрами кластеризації. Далі решта вузлів приєднуються до найближчого центру за відстанню, забезпечуючи таким чином поділ мережі на групи, кожна з яких має свій керуючий вузол (центр). Алгоритм завершується, коли всі вузли отримали своє кластерне приписування. Результатом є поділ вихідного графа на кілька груп, всередині яких вузли максимально щільно взаємопов'язані, а різниця між

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

центрами кластерів відповідає заданим критеріям, що забезпечує оптимальне розміщення контролерів для управління мережею.

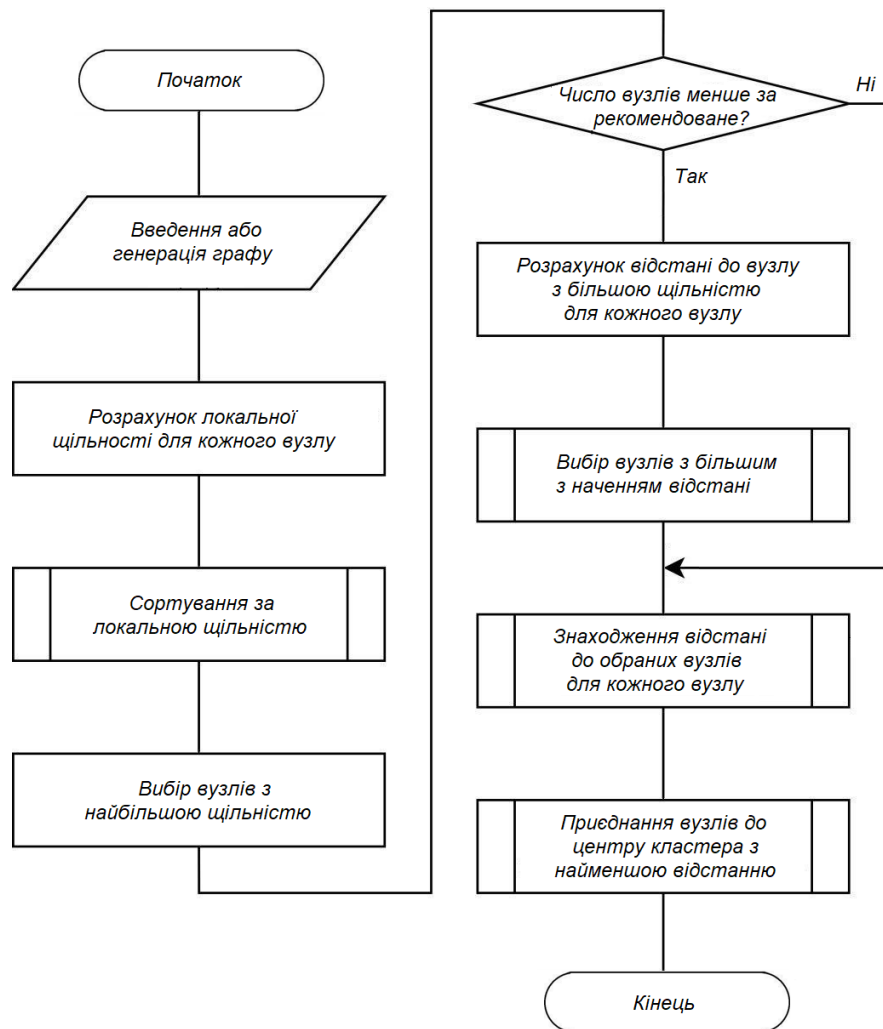


Рисунок 1.17. БСА застосунку для моделювання роботи SDN

## 1.8 Реалізація застосунку для моделювання роботи SDN

Розроблений застосунок призначено для моделювання алгоритму розподілу навантаження до контролерів SDN, а також для інтерактивного редагування мережевої топології. Програма написана мовою Java із використанням бібліотеки JavaFX для візуального представлення графу мережі, що забезпечує користувачеві зручний та інтуїтивний інтерфейс.

Інтерфейс застосунку розташовано у верхній частині головного вікна програми у вигляді меню, що містить три основні розділи: «Файл», «Редагувати» та «Моделювати». Меню «Файл» надає засоби для загального управління програмою. За допомогою його пунктів користувач може створити новий граф,

зберегти поточну конфігурацію, відкрити збережений файл, згенерувати граф відповідно до заданих параметрів або завершити роботу застосунку. Для полегшення роботи доступні гарячі клавіші (Ctrl+N, Ctrl+S, Ctrl+O, Ctrl+G, Ctrl+E).

Меню «Редагувати» націлено на безпосереднє управління мережею. Завдяки цьому модулю користувач може додавати нові вузли до графу, видаляти існуючі або переміщувати їх, а також працювати з зв'язками між вузлами – створювати, редагувати вагу або видаляти посилання. Для швидкого доступу до відповідних функцій призначено окремі гарячі клавіші (цифри 1–6), що дозволяють перемикатися між режимами роботи без необхідності звертатися до меню.

Меню «Моделювати» використовується для запуску і контролю процесу симуляції роботи SDN. За його допомогою можна перевірити коректність побудованого графу, вибрати режим моделювання (без обмежень або з врахуванням визначених обмежень), встановити конкретний алгоритм моделювання (що, наприклад, дозволяє порівняти результати роботи розробленого методу з іншими підходами), а також відобразити результати кластеризації, параметри розподілу контролерів та кінцеві позиції їх розташування. Для цього у меню передбачено окремі пункти, що активуються за допомогою комбінацій Shift + C, Shift + L, Shift + E, Shift + A, Shift + R, Shift + P, Shift + K, а загальне перемикання здійснюється через клавішу Shift.

Практичний користувацький досвід роботи із застосунком забезпечується через інтуїтивно зрозумілу панель кнопок, розташовану зліва від головного вікна. Тут розміщено елементи управління для операцій над вузлами (додавання, видалення, переміщення) та зв'язками (додавання, видалення, редагування ваги). Щоб виконати певну операцію, користувачу достатньо активувати відповідний режим через меню або кнопку та здійснити клік безпосередньо на графічній області. Наприклад, для додавання вузла користувач переходить у режим додавання, натискає у потрібній точці робочої області, після чого новий вузол автоматично з'являється згенерованим з унікальним ідентифікатором.

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

Аналогічним способом реалізовано функції видалення та переміщення.

Розроблений застосунок підтримує роботу з файлами, що дозволяє користувачу зберігати поточний стан моделі мережі, завантажувати раніше збережені конфігурації, а також генерувати нові графи відповідно до заданих параметрів. Це значно полегшує тестування різних сценаріїв моделювання роботи SDN, адже користувач може оперативно переглядати результати та проводити аналіз отриманих даних.

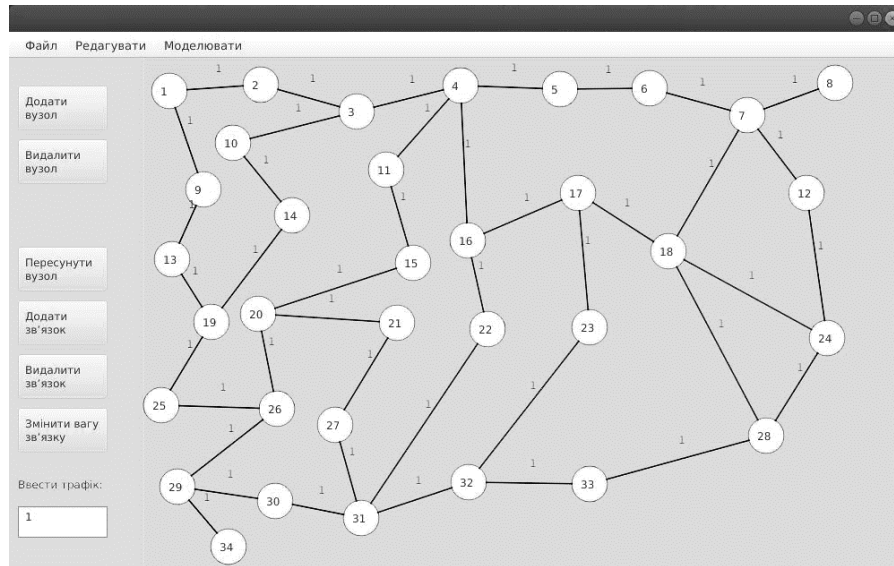


Рисунок 1.18. Моделювання початкової топології SDN-мережі

Для відображення результатів моделювання застосунок використовує декілька режимів.

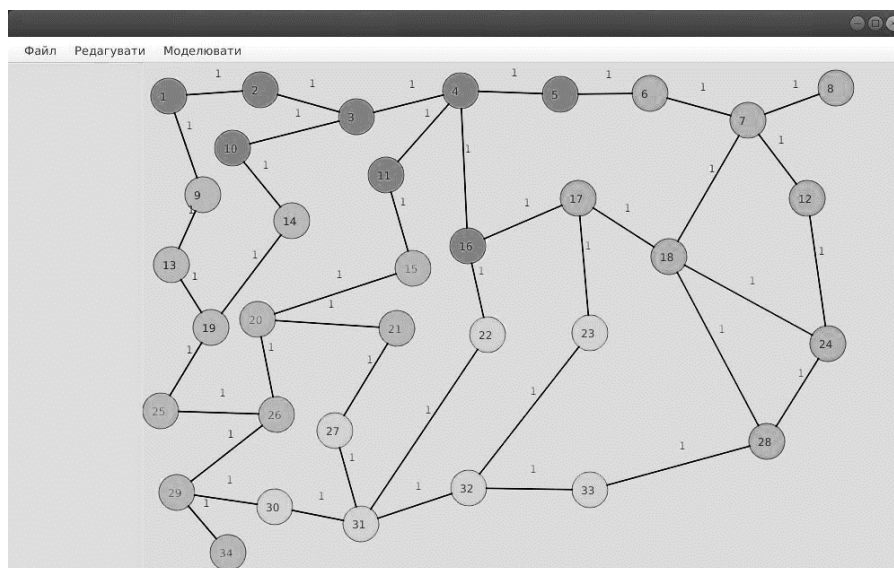


Рисунок 1.19. Моделювання розподілу навантаження у SDN-мережі

На рис. 1.18 демонструються результати створення початкової топології мережі, на рис. 1.19 – результати розподілу навантаження у SDN-мережі, на рис. 1.20 – результати моделювання, а на рис. 1.21 – схему розташування контролерів у створеній моделі мережі. Ці візуальні результати дозволяють користувачу швидко оцінити ефективність моделювання та внести необхідні корективи.

№	ρ	dc
1	4	1
2	8	1
3	8	1
4	10	3
5	6	1
6	6	1
7	8	1
8	4	1
9	4	1
10	5	1
11	6	1
12	6	1

№	ρ	dc
13	5	1
14	5	1
15	5	1
16	9	1
17	8	1
18	9	3
19	6	2
20	7	1
21	5	1
22	7	1
23	6	1
24	6	1

№	ρ	dc
25	6	1
26	8	3
27	6	1
28	7	1
29	6	1
30	7	1
31	9	3
32	8	1
33	6	1
34	3	1

Рисунок 1.20. Відображення результатів моделювання у SDN-мережі

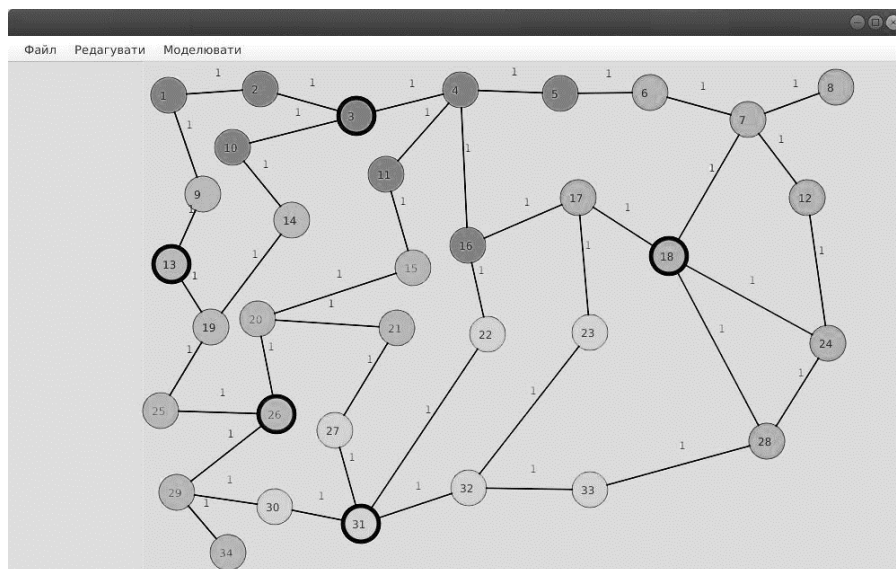


Рисунок 1.21. Моделювання розташування контролерів у SDN-мережі

## 1.9 Аналіз результатів моделювання та оцінка ефективності алгоритмів розподілу навантаження

Під час тестування модифікованого алгоритму розподілу навантаження та кластеризації для SDN-мережі було проведено ряд експериментальних вимірювань із використанням генератора випадкових мережевих графів. Модель

тестувалася на графах із збільшенням кількості вузлів із кроком у двадцять п'ять одиниць, при цьому всі трафікові з'єднання приймалися як одиничні значення. Отримані результати дозволяють порівняти ефективність запропонованого методу з алгоритмами кластеризації РОСО та k-середніх за кількома ключовими критеріями: швидкістю вирішення задачі розгортання контролерів, затримкою розподілу сервісного трафіку, а також насиченням топології контролерами при обмеженнях на трафік та кількість вузлів.

Кількість вершин	Алгоритм моделювання		
	Модифікований	k-середніх	РОСО
25	0,145	0,121	0,213
50	0,345	0,249	0,346
75	0,687	0,784	1,234
100	1,237	1,456	1,983
125	1,751	1,876	4,305
150	2,221	2,687	6,217
175	2,987	3,456	8,329
200	3,415	4,324	9,299
225	3,769	5,607	10,845
250	4,681	6,809	13,657

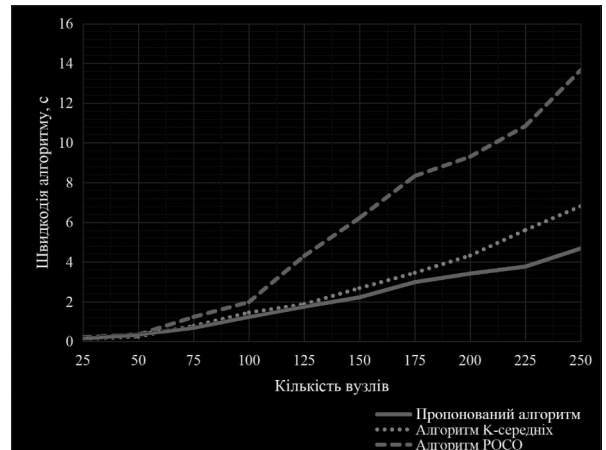


Рисунок 1.22. Ефективність алгоритмів розподілу навантаження в залежності від кількості вузлів мережі SDN

За даними таблиці і графіка на рис. 1.22, для мереж з до 100 вузлів різниця часового показника між алгоритмами є досить незначною. Проте при збільшенні кількості вузлів різниця починає суттєво зростати. Модифікований алгоритм демонструє нижчий час виконання порівняно з РОСО, який характеризується вищим порядком поліноміальної складності. При цьому алгоритм k-середніх, хоч і є досить ефективним для малих топологій, поступається запропонованому методу на великих графах, що свідчить про повільне зростання триваючих витрат у розробленому підході.

Дослідження затримки розподілу сервісного трафіку, представлене на рис. 1.23, показало, що значення затримок для різних алгоритмів коливаються залежно від кількості вершин. Незважаючи на те, що у деяких тестових сценаріях алгоритми можуть давати подібні результати, кращі показники демонструє модифікований метод – переважно він забезпечує нижчі затримки при передачі сервісних повідомлень. Це свідчить про ефективне балансування навантаження,

завдяки оптимізації кластеризації мережі на підмережі, де внутрішня зв'язність вузлів є максимально інтенсивною.

Кількість вершин	Алгоритм моделювання		
	Модифікований	k-середніх	POCO
25	6	7	8
50	8	7	7
75	5	6	9
100	8	12	14
125	9	8	9
150	10	9	11
175	11	9	10
200	10	11	12
225	15	18	17
250	21	20	23

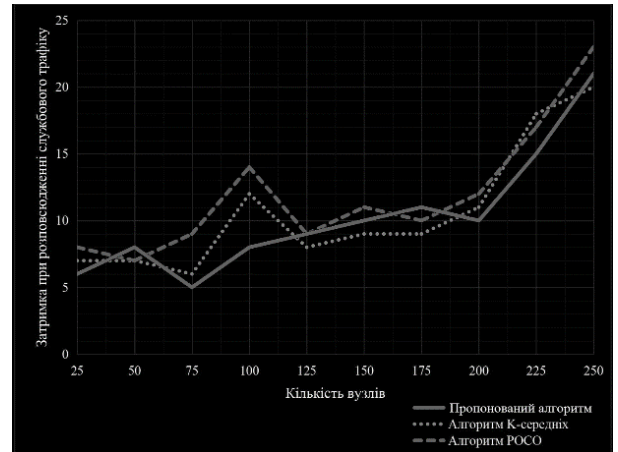


Рисунок 1.23. Затримки розподілу сервісного трафіку в залежності від кількості вузлів мережі SDN

Аналіз насичення топології контролерами у випадках, коли обмежено трафік роутерів (рис. 1.24), дозволив оцінити, як ефективно кожен алгоритм забезпечує покриття мережі контролерами.

За результатами експериментів, зростання кількості контролерів супроводжується збільшенням насичення топології, і модифікований алгоритм при цьому показує дещо вищі значення, що свідчить про його здатність рівномірніше розподіляти потік даних між зонами. Подібний підхід значно знижує ризик перевантаження окремих елементів і дозволяє підтримувати оптимальний рівень управління навіть при великих значеннях мережевих параметрів.

Кількість вершин	Алгоритм моделювання		
	Модифікований	k-середніх	POCO
5	0,19	0,15	0,16
10	0,34	0,29	0,32
15	0,55	0,48	0,51
20	0,76	0,63	0,64
25	0,91	0,77	0,83
30	0,95	0,89	0,93
35	0,96	0,95	0,94
40	0,98	0,97	0,97
45	0,99	0,99	0,99
50	1	1	1

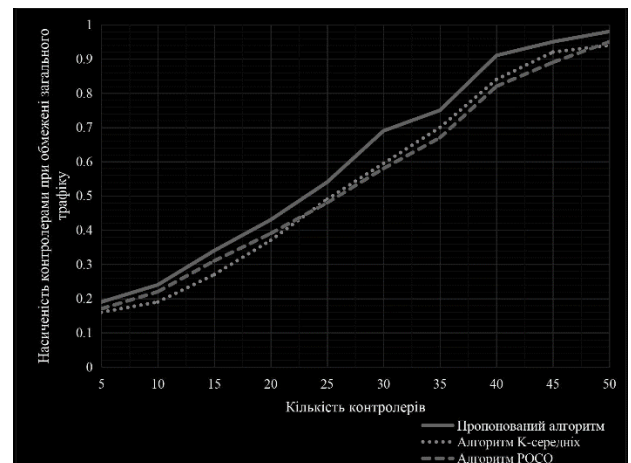


Рисунок 1.24. Насичення топології контролерами при обмеженні потоку даних роутерів мережі SDN

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

Також проведено аналіз насичення топології контролерами з урахуванням обмеження на загальну кількість роутерів. Результати показують, що при збільшенні кількості вузлів мережі всі алгоритми прагнуть до максимального рівня насиченості (значення 1), але модифікований підхід досягає цієї мети швидше, що свідчить про більш ефективне розміщення контролерів. Це означає, що застосування запропонованого методу дозволяє оптимізувати використання мережевих ресурсів та забезпечити високий рівень доступності послуг керування навіть при жорстких обмеженнях.

Порівняльний аналіз результатів тестування засвідчує, що при збільшенні масштабу мережі (кількості вершин) різниця у швидкості розгортання контролерів та затримках у розподілі сервісного трафіку між розробленим алгоритмом, класичними методом k-середніх і алгоритмом РОСО стає все вираженішою. Модифікований алгоритм демонструє найкращі показники як за час обчислення, так і за ефективністю маршрутизації — при цьому відмінність від k-середніх зростає поліноміально, а алгоритм РОСО вирізняється значно вищою складністю.

Результати моделювання, представлені на відповідних графіках (рис. 1.22–1.24), дають підстави зробити висновок про вищу ефективність модифікованого алгоритму розподілу навантаження для SDN. Запропонований метод забезпечує швидше виконання завдань, менші затримки розподілу сервісного трафіку, а також більш ефективно насичення топології контролерами при обмеженому трафіку та у великих мережевих системах. Ці результати свідчать про перспективність і доцільність впровадження модифікованого алгоритму в сучасні SDN-архітектури для забезпечення високої продуктивності та стійкості мережі.

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

## 2 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

Праця є однією з ключових складових людської діяльності, спрямованої на створення матеріальних благ. Вона виконує дві основні функції: забезпечує засоби до існування та є способом самовираження і самореалізації особистості. Перша функція проявляється через усвідомлений вибір людиною виду діяльності, її відповідність внутрішнім переконанням та моральне задоволення від виконуваної роботи.

Ергономічне забезпечення покликане покращувати взаємодію між людиною та технікою на всіх етапах—від створення й виробництва до експлуатації та утилізації технічних систем. Це досягається шляхом застосування комплексу продуманих ергономічних рішень, що враховують логічну послідовність процесів, фізичні та психологічні особливості людини, а також ефективність технологій.

У цьому розділі дипломного проекту акцент зроблено на аспектах охорони праці програміста, що мають ключове значення для забезпечення комфортних і безпечних умов роботи.

### 2.1 Аналіз небезпечних і шкідливих факторів, що впливають на програміста

Тривала та інтенсивна робота з персональним комп'ютером може призводити до розвитку різних захворювань. Постійне перебування перед екраном сприяє виникненню психологічного стресу, порушень функціонування центральної нервової системи, а також проблем із верхніми дихальними шляхами. Вплив низькочастотних електромагнітних полів у поєднанні з іншими негативними факторами може збільшувати ризик розвитку онкологічних захворювань, зокрема лейкемії. Крім того, електростатичне поле монітора притягує пил, який може спричиняти дерматологічні проблеми, загострення астматичних симптомів та подразнення слизових оболонок.

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

## **2.2 Гігієнічні вимоги до виробничого середовища.**

До основних заходів захисту на робочому місці програміста належать ефективна вентиляція, належне штучне освітлення та якісна звукоізоляція. Дотримання встановлених нормативів щодо рівня запиленості, температури повітря, рівня шуму та освітленості дозволяє створити комфортне середовище для продуктивної роботи.

Щоб забезпечити максимальну ефективність праці, необхідно враховувати всі ці фактори при облаштуванні робочого простору. Збалансовані умови допомагають не лише зберегти здоров'я, а й підвищити концентрацію, знизити рівень стресу та покращити загальну продуктивність.

### **2.2.1 Вимоги до приміщення**

Об'ємно-планувальні рішення будівель та приміщень для роботи з ВДТ мають відповідати вимогам ДСанПІН 3.3.2.007-98. Розміщення робочих місць з ВДТ ЕОМ і ПЕОМ у підвальних приміщеннях, на цокольних поверхах заборонено. Площа на одне робоче місце становить не менше 6,0 м<sup>2</sup>, а об'єм – не менше ніж 20,0 м<sup>3</sup>. У приміщеннях слід щоденно робити вологе прибирання. Вони повинні бути оснащені аптечками першої медичної допомоги. При приміщеннях мають бути обладнані побутові приміщення для відпочинку.

### **2.2.2 Освітлення**

Робочі кімнати і кабінети повинні мати природне освітлення. В інших приміщеннях допускається штучне освітлення. У тих випадках, коли одного природного освітлення не вистачає, встановлюється сполучене освітлення. При цьому додаткове штучне освітлення застосовується не тільки в темне, але та у світлий час доби. Раціональне колірне оформлення приміщення впливає на нервову систему людини, його настрій і в кінцевому рахунку на продуктивність праці. Норма для необхідної освітленості робочого місця становить 300-500 лк.

### **2.2.3 Шум**

Для зменшення рівня шуму у джерелі його випромінювання можна використовувати пружні прокладки, що встановлюються між основою

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

обладнання та опорною поверхнею. Як такі прокладки застосовують матеріали, здатні поглинати вібрації—гума, повсть, пробка, а також спеціальні амортизатори.

Щоб знизити шум від настільних пристроїв, рекомендується використовувати м'які килимки із синтетичних матеріалів. Крім того, для додаткового поглинання вібрацій на ніжки столів, де розташована техніка, можна встановлювати прокладки з гуми або повсті завтовшки 6–8 мм.

Оптимальний рівень шуму для комфортної розумової праці, що потребує концентрації, не повинен перевищувати 50 дБ. Дотримання цих рекомендацій сприяє створенню сприятливого робочого середовища, яке мінімізує вплив сторонніх звуків і дозволяє зосередитися на виконанні завдань.

#### **2.2.4 Мікроклімат**

В процесі трудової діяльності людина знаходиться в постійній тепловій взаємодії з виробничим середовищем. Посилення енерговитрат і обміну речовин, при виконанні роботи викликає в організмі працівника збільшення теплотворення, яке відображається на його терморегуляції.

Одним з важливих складових мікроклімату є концентрація іонів в повітрі робочої зони. Дослідження показали, що в процесі роботи ВДТ протягом зміни концентрація іонів в повітрі робочої зони користувачів зазнає значні зміни.

Для забезпечення оптимальних мікрокліматичних умов в будь-який період року для приміщень в яких розташовані комп'ютеризовані робочі місця повинно бути виконано:

- раціональне розміщення технологічного обладнання (обладнання яке є джерелом тепла, бажано розміщувати безпосередньо біля зовнішніх стін будівлі і в одну низку на такій відстані один від одного, щоб теплові потоки від них не перехрещувалися на робочих місцях);
- опалювання і кондиціонування повітря (найпоширеніші способи нормалізації мікроклімату у виробничих приміщеннях, забезпечують нормальні теплові умови в холодний період року у великогабаритних і полегшених промислових будівлях);

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

- раціоналізація режимів праці і відпочинку (досягається скороченням тривалості робочого часу за рахунок додаткових перерв, створенням умов для ефективного відпочинку в приміщеннях з нормальними метеорологічними умовами); теплоізоляція обладнання і захисних екранів (як теплоізоляційні матеріали широко використовують:
  - азбест, азбоцемент, мінеральну вату, склотканина , керамзит, пінопласт);для підтримки допустимих значень мікроклімату і концентрації позитивних і негативних іонів необхідно передбачити установки або прилади зволоження та / або штучної іонізації, кондиціонування повітря.

### 2.2.5 Вимоги до організації робочого місця працівника

Робоче місце програміста має бути організоване таким чином, щоб забезпечити як безпеку, так і високу продуктивність праці. Важливим аспектом є правильне розташування всіх його елементів відповідно до антропометричних, фізичних і психологічних вимог. Врахування характеру роботи також має ключове значення для створення комфортних умов.

Ергономічні принципи проектування таких робочих місць включають оптимальну висоту робочої поверхні, достатній простір для ніг та продумане розташування документів і обладнання. Важливими факторами є наявність підставок для документів, можливість їх зручного розміщення, а також правильна дистанція від очей користувача до екрану, клавіатури та інших робочих елементів.

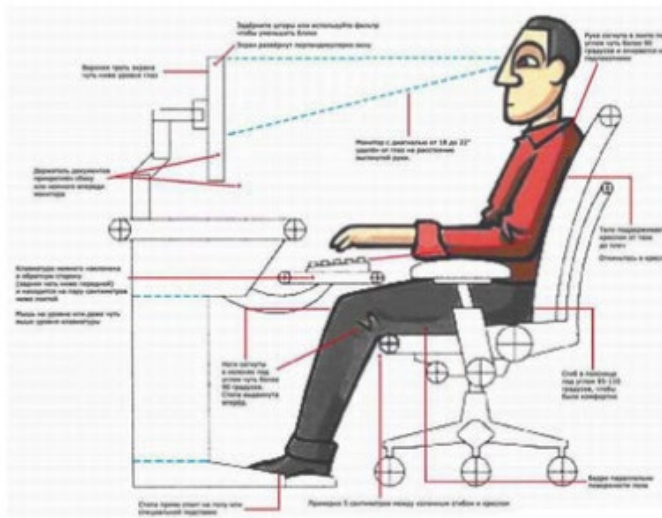


Рисунок 2.1. Організація робочого місця оператора

Комфорт програміста також залежить від характеристик робочого крісла, його регульованості та підтримки постави, а також від типу та покриття поверхні робочого столу. Дотримання цих вимог сприяє зниженню навантаження на організм і покращує загальну ефективність роботи.

## 2.6 Електробезпека

Обладнання ЕОМ, що належить до електричних установок, може становити значну небезпеку для людини. Під час експлуатації або проведення профілактичних робіт можливий контакт із струмоведучими елементами, що перебувають під напругою. Особливість електроустановок полягає в тому, що пошкоджені провідники або корпуси обладнання, які опинилися під напругою через пробій ізоляції, не мають видимих чи звукових сигналів попередження. Реакція людини на електричний струм виникає лише тоді, коли струм проходить через її тіло.

## 2.7 Пожежна безпека

Пожежна безпека—це стан об'єкта, за якого знижено ризик виникнення та поширення пожежі, а також мінімізовано вплив небезпечних факторів на людей і майно. Забезпечення пожежної безпеки приміщень з електричними мережами регулюється нормами ГОСТ 12.1.033-81 та ГОСТ 12.1.004-85, які встановлюють вимоги до безпечної експлуатації електрообладнання.

Робота оператора ЕОМ повинна проводитися у приміщенні, що відповідає категорії Д пожежної безпеки—тобто середовищі, де знаходяться негорючі речовини та матеріали у холодному стані. Для протипожежного захисту такі приміщення оснащуються вуглекислотними або порошковими вогнегасниками.

- У разі виникнення пожежі важливо:
- негайно відключити електроживлення, щоб уникнути додаткових ризиків.
- Викликати пожежну службу за номером 101.
- Евакуювати людей відповідно до плану евакуації.
- Розпочати ліквідацію пожежі, використовуючи засоби пожежогасіння.

Дотримання цих заходів сприяє створенню безпечного робочого простору та ефективному реагуванню у випадку надзвичайних ситуацій.

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

## ВИСНОВКИ

У ході виконання кваліфікаційної роботи було досліджено модифікований алгоритм розподілу навантаження для контролерів програмно-визначуваних мереж (SDN). Основна увага приділялася як математичному моделюванню алгоритму, так і його практичній реалізації у вигляді застосунку, що дозволяє моделювати роботу мережі за допомогою графічного представлення.

Проведені експерименти показали, що модифікований алгоритм забезпечує кращу продуктивність при розгортанні контролерів, порівняно з традиційними методами, зокрема з алгоритмами k-середніх і РОСО. Вимірювання часу вирішення задачі, представлені у таблицях та відповідних графіках, свідчать про зниження обчислювальних витрат при збільшенні кількості вузлів мережі. Це дозволяє ефективно масштабувати систему навіть для великих мереж. Аналіз затримок розподілу сервісного трафіку показав, що модифікований алгоритм забезпечує більш низькі значення затримок у порівнянні з конкуруючими методами. Значення, отримані в результаті аналізу насичення топології контролерами (як при умовах обмеження трафіку, так і при обмеженні кількості вузлів), свідчать про те, що модифікований алгоритм забезпечує більш рівномірне розподілення навантаження всередині мережі. Це значною мірою сприяє зниженню ризику перевантаження окремих елементів та підвищенню загальної стійкості мережевої архітектури.

Розроблений застосунок адаптовано для роботи під операційною системою Windows, що забезпечує користувачам можливість редагування, збереження та аналізу мережевих графів. Використання об'єктно-орієнтованої мови Java та бібліотеки JavaFX забезпечило високу гнучкість та масштабованість програми, а також дає можливість подальшого розширення її функціональності за рахунок додавання нових алгоритмів розподілу навантаження.

Отримані результати моделювання свідчать про поліпшені показники швидкості обчислень, зниження затримок у розподілі навантаження, а також більш ефективне насичення топології мережі, підтверджують доцільність впровадження даного методу в сучасні SDN-системи.

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

# ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Коваль Є.М. Програмно-визначувані мережі: сучасні підходи та технології / Є.М. Коваль. – Київ: ТОВ «Інфотех», 2022. – 312 с.
2. Литвинов І.П. Математичне моделювання та оптимізація мережевих систем / І.П. Литвинов. – Харків: Харківський національний університет імені В. Н. Каразіна, 2020. – 278 с.
3. Мельник Т.О. SDN: інноваційні підходи до управління мережами / Т.О. Мельник. – Одеса: Одеський національний університет, 2021. – 245 с.
4. Сидоренко В.П. Оптимізація мережевих систем: методи розподілу навантаження / В.П. Сидоренко. – Львів: Видавничий дім Львівського національного університету, 2023. – 310 с.
5. Шевченко О.М. Телекомунікаційні технології: сучасний стан і перспективи розвитку / О.М. Шевченко. – Київ: ТОВ «ТЕРРА», 2021. – 350 с.
6. Новак В.С. Сучасні тенденції розподілу навантаження в мережевих системах / В.С. Новак. – Харків: «Науковий світ», 2022. – 260 с.
7. Петрова Л.А. Алгоритми кластеризації в програмно-визначуваних мережах / Л.А. Петрова. – Львів: Видавництво ЛПІ, 2023. – 225 с.
8. Гуменюк О.М. Моделювання мережевих систем: методи аналізу та оптимізації / О.М. Гуменюк. – Київ: «Наукова думка», 2022. – 300 с.
9. Власенко М.Ю. Інформаційні технології в телекомунікаціях: сучасні підходи / М.Ю. Власенко. – Одеса: Одеський національний університет імені І.І. Мечникова, 2024. – 280 с.
10. Романчук І.Б. Теорія графів та її застосування в комп'ютерних мережах / І.Б. Романчук. – Львів: «Львівський Політехнічний Інститут», 2023. – 230 с.
11. Інтернет-ресурс: «SDN у сучасних телекомунікаційних системах» [Електронний ресурс]. – Режим доступу: <http://www.sdn.ua/article/sdn-som> (Дата звернення: 01.04.2025).
12. Інтернет-ресурс: «Алгоритми розподілу навантаження для програмно-визначуваних мереж» [Електронний ресурс]. – Режим доступу: <https://www.it-tech.com.ua/algorithmy-rozpodilu-sdn> (Дата звернення: 01.04.2025).

					<b>БКС 29. 04 000. 00 КРБ ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

## ДОДАТОК А. Фрагмент коду мовою Java застосунку для моделювання роботи SDN

```
package sdn.model;
import java.util.Arrays;
import java.util.LinkedList;
/**
 * Клас SDNLoadBalancer виконує моделювання розподілу навантаження
 * шляхом кластеризації мережесвих вузлів згідно з алгоритмом,
 * що аналізує локальну щільність ( $\rho$ ) та відстань до вузла з більшою щільністю ( $\delta$ ).
 */
public class SDNLoadBalancer {
    private int[][] links; // Матриця зв'язків (не використовується прямо для щільності)
    private double[][] distances; // Матриця відстаней між вузлами
    private double[] localDensity; // Локальна щільність  $\rho$  для кожного вузла
    private double[] delta; // Відстань  $\delta$  до найближчого вузла з вищою щільністю
    private int[] clusterAssignment; // Призначення вузлів до кластерів (ідентифікатор кластера)
    private int numNodes; // Кількість вузлів у графі
    private double dc; // Поріг відстані для врахування сусідніх з'єднань у розрахунку щільності
    /**
     * Конструктор моделі.
     * @param links матриця зв'язків між вузлами
     * @param distances матриця відстаней між вузлами
     * @param dc максимальна відстань для врахування сусідніх вузлів у розрахунку щільності
     */
    public SDNLoadBalancer(int[][] links, double[][] distances, double dc) {
        this.links = links;
        this.distances = distances;
        this.numNodes = links.length;
        this.dc = dc;
        this.localDensity = new double[numNodes];
        this.delta = new double[numNodes];
        this.clusterAssignment = new int[numNodes];
        Arrays.fill(clusterAssignment, -1); // Спочатку вузли не приписані до жодного кластеру.
        computeLocalDensity();
        computeDelta();
        assignClusters();
    }
    /**
     * Обчислює локальну щільність  $\rho$  для кожного вузла.
     * Для кожного вузла сумується кількість інших вузлів,
     * відстань до яких менша за  $dc$ .
     */
    private void computeLocalDensity() {
        for (int i = 0; i < numNodes; i++) {
            double rho = 0.0;
            for (int j = 0; j < numNodes; j++) {
                if (i != j && distances[i][j] < dc) {
                    // За умовою  $kcl = 1$ , якщо  $(dc - distance) > 0$ .
                    rho += 1.0;
                }
            }
            localDensity[i] = rho;
        }
    }
}
```

```

/**
 * Обчислює  $\delta$  для кожного вузла – мінімальну відстань до вузла,
 * який має вищу локальну щільність.
 * Якщо вузол має вищу щільність,  $\delta$  встановлюється рівним макс. відстані у мережі.
 */
private void computeDelta() {
    double maxDistance = 0;
    // Обчислення максимальної відстані між вузлами
    for (int i = 0; i < numNodes; i++) {
        for (int j = 0; j < numNodes; j++) {
            if (distances[i][j] > maxDistance) {
                maxDistance = distances[i][j];
            }
        }
    }
    // Обчислення  $\delta$  для кожного вузла
    for (int i = 0; i < numNodes; i++) {
        double minDelta = Double.MAX_VALUE;
        for (int j = 0; j < numNodes; j++) {
            if (localDensity[j] > localDensity[i] && distances[i][j] < minDelta) {
                minDelta = distances[i][j];
            }
        }
        // Якщо вузол має максимальну щільність, призначаємо  $\delta$  максимальній відстані
        if (minDelta == Double.MAX_VALUE) {
            delta[i] = maxDistance;
        } else {
            delta[i] = minDelta;
        }
    }
}
/**
 * Вузли з високими значеннями  $\delta$  (понад певний поріг) вважаються центрами кластерів,
 * а решта вузлів приєднуються до найближчого з центрів.
 */
private void assignClusters() {
    double deltaThreshold = computeDeltaThreshold();
    int clusterId = 0;
    // Визначення центрів кластерів
    for (int i = 0; i < numNodes; i++) {
        if (delta[i] >= deltaThreshold) {
            clusterAssignment[i] = clusterId;
            clusterId++;
        }
    }
    // Призначення решти вузлів до відповідних кластерів:
    for (int i = 0; i < numNodes; i++) {
        if (clusterAssignment[i] == -1) {
            int nearestCenter = -1;
            double minDist = Double.MAX_VALUE;
            for (int j = 0; j < numNodes; j++) {
                if (localDensity[j] > localDensity[i] && clusterAssignment[j] != -1 &&
                    distances[i][j] < minDist) {
                    minDist = distances[i][j];
                    nearestCenter = j;
                }
            }
        }
    }
}

```

```

    }
    if (nearestCenter != -1) {
        clusterAssignment[i] = clusterAssignment[nearestCenter];
    } else {
// Якщо вузол не знайшов підходящого сусіда, він самостійно створює новий кластер
        clusterAssignment[i] = clusterId;
        clusterId++;
    }
}
}
}
}
}
}
**
* Обчислює поріг для  $\delta$ . Наприклад, можна використати середнє значення  $\delta$  по всім вузлам.
* @return порогове значення  $\delta$ .
*/
private double computeDeltaThreshold() {
    double sum = 0;
    for (double d : delta) {
        sum += d;
    }
    return sum / numNodes;
}
}
**
* Виводить результати кластеризації: для кожного вузла показує локальну щільність,
*  $\delta$  та номер кластера.
*/
public void printClusterAssignments() {
    System.out.println("Вузол\tЩільність  $\rho$ \t $\delta$ \tКластер");
    for (int i = 0; i < numNodes; i++) {
        System.out.printf("%d\t%.2f\t%.2f\t%d\n", i, localDensity[i], delta[i],
clusterAssignment[i]);
    }
}
}
// Метод для отримання призначення кластерів
public int[] getClusterAssignments() {
    return clusterAssignment;
}
}
// Головний метод для демонстрації роботи моделі
public static void main(String[] args) {
    // Приклад: 5 вузлів, матриця зв'язків не використовується прямо для щільності,
// але для сумісності параметр links передається як масив.
    int numNodes = 5;
    int[][] links = new int[numNodes][numNodes];
    // Матриця відстаней між вузлами
    double[][] distances = {
        {0, 10, 15, 20, 25},
        {10, 0, 35, 25, 30},
        {15, 35, 0, 30, 20},
        {20, 25, 30, 0, 15},
        {25, 30, 20, 15, 0}
    };
    // Встановлюємо максимально допустиму відстань dc
    double dc = 20;
    SDNLoadBalancer balancer = new SDNLoadBalancer(links, distances, dc);
    balancer.printClusterAssignments();
}
}

```

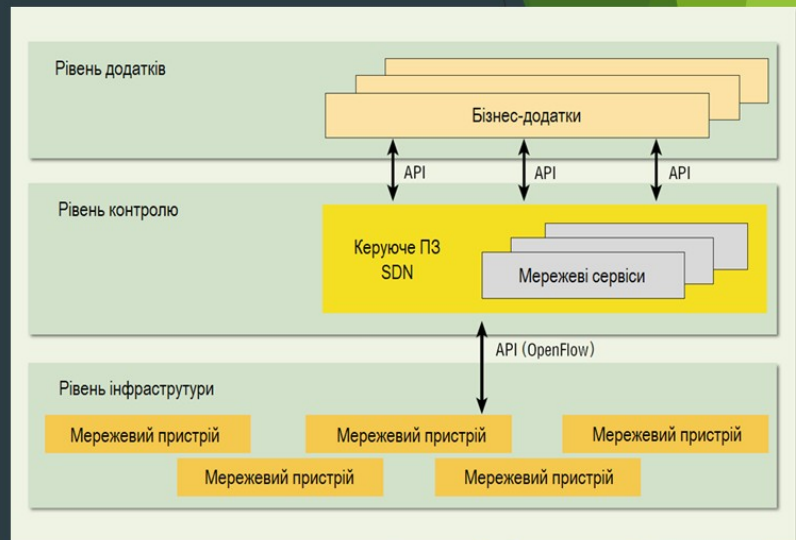
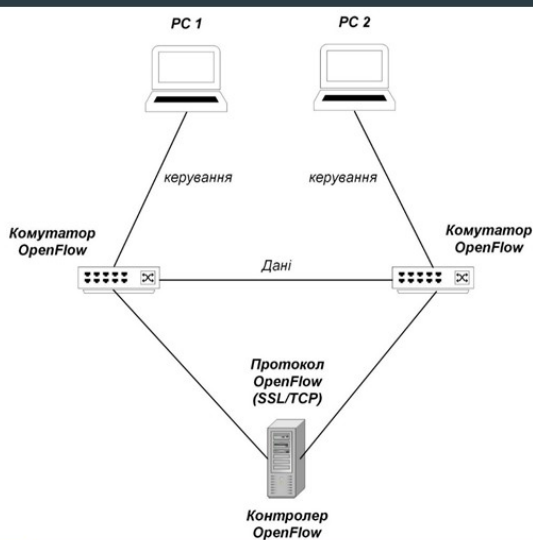
# ДОДАТОК Б. Слайди мультимедійної презентації



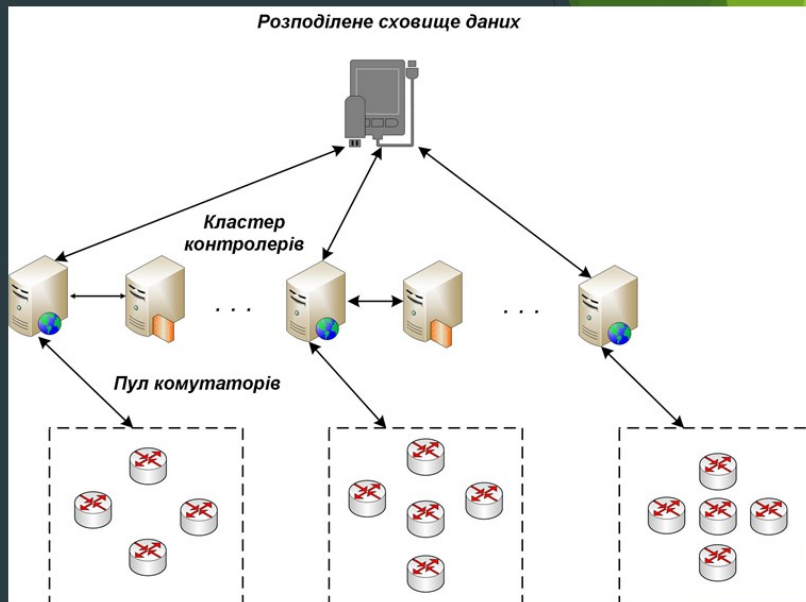
## Аналіз ефективності алгоритмів розподілу навантаження до контролерів SDN

Васильюк Олександр, гр.2БКС-29

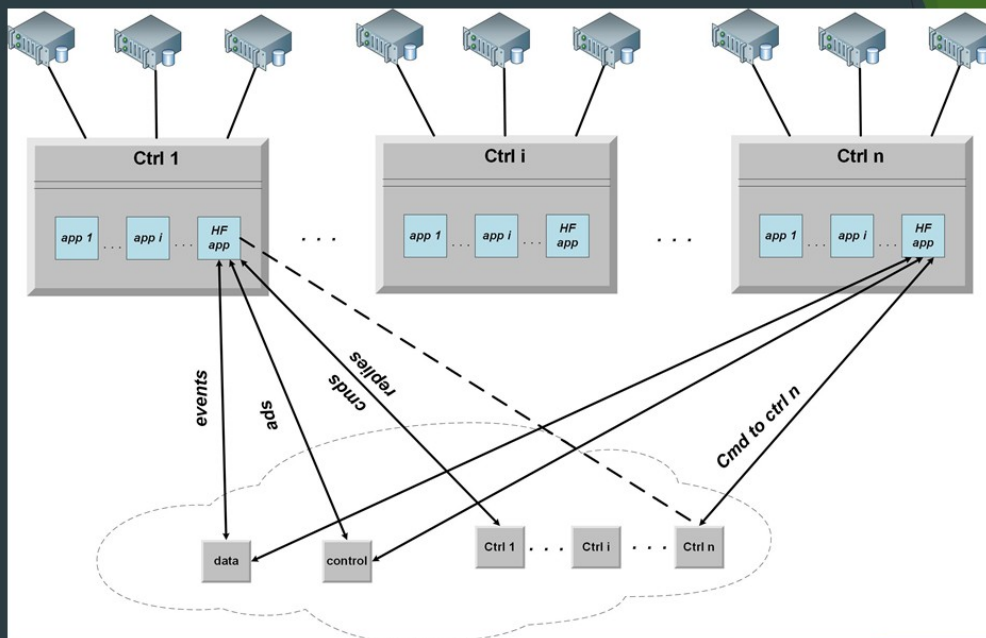
### Архітектура мережі SDN за протоколом OpenFlow



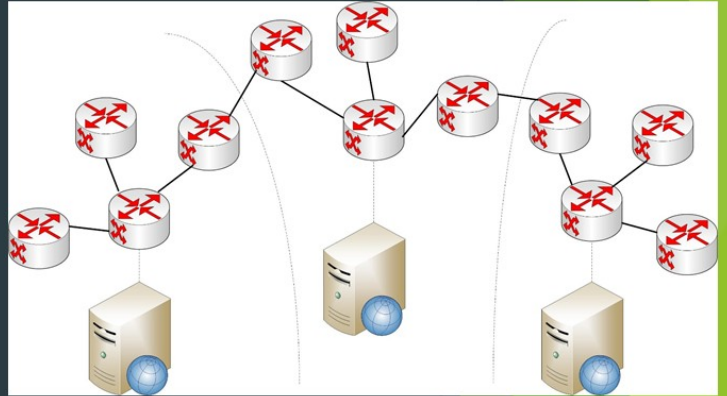
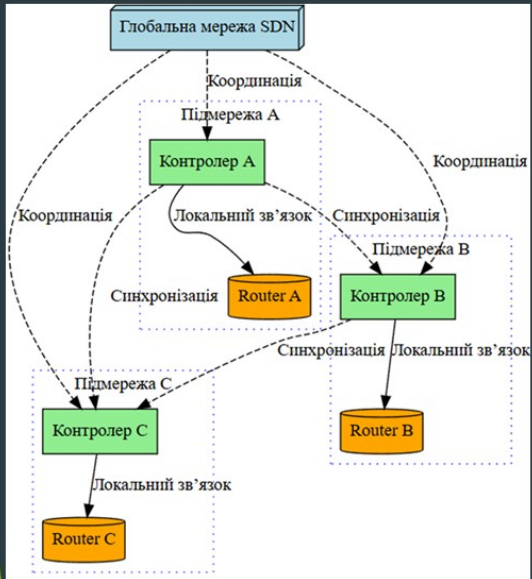
# SDN-мережа із розподіленими контролерами



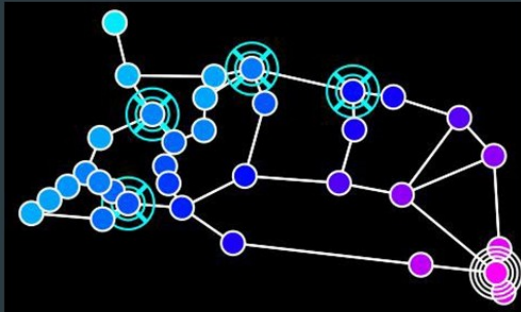
# Архітектура технології HyperFlow



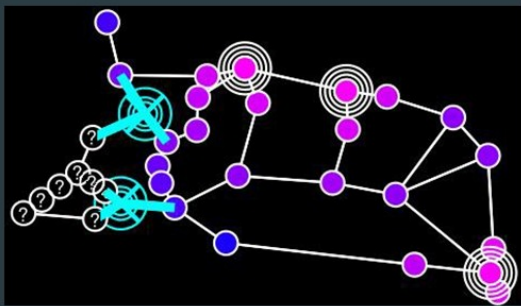
# Кластеризація у SDN-мережі



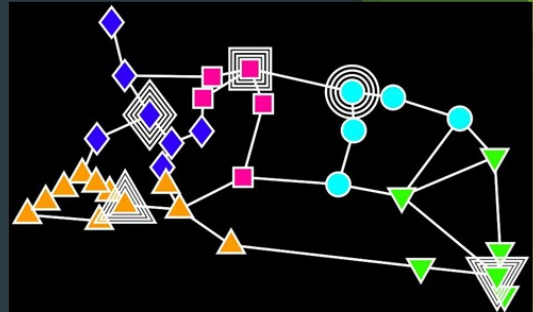
Вихід з ладу деяких мережевих контролерів



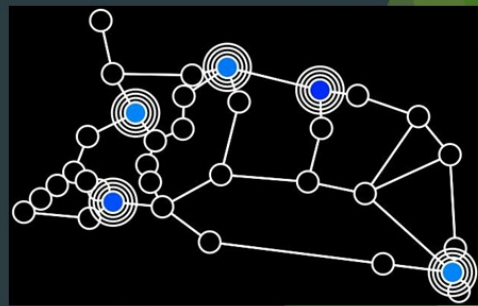
Вихід з ладу деяких вузлів та з'єднань у мережі



Організація мережі без урахування навантаження контролерів



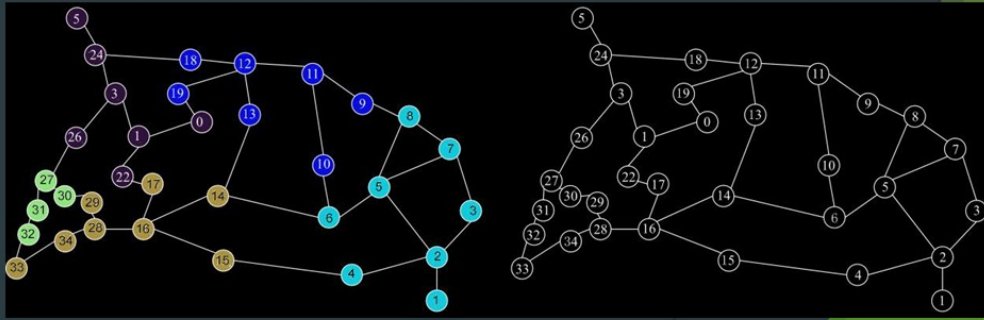
Затримка передачі між мережевими контролерами



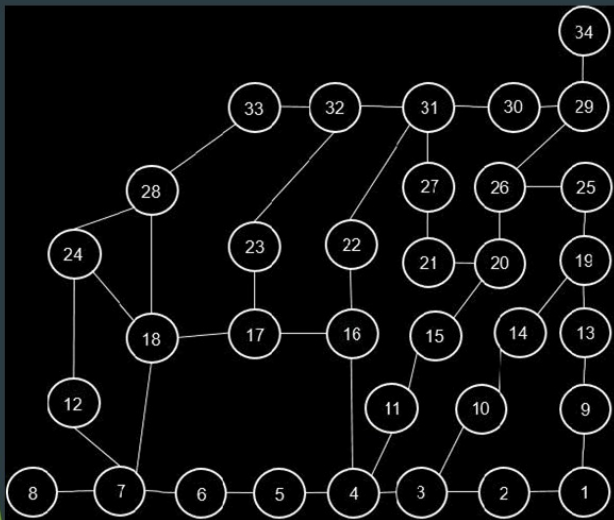
## Розташування одного і п'яти контролерів для оптимальної середньої та максимальної затримок



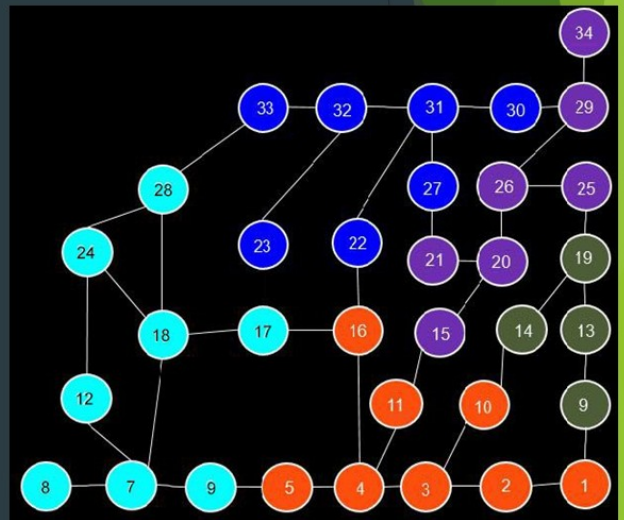
## Кластеризація мережі за алгоритмом на основі щільності



## Приклад графу для моделювання SDN-мережі



## Моделювання модифікованого алгоритму розподілу навантаження



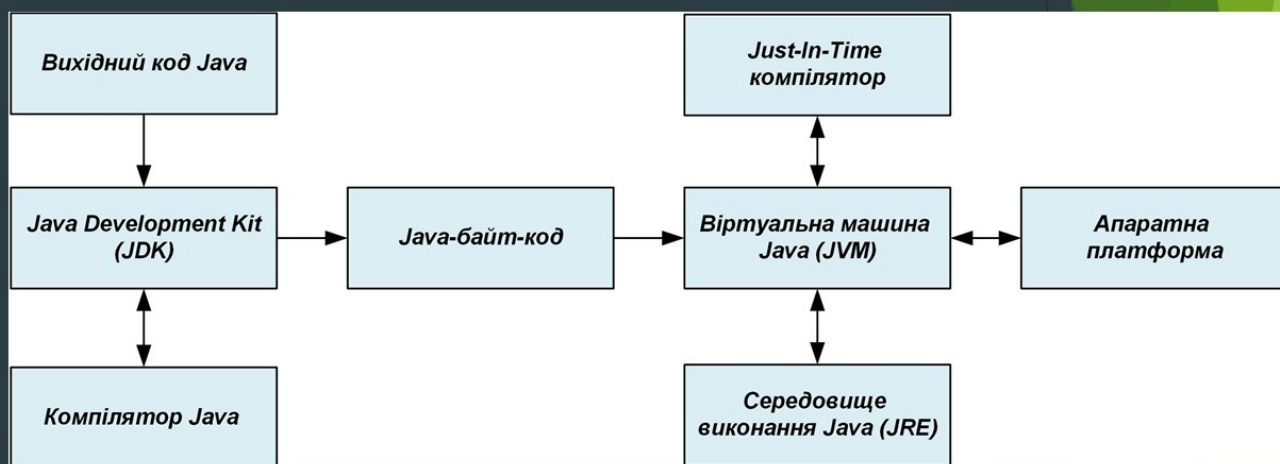
## Результати моделювання з використанням модифікованого алгоритму

<i>id</i> вузла	$\rho$	$\delta$	<i>id</i> вузла	$\rho$	$\delta$
1	4	1	18	9	3
2	8	1	19	6	2
3	8	1	20	7	1
4	10	3	21	5	1
5	6	1	22	7	1
6	6	1	23	6	1
7	8	1	24	6	1
8	4	1	25	6	1
9	4	1	26	8	3
10	5	1	27	6	1
11	6	1	28	7	1
12	6	1	29	6	1
13	5	1	30	7	1
14	5	1	31	9	3
15	5	1	32	8	1
16	9	1	33	6	1
17	8	1	34	3	1

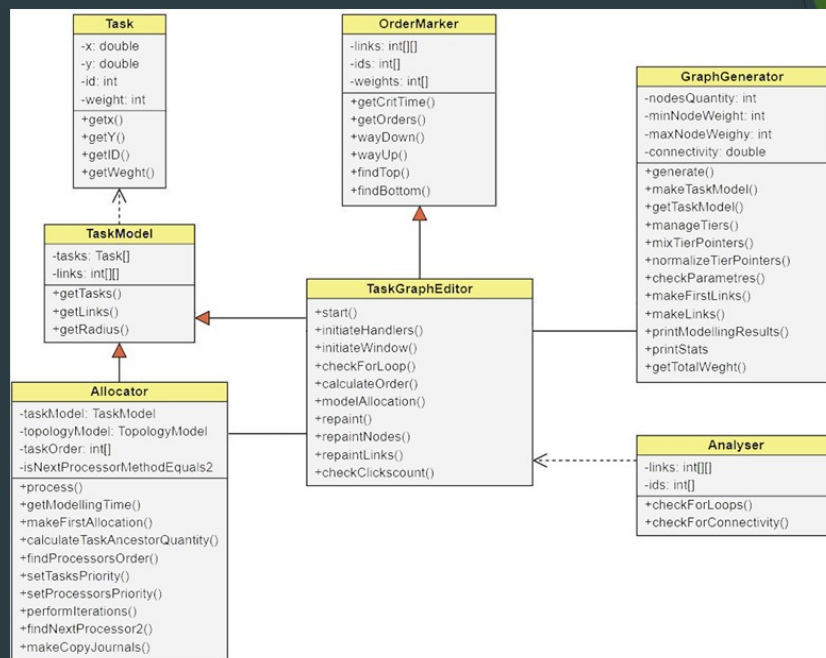
## Розподіл локальної щільності по відстані до вузлів із більшою щільністю



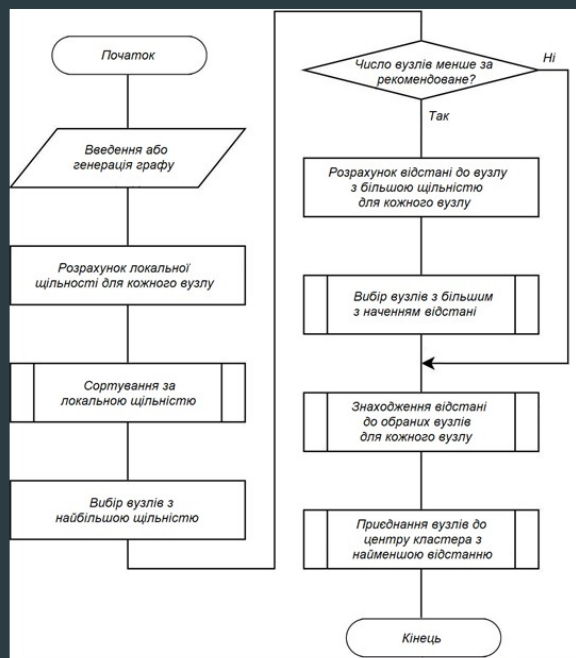
## Схема компіляції програми (мова Java)



# Діаграма класів застосунку для моделювання роботи SDN-мережі



# БСА застосунку для моделювання роботи SDN-мережі



## Інтерфейс застосунку для моделювання роботи SDN-мережі

The screenshot shows the application window titled "Моделювання алгоритму кластеризації з розподілом щільності розташування контролерів". The main area displays a network graph with 34 nodes (numbered 1-34) and edges. A toolbar on the left contains buttons for "Додати вузол", "Видалити вузол", "Пересунути вузол", "Додати зв'язок", "Видалити зв'язок", and "Змінити вагу зв'язку". Below the toolbar is a "Ввести графік:" field with the value "1".

## Результати обчислення параметрів моделювання

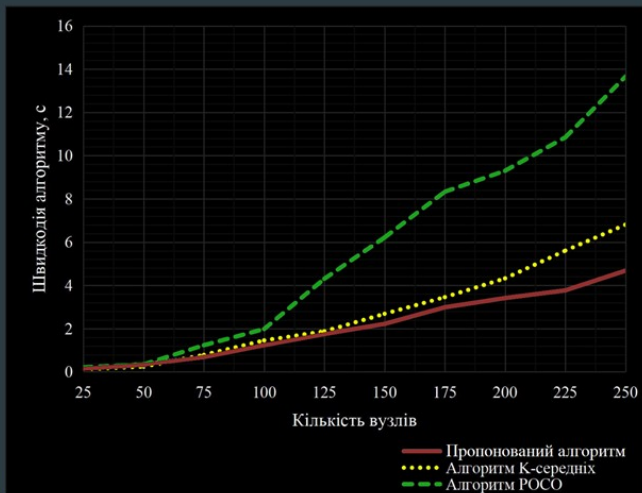
The screenshot shows the application window displaying the results of the calculation. It features three tables, each with three columns: "№" (Node ID), "ρ" (Density), and "dc" (Distance to Controller). The data is as follows:

№	ρ	dc
1	4	1
2	8	1
3	8	1
4	10	3
5	6	1
6	6	1
7	8	1
8	4	1
9	4	1
10	5	1
11	6	1
12	6	1

№	ρ	dc
13	5	1
14	5	1
15	5	1
16	9	1
17	8	1
18	9	3
19	6	2
20	7	1
21	5	1
22	7	1
23	6	1
24	6	1

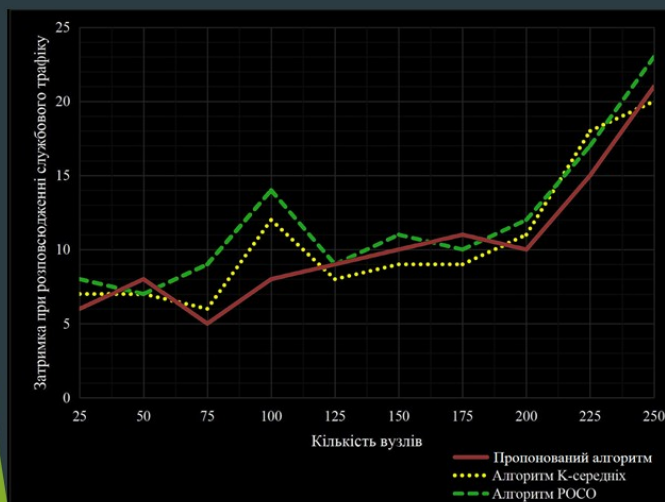
№	ρ	dc
25	6	1
26	8	3
27	6	1
28	7	1
29	6	1
30	7	1
31	9	3
32	8	1
33	6	1
34	3	1

## Залежність продуктивності алгоритмів розподілу навантаження до контролерів SDN від кількості вузлів мережі



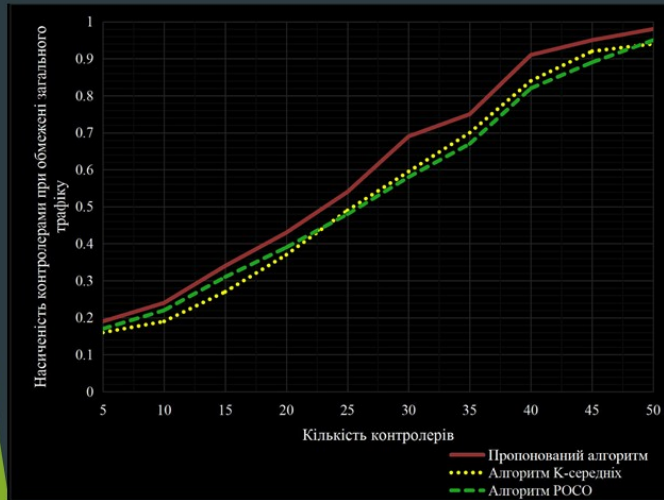
Кількість вершин	Алгоритм моделювання		
	Модифікований	k-середніх	РОСО
25	0,145	0,121	0,213
50	0,345	0,249	0,346
75	0,687	0,784	1,234
100	1,237	1,456	1,983
125	1,751	1,876	4,305
150	2,221	2,687	6,217
175	2,987	3,456	8,329
200	3,415	4,324	9,299
225	3,769	5,607	10,845
250	4,681	6,809	13,657

## Залежність затримки розподілу сервісного трафіку від кількості вузлів мережі



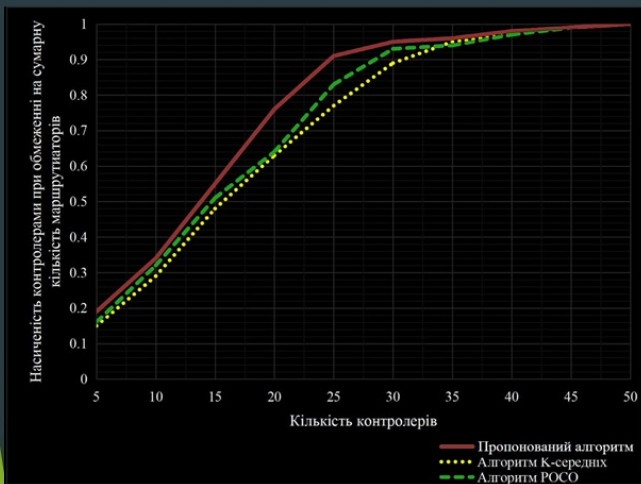
Кількість вершин	Алгоритм моделювання		
	Модифікований	k-середніх	РОСО
25	6	7	8
50	8	7	7
75	5	6	9
100	8	12	14
125	9	8	9
150	10	9	11
175	11	9	10
200	10	11	12
225	15	18	17
250	21	20	23

## Насичення мережі контролерами при обмеженні трафіку маршрутизаторів мережі



Кількість контролерів	Алгоритм моделювання		
	Модифікований	k-середніх	РОСО
5	0,19	0,16	0,17
10	0,24	0,19	0,22
15	0,34	0,27	0,31
20	0,43	0,37	0,39
25	0,54	0,49	0,48
30	0,69	0,595	0,58
35	0,75	0,7	0,67
40	0,91	0,84	0,82
45	0,95	0,92	0,89
50	0,98	0,94	0,95

## Насичення топології контролерами при обмеженні на сумарну кількість маршрутизаторів



Кількість вершин	Алгоритм моделювання		
	Модифікований	k-середніх	РОСО
5	0,19	0,15	0,16
10	0,34	0,29	0,32
15	0,55	0,48	0,51
20	0,76	0,63	0,64
25	0,91	0,77	0,83
30	0,95	0,89	0,93
35	0,96	0,95	0,94
40	0,98	0,97	0,97
45	0,99	0,99	0,99
50	1	1	1

**РЕЦЕНЗІЯ**

на кваліфікаційну роботу здобувача (здобувачки) освіти  
відділення комп'ютерних систем

Василюка Олександра Володимировича

(прізвище, ім'я та по батькові)

Спеціальність 123 "Комп'ютерна інженерія"

Освітньо-професійна програма «Комп'ютерна інженерія»

Керівник кваліфікаційної роботи Закроєв Юрій Михайлович

(прізвище, ім'я та по батькові)

Тема кваліфікаційної роботи Аналіз ефективності алгоритмів розподілу навантаження до контролерів SDN

Обсяг розрахунково-пояснювальної записки 69 сторінок

Обсяг графічної (презентаційної) частини 18 аркушів (слайдів)

**ХАРАКТЕРИСТИКА КВАЛІФІКАЦІЙНОЇ РОБОТИ**

а) заключення про ступінь відповідності виконаного кваліфікаційної роботи завданню

Представлена на рецензію кваліфікаційна робота бакалавра повністю відповідає меті випускної роботи та технічному завданню. Тематика кваліфікаційної роботи є актуальною для своєї галузі та присвячена аналізу ефективності алгоритмів розподілу навантаження до контролерів SDN.

б) характеристика виконання кожного розділу кваліфікаційної роботи

Кваліфікаційна робота складається зі вступу, двох розділів, висновків, переліку використаних джерел. У основному розділі виконано аналіз архітектури SDN-мереж; аналіз алгоритмів розподілу навантаження; огляд методів розподілу навантаження до контролерів SDN; адаптація математичної моделі розподілу навантаження; реалізація модифікованого алгоритму розподілу навантаження; розробка діаграми класів та блок-схеми алгоритму застосунку для моделювання роботи SDN; реалізація застосунку для моделювання роботи SDN

в) оцінка якості виконання пояснювальної записки та графічної частини кваліфікаційної роботи

Графічна частина виконана на достатньо високому рівні у вигляді презентації із використанням офісного пакету Microsoft PowerPoint та Visio. Пояснювальна записка виконана охайно та у відповідності до норм оформлення документів із використанням офісного пакету Microsoft Word. Загальна якість виконання документації – добра, академічного плагіату ідей у роботі не виявлено

г) перелік позитивних якостей кваліфікаційної роботи

*У роботі проаналізовано як класичні алгоритми розподілу навантаження (наприклад, метод к-середніх), так і сучасні підходи (кластеризація за щільністю, алгоритми, орієнтовані на стійкість до відмов), що дає можливість побачити повну картину проблеми*

д) основні недоліки кваліфікаційної роботи

*Експериментальна частина базується переважно на симуляційному аналізі з використанням генераторів випадкових мережевих графів. Для повнішої оцінки ефективності пропонуваного алгоритму доцільно провести додаткові дослідження або порівняльні тести на даних реальних SDN-середовищ.*

Оцінка розрахункової частини Відмінно

Оцінка графічної частини Добре

Загальна оцінка Відмінно

Прізвище, ім'я, по батькові рецензента к.т.н. Шibaєва Наталія Олегівна

Місце роботи і посада рецензента Національний університет «Одеська політехніка», доцент кафедри інформаційних технологій

Підпис:

« 23 »

2025 р.



ВСП «Одеський технічний фаховий коледж ОНТУ»

## ВІДГУК

керівника про кваліфікаційну роботу бакалавра

*Василюка Олександра Володимировича*

(прізвище, ім'я та по батькові)

Спеціальність 123 "Комп'ютерна інженерія"

Тема кваліфікаційної роботи Аналіз ефективності алгоритмів розподілу навантаження до контролерів SDN

### ХАРАКТЕРИСТИКА КВАЛІФІКАЦІЙНОЇ РОБОТИ

а) Обсяг і якість виконання роботи (графічного матеріалу і розрахунково-пояснювальної записки) Випускна робота виконана відповідно технічному завданню. Пояснювальна записка до випускної роботи містить 69 сторінок. У пояснювальній записці розглянуто проблему ефективності алгоритмів розподілу навантаження до контролерів SDN, кластеризації та розташування контролерів програмно-визначуваних мереж. Графічна частина складається з 18 слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра, розробку виконано у повному обсязі.

б) Самостійність роботи Протягом виконання випускної бакалаврської роботи Василюк Олександр поступово та послідовно виконувала всі етапи, проявила ініціативу у створенні загальної концепції та реалізації випускної роботи. Всі роботи вона виконувала самостійно, з оглядом на рекомендації керівника.

в) Теоретична підготовка здобувача освіти \_\_\_\_\_

*Василюк Олександр під час роботи над випускною бакалаврською роботою вивчив достатню кількість літературних джерел за даною тематикою.*

*Вважаю, що теоретична підготовка здобувача освіти добра і він готовий до захисту роботи.*

г) Вміння розв'язувати виробничі і конструкторські питання на базі останніх досліджень науки і техніки, передових методів виробництва \_\_\_\_\_

*Під час виконання роботи Василюк Олександр мав змогу самостійно приймати окремі рішення з виконання програмної частини роботи та показав вміння організовано працювати над поставленою задачею, складати та оформлювати презентацію проекту, користуючись сучасними комп'ютерними програмними засобами, такими як Java JRE 8 та Java SDK.*

Оцінка розрахункової частини \_\_\_\_\_ *Відмінно*

Оцінка графічної частини \_\_\_\_\_ *Добре*

Загальна оцінка \_\_\_\_\_ *Відмінно*

Прізвище, ім'я, по батькові \_\_\_\_\_ *Закроєв Юрій Михайлович*

Місце роботи і посада керівника роботи \_\_\_\_\_ *ТОВ "БІГ ВОШ", директор*

Підпис \_\_\_\_\_

« 20 »

06

2024 р.

**ДОЗВІЛ  
НА РОЗМІЩЕННЯ  
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

**Василюк О.В.**,  
здобувач освіти гр. 2БКС-29, та

**Закроєв Ю.М.**,  
керівник дипломного проекту,

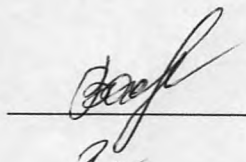
не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи бакалавра на тему:

**«Аналіз ефективності алгоритмів розподілу навантаження до контролерів SDN» (автор роботи – Василюк О.В., керівник роботи – Закроєв Ю.М.)**

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2025 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

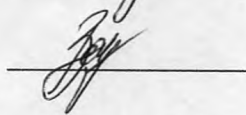
Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець



/ Василюк О.В. /

Керівник



/ Закроєв Ю.М. /

«16» червня 2025 р.

# Д О В І Д К А


кафедри комп'ютерної інженерії  
про допуск до захисту кваліфікаційної роботи  
здобувача (здобувачки) освіти II курсу  
відділення комп'ютерних систем групи 2БКС-29

*Василюка Олександра Володимировича*

на тему Аналіз ефективності алгоритмів розподілу навантаження  
до контролерів SDN

Висновок відповідальної особи за проведення нормоконтролю:

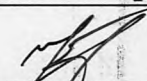
пояснювальна записка до кваліфікаційної роботи виконана з несуттєвими  
порушеннями ДСТУ та оформлена відповідно до вимог Положення про  
дипломне проектування

  
(підпис)

20.06.2025  
(дата)

Петрашова В.І.  
(П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного  
плагіату згідно звіту про перевірку від 31.05.2025 р. значення коефіцієнту  
подібності в роботі становить 11,52%, коефіцієнт цитування – 0,80%.

  
(підпис)

20.06.2025  
(дата)

Краснокутська К.Г.  
(П.І.Б.)

**Попередня експертиза (малий захист) кваліфікаційної роботи**

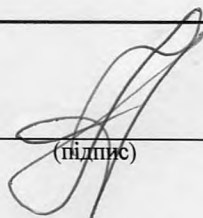
здобувача (здобувачки) освіти

Василюка О.В.  
(П.І.Б.)

проведена « 20 » червня 2025 р.

Висновки Пояснювальна записка до кваліфікаційної роботи виконана у  
повному обсязі. Випускна кваліфікаційна робота відповідає вимогам  
Положення про дипломне проектування та рекомендована до захисту.

Зав. кафедри КІ

  
(підпис)

Іванова Л.В.  
(П.І.Б.)

## Звіт подібності

## метадані

Назва організації

Odesa Technical Professional College of Odesa National University of Technology

Заголовок

Аналіз ефективності алгоритмів розподілу навантаження до контролерів SDN

Автор

Науковий керівник / Експерт

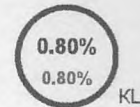
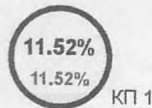
Васильюк Олександр Володимирович Закроєв Юрій Михайлович

підрозділ

Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

## Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

12738

Кількість слів

108405

Кількість символів

## Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		23
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		51

## Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копію тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

## 10 найдовших фраз

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	Копію тексту
1	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/d5a3d14f-d5cb-460f-9c49-cba3f9d50554/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/d5a3d14f-d5cb-460f-9c49-cba3f9d50554/content</a>	109 0.86 %
2	<a href="https://thelib.info/psihologiya/190888-vpliv-parametriv-mikroklimatu-na-zahvorjuvanist-koristuvachiv-personalnogo-komp-jutera/">https://thelib.info/psihologiya/190888-vpliv-parametriv-mikroklimatu-na-zahvorjuvanist-koristuvachiv-personalnogo-komp-jutera/</a>	82 0.64 %
3	<a href="https://card-file.ontu.edu.ua/bitstreams/11562741-24e6-4201-bc41-a00c8013fca1/download">https://card-file.ontu.edu.ua/bitstreams/11562741-24e6-4201-bc41-a00c8013fca1/download</a>	77 0.60 %
4	<a href="https://thelib.info/psihologiya/190888-vpliv-parametriv-mikroklimatu-na-zahvorjuvanist-koristuvachiv-personalnogo-komp-jutera/">https://thelib.info/psihologiya/190888-vpliv-parametriv-mikroklimatu-na-zahvorjuvanist-koristuvachiv-personalnogo-komp-jutera/</a>	66 0.52 %

5	<a href="https://card-file.ontu.edu.ua/bitstreams/361286d7-8a03-4221-ad05-db5133ab5f79/download">https://card-file.ontu.edu.ua/bitstreams/361286d7-8a03-4221-ad05-db5133ab5f79/download</a>	50 0.39 %
6	<a href="https://lektcii.org/2-52678.html">https://lektcii.org/2-52678.html</a>	43 0.34 %
7	<a href="https://card-file.ontu.edu.ua/bitstreams/361286d7-8a03-4221-ad05-db5133ab5f79/download">https://card-file.ontu.edu.ua/bitstreams/361286d7-8a03-4221-ad05-db5133ab5f79/download</a>	32 0.25 %
8	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/21ac499a-a9e9-4137-810c-5f21a0318048/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/21ac499a-a9e9-4137-810c-5f21a0318048/content</a>	32 0.25 %
9	<a href="https://thelib.info/psihologiya/190888-vpliv-parametriv-mikroklimatu-na-zahvorjuvanist-koristuvachiv-personalnogo-komp-jutera/">https://thelib.info/psihologiya/190888-vpliv-parametriv-mikroklimatu-na-zahvorjuvanist-koristuvachiv-personalnogo-komp-jutera/</a>	31 0.24 %
10	<a href="https://card-file.ontu.edu.ua/bitstreams/0e6c3361-ffb-4469-86a1-fe84a1fe21cd/download">https://card-file.ontu.edu.ua/bitstreams/0e6c3361-ffb-4469-86a1-fe84a1fe21cd/download</a>	24 0.19 %

### з домашньої бази даних (0.05 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Створення web-застосунку цифрового помічника з використанням Open AI 5/28/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	6 (1) 0.05 %

### з програми обміну базами даних (0.75 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Спосіб кластеризації мобільних SDN мереж 3/15/2025 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute)	46 (4) 0.36 %
2	123_Kohan_Vladyslav_Vladyslavovych 6/27/2024 Odessa I.I.Mechnikov National University (Odessa I.I.Mechnikov National University)	27 (2) 0.21 %
3	Розробка волоконно-оптичного транспортного кільця з використанням обладнання хвильового мультиплексування 12/24/2022 State University of Telecommunications (NHIT)	13 (1) 0.10 %
4	Спосіб кластеризації мережі SDN з урахуванням розподілу щільності зв'язків та потоків даних 3/15/2025 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute)	10 (1) 0.08 %

### з Інтернету (10.72 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	<a href="https://card-file.ontu.edu.ua/bitstreams/0e6c3361-ffb-4469-86a1-fe84a1fe21cd/download">https://card-file.ontu.edu.ua/bitstreams/0e6c3361-ffb-4469-86a1-fe84a1fe21cd/download</a>	587 (51) 4.61 %
2	<a href="https://thelib.info/psihologiya/190888-vpliv-parametriv-mikroklimatu-na-zahvorjuvanist-koristuvachiv-personalnogo-komp-jutera/">https://thelib.info/psihologiya/190888-vpliv-parametriv-mikroklimatu-na-zahvorjuvanist-koristuvachiv-personalnogo-komp-jutera/</a>	179 (3) 1.41 %
3	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/d5a3d14f-d5cb-460f-9c49-cba3f9d50554/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/d5a3d14f-d5cb-460f-9c49-cba3f9d50554/content</a>	109 (1) 0.86 %
4	<a href="https://card-file.ontu.edu.ua/bitstreams/11562741-24e6-4201-bc41-a00c8013fca1/download">https://card-file.ontu.edu.ua/bitstreams/11562741-24e6-4201-bc41-a00c8013fca1/download</a>	103 (4) 0.81 %
5	<a href="https://card-file.ontu.edu.ua/bitstreams/361286d7-8a03-4221-ad05-db5133ab5f79/download">https://card-file.ontu.edu.ua/bitstreams/361286d7-8a03-4221-ad05-db5133ab5f79/download</a>	102 (5) 0.80 %

6	<a href="https://leksii.org/2-52678.html">https://leksii.org/2-52678.html</a>	56 (2) 0.44 %
7	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/e86ba9fc-9135-43bb-922a-10bf0bce46b5/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/e86ba9fc-9135-43bb-922a-10bf0bce46b5/content</a>	41 (4) 0.32 %
8	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/21ac499a-a9e9-4137-810c-5f21a0318048/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/21ac499a-a9e9-4137-810c-5f21a0318048/content</a>	39 (2) 0.31 %
9	<a href="http://rma350.github.io/2012/06/13/all-pairs-shortest-path-in-parallel-with-floyd-warshall-in-java.html">http://rma350.github.io/2012/06/13/all-pairs-shortest-path-in-parallel-with-floyd-warshall-in-java.html</a>	35 (3) 0.27 %
10	<a href="https://card-file.ontu.edu.ua/bitstreams/5240e379-7721-49f0-8ee8-27140b0b473a/download">https://card-file.ontu.edu.ua/bitstreams/5240e379-7721-49f0-8ee8-27140b0b473a/download</a>	20 (2) 0.16 %
11	<a href="https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download">https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download</a>	19 (1) 0.15 %
12	<a href="https://card-file.ontu.edu.ua/bitstreams/62baa43e-b968-4993-bb54-8cf8761a89b2/download">https://card-file.ontu.edu.ua/bitstreams/62baa43e-b968-4993-bb54-8cf8761a89b2/download</a>	15 (2) 0.12 %
13	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/f5042058-9544-4ac7-bd33-42bd733c8e6b/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/f5042058-9544-4ac7-bd33-42bd733c8e6b/content</a>	14 (2) 0.11 %
14	<a href="https://card-file.ontu.edu.ua/bitstreams/f789da43-3034-4ad8-bf34-640a47414f93/download">https://card-file.ontu.edu.ua/bitstreams/f789da43-3034-4ad8-bf34-640a47414f93/download</a>	11 (1) 0.09 %
15	<a href="https://card-file.ontu.edu.ua/server/api/core/bitstreams/c5cd348b-fc64-4a25-9a5b-6cc8d62db909/content">https://card-file.ontu.edu.ua/server/api/core/bitstreams/c5cd348b-fc64-4a25-9a5b-6cc8d62db909/content</a>	11 (1) 0.09 %
16	<a href="https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download">https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download</a>	10 (1) 0.08 %
17	<a href="https://card-file.ontu.edu.ua/bitstreams/d42aac6d-ab01-4a74-b9cb-ced2a9eff719/download">https://card-file.ontu.edu.ua/bitstreams/d42aac6d-ab01-4a74-b9cb-ced2a9eff719/download</a>	8 (1) 0.06 %
18	<a href="https://essuir.sumdu.edu.ua/bitstream/123456789/95439/1/Nahornyi_masters_thesis.pdf">https://essuir.sumdu.edu.ua/bitstream/123456789/95439/1/Nahornyi_masters_thesis.pdf</a>	6 (1) 0.05 %

**Список прийнятих фрагментів (немає прийнятих фрагментів)**

ПОРЯДКОВИЙ НОМЕР

ЗМІСТ

КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»  
Освітньо-професійна програма: «Комп'ютерна інженерія» Група: 2БКС-29

КВАЛІФІКАЦІЙНА  
РОБОТА

здобувача освіти денної форми навчання БКС. 29.04.000. КРБ

ВАСИЛЮКА  
ОЛЕКСАНДРА  
ВОЛОДИМИРОВИЧА

м. Одеса  
2025 р. МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»  
Освітньо-професійна програма: «Комп'ютерна інженерія»  
Група: 2 БКС-29

ПОЯСНЮВАЛЬНА ЗАПИСКА До кваліфікаційної роботи бакалавра на тему: \_\_\_\_\_

\_\_\_\_\_ Проектний матеріал складається з  
пояснювальної записки на \_\_\_\_\_ сторінках та графічного (презентаційного) матеріалу на \_\_\_\_\_ аркушах (слайдах) Виконавець  
\_\_\_\_\_ (Василюк О.В.)

Керівник проекту \_\_\_\_\_ (Закроев Ю.М.)

Консультанти: з розділу охорони праці та техніки безпеки \_\_\_\_\_ (Чорновоп Н.І.) з нормоконтролю  
\_\_\_\_\_ (Петрашова В.І.) старший консультант \_\_\_\_\_ (Кривченко Ю.В.) До  
захисту допущений \_\_\_\_\_