*DOI: 10.25140/2411-5363-2023-3(33)-141-149*
*УДК 004.732*

***Вадим Щур[1], Юрій Кулаков[2]***

[1]аспірант, асистент кафедри обчислювальної техніки
Національного технічного університету України
«Київського політехнічного інституту імені Ігоря Сікорського» (Київ, Україна)
**E-mail:** vadimmshchur@gmail.com. **ORCID**: https://orcid.org/0000-0001-8925-4813

[2] професор, доктор технічних наук
Національного технічного університету України
«Київського політехнічного інституту імені Ігоря Сікорського» (Київ, Україна)
**E-mail:** ya.kulakov@gmail.com **ORCID**: https://orcid.org/0000-0002-8981-5649

# ОПТИМІЗОВАНИЙ АДАПТИВНИЙ МЕТОД БАЛАНСУВАННЯ НАВАНТАЖЕННЯ В МЕРЕЖАХ SDN НА ОСНОВІ ANT COLONY OPTIMIZATION

*У сучасних програмно-визначених мережах (SDN) забезпечення ефективного балансування навантаження є ключовим завданням для оптимального використання ресурсів та забезпечення стабільної якості обслуговування. Для досягнення цих цілей у цій статті ми пропонуємо покращений метод балансування навантаження для мереж SDN, заснований на мурашиному алгоритмі з динамічним налаштуванням параметрів.*

*Запропонований метод демонструє високу ефективність в умовах змінної динаміки мережі та різноманітного навантаження на вузли. Його основною перевагою є здатність адаптуватися до мінливих умов навантаження і трафіку в режимі реального часу. Алгоритм безперервно аналізує навантаження на вузли і динамічно коригує вагові коефіцієнти для забезпечення оптимального розподілу трафіку.*

*Запропонований метод виділяється своєю здатністю ефективно підтримувати баланс навантаження при різноманітних викликах і навантаженнях, що робить його потужним інструментом забезпечення надійності і продуктивності в мережах SDN.*

***Ключові слова:*** *програмно-визначені мережі; балансування навантаження; мережі SDN; адаптивний мурашиний метод; динамічне налаштування параметрів.*

*Табл.: 1. Рис.: 1. Бібл.: 10.*

**Актуальність теми дослідження.** Зростання обсягів даних і підвищення вимог до продуктивності мереж у сучасному світі ставлять перед інженерами та дослідниками завдання підвищити продуктивність комп'ютерних мереж. Від реагування на пікові навантаження до забезпечення швидкості передачі даних і мінімізації затримок, ефективне управління мережевими ресурсами стає критично важливим.

Програмно-визначені мережі представляють новий підхід до побудови мереж, де контроль і управління мережею відокремлені від фізичної інфраструктури. Така архітектура дозволяє швидше розгортати нові послуги, зменшити витрати на обслуговування, а також покращити масштабованість і гнучкість.

Однак впровадження мереж SDN створює нові виклики, зокрема, пов'язані з балансуванням навантаження. Забезпечення рівномірного розподілу трафіку між вузлами мережі стає ключовим завданням для підтримки стабільної швидкості передачі даних і запобігання перевантаженню окремих компонентів мережі.

Для вирішення цього завдання існує безліч методів і підходів, серед яких балансування навантаження, використання резервних маршрутів, адаптивні алгоритми та інші. Однак більшість з існуючих підходів мають свої обмеження, особливо коли мова йде про адаптацію до змін в мережі та навантаженні.

**Постановка проблеми.** Інтеграція мереж SDN призводить до нових викликів, зокрема, пов'язаних з балансуванням навантаження. Забезпечення рівномірного розподілу трафіку між вузлами мережі стає ключовим завданням для підтримки стабільної швидкості передачі даних і запобігання перевантаженню окремих компонентів мережі. Для вирішення цієї проблеми було запропоновано багато методів та підходів, таких як балансування навантаження, використання резервних маршрутів, адаптивні алгоритми тощо.

**Аналіз останніх досліджень і публікацій.** Незважаючи на значні досягнення в області балансування навантаження в програмно-визначених мережах, більшість існуючих методів мають певні обмеження та недоліки. Огляд найбільш поширених недоліків, що наведений в таблиці, допоможе зрозуміти необхідність вдосконалення та розробки нових підходів.

*Таблиця 1 – Найпоширеніші недоліки методів балансування*

| **Недоліки** | **Огляд робіт авторів з зазначеним методом та пояснення щодо його використання** |
|---|---|
| Статичні параметри | Багато сучасних методів балансування навантаження, таких як Round Robin, описаний авторами в [4], використовують статичні параметри. Це призводить до недостатньої адаптації до змін в мережі, оскільки такі параметри залишаються незмінними протягом тривалого часу. В результаті гнучкість балансування навантаження обмежена, що може призвести до неоптимальних результатів. |
| Неврахування навантаження | В роботі [5] автори розглядають алгоритм "Least Connections" , але не враховують різницю в навантаженні між вузлами, тому можуть розподіляти трафік без урахування поточного навантаження на вузли. Це призводить до перевантаження одних вузлів і недовикористання інших, що може погіршити продуктивність системи. |
| Брак гнучкості | Алгоритм "Weighted Round Robin", запропонований авторами [6], має обмежену гнучкість у виборі маршрутів, і може не забезпечити достатньої оптимізації балансування навантаження та уникнення заторів. Обмежена можливість маневрування маршрутами обмежує потенціал оптимізації. |
| Обчислювальні витрати | Деякі складні методи, такі як "Adaptive Load Balancing" описане в [7], можуть вимагати значних обчислювальних ресурсів для виконання розрахунків і прийняття рішень. Висока обчислювальна складність може вплинути на продуктивність мережі і зробити її менш ефективною. |
| Вразливість до атак | «Random Load Balancing» запропонований авторами в [8], може бути вразливим до атак. Основною причиною такої вразливості є те, що метод випадковим чином розподіляє трафік між вузлами без урахування їх поточного навантаження. Також через випадковість розподілу трафіку метод Random Load Balancing може призвести до нерівномірного розподілу навантаження між вузлами. Певні вузли можуть бути перевантажені, в той час як інші можуть бути недостатньо завантажені. Це може створити можливість для атак, таких як DDoS, спрямованих на конкретні вузли. |

Джерело: розроблено авторами

Тому для подолання цих обмежень і недоліків важливо розробляти нові методи балансування навантаження, які є гнучкими, адаптивними і відповідають вимогам сучасних мереж SDN.

**Метою статті** є розробка та представлення нового методу балансування навантаження для програмно-визначених мереж на основі мурашиного алгоритму з динамічним налаштуванням параметрів. Ключовою метою цього методу є досягнення оптимального розподілу трафіку та підвищення продуктивності мережі в умовах змінного навантаження та мережевої динаміки.

Дана робота спрямована на вирішення актуальної проблеми балансування навантаження в мережах SDN та внесення значного внеску в підвищення продуктивності, надійності та ефективності сучасних мереж.

**Виклад основного матеріалу.** Метод «Adaptive Ant Colony» - один з методів балансування навантаження в програмно-конфігурованих мережах (SDN), в основі якого лежить мурашиний алгоритм з елементами адаптації. У цьому методі мурашині агенти

моделюють поведінку мурах в природі, які взаємодіють один з одним і навколишнім середовищем за допомогою феромонів. Феромони використовуються для визначення вибору оптимальних маршрутів для передачі трафіку в мережі.

Однією з головних переваг методу Adaptive Ant Colony є його здатність пристосовуватися до змін у мережі та навантаження. Для досягнення цієї адаптивності вводяться додаткові параметри, які регулюють вагу феромонів і швидкість їх випаровування. Ці параметри автоматично підлаштовувалися в залежності від стану мережі та обсягу трафіку.

У статті [10] описано детальний алгоритм роботи методу адаптивної мурашиної колонії, а саме розрахунок оновлених ваг феромонів та швидкості випаровування, що забезпечує збалансованість маршрутів для трафіку

Формула для оновлення маси феромонів:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

Формула для оновлення швидкості випаровування:

$$\rho_{ij}(t+1) = (1-\alpha) \cdot \rho_{ij}(t) + \alpha \cdot \frac{f_{ij}(t)}{f_{ij}\ (t)}$$

де:

$\tau_{ij}(t)$ вага феромону на ребрі $(i,j)$ у момент часу $t$.

$\rho$ – швидкість випаровування феромону.

$\Delta\tau_{ij}(t)$ приріст феромону на ребрі $(i,j)$ у момент часу $t$.

$\alpha$ – коефіцієнт впливу інтенсивності трафіку на зміну швидкості випаровування

$f_{ij}(t)$ інтенсивність трафіку на ребрі $(i,j)$ у момент часу $t$.

Алгоритм Adaptive Ant Colony має свої обмеження і недоліки, які можуть вплинути на його ефективність в певних умовах. Серед них - чутливість до початкових значень параметрів, що може призвести до неадекватної поведінки та обмеженої збіжності алгоритму. Висока обчислювальна складність алгоритму також може створювати перешкоди, особливо у великих мережах або при обмежених ресурсах [9]. У цьому випадку алгоритм може стати вразливим до застрявання на певних ребрах або вузлах, що призводить до неоптимальних рішень.

Крім того, залежність від інтенсивності трафіку може обмежити адаптивність алгоритму, а також обмежити його здатність балансувати феромонну регуляцію. Алгоритм також не завжди може ефективно адаптуватися до раптових динамічних змін у мережі або навантаженні, що може призвести до затримок у реагуванні. В умовах обмеженого обсягу даних алгоритм також може давати неточні рішення, оскільки його рішення ґрунтуються на статистичних даних.

Метод «Enhanced Adaptive Ant Colony» є однією з найсучасніших стратегій балансування навантаження в програмно-визначених мережах (SDN). В роботі [9] автори запропонували цей метод як спосіб підвищення адаптивності та ефективності алгоритму Adaptive Ant Colony .

Головною особливістю Enhanced Adaptive Ant Colony є її здатність динамічно адаптуватися до змін у мережі та навантаженні. Для досягнення такої адаптивності автори вводять додаткові параметри, які регулюють вагу феромонів та швидкість їх випаровування. Ці параметри можуть автоматично коригуватися залежно від стану мережі та обсягу трафіку [9]. Це дозволяє алгоритму ефективно реагувати на зміни та оптимально розподіляти навантаження між вузлами мережі.

Для цього вводяться додаткові параметри, які регулюють вагу феромонів та швидкість їх випаровування. Ці параметри автоматично коригуються в залежності від стану

мережі та обсягу трафіку. Формули для розрахунку оновлених ваг феромонів і швидкості випаровування виглядають наступним чином:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$
$$\tau_{ij}(t) = \left(1/L_k(t)\right) * Q$$

де $L_k(t)$ довжина маршруту $k$ в момент часу $t$, $Q$ константа, яка визначає кількість феромону, що виділяється мурахою.

Покращення, запроваджене Enhanced Adaptive Ant Colony, полягає в більш точному адаптивному управлінні параметрами. Це дозволяє алгоритму ефективніше адаптуватися до змін, забезпечуючи краще балансування навантаження, менші обчислювальні витрати та вищу надійність, ніж попередні методи.

Метод Enhanced Adaptive Ant Colony дійсно покращує алгоритм Adaptive Ant Colony [10], але він також має свої обмеження та недоліки. Одним з найважливіших недоліків є висока обчислювальна складність методу. Це може призвести до значного збільшення обчислювальних ресурсів та часу виконання алгоритму, особливо у розгалужених мережах з великою кількістю вузлів та різноманітних маршрутів.

Додатковим недоліком може бути недостатня реакція на раптові зміни в мережі. Оскільки адаптація базується на історичних даних і статистиці, алгоритм може неефективно реагувати на ситуації, які не були передбачені попередніми даними.

Крім того, Enhanced Adaptive Ant Colony може бути вразливим до атак, подібних до методу Adaptive Ant Colony [9]. Наприклад, атаки, які надсилають штучний трафік, можуть спотворити статистику і призвести до неправильного розподілу навантаження.

Загалом, Enhanced Adaptive Ant Colony є першим кроком на шляху до підвищення адаптивності та ефективності балансування навантаження в програмно-визначених мережах, але його обмеження та недоліки слід враховувати при подальшій розробці та вдосконаленні алгоритмів балансування навантаження.

Концепція програмно-визначених мереж (SDN) являє собою новий підхід до побудови мереж, де управління та контроль мережі відокремлені від фізичної інфраструктури. Така архітектура дозволяє швидше розгортати нові послуги, знизити витрати на обслуговування, а також підвищити масштабованість і гнучкість.

Відмінною рисою SDN є те, що мережа організовується і управляється на програмному рівні за допомогою віртуальних комутаторів і центрального SDN-контролера [1,2] (рис. 1).

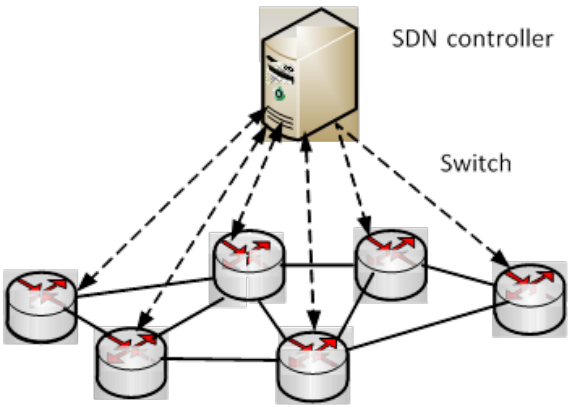


*Рис. 1. Структура мережі SDN*

Джерело: розроблено авторами

Існують різні рішення для балансування навантаження в програмно-визначених мережах (SDN), але деякі обмеження все ще залишаються актуальними, що вимагає розробки більш ефективних підходів. Серед таких рішень ми зосередимо увагу на алгоритмі Enhanced Adaptive Ant Colony, який слугуватиме основою для запропонованих нами вдосконалень.

Алгоритм Enhanced Adaptive Ant Colony базується на концепції мурашиного алгоритму [3]. У цьому методі колонія віртуальних мурах рухається мережею, відкладаючи та випромінюючи феромони для комунікації. Це орієнтує наступних мурах на вибір маршрутів з більшою концентрацією феромонів, що певною мірою сприяє балансуванню навантаження. Однак базовий алгоритм має свої недоліки, такі як статичні значення параметрів, обмежена гнучкість у виборі маршруту та чутливість до змін у мережі.

Нашою метою є вдосконалення алгоритму Enhanced Adaptive Ant Colony шляхом внесення динамічних змін до його ключових параметрів, що дозволить йому адаптуватися до реальних умов мережі. Ця вдосконалена версія має на меті підвищити точність балансування навантаження, ефективно реагувати на зміни в навантаженні мережі та забезпечити більш гнучке рішення для сучасних SDN-середовищ.

Запропонований метод балансування навантаження в програмно-визначених мережах базується на мурашиному алгоритмі та враховує динамічне налаштування параметрів. Основні принципи та етапи методу, а також формули для кожного етапу покроково описані нижче.

Мурашиний алгоритм базується на взаємодії агентів, які імітують поведінку мурах при пошуку шляхів до джерел їжі. Ключовою ідеєю є здатність агентів обмінюватися інформацією про якість поточних шляхів і використовувати цю інформацію для вибору оптимального маршруту [5].

Для подолання обмежень базового алгоритму ми розробили кроки оптимізації, які дозволяють алгоритму адаптуватися до мінливих умов мережі. Розглянемо детальніше кожен з цих кроків, починаючи з його ініціалізації.

**Крок 1: Ініціалізація**

На цьому кроці ми налаштовуємо алгоритм і готуємо його до виконання.

1. Визначимо $N$ як кількість вузлів мережі
2. Ініціалізуємо рівні феромонів на всіх можливих маршрутах:

$$\tau_{ij} = \tau_0,$$

де i та j - вузли мережі, $\tau_{ij}$ - рівень феромону між вузлами i та j, $\tau_0$ початковий рівень феромону

3. Визначення агентів на вихідних вузлах мережі, звідки йде вихідне навантаження. Кожен агент ідентифікує свій поточний вузол і призначений вузол.

**Крок 2: Динамічне налаштування параметрів**

Для того, щоб адаптуватися до змін і досягти найбільш ефективного балансування навантаження, ми використовуємо динамічні налаштування параметрів.

1. Розраховуємо поточне навантаження на мережу, load_factor.
2. Визначаємо коефіцієнти для ваг параметрів α та β:

$$\alpha = \alpha_{base} \times \frac{load\_factor}{max\_load},$$

$$\beta = \beta_{base} \times \frac{max\_load}{load\_factor}$$

де $\alpha_{base}$ та $\beta_{base}$ - початкові значення параметрів, max_load - максимально можливий ступінь навантаження.

**Крок 3: Покращений вибір маршруту**

Для покращення збіжності та точності балансування ми використовуємо покращений вибір маршруту з "елітними маршрутами".

1. Коли агент вибирає маршрут, ми використовуємо ймовірність вибору маршруту:

$$P_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_k \left(\tau_{ik}^{\alpha} \cdot \eta_{ik}^{\beta}\right)}.$$

2. Після вибору маршруту оновіть рівень феромонів на ньому додатковим збільшенням:

$$\Delta\tau_{ij} = \frac{load\_factor}{max\_load} \times \tau_{ij}.$$

**Крок 4: Адаптивне оновлення феромонів**

Замість статичного коефіцієнта випаровування ми використовуємо адаптивний коефіцієнт, який залежить від ступеня навантаження:

$$\rho = \rho_{base} + (load\_factor - 1) \times \rho_{base}.$$

Це дозволяє ефективніше реагувати на зміни навантаження і підтримувати баланс навантаження на оптимальному рівні.

**Крок 5: Динамічне налаштування параметрів**

Щоб забезпечити адаптацію до змін у мережі та максимально ефективне балансування навантаження, ми використовуємо динамічне регулювання налаштувань (Крок 2).

1. Розраховуємо поточне навантаження на мережу, load_factor.
2. Визначаємо коефіцієнти для ваг параметрів α та β:

$$\alpha = \alpha_{base} \times \frac{load\_factor}{max\_load},$$
$$\beta = \beta_{base} \times \frac{max\_load}{load\_factor},$$

де $\alpha_{base}$ та $\beta_{base}$ - початкові значення параметрів, max_load - максимально можливий ступінь навантаження..

**Крок 6: Динамічне регулювання налаштувань**

Ми використовуємо динамічне налаштування параметрів, щоб забезпечити адаптивність алгоритму до змін у мережі.

1. Визначаємо коефіцієнт адаптивності для феромонів:

$$\phi = \frac{load\_factor}{max\_load}.$$

2. Адаптивно змінюємо параметр випаровування:

$$\rho = \rho_{base} + (\phi - 1) \times \rho_{base}.$$

3. Змінення значення параметрів $\alpha$ та $\beta$:

$$\alpha = \alpha_{base} \times \phi\,,$$
$$\beta = \beta_{base} \times (1 - \phi).$$

**Крок 7: Пошук найкращого рішення**

Після кількох ітерацій алгоритму кожен агент матиме кілька можливих маршрутів. З них ми виберемо найкращий, враховуючи як феромони, так і вагу вузлів.

1. Визначаємо оцінку для кожного маршруту агента:

$$evaluation = \frac{pheromone\_level \times node\_weight}{path\_length}.$$

2. Обираємо маршрут з найбільшою кількістю балів як найкращий.

Використання динамічних налаштувань параметрів дозволить адаптувати параметри алгоритму в режимі реального часу. Замість статичних значень параметри алгоритму тепер автоматично змінюються в залежності від змін в мережі та навантаження. Це дозволяє алгоритму ефективно реагувати на зміни, підтримуючи оптимальний розподіл навантаження навіть у мінливих умовах.

Друге важливе поліпшення стосується вдосконалення процесу вибору маршруту для віртуальних мурах. Шляхи тепер враховують більше факторів, включаючи поточне навантаження на вузли та шляхи. Це дозволяє мурахам обирати найкращі маршрути, виходячи з поточних обставин, забезпечуючи рівномірний розподіл трафіку та уникаючи заторів.

Підвищена адаптивність - третій ключовий аспект оптимізованого алгоритму. Впровадження динамічного налаштування параметрів у поєднанні з удосконаленим процесом вибору маршруту зробило алгоритм більш адаптивним до динамічних змін у мережі. Він здатний ефективно адаптуватися до коливань навантаження та змін у структурі мережі.

Ці вдосконалення разом сприяють кращому балансуванню навантаження в мережах SDN. Забезпечуючи точний розподіл трафіку, запобігаючи перевантаженню окремих вузлів і забезпечуючи високу продуктивність мережі, оптимізований алгоритм Enhanced Adaptive Ant Colony стає потужним інструментом для підтримки стабільної та ефективної мережі з програмованою структурою.

Запропонований алгоритм порівнювався з Round Robin [4], Least Connections [5] і Adaptive Load Balancing [7], де всі алгоритми були перероблені на основі опублікованих псевдокодів, формулювань і блок-схем. Кожен алгоритм виконується 10 раз для кожного діапазону несправностей, і середнє значення береться для більш точного вимірювання.

Список показників продуктивності включає час виконання, середній час виконання і затримку на завдання, балансування навантаження і коефіцієнт успішності виконання.

В даному дослідженні зосередимось на часу виконання та середньому часу виконання.

Час виконання для всіх алгоритмів майже однаковий, коли немає помилок. Однак у міру збільшення частоти відмов час виконання для запропонованого алгоритму є найнижчим серед всіх (380 000мс на 50%). В той же час показник Round Robin склав 1 200 000мс, Least Connections - 800 000мс, Adaptive Load Balancing - 450 000мс.

Ситуація з середнім часом виконання очікувано така ж - запропонований алгоритм закінчує свою роботу швидше на 15-20% в порівнянні з іншими алгоритмами за рахунок динамічного розподілу параметрів.

В наступних дослідженнях оцінимо алгоритм по іншим критеріях продуктивності.

**Висновки.** У цій статті ми представляємо оптимізований метод балансування навантаження в мережах SDN, заснований на вдосконаленому алгоритмі адаптивних мурашиних колоній. Сучасні мережі стикаються зі зростаючим обсягом даних та різноманітними вимогами, що загрожує їх ефективності та стабільності. Для подолання цих викликів балансування навантаження в мережах стає критично важливим завданням.

Заснований на мурашиному алгоритмі, запропонований метод використовує ідеї адаптивності та динамічного налаштування параметрів для досягнення оптимального балансування навантаження. Цей метод дозволяє мережі адаптуватися до змін навантаження та забезпечити рівномірний розподіл даних між вузлами. Адаптивність алгоритму дозволяє підтримувати стабільний рівень продуктивності навіть в умовах динамічних змін навантаження.

Процес балансування навантаження включає кілька етапів, таких як ініціалізація, динамічне налаштування параметрів, покращений вибір маршруту, адаптивне оновлення

феромонів та пошук найкращого рішення. Кожен з цих етапів виконує важливу функцію в забезпеченні ефективного балансування навантаження.

Метод Enhanced Adaptive Ant Colony є важливим кроком на шляху до забезпечення відмовостійкості та оптимальної продуктивності мереж SDN. Враховуючи динаміку розвитку сучасних мереж та зростаючі вимоги до їх продуктивності, він дає можливість ефективно вирішувати проблему навантаження та забезпечувати надійну і швидку передачу даних.

Отже, оптимізований метод Enhanced Adaptive Ant Colony відкриває перспективи для подальших досліджень в області балансування навантаження в мережах SDN та розширення можливостей алгоритму для вирішення різноманітних задач в мережевих середовищах.

## Список використаних джерел

[1] Machine learning for network automation: Overview, architecture, and applications / A. Garcia-Saavedra, P. Serrano, A. Banchs, X. Costa-Perez // IEEE Communications Magazine. – 2018. – Vol. 56, № 3. – P. 11-17.

[2] Resilient SDN traffic engineering: A survey / L. Tang, S. Han, H. Zhang, M. Gerla // IEEE Communications Surveys & Tutorials. – 2016. – Vol. 18, № 4. – P. 2682-2706.

[3] Dorigo, M. Ant system: optimization by a colony of cooperating agents / M. Dorigo, V. Maniezzo, A. Colorni // IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). – 1996. – Vol. 26, № 1. – P. 29-41.

[4] Jamal, M. A comparative study of software defined networking load balancer algorithms / M. Jamal, A. S. Siddiqui // IEEE Access. – 2020. – № 8. – P. 155905-155919.

[5] Ghorbani, S. An efficient load balancing algorithm for software-defined networking / S. Ghorbani, M. R. Sama, B. Abolhasani // In Proceedings of the 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing. – 2015. – P. 80-85.

[6] Chen, C. C. A weighted round robin algorithm for software defined networkin / C. C. Chen, M. Chen // In Proceedings of the 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing. – 2015. – P. 138-143.

[7] Data center network virtualization: A survey. / M. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, G. Rabbani, Q. Zhang // IEEE Communications Surveys & Tutorials. – 2013. № 15(3). – P. 1614-1634.

[8] Load balancing strategy for SDN based on improved ant colony algorithm / Y. Ren, H. Luo, D. Xiang, L. Zeng, Y. Pan // IEEE Access. – 2019. – № 7. – P. 151536-151543.

[9] Mehmood, R. Enhanced dynamic ant colony load balancing algorithm for SDN / R. Mehmood, F. Ahmed // Journal of Ambient Intelligence and Humanized Computing. – 2020. – Vol. 11. – P. 5907-5921.

[10] Білий, В. С. Адаптивний метод балансування навантаження в програмно-керованих мережах / В. С. Білий, І. І. Пупськин // Інформаційні технології та комп'ютерна інженерія. – 2019. – № 4 (44).

## References

[1] Garcia-Saavedra, A., Serrano, P., Banchs, A., Costa-Perez, X. (2018). Machine learning for network automation: Overview, architecture, and applications. *IEEE Communications Magazine, 56(3),* 11-17. DOI: 10.1109/MCOM.2018.1700980.

[2] Tang, L., Han, S., Zhang, H., Gerla, M. (2016). Resilient SDN traffic engineering: A survey. *IEEE Communications Surveys & Tutorials, 18(4),* 2682-2706. DOI: 10.1109/COMST.2016.2581599.

[3] Dorigo, M., Maniezzo, V., Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1),* 29-41. DOI: 10.1109/3477.484436.

[4] Jamal, M., Siddiqui, A. S. (2020). A comparative study of software defined networking load balancer algorithms. *IEEE Access, 8,* 155905-155919.

[5] Ghorbani, S., Sama, M. R. B., Abolhasani, M. (2015). An efficient load balancing algorithm for software-defined networking. *Proceedings of the 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing* (pp. 80-85).

[6] Chen, C. C., & Chen, M. (2015). A weighted round robin algorithm for software defined networking. *Proceedings of the 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing* (pp. 138-143).

[7] Bari, M. F., Boutaba, R., Esteves, R., Granville, L. Z., Podlesny, M., Rabbani, G., & Zhang, Q. (2013). Data center network virtualization: A survey. *IEEE Communications Surveys & Tutorials, 15(3),* 1614-1634.

[8] Ren, Y., Luo, H., Xiang, D., Zeng, L., & Pan, Y. (2019). Load balancing strategy for SDN based on improved ant colony algorithm. *IEEE Access, 7,* 151536-151543.

[9] Mehmood, R., Ahmed, F. (2020). Enhanced dynamic ant colony load balancing algorithm for SDN. *Journal of Ambient Intelligence and Humanized Computing, 11,* 5907-5921. DOI: 10.1007/s12652-020-01906-4.

[10] Bilyi, V. S., Pupskyn, I. I. (2019). Adaptive method of load balancing in software-controlled networks. Information technologies and computer engineering, 4(44).

Отримано 15.09.2023

*UDC 681.3*

***Vadym Shchur[1], Yurii Kulakov[2]***

[1]PhD Student of the Department of Computer Engineering
National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute» (Kyiv, Ukraine)
**E-mail:** vadimmshchur@gmail.com. **ORCID**: https://orcid.org/0000-0001-8925-4813

[2] Doctor of technical sciences, Professor of the Department of Computer Engineering
National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute» (Kyiv, Ukraine)
**E-mail:** ya.kulakov@gmail.com **ORCID**: https://orcid.org/0000-0002-8981-5649

# OPTIMISED ADAPTIVE LOAD BALANCING METHOD IN SDN NETWORKS BASED ON ANT COLONY OPTIMISATION

*This paper presents an important contribution to the field of software-defined networking (SDN) through the development and improvement of a load balancing method that uses adaptive ant colonies. Modern networks face significant challenges, such as the growth of data and the diversity of user requirements, which can significantly affect their efficiency and reliability. In this context, load balancing becomes an extremely important task.*

*The proposed method is based on the ideas of the ant algorithm and uses adaptability and dynamic parameter tuning to achieve optimal load balancing. This allows the network to adapt to changes in load and ensures an even distribution of data between nodes. An important advantage of this method is its ability to maintain stable performance even in situations where the load changes dynamically.*

*The load balancing process involves several key steps, such as initialisation, dynamic parameter tuning, improved route selection, adaptive pheromone updating, and finding the best solution. Each of these stages is aimed at providing efficient and reliable load balancing.*

*The proposed Enhanced Adaptive Ant Colony method is an important step towards improving fault tolerance and optimal performance in SDN networks. Given the rapid development of modern networks and the growth of their performance requirements, this method opens up opportunities for efficient load balancing and ensuring reliable and fast data exchange.*

*In general, the optimised Enhanced Adaptive Ant Colony method opens up prospects for further research in the field of load balancing in SDN networks and extending the capabilities of the algorithm to solve various problems in network environments. It introduces a new approach to ensuring the reliability and performance of SDN networks and can become an important tool for network engineers and researchers in this area.*

***Keywords:*** *software-defined networks; load balancing; SDN networks; adaptive ant colony method; dynamic parameter adjustment.*

*Table: 1. Fig.: 1. References: 10.*