

# ПІДСИСТЕМА ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧ ОПТИМІЗАЦІЇ ПАРАМЕТРІВ КОМП'ЮТЕРНИХ МЕРЕЖ З ВИКОРИСТАННЯМ МУРАШКОВИХ АЛГОРИТМІВ

**Варагаш М.В.**

*Тернопільський національний економічний університет, студент*

## **I. Постановка задачі**

Для забезпечення високопродуктивної роботи комп'ютерних мереж необхідно визначити оптимальні її параметри. Одним із методів такої оптимізації є використання нових алгоритмів маршрутизації пакетів даних у комп'ютерній мережі. Маршрутизація це важлива і складна задача. Важлива, тому що, вона має значний вплив на загальну продуктивність мережі, а складна тому що, характеристики мереж, такі як навантаження трафіку та топологія мережі, можуть змінюватися як стохастичним способом, так і бути змінними у часі. Отже, мурашкові алгоритми виглядають багатообіцяючими для розв'язання такого роду задач, забезпечуючи високий рівень відмовостійкості.

## **II. Мета роботи**

Метою даної роботи є оптимізація процесу маршрутизації даних в комп'ютерній мережі з використанням принципів роботи мурашкових колоній.

## **III. Алгоритми маршрутизації**

Алгоритми маршрутизації можуть бути класифіковані, як централізовані в порівнянні з розподіленими так і статичні в порівнянні з адаптивними. У централізованих алгоритмах головний контролер забезпечує оновлення таблиці маршрутизації кожного вузла і виконання маршрутизації. Такі алгоритми можуть використовуватися тільки в особливих випадках і для малих мереж. Навпроти, у розподіленій маршрутизації обчислення маршрутів виконується шляхом сумісного використання середовища мережевих вузлів, які обмінюються необхідною інформацією.

В статичній маршрутизації, шлях обраний пакетом даних, визначений тільки на основі його джерела і місця призначення без зв'язку з поточним мережевим трафіком. Обраний шлях зазвичай мінімальний, відповідно до деякого критерію вартості. Адаптивна маршрутизація, в принципі, більш приваблива, тому що може адаптувати правила маршрутизації до часу та просторово змінних умов трафіку. Як недолік, адаптивні алгоритми можуть викликати кругові шляхи та значні зміни в продуктивності.

З точки зору оптимізації, алгоритми маршрутизації поділяють на алгоритми оптимальної маршрутизації та маршрутизації найкоротшого шляху.

Оптимальна маршрутизація має на меті оптимізувати функцію потоків всіх індивідуальних зв'язків (переважно ця функція це сума присвоєних затрат на основі середніх пакетних затримок).

У маршрутизації найкоротшого шляху немає глобальної функції для оптимізації, її ціль полягає у знаходженні найкоротшого шляху між двома вузлами, де витрати лінії зв'язку обчислюються (статично чи адаптивно) відповідно до деякого статистичного опису потоку трафіку, який проходить через лінії зв'язку. З огляду на різний вміст, який заноситься в таблиці маршрутизації, маршрутизацію найкоротшого шляху можна розподілити на два класи: дистанційно-векторні алгоритми та алгоритми стану каналу.

Дистанційно-векторні алгоритми використовують таблиці маршрутизації, які складаються з трьох складових (місце призначення, оціночна відстань, наступний перехід), визначені для всіх вузлів призначення в мережі і для всіх сусідніх вузлів маршрутизатора. Отримавши від деякого сусіднього вузла вектор дистанцій, до відомих йому мереж, маршрутизатор нарощує компоненти вектора на величину відстані від себе до даного сусіднього вузла. Крім того, він доповнює вектор інформацією про невідомі йому самому інші мережі, про які він дізнався безпосередньо чи з повідомлень інших маршрутизаторів. Оновлене значення вектора, маршрутизатор розсилає сусіднім маршрутизаторам. Такі алгоритми добре працюють тільки в малих мережах, у великих мережах вони періодично завантажують лінії зв'язку інтенсивним трафіком.

Алгоритми стану каналу використовують таблиці маршрутизації з набагато більшою кількістю інформації, порівняно з дистанційно-векторними алгоритмами. Фактично, в ядрі даного алгоритму є розподілена і тиражована база даних. Ця база це динамічна карта цілої мережі, яка описує в деталях всі її складові. Використовуючи цю базу, кожен вузол обчислює свій найкращий шлях

використовуючи алгоритм Дейкстри, і тоді використовуючи цей найкращий шлях будує таблицю маршрутизації.

#### IV. Мурашковий алгоритм маршрутизації

Мурашковий алгоритм зручно описувати з точки зору двох наборів штучних мурашок, які рухаються у двох напрямках: вперед  $F$  і назад  $B$ . Мурахи в кожному наборі володіють однією і тією ж структурою, але вони по іншому розташовані в середовищі, тобто вони можуть знайти різні входи і видати відмінні важливі виходи. Мурахи спілкуються непрямим способом, згідно парадигми стігметрії, через інформацію яку вони зчитують і записують на мережевих вузлах, які вони відвідують.

Даний алгоритм може бути розділений на дві основні фази: побудова рішення і оновлення структури даних.

##### 1. Побудова рішення

Через рівномірний інтервал  $dt$  від кожного мережевого вузла  $s$ , мурашка  $F_{s-d}$  запущена до вузла призначення  $d$ , щоб знайти до нього найкоротший шлях та дослідити стан навантаження мережі на їхньому шляху. Мурашки  $F$  поширюються в тій же черзі, що й пакети даних, таким чином вони зазнають того ж самого навантаження трафіку. Протягом просування мурашки до її цілі, вона запам'ятовує весь її шлях та стан трафіку. Ідентифікатор кожного відвіданого вузла  $i$  та час, що пройшов з моменту запуску до моменту досягнення  $i$ -го вузла зберігаються в стек пам'яті  $S_{s-d}(i)$ .

Мураха будує шлях виконуючи наступні кроки:

1.1. В кожному вузлі  $i$ , кожна прохідна мураха, що рухається в напрямку кінцевого вузла  $d$ , вибирає вузол  $j$  серед тих сусідніх вузлів, які ще не були відвідані, або серед усіх сусідніх, якщо всі вони вже були відвідані. Сусідній вузол  $j$  обирається з ймовірністю  $P_{ijd}$ , яка обчислюється як нормалізована сума феромонів  $t_{ijd}$  з евристичним значенням  $h_{ij}$ , яке отримується при обчисленні стану (довжини) лінії зв'язку між  $j$ -им та  $i$ -им вузлами.

$$P_{ijd} = \frac{T_{ijd} + ah_{ij}}{1 + a(|N_i| - 1)}. \quad (1)$$

Евристичне значення  $h_{ij}$  – нормалізоване, в межах  $[0,1]$ ;  $q_{ij}$  – значення функції довжини черги на лінії зв'язку між вузлом  $i$  та  $j$ .

$$h_{ij} = 1 - \frac{q_{ij}}{\sum_{l=1}^{|N_i|} q_{il}}. \quad (2)$$

Параметр  $a$  це коефіцієнт який задає важливість евристичного значення відносно значень феромону, який зберігається в матриці феромонів  $T$ . Значення  $h_{ij}$  відображає миттєвий стан черги вузла  $i$ , припускаючи, що процес черги майже стаціонарний або повільно змінюється,  $h_{ij}$  дає кількісний показник, який зв'язаний з часом очікування в черзі. Значення феромона, з другої сторони, є результатом постійного процесу навчання. Використання цих значень дозволяє алгоритму бути більш "реактивним".

1.2. Якщо знайдено цикл, тобто якщо мурашка повертається в уже відвіданий вузол, то вузли циклу видаляються.

1.3. Коли вузол призначення  $d$  досягнуто, агент  $F_{s-d}$  генерує іншого агента  $B_{d-s}$  (мураха  $B$ ), а сам видаляється. Мураха  $F$  також видаляється, якщо її життєвий цикл більший ніж  $max\_life$  (це параметр алгоритму).

1.4. Мураха  $B$  повертається тим же самим шляхом, що й мураха  $F$ , але в протилежному напрямку. Мурахи  $B$  також використовують черги з високим пріоритетом, оскільки їхня основна ціль це як найшвидше оновлення матриць феромону.

##### 2. Оновлення структури даних

Прибуваючи до вузла  $i$  з сусіднього вузла, мурашка  $B$  оновлює дві основні структури даних вузла, локальну модель трафіку  $M_i$  та матрицю феромонів  $T_i$ , для всіх записів відповідно до кінцевого вузла  $d$ .  $M_i$  оновлюється значеннями, які містяться в пам'яті мурашки. Оновлення матриці феромонів  $T_i$  проводиться шляхом збільшенням  $T_{ifd'}$  (у випадку коли феромон вказує вибрати сусіда  $f$  коли вузол призначення є  $d'$ ) і зменшенням, нормалізацією інших феромонів  $T_{ijd'}, j \in N_i, j \neq f$

#### V. Результати

На основі приведенного вище мурашкового алгоритму в середовищі ns2 (Network Simulator 2) було виконано його реалізацію та проведено порівняння з іншими алгоритмами маршрутизації

(AODV, OSPF, DSR) за такими параметрами як пропускна здатність мережі та час передачі пакетів (рисунки 1-2).

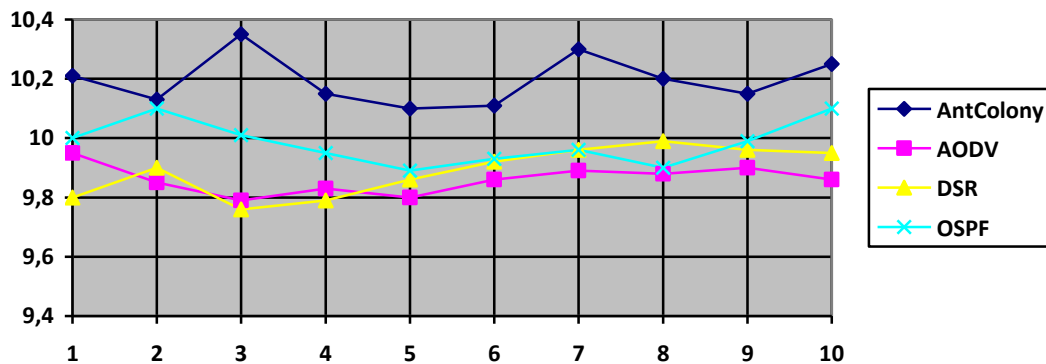


Рисунок 1 – Графік пропускної здатності мережі

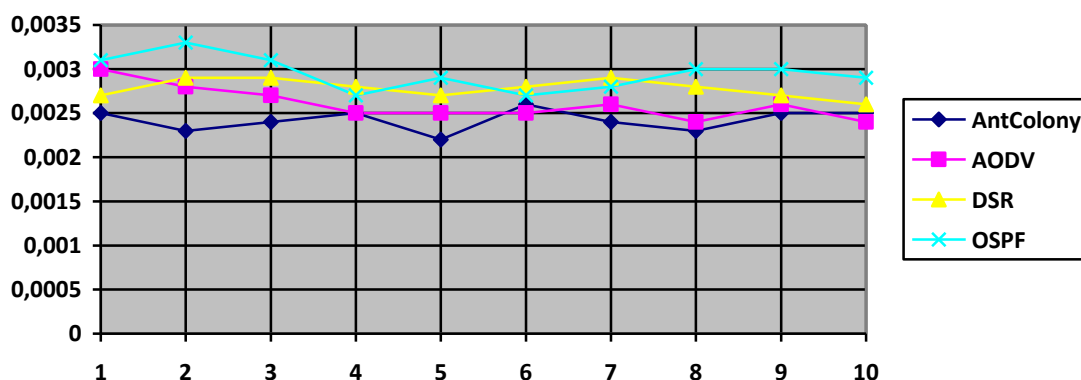


Рисунок 2 – Графік часу затримки передачі пакетів

В результаті проведеного порівняння було отримано вищу пропускну здатність комп'ютерної мережі у середньому 10,195 Мбіт/с, що на 2% краще ніж інші алгоритми, а також менший час затримки передачі пакетів даних у середньому 2,42 мс, що на 17% краще в порівнянні з іншими.

### Висновки

В даній роботі було реалізовано мурашковий алгоритм маршрутизації даних в комп'ютерній мережі з використанням мережевого симулятора Network Simulator, що дало змогу перевірити ефективність використання мурашкового алгоритму порівняно з іншими алгоритмами маршрутизації. У результаті досліджень було отримано кращі показники пропускної здатності комп'ютерної мережі та меншу затримку часу передачі пакетів даних.

### Список використаних джерел

1. Dorigo M. Ant Colony Optimization. / M. Dorigo, T. Stützle - A Bradford book. 2004. - 321 с.
2. Штовба С. Д. Мурашині алгоритми оптимізації / С. Д. Штовба, О. М. Рудий // Вісник ВПІ. - 2004. - №4. - С. 62-69
3. Kambayashi Y. A Review of Routing Protocols Based on Ant-Like Mobile Agents / Y. Kambayashi // Algorithms. - 2013. - №6. - С. 442-456
4. Суботін С. О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей: Монографія / С. О. Суботін, А. О. Олійник, О. О. Олійник // Метод мурашиних колоній. - Запоріжжя: ЗНТУ, 2009. - С. 292-313