

УДК 004.724.4

## МОДИФІКОВАНИЙ АЛГОРИТМ БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ДЛЯ ЧАСТКОВО-ПЕРЕСІЧНИХ МАРШРУТІВ ПЕРЕДАЧІ ДАНИХ

аспірант, Калюжний О. О.

*e-mail: sleeper253@gmail.com*

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Україна, Київ

*В даній роботі запропоновано модифікований алгоритм балансування навантаження, за допомогою якого можна оптимізувати навантаження на мережу та зменшити затримку передачі даних. Даний алгоритм спрямований на балансування навантаження між частково-пересічними маршрутами передачі даних. В основі лежить модифікований алгоритм ECMP з корекцією навантаження на мережу та врахуванням довжини можливих шляхів. Даний алгоритм є оптимальним при централізованому керуванні мережею, а тому запропоновано його використання у SDN мережах. Проведено тестування даної розробки для порівняння з існуючим рішенням та представлено переваги даної модифікації.*

*Ключові слова: багатошляхова маршрутизація, частково-пересічні маршрути, балансування навантаження, ECMP алгоритм, SDN мережі.*

*аспірант, Калюжний А. О. Модифицированный алгоритм балансировки нагрузки для частично-пересекающихся маршрутов передачи данных / Национальный технический университет Украины «Киевский политехнический институт им. Игоря Сикорского», Украина, Киев.*

*В данной работе предложен модифицированный алгоритм балансировки нагрузки, с помощью которого можно*

*оптимизировать нагрузку на сеть и уменьшить задержку передачи данных. Данный алгоритм направлен на балансировку нагрузки между частично-пересекающимися маршрутами передачи данных. В основе лежит модифицированный алгоритм ECMP с коррекцией нагрузки на сеть и учетом длины возможных путей. Данный алгоритм является оптимальным при централизованном управлении сетью, а поэтому предложено его использование в SDN сетях. Проведено тестирование данной разработки для сравнения с иснучим решением и представлены преимущества данной модификации.*

*Ключевые слова: багатошляхова маршрутизація, частично-рядові маршрути, балансировка нагрузки, ECMP алгоритм, SDN сети.*

*O. O. Kaliuzhnyi, postgraduate student, Modified load balancing algorithm for partially-overlapping data transmission routes / National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, Kyiv.*

*In this paper we propose a modified load balancing algorithm, which can be used to optimize network load and reduce data transmission delay. This algorithm is aimed at load balancing between partially-overlapping data transmission routes. It is based on a modified ECMP algorithm with correction of the load on the network and taking into account the length of possible paths. This algorithm is optimal for centralized network management, and therefore its use in SDN networks is proposed. Testing of this development is carried out for comparison with the existing solution and the advantages of this modification are presented.*

*Keywords: multipath routing, partially-overlapping routes, load balancing, ECMP algorithm, SDN networks.*

**Вступ.** На сьогоднішній день у світі відбувається активний розвиток мережних технологій, що спричинений збільшенням кількості пристроїв підключених до глобальної мережі. Це призводить до розширення вимог до якості обслуговування у системах передачі даних. Під час роботи, всередині мережі можуть змінюватись певні параметри, такі як навантаження, пропускна здатність ліній зв'язку, топологія та інші. Це може спричинити появу перенавантаження каналів передачі даних в певних частинах мережі, що в свою чергу знижує ефективність використання доступних ресурсів мережі та якості обслуговування. Одним із варіантів вирішення даної проблеми є передача даних по декільком шляхам одночасно, що дозволяє більш ефективно розподілити навантаження на мережу. Однак використовуючи багатошляхову маршрутизацію ми стикаємось з питанням оптимального розподілу потоку даних між доступними маршрутами.

В даній роботі представлено модифікацію алгоритму балансування навантаження спрямованого на використання частково-пересічних шляхів передачі даних, що дозволяє покращити використання доступних ресурсів мережі.

**Постановка проблеми.** Під час управління мережевим трафіком основним питанням є оптимальне використання доступної пропускної здатності мережі зі збереженням мінімальної затримки передачі даних. При використанні багатошляхової маршрутизації є декілька можливих шляхів передачі, котрі можуть бути різної довжини та мати різну пропускну здатність.

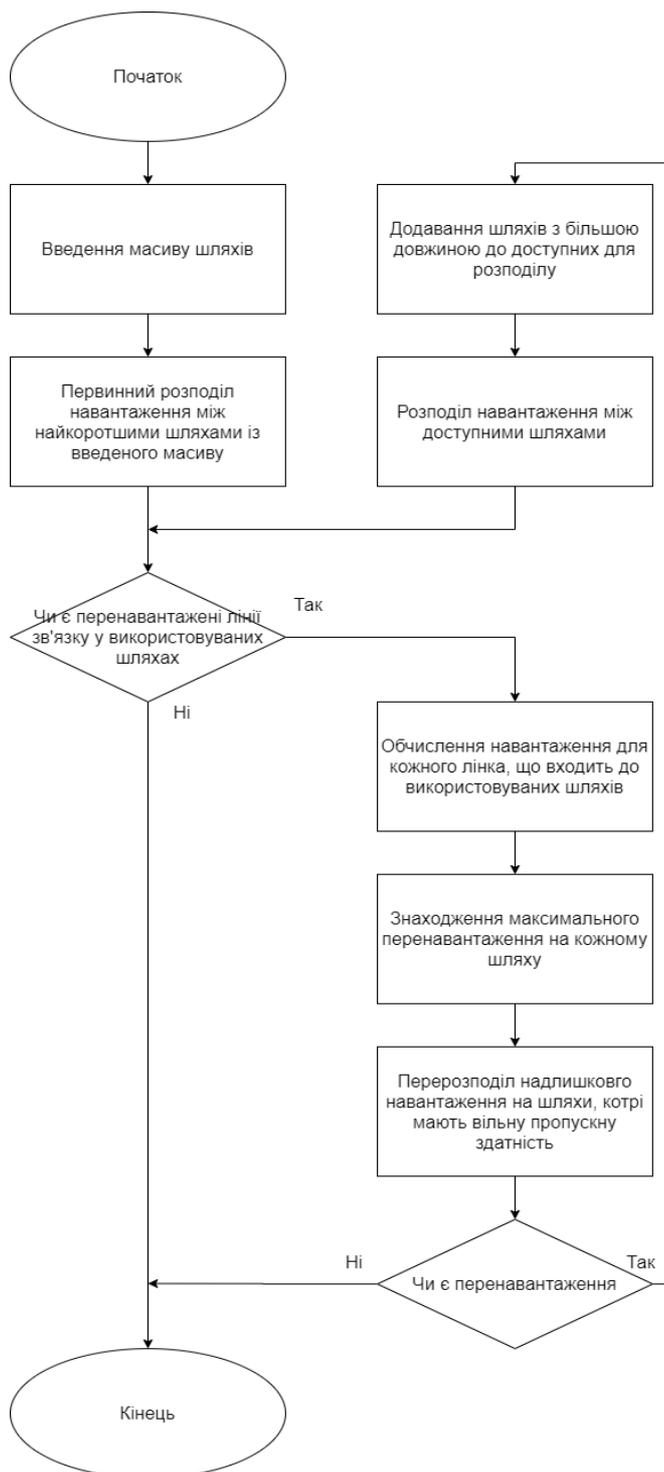
Для розподілу навантаження між множиною маршрутів найчастіше використовуються алгоритми VLB[1] та ECMP[2]. Однак дані алгоритми призначені для балансування навантаження між шляхами рівної довжини і вони у своєму базовому вигляді не

враховують поточні параметри навантаження на мережу. Щоб врахувати доступну пропускну здатність ліній зв'язку, оптимальним є використання технології SDN[3], що реалізує централізоване управління мережею та має інформацію про все що відбувається всередині неї. У роботі [4] було запропоновано модифікацію алгоритму ECMP, що враховував поточне навантаження на мережу. Однак він не брав до уваги довжину доступних шляхів передачі даних, що спричиняло появу проблеми збільшення часу передачі потоку даних.

Щоб вирішити дану проблему, в даній роботі запропоновано модифікацію алгоритму, що намагається використовувати більш короткі доступні шляхи передачі даних та рівномірно розподіляє навантаження між ними.

**Опис розробленого алгоритму.** Блок-схема модифікованого алгоритму зображена на рисунку 1.

Вхідними даними алгоритму є масив частково-пересічних маршрутів між двома вузлами мережі. Далі виконується сортування шляхів за довжиною від найкоротшого до найдовшого. Наступним кроком алгоритму є первинний рівномірний розподіл навантаження між найкоротшими шляхами. Потім виконується перевірка на перевантаженість ланок мережі і якщо таких немає, то алгоритм закінчує роботу. У випадку перевищення пропускну здатності ліній зв'язку, відбувається ще один перерозподіл між вільними шляхами на даному кроці алгоритму.

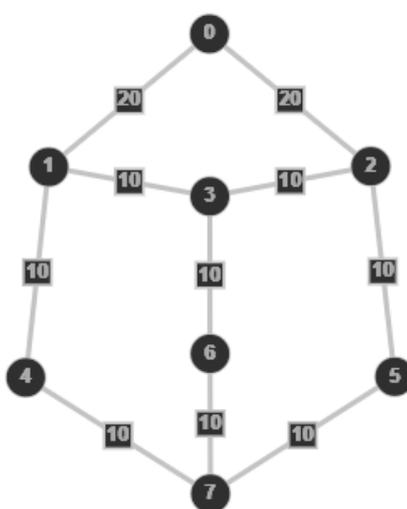


**Рис. 1. Блок-схема роботи алгоритму**

У випадку коли перерозподіл не вирішив проблему, алгоритм додає більш довгі шляхи (із введеного масиву на початку роботи) та знову намагається розподілити навантаження та провести корекцію за

необхідності. Даний цикл додавання маршрутів повторюється до успішного розподілу навантаження або до повного використання введеного масиву шляхів.

Далі приведено приклад роботи розробленого алгоритму та його порівняння з базовою версією. Для пошуку шляхів буде використовуватись алгоритм представлений у роботі [5]. На рисунку 2 зображено приклад топології мережі.



**Рис. 2. Топологія для демонстрації роботи розробленого алгоритму**

Пропускна здатність ліній зв'язку - 10 та 20 mb/s. Для наглядної демонстрації роботи припустимо, що потрібно пропустити потік 10 mb/s від вузла 0 до вузла 7.

Частково-пересічні шляхи що знайдені:

$$R1 = 0 - 1 - 4 - 7,$$

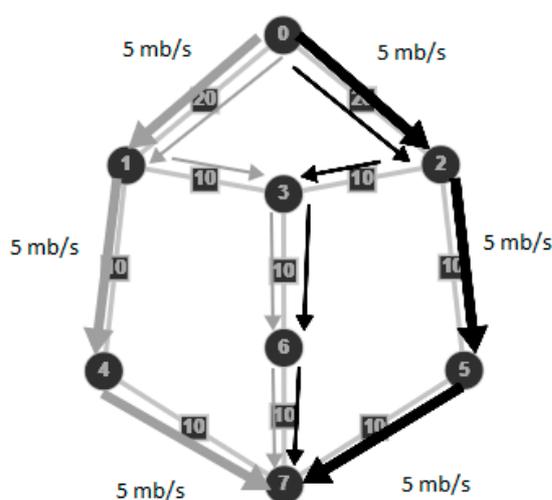
$$R2 = 0 - 2 - 5 - 7,$$

$$R3 = 0 - 1 - 3 - 6 - 7,$$

$$R4 = 0 - 2 - 3 - 6 - 7.$$

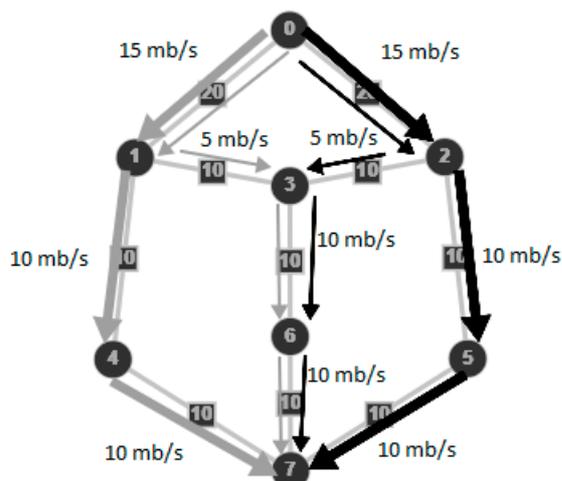
При використанні алгоритму представленого у даній роботі навантаження розподілиться рівномірно між шляхами R1 та R2. Довжина максимального шляху – 3 переходи.

У випадку використання базової версії алгоритму навантаження рівномірно розподілилось би між усіма маршрутами R1 – R4 і максимальна довжина використаних шляхів дорівнювала була – 4 переходи



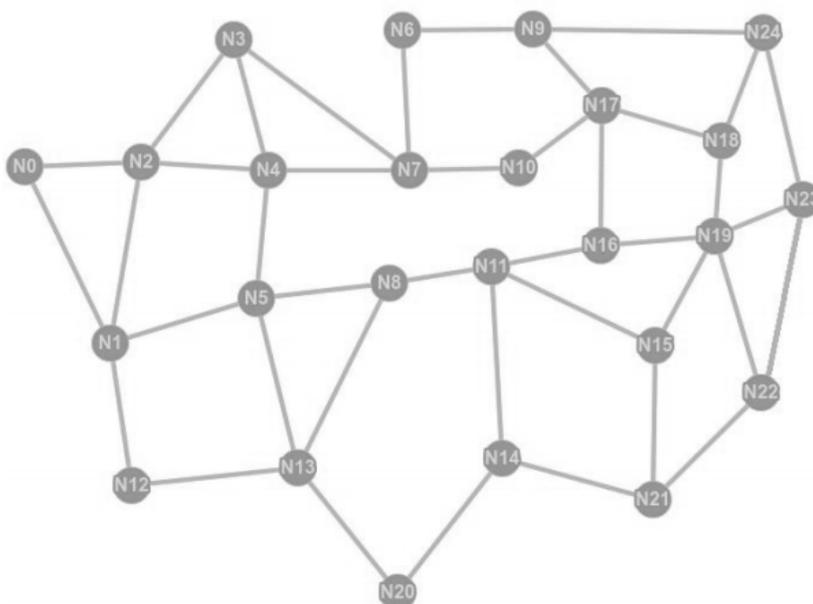
**Рис. 3. Стан мережі після розподілу навантаження 10 mb/s**

У випадку якщо потрібно пропустити потік даних 30 mb/s, даний алгоритм після первинного розподілу навантаження між найкоротшими шляхами знаходить навантаження яке неможливо скорегувати (сумарна пропускна здатність R1 та R2 – 20mb/s). Після цього відбувається додавання більш довгих шляхів із введеного масиву (R3 та R4) та відбувається нова спроба розподілу навантаження. Результат роботи можна побачити на рисунку 4.



**Рис. 4. Стан мережі після розподілу навантаження 30 mb/s**

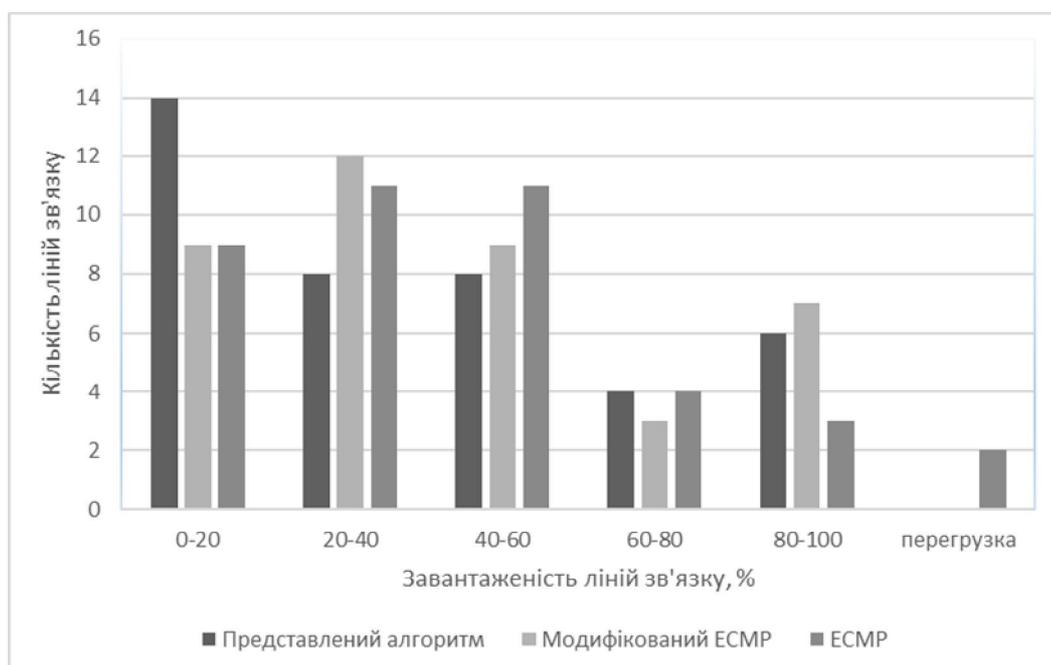
**Тестування.** Граф тестової мережі зображений на рисунку 5. Пропускна здатність ліній зв'язку дорівнює 50mb. Для знаходження можливих шляхів використовується алгоритм описаний у роботі [5]. В процесі тестування було прокладено маршрути між вузлами: (12, 24), (3, 14) (21, 24), (3, 12). Виділення пропускної здатності відбувається в порядку задання маршрутів, а розмір потоку даних — 40 mb/s.



**Рис. 5. Граф мережі на якій проводиться тестування**

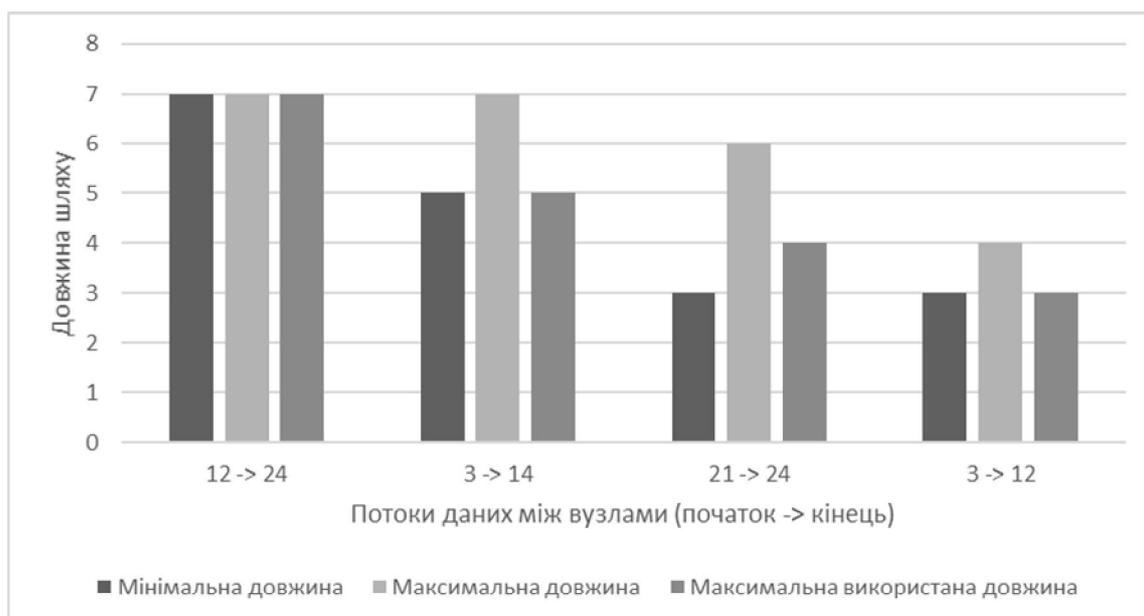
На рис. 6 зображено порівняння розподілу навантаження між класичним ЕСМР, модифікованим ЕСМР з роботи [4] та алгоритмом представленим у даній роботі, що враховує довжину маршрутів.

Як бачимо з графіку, розроблений алгоритм показує більш усереднені результати у порівнянні з базовою версією. Акцент навантаження змістився з 20%-60% до 60%-100% та 0%-20%, тобто алгоритм спочатку намагається максимально навантажити найкоротші шляхи, а при неможливості цього починає розділяти потік даних між більш довгими. Через це збільшилась кількість мінімально та максимально завантажених ланок. Через те, що даний алгоритм базується на модифікації ЕСМР, у нього взагалі відсутні перенавантажені лінії зв'язку.



**Рис. 6. Розподіл навантаження у мережі**

На рисунку 7 показано максимальна використана довжина шляху для кожного маршруту тестування.



**Рис. 7. Довжини використаних шляхів**

Як видно з графіку, при наявності більш коротких шляхів, алгоритм намагається використати лише їх, без залучення більш довгих. Наприклад для потоку між вузлами 21 та 24 знайдено декілька шляхів. Мінімальна довжина знайденого шляху - 3 хопи, максимальна – 6 (також було знайдено декілька шляхів довжиною - 4). Після неможливості розподілити потік через найкоротші шляхи, алгоритм починає обирати більш довші, до моменту поки не зможе оптимально розподілити навантаження. Через обрання більш коротки маршрутів виникає зменшення затримки передачі даних між вузлами мережі.

**Висновки.** Даний алгоритм був розроблений з метою покращення роботи балансувальника навантаження при роботі з частково-пересічними маршрутами. Базова версія даного алгоритму має досить великий недолік у вигляді використання досить довгих маршрутів передачі даних у випадках коли в цьому немає необхідності, а саме у випадках можливості розподілити потік між

більш короткими шляхами. Як видно з результатів тестування, даний недолік вирішено у алгоритмі представленому в даній роботі, без втрати якості балансування. Тому можна сказати, що розроблений алгоритм вирішує поставлену проблему.

***Література:***

1. W. J. Dally and B. Towles (2004). Principles and Practices of Interconnection Networks. Morgan Kaufmann Publisher, 2004
2. Hopps, C. E. (2000). Analysis of an equal-cost multi-path algorithm  
<<https://tools.ietf.org/html/rfc2992>>
3. Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2014). A survey on software-defined networking. IEEE Communications Surveys & Tutorials, 17(1), 27-51.
4. Калюжний О. О. (2019). Алгоритм балансування навантаження для використання з частково-пересічними шляхами передачі даних. Міжнародний науковий журнал "Науковий огляд", 6(59), Київ, ТК Меганом.
5. Калюжний О. О., Кулаков Ю. О., Диброва М. О. (2018). Спосіб організації багатозляхової маршрутизації з використанням частково-пересічних шляхів. The International Conference on Security, Fault Tolerance, Intelligence: зб. наук. праць міжнародної науково-технічної конференції., 2018р., 310-314.

***References:***

1. W. J. Dally and B. Towles (2010). Principles and Practices of Interconnection Networks. Morgan Kaufmann Publisher, 2004
2. Hopps, C. E. (2000). Analysis of an equal-cost multi-path algorithm  
<<https://tools.ietf.org/html/rfc2992>>

3. Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2014). A survey on software-defined networking. *IEEE Communications Surveys & Tutorials*, 17(1), 27-51.
4. Kaliuzhnyi O.O. (2019). Alhorytm balansuvannia navantazhennia dlia vykorystannia z chastkovo-peresichnymy shliakhamy peredachi danykh. *Mizhnarodnyi naukovyi zhurnal "Naukovyi ohliad"*, 6(59), Kyiv, TK Meganom.
5. Kaliuzhnyi O. O., Kulakov Yu. O., Dybrova M. O. (2018) Sposib orhanizatsii bahatoshliakhovoi marshrutyzatsii z vykorystanniam chastkovo-peresichnykh shliakhiv. *The International Conference on Security, Fault Tolerance, Intelligence: zb. nauk. prats mizhnarodnoi naukovo-tekhnichnoi konferentsii.*, 2018r., 310-314