

## СПОСІБ МАРШРУТИЗАЦІЇ У КОМП'ЮТЕРНИХ МЕРЕЖАХ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ SDN

Дана стаття присвячена способу формування множини непересічних шляхів в комп'ютерних мережах на основі технології SDN. Демонструється детальний алгоритм роботи способу та описуються його переваги та недоліки, приводиться приклад роботи для довільно визначеного графу мережі. На основі розробленого додатку для моделювання мережі тестуються якісні характеристики утвореного алгоритму на практиці.

The subject of the article is way of routing at computer networks with using SDN technology. A detail algorithm of work of this way is demonstrated, benefits and drawbacks of it are given, also an example of work for defined at random network graph is shown. On a basis of designed application for modelling of network the qualitative characteristics of the created algorithm in practice.

**Ключові слова:** SDN, комп'ютерна мережа, граф, алгоритм, маршрутизація.

### 1. Введення

Нинішній спосіб керування мережами у великих масштабах є неефективним. Мережа конфігурується через командну строку або через веб-браузер для кожного вузла. Навіть з використанням сучасних веб-інтерфейсів на новіших комутаторах, вузли конфігуруються індивідуально і залишаються статичними. Однак, з допомогою SDN, користувачі можуть керувати, налаштовувати та контролювати мережі з допомогою окремих контролерів. Це забезпечує систему, у якій керування різними вузлами відбувається через один пристрій, а не їх множини, як раніше.

Також, на даний момент, ситуація з алгоритмами маршрутизації та їх ефективністю не є однозначною. Серед основних використовуваних алгоритмів можна виділити такі як алгоритм Дейкстри, модифікований алгоритм Дейкстри, «гілок і границь», «хвильовий» алгоритм, «зустрічної хвилі», алгоритм Беллмана-Форда [1]. Проте, більшість з них має ряд таких недоліків, як велика часова складність, лінійність або лише часткове розпаралелювання, відсутність гарантії побудови хоча б одного шляху у результаті роботи, велика ресурсоемність, яка займається кожним пристроєм-вузлом, що утворює множини шляхів. Також суттєвим недоліком усіх цих способів маршрутизації є невикористання інформації про утворені шляхи для побудови шляхів між внутрішніми вузлами. Розроблений спосіб маршрутизації покликаний подолати ці недоліки завдяки поєднанню

централізованого керування у SDN та модифікації та адаптації «хвильового алгоритму».

### 2. Постановка задачі

З урахуванням вказаних недоліків вище описаних методів формування множини непересічних шляхів необхідно розробити алгоритм, що буде відповідати наступним вимогами:

- 1) Часова складність менша за складність алгоритму Дейкстри;
- 2) Врахування метрики під час роботи алгоритму;
- 3) Гарантія знаходження принаймні одного шляху між початковою і кінцевою вершинами
- 4) Можливість використати результати обчислень попередніх шляхів для формування наступних маршрутів між іншими вершинами

### 3. Опис способу маршрутизації

Часова складність формування багатоканального віртуального шляху, що складається з  $q$  фізичних каналів за допомогою комбінаторних алгоритмів, наприклад, алгоритму Дейкстри дорівнює  $O(N^2)$  [2]. Теоретично зменшити часову складність можна за рахунок виключення операції перебору варіантів на основі методу «гілок і границь», проте даний алгоритм не дозволяє враховувати метрики шляху при організації багатошляхової маршрутизації. У зв'язку з цим, в статті була

запропонована модифікація хвильового алгоритму [3] багатошляхової маршрутизації, алгоритм якої наведено нижче.

1. Begin;
2. For Start node B0 make a set of connected nodes Bs and declare it as Wave set;
3. Make a set of ways W from node B0 to nodes of Bs;
4. Make a set of Working nodes as last nodes of ways of W;
5. Make a set of Current nodes = Wave set;
6. Make new Wave set from Current nodes to connected nodes to them;
7. Sort Working nodes on condition of outer connectivity;
8. Choose first from Working nodes and make hop to node with least connectivity from Wave set;
9. Is the hop done to End node? If “yes” – continue, else – go to 17;
10. Mark this node as inactive;
11. Does this node have any non Wave set active nodes? If “yes” – continue, else go to 18;
12. Does this node have any Wave set active nodes? If “yes” – continue, else go to 14;
13. Make hop to active node with least connectivity from Wave set, go to 9;
14. Cut off this node from current way, make current node previous;
15. Is this node in Wave set? If “yes” – go to 9, else continue;
16. Way is failed, delete it from W, go to 18;

17. Finish current way;
18. Is W empty? If “yes” – go to 21, else continue;
19. Does Wave set have active nodes? If “yes” – go to 8, else continue;
20. Are there any active nodes except of End node? If yes – go to 4, else continue;
21. End.

Особливістю даного модифікованого алгоритму є можливість сформувати всі можливі непересічні шляхи з частковим урахуванням їх метрик і оптимізувати їх за допомогою алгоритму пошуку в ширину. Часова складність даного алгоритму обчислюється за формулою [4]:

$$O\left(\sum_{i=1}^k \sum_{j=1}^n (N_{N_{ij}})^2\right), \quad (1)$$

де  $k$  – кількість непересічних шляхів;

$n$  – кількість вершин у шляху  $L_i$ ;

$N_{N_{ij}}$  – зв'язність вершини  $N_j$  шляху  $L_i$ .

Розділення роботи алгоритму на етапи формування множини непересічних шляхів і їх оптимізації дозволяє вводити питання раціональності оптимізації в конкретних умовах і конкретних ресурсах, а також надає можливість надсилати пакети до вузла призначення до завершення процесу оптимізації заради економії часу [5].

#### 4. Приклад роботи алгоритму

Розглянемо приклад роботи модифікованого хвильового алгоритму формування множини непересічних шляхів між початковою вершиною  $v_1$  та кінцевою вершиною  $v_{17}$  (рис. 1.1)

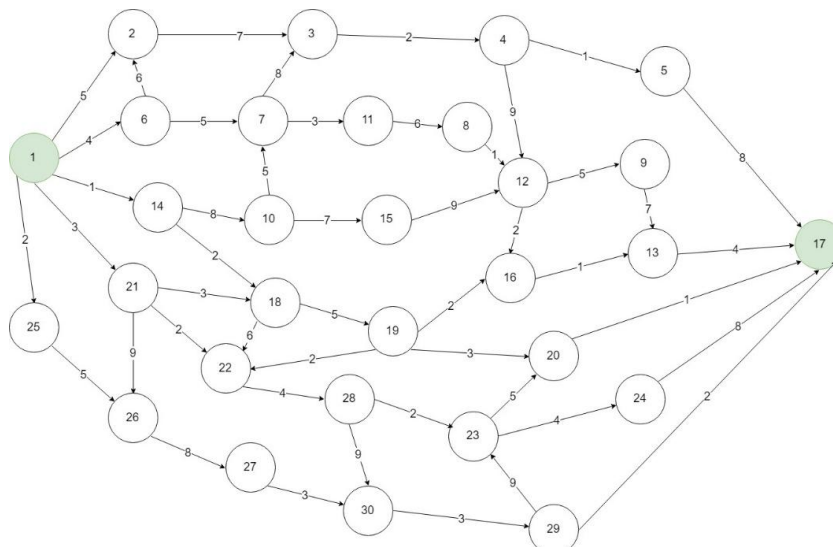
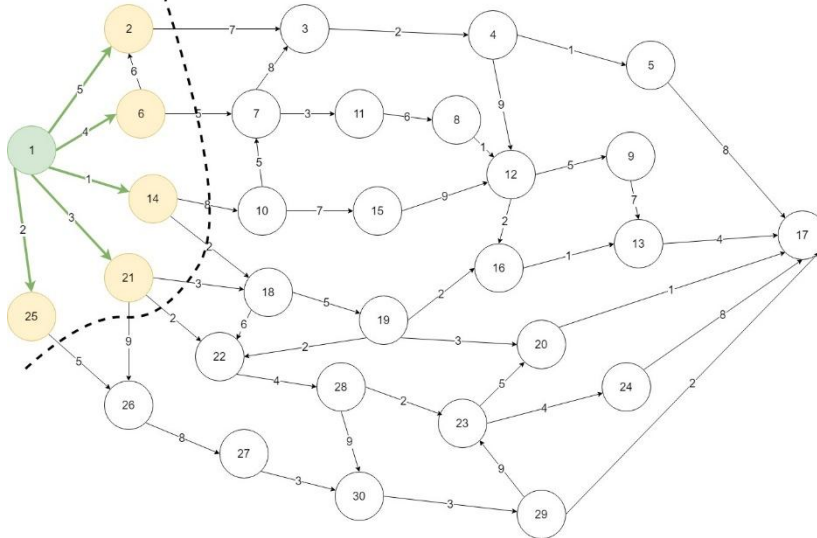


Рис. 1.1 Граф мережі

На початковому етапі будується множина суміжних вершин до початкової. Так, у даному графі до вершини  $v_0$  суміжними будуть вершини  $v_2, v_6, v_{14}, v_{21}, v_{25}$ . Далі будуються шляхи з початкової до даних суміжних шляхів:

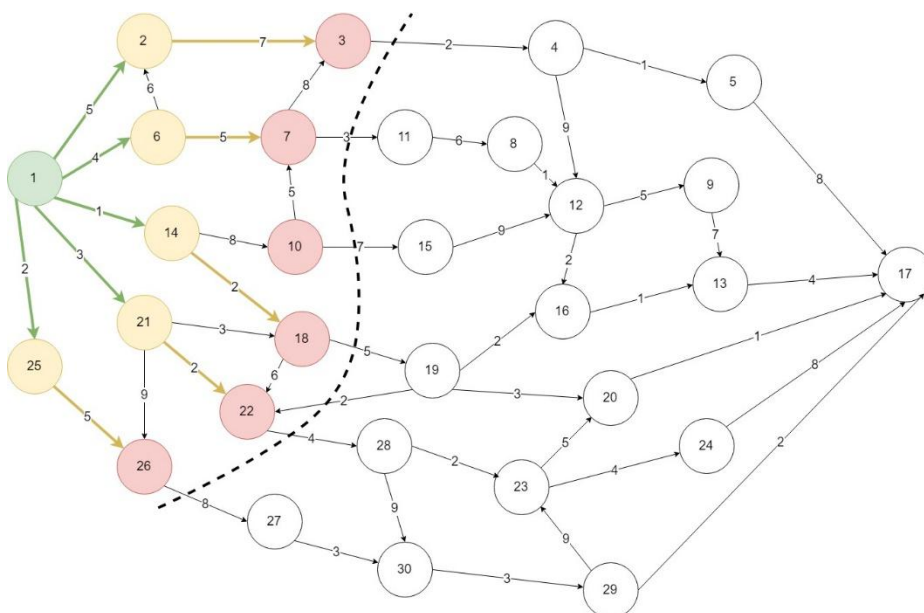
$P_1=\{v_1, v_2\}$ ,  $P_2=\{v_1, v_6\}$ ,  $P_3=\{v_1, v_{14}\}$ ,  $P_4=\{v_1, v_{21}\}$ ,  $P_5=\{v_1, v_{25}\}$  (рис 1.2). Множина суміжних вершин до початкової є множиною вершин першого рівня  $L_1=\{v_2, v_6, v_{14}, v_{21}, v_{25}\}$ .



**Рис. 1.2 Початковий рівень формування шляхів**

На наступному кроці шляхи продовжуються до другого рівня  $L_2=\{v_3, v_7, v_{18}, v_{22}, v_{26}\}$ . Це відбувається при дотриманні наступного правила: першим відбувається перехід з того шляху, остання вершина якого має найменшу зовнішню зв'язність у вершину що також має найменшу зв'язність. У випадку, коли якісь вершини з яких відбувається перехід мають однакові степені зв'язності, вершина обирається довільно. Якщо ж така ситуація

складається з вершинами, у які може відбутись перехід, то пріоритет віддається переходу з меншою метрикою. У даному випадку, шляхи продовжуються у такій послідовності (рис. 1.3) :  $P_1=\{v_1, v_2, v_3\}$ ,  $P_2=\{v_1, v_6, v_7\}$ ,  $P_3=\{v_1, v_{14}, v_{18}\}$ ,  $P_4=\{v_1, v_{21}, v_{22}\}$ ,  $P_5=\{v_1, v_{25}, v_{26}\}$ , оскільки вершини  $v_2, v_6, v_{21}, v_{25}$  мають однаковий степінь зовнішньої зв'язності (1), а вершина  $v_{14}$  має степінь (2). У вершині  $v_{14}$  вибір переходу відбувається за критерієм меншої метрики.



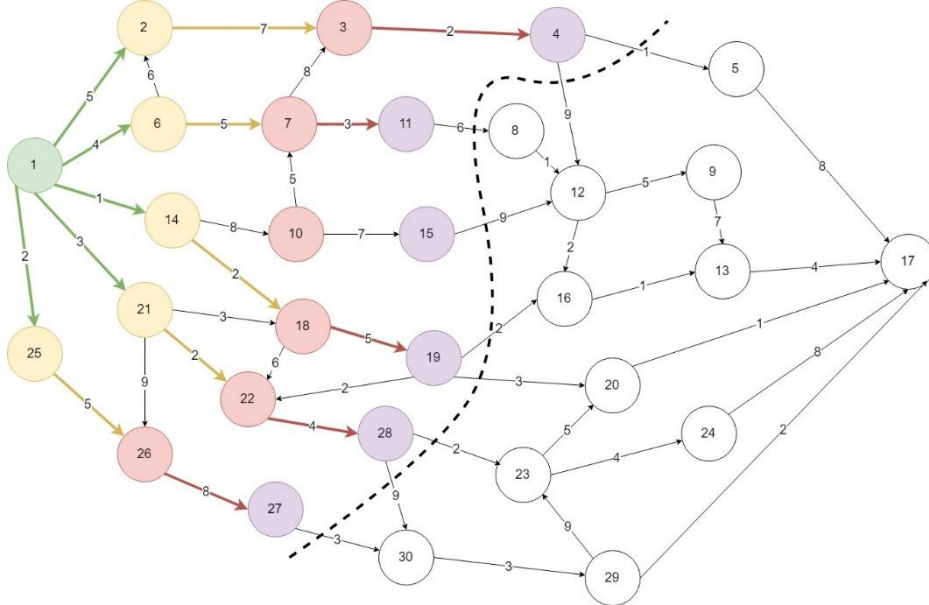
**Рис. 1.3 Другий етап формування шляхів**

На третьому кроці шляхи продовжуються до вершин з множини вершин третього рівня

$L_3=\{v_4, v_{11}, v_{15}, v_{19}, v_{28}, v_{27}\}$  (рис. 1.4). Утворюються шляхи наступним чином:  $P_2=\{v_1,$

$v_6, v_7, v_{11}$ },  $P_5=\{v_1, v_{25}, v_{26}, v_{27}\}$ ,  $P_1=\{v_1, v_2, v_3, v_4\}$ ,  $P_3=\{v_1, v_{14}, v_{18}, v_{19}\}$ ,  $P_4=\{v_1, v_{21}, v_{22}, v_{28}\}$ , останні вершини яких мають такі степені зв'язності відповідно: (1), (1), (2), (2), (2). Як і на

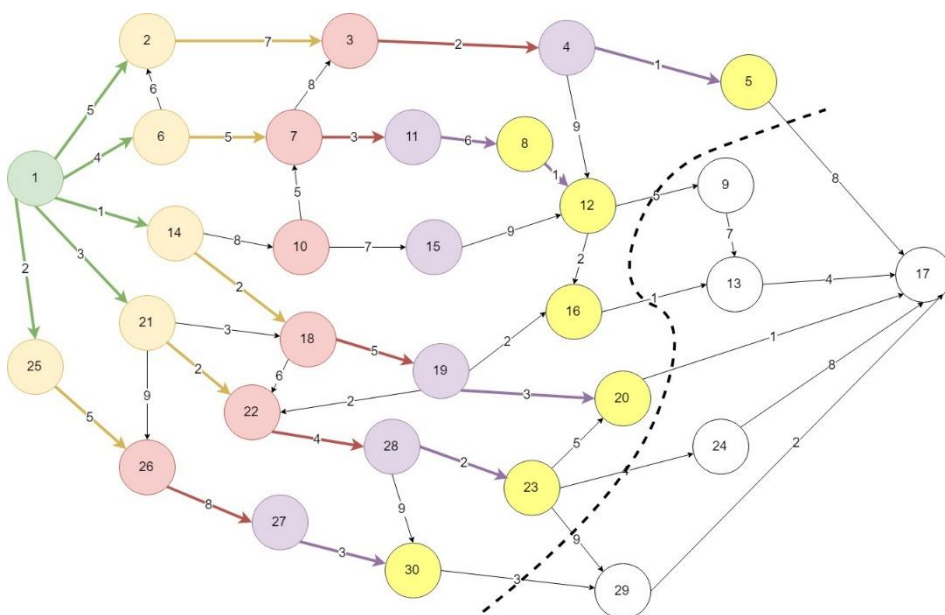
попередньому кроці, вибір вершини при степені більше одиниці відбувається за критерієм найменшої метрики.



**Рис. 1.4 Третій етап формування шляхів**

На четвертому кроці формується множина вершин відповідного рівня  $L_4=\{v_5, v_8, v_{12}, v_{16}, v_{20}, v_{23}, v_{30}\}$  (рис. 1.5). Спочатку продовжується шлях  $P_5=\{v_1, v_{25}, v_{26}, v_{27}, v_{30}\}$ . Далі доходить черга до маршруту  $P_2$ . Оскільки вершина  $v_{11}$  має єдиний перехід у вершину  $v_8$ , то цей перехід і відбувається –  $P_2=\{v_1, v_6, v_7, v_{11}, v_8\}$ . Проте вершина  $v_8$  має перехід лише у вершину, що також відноситься до множини вершин  $L_4$  – вершину  $v_{12}$ . Через це відбувається

продовження шляху до даної вершини з очікуванням того, що ця вершина має суміжні вершини не з хвили, або принаймні незайняті жодним іншим маршрутом вершини з хвили. Оскільки вершина  $v_{12}$  має перехід до вершини  $v_9$ , то перехід затверджується і шлях продовжується –  $P_2=\{v_1, v_6, v_7, v_{11}, v_8, v_{12}\}$ . Надалі відбувається продовження шляхів  $P_1=\{v_1, v_2, v_3, v_4, v_5\}$ ,  $P_3=\{v_1, v_{14}, v_{18}, v_{19}, v_{20}\}$ ,  $P_4=\{v_1, v_{21}, v_{22}, v_{28}, v_{23}\}$



**Рис. 1.5 Четвертий етап формування шляхів**

На п'ятому кроці у множині вершин п'ятого рівня  $L_5 = \{v_{17}, v_9, v_{13}, v_{24}, v_{29}\}$  (рис. 1.6) потрапляє кінцева вершина  $v_{17}$ . Вона може бути досягнута з шляху  $P_1$ , тобто тепер цей шлях сформовано –  $P_1 = \{v_1, v_2, v_3, v_4, v_5, v_{17}\}$ . Далі формується остаточно шлях  $P_3 = \{v_1, v_{14}, v_{18}, v_{19}, v_{20}, v_{17}\}$ . Наступним відбувається перехід у вершину  $v_{24}$  зі шляху  $P_4 = \{v_1, v_{21}, v_{22}, v_{28}, v_{23}\}$ . Проте знову відбувається ситуація, аналогічна випадку з четвертого етапу: вершина  $v_{24}$  серед суміжних вершин має лише вершину з хвилі (що по сумісництву є кінцевою вершиною). Тому відбувається додатковий перехід у вершину  $v_{17}$  і шлях  $P_4 = \{v_1, v_{21}, v_{22}, v_{28},$

$v_{23}, v_{24}, v_{17}\}$  завершується. Таким самим чином шлях  $P_5$  продовжується до вершини  $v_{29}$ , а далі до кінцевої вершини  $v_{17}$  –  $P_5 = \{v_1, v_{25}, v_{26}, v_{27}, v_{30}, v_{29}, v_{17}\}$ . Шлях  $P_2$  продовжується по аналогічному сценарію, єдина відмінність, що після переходу у вершину  $v_9$  відбувається додатковий перехід у вершину хвилі  $v_{13}$ , що також не задовольняє умові наявності суміжної вершини, що не входить у множину  $L_5$ , тому звідти відбувається ще один додатковий перехід у вершину  $v_{17}$ , що є кінцевою і відповідає умові завершення шляху,  $P_2 = \{v_1, v_6, v_7, v_{11}, v_8, v_{12}, v_9, v_{13}, v_{17}\}$ .

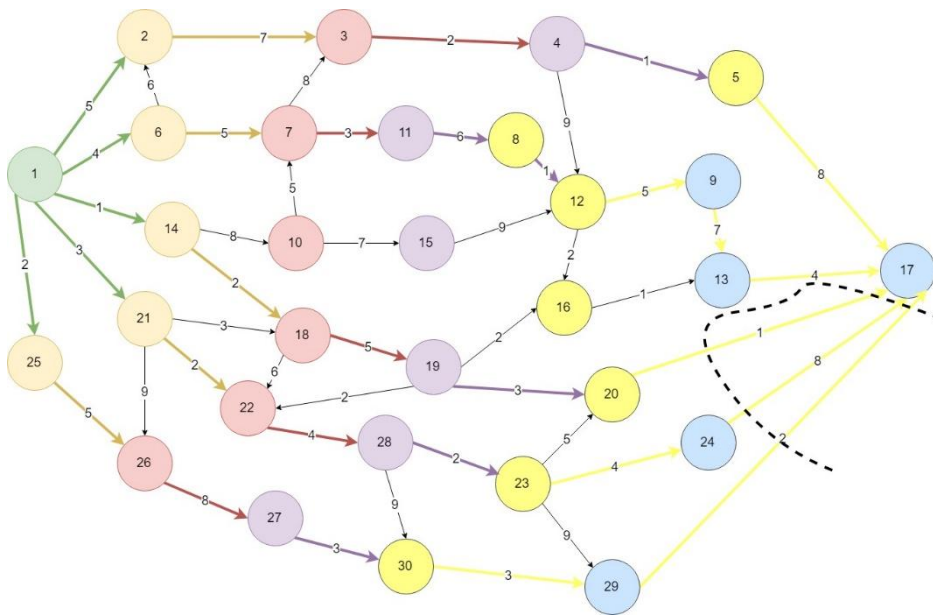


Рис. 1.6 Останній етап формування шляхів

У результаті було побудовано п'ять непересічних шляхів за п'ять кроків основної частини алгоритму. Даний алгоритм може забезпечити побудову принаймні одного шляху між парою вершин; має часову складність меншу, аніж  $O(N^2)$ ; потенційно може бути розпаралеленим на кількість потоків, що не перевищує кількості сформованих шляхів на першому кроці. Алгоритм може вибирати оптимальний шлях за критерієм метрики не в шкоду до непересічності шляхів, тому потребує додаткового алгоритму оптимізації, що дозволить забезпечити мінімальну довжину попередньо прокладених маршрутів. Згідно з формулою (1) часова складність алгоритму для даної конфігурації обраховується наступним чином.

У утворених шляхів  $P_1 = \{v_1, v_2, v_3, v_4, v_5, v_{17}\}$ ,  $P_2 = \{v_1, v_6, v_7, v_{11}, v_8, v_{12}, v_9, v_{13}, v_{17}\}$ ,  $P_3 = \{v_1, v_{14}, v_{18}, v_{19}, v_{20}, v_{17}\}$ ,  $P_4 = \{v_1, v_{21}, v_{22}, v_{28}, v_{23}\}$ ,  $P_5 = \{v_1, v_{25}, v_{26}, v_{27}, v_{30}, v_{29}, v_{17}\}$  вершини мають

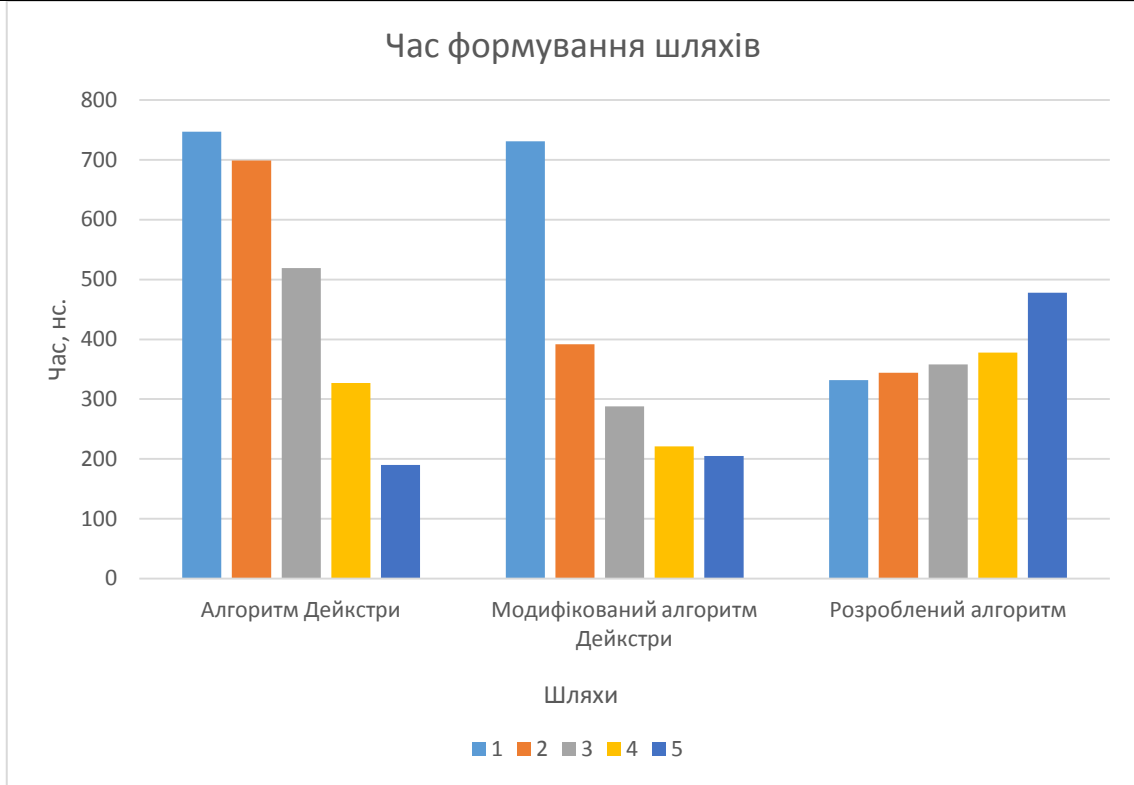
такі кількості сусідів, як  $L_1 = \{5, 3, 3, 3, 2, 5\}$ ,  $L_2 = \{5, 3, 4, 2, 2, 5, 2, 3, 5\}$ ,  $L_3 = \{5, 3, 4, 4, 3, 5\}$ ,  $L_4 = \{5, 4, 4, 3, 4, 2, 5\}$ ,  $L_5 = \{5, 2, 3, 2, 3, 3, 5\}$ . Після підсумування квадратів цих величин отримуємо часову складність, яка рівна 494.

## 5. Тестування способу маршрутизації

Для тестування програми було обрано вище описаний граф з 30 вершин.

Для досягнення чистоти результатів і зменшення похибок при вимірюваннях було проведено по 1000 тестів роботи розробленого алгоритму, алгоритму Дейкстри і модифікованого алгоритму Дейкстри. Стартовою вершиною обрано вершину №1 з степінню 5, кінцевою – вершину №17 з степінню 5.

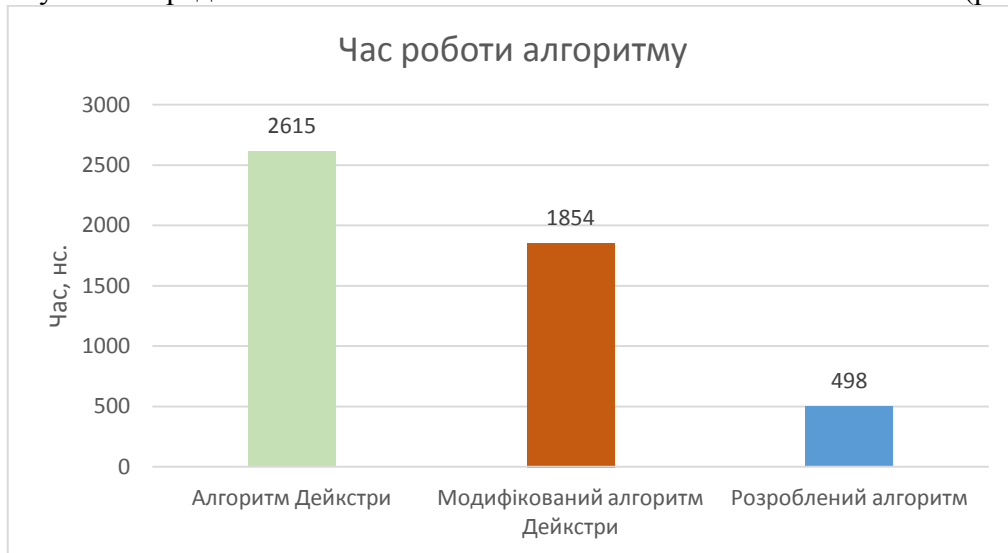
На рис. 1.7 представлено середнє значення часової складності формування 5 шляхів для зазначених алгоритмів.



**Рисунок 1.7 Середній час роботи алгоритмів маршрутизації**

Виходячи з результатів виконання алгоритмів чітко видно основну відмінність і перевагу розробленого способу маршрутизації – одночасна побудова шляхів суттєво зменшує часову складність, різниця у часі побудови чергового шляху з попереднім мінімальна. У

той же час обидві варіації алгоритму Дейкстри будують свої шляхи та групи шляхів послідовно і кожен етап побудови залежить від попереднього. Через це загальний час виконання розробленого способу значно менший за час виконання інших (рис 1.8).



**Рисунок 1.8 Загальний час роботи алгоритму**

Для даної конфігурації тестів приріст ефективності досягнув 425% у порівнянні з алгоритмом Дейкстри і 272% у порівнянні з модифікованим алгоритмом Дейкстри. Під час тестування даного графу для інших пар вершин приріст ефективності варіювався від 324% до

486% у порівнянні з алгоритмом Дейкстри і від 190% до 351% у порівнянні з його модифікацією. Варто зазначити, що розроблений алгоритм не гарантує формування максимально можливої кількості шляхів, і це

може бути його недоліком у порівнянні з іншими.

## 6. Висновки

У статті описано алгоритм, що задовольняє поставлені умови та інтегрується з рішеннями технології SDN, використовуючи її переваги. Було надано загальні відомості про багатошляхову маршрутизацію в цілому та програмно-конфігуровані мережі. Був проведений огляд найпопулярніших алгоритмів маршрутизації, визначені їх головні переваги та

проаналізовано недоліки. Для тестування розробленого способу створено систему моделювання маршрутизації. На її базі було показано, що розроблений алгоритм дозволяє істотно зменшити реальну часову складність у порівнянні з розглянутими алгоритмами, реальні і теоретичні часові складності збігаються у межах допустимої похибки.

Розроблений спосіб багатошляхової маршрутизації може бути ефективно використаний у реальних SDN мережах після його адаптації під конкретну систему.

## Список використаної літератури

1. Shafi'i Muhammad Abdulhamid. On-Demand Grid Provisioning Using Cloud Infrastructures and Related Virtualization Tools: A Survey and Taxonomy / Shafi'i Muhammad Abdulhamid, Muhammad Shafie Abd Latiff, Mohammed Bakri Bashir // International Journal of Advanced Studies in Computer Science and Engineering (IJASCSE), Volume 3, Issue 1, February 2014, p. 49-59.
2. N. Mohan Krishna Varma. Extending Grid Infrastructure Using Cloud Computing / N. Mohan Krishna Varma, and Eunmi Choi // Ubiquitous Information Technologies and Applications, Volume 2014, November 2012, p. 507-516.
3. Кулаков Ю.А. Способ формирования структуры виртуальной Grid-системы / Кулаков Ю.А., Олещенко А.Ю. // Вісник університету «Україна» №2, 2011. С.71-75.
4. Hamada Alshaer. An Overview of Network Virtualization and Cloud Network as a Service / International Journal of Network Management 1 December 2014, in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nem.1882.
5. Faizul Bari. Data center network virtualization: a survey / Faizul Bari, Raouf Boutaba, Rafael Esteves, Lisandro Zambenedetti Granville, Maxim Podlesny, Golam Rabbani, Qi Zhang, Mohamed Faten Zhani // Communications Surveys and Tutorials, Volume 15, Issue 2, September 2012, p. 909-928.