УДК 004.735
DOI https://doi.org/10.32782/2663-5941/2025.1.2/17

***Киричек Г.Г.***
Національний університет «Запорізька політехніка»

***Пестов О.Д.***
Національний університет «Запорізька політехніка»

***Тягунова М.Ю.***
Національний університет «Запорізька політехніка»

# МАСШТАБНА ДЕЦЕНТРАЛІЗОВАНА MESH МЕРЕЖА

*В роботі досліджується масштабованість експериментальної схеми маршрутизації Yggdrasil щодо логіки, ліміту переходів, використання процесора та пам'яті. Експерименти з демоном маршрутизації проведено на різного розміру топологіях із використанням середовищ meshnet-lab і coreemu-lab. Виявлено певні особливості використання даної схеми маршрутизації як основи масштабної децентралізованої mesh мережі. Метою роботи є проведення досліджень, визначення доцільності використання схеми маршрутизації Yggdrasil, як основи масштабної децентралізованої mesh мережі. Об'єктом дослідження є процес проєктування масштабної децентралізованої mesh мережі на основі схеми маршрутизації Yggdrasil. Предметом є моделі, методи та програмні засоби реалізації масштабної децентралізованої mesh мережі на основі Yggdrasil. У рамках дослідження запропоновано нові підходи до розгортання масштабної децентралізованої mesh мережі як альтернативи ієрархічним мережами інтернет-провайдерів. Авторами визначено збільшення використання процесору на довгих гілках остовного дерева і аномалії трафіку, які внесені блукаючим кореневим вузлом та запропоновано попереджувальний майнінг малих ключів для детермінованого розміщення кореневих вузлів, також наведено модель системи та обґрунтовано використання стандарту IEEE 802.11s з вимкненою маршрутизацією як базової технології канального рівня. А також отримано орієнтовну структуру масштабної децентралізованої mesh мережі та технічні вимоги до її вузлів. Результати показали, що Yggdrasil значно перевершує інші протоколи mesh маршрутизації, такі як OLSR і Babel в обмеженні переходів, визначаючи параметри використання системних ресурсів. Експерименти демонструють типовий час конвергенції, а також тенденції використання пам'яті та процесора Yggdrasil і підтверджують їх масштабування з глибиною мережі, а не з її загальним розміром. Тому Yggdrasil є перспективним методом розгортання великомасштабних децентралізованих сітчастих мереж незалежних вузлів і альтернативою централізованим ієрархічним мережам.*

***Ключові слова:*** *масштабованість, mesh мережа, Yggdrasil, маршрутизація, IEEE 802.11s.*

**Постановка проблеми.** В наші дні централізовані програмно-визначені мережі стають все більш популярними серед інтернет-провайдерів завдяки перевагам у автоматизованому віддаленому налаштуванні обладнання, моніторингу, контролі, простоті розгортання, використанні апаратного забезпечення типу «white box» тощо [1]. Ці переваги є можливими за рахунок надійності мережі та стійкості до збоїв, бо централізація неминуче додає системі єдині точки відмови [2]. Є багато прикладів того, наскільки ненадійними є централізовані ієрархічні системи зв'язку, що вимагає іншого підходу – сітчастої (mesh) мережі, де межа між маршрутизаторами та кінцевими пристроями стає розмитою. Кожен вузол мережі функціонує незалежно, встановлюючи зв'язки з іншими вузлами поблизу з метою маршрутизації трафіку в мережі не покладаючись на центральний орган [1, 2]. Потенційним вирішенням проблем, які вникають є використання експериментальної схеми маршрутизації Yggdrasil [3], основна суть якої полягає у використанні адресації на основі відкритих ключів Ed25519 [4] і самоорганізованого остовного дерева мережі. Останнє схоже на CottonCandy, за винятком того, що замість визначеного шлюзу коренем дерева стає вузол із найнижчим значенням ключа, і може бути автоматично заміненим у разі збою. Пошук вузлів виконується на вимогу і за допомогою широкомовних розсилок, які фільтруються bloom-фільтрами. Результатом такого пошуку є отримання координат вузла, який приймає дані, відносно поточного кореня мережі, за якими надсилається трафік. Для визначення напрямку трафіку, на кожному

наступному вузлі застосовується опортуністична жадібна маршрутизація [5], пакети надсилаються через з'єднання, що наблизить їх до вказаних координат.

Через те, що схема маршрутизації Yggdrasil є відносно новою та все ще перебуває на альфастадії розробки, існує не так багато робіт, які охоплюють її функціональність або просто наводять, як приклад. Автори наукових робіт представляють результати застосування порівняльних тестів при використанні оверлейної тестової мережі Yggdrasil як інструменту обходу NAT при віддаленому керуванні, але без тестування масштабованості самої схеми маршрутизації [6]. Інші автори порівнюють Yggdrasil із іншими мережними протоколами маршрутизації в ряді бенчмарків, де лише один з них підтримує масштабування понад 100 вузлів [7].

**Аналіз останніх досліджень і публікацій.** Дослідження з питань маршрутизації для великих mesh мереж зосереджені на її реалізації для IoT та бездротових сенсорних мереж, з урахуванням оптимізації енергоспоживання і віддаленості за рахунок пропускної здатності та безпеки [8]. Наприклад, мережа на основі CottonCandy працює за принципом організації вузлів в остовне дерево (spanning tree) від вузла Інтернет-шлюзу, призначеного коренем дерева, при цьому уникаючи зіткнень завдяки використанню рекурсивних запитів даних під час робочих циклів (рис. 1) [9].

Додатковим засобом уникнення зіткнень є використання вузлами різних каналів відповідно до остовного дерева – кожен вузол прослуховує батьківський канал на предмет рекурсивних запитів, а пропагує їх за власним каналом, відмінним від батьківського (рис. 2).

Хоча підхід добре працює при невеликих обсягах сенсорних даних, він не підходить для спонтанного зв'язку між вузлами в мережі загального призначення. Крім того, наявність чітко визначеного кореневого вузла додає до мережі єдину точку збою, що перешкоджає загальній меті децентралізації.

Ямамото та інші автори пропонують протокол маршрутизації VORTEX – підхід, заснований на опортуністичній маршрутизації ієрархічної структури рівнів, яка встановлена на початковому етапі (рис. 3) [10]. Рівні розподіляються за наступним алгоритмом: вузли періодично надсилають HELLO-пакети та прослуховують ефір на предмет отримання пакетів від інших вузлів; з кількості унікальних ID у HELLO-пакетах кожен вузол має кількість сусідів та їх ступені; якщо вузол має найбільшу ступінь, він підвищує свій ступінь до 1, інакше надсилає UTR (запит підвищення рівня) вузлу з найбільшим ступенем; отримувач UTR підвищує свій рівень до 1; аналогічно серед вузлів першого рівня обираються вузли другого рівня.

Для пари кінцевих вузлів мережа виглядає як «чорна скринька», а проміжні вузли просто пересилають пакети вздовж ієрархії, поки вони не досягнуть одержувача. Пакети проходять кількома шляхами для забезпечення надійності. Необхідність здійснювати виявлення маршруту відпадає, однак мережа переповнюється додатковим трафіком, який знижує її продуктивність.

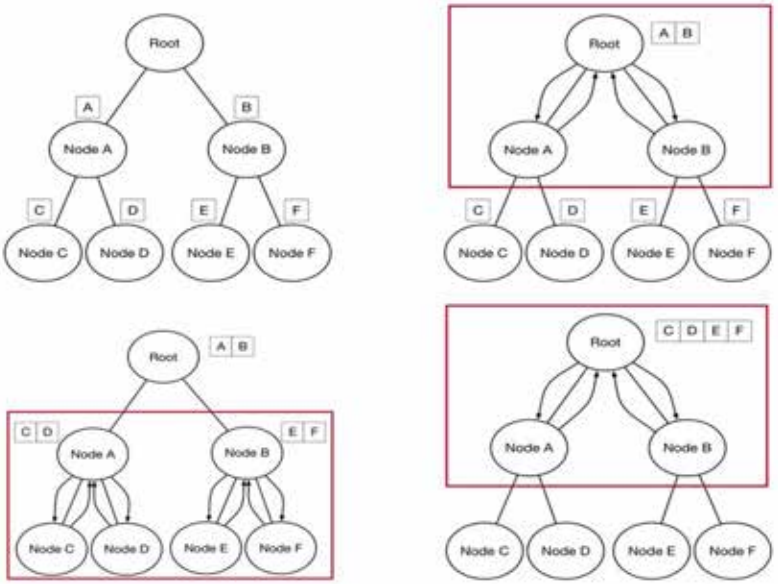


**Рис. 1. Рекурсивне збирання даних у остовному дереві CottonCandy [9]**

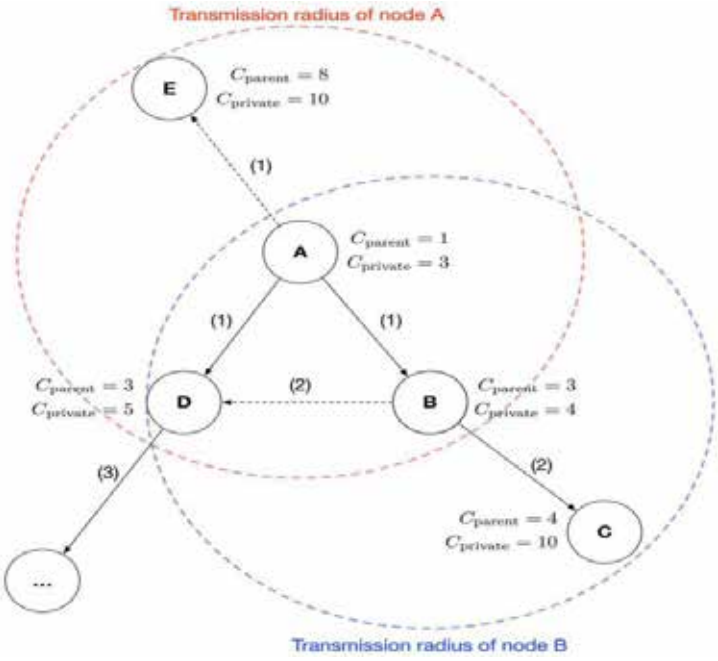


**Рис. 2. Канальний розподіл CottonCandy [9]**

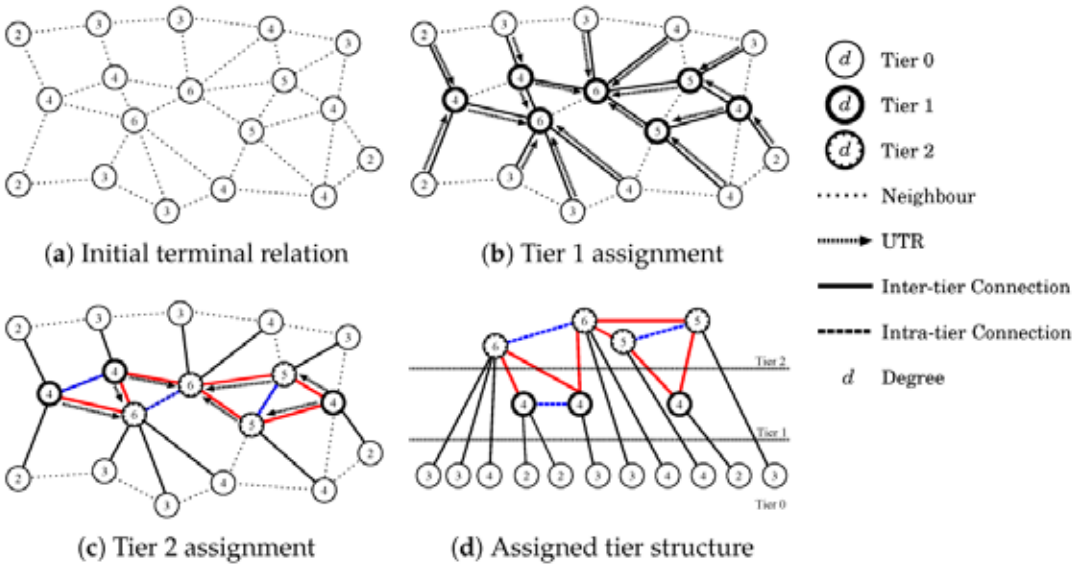


**Рис. 3. Розподіл вузлів за рівнями у VORTEX [10]**

Прабу та інші [11] мають новий підхід до mesh маршрутизації – концепцію «теплових слідів» і представляють її за допомогою bloom-фільтрів на кількох рівнях градієнта інтенсивності (рис. 4). Під час пошуку маршруту вузли випадковим чином «переглядають» мережу, поки не досягнуть «тепла» вузла призначення у bloom-фільтрі останнього рівня, а потім слідують за ним у напрямку збільшення інтенсивності (рис. 5). Інші вузли можуть слідувати до місця призначення, повторно використовуючи маршрут. При цьому знайдений маршрут поступово оптимізується під час використання (рис. 6).

Основним недоліком цього підходу є його великомасштабна продуктивність, пов'язана з випадковим пошуком із малою кількістю рівнів градієнта «тепла». Тому щоб знайти пункт призначення, може знадобитися пройти всю мережу, що призведе до збільшеного використання пам'яті.

**Постановка завдання.** Метою роботи є проведення досліджень, визначення доцільності використання схеми маршрутизації Yggdrasil, як

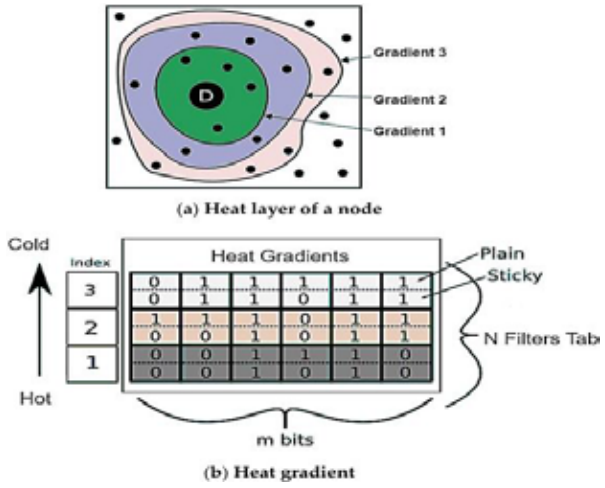


**Рис. 4. Градієнт інтенсивності «тепла» вузла D bloom-фільтрами [11]**

основи масштабної децентралізованої mesh мережі. Об'єктом дослідження є процес проєктування масштабної децентралізованої mesh мережі на основі схеми маршрутизації Yggdrasil. Предметом є моделі, методи та програмні засоби реалізації масштабної децентралізованої mesh мережі на основі Yggdrasil. У рамках дослідження потрібно визначити збільшення використання процесору на довгих гілках остовного дерева і аномалії трафіку, які вносяться блукаючим кореневим вузлом та запропонувати попереджувальний майнінг малих ключів для детермінованого розміщення кореневих вузлів, а також навести модель системи та обґрунтувати використання стандарту IEEE 802.11s із вимкненою маршрутизацією як базової технології канального рівня, отримати орієнтовну структуру масштабної децентралізованої mesh мережі та технічні вимоги до її вузлів. Експериментально продемонструвати, що Yggdrasil значно перевершує інші протоколи маршрутизації сітчастої мережі в обмеженні переходів, визначаючи параметри використання системних ресурсів та довести, що Yggdrasil є перспективним методом розгортання децентралізованих сітчастих мереж незалежних вузлів і альтернативою централізованим ієрархічним мережам.

**Виклад основного матеріалу.** При прийнятті рішення про доречність схеми маршрутизації важливим є масштабування, використання системних ресурсів і зростання часу конвергенції із розміром мережі. Випробування проводилися із використанням середовища meshnet-lab, враховуючи його гнучкість та можливості емуляції великомасштабних топологій за допомогою мережевого простору імен Linux (Linux network namespaces) [12] та автоматизації процесу за допомогою Python

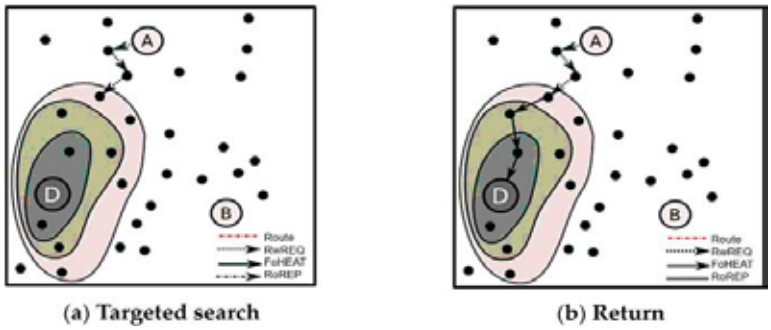


**Рис. 5. Пошук вузла D за його «теплом» [11]**

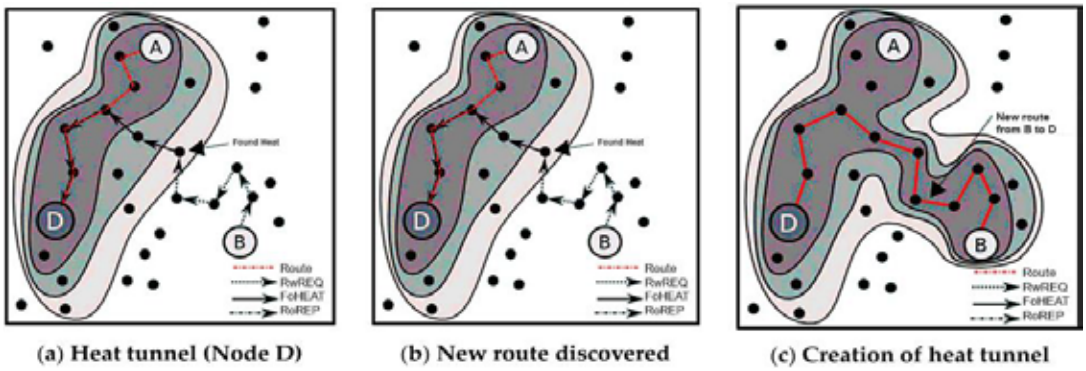


**Рис. 6. Повторне використання «теплового тунелю» вузлом B [11]**

[13]. Однак, слід зазначити, що хост-система, на якій проводилися тести, не може обробляти більше ніж 750 вузлів одночасно, оскільки після цього в системі закінчується пам'ять і починають використовуватися додаткові ресурси. Процедура є покроковою:

– підготовка топології згенерованими файлами JSON, включеними до meshnet-lab та обмеження пропускної здатності з'єднань до 100 Мбіт/с;

– запуск демонів маршрутизації на вузлах;

– запуск скрипту моніторингу системних ресурсів (кожну секунду здійснюється запис використання ЦП та резидентної пам'яті);

– очікування 30 сек., поки вузли знаходять один одного;

– протягом наступних 30 секунд надсилаються N унікальних випадкових пінгів із кількістю переходів 2 і дедлайном у 30 сек., де N є кількістю з'єднань;

– очікування 15 секунд;

– повторюються попередні кроки п'ять разів;

– запуск сценарію збору інформації про вузли (необхідна для визначення кореневого вузла під час обробки результатів);

– зупинка демонів, сценаріїв моніторингу ресурсів і очищення мережі;

– повторення процедури на більшій топології, доки не надійде менше 40% пінгів або не буде застосовано більше 750 вузлів.

Цей алгоритм, як більш реалістичний і найгірший сценарій відповідно, перевірено на топології випадкового дерева і лінійній топології. Крім того, з метою дослідження обсягу пам'яті, що використовується в конвергентній мережі, проведено тривалі за часом експерименти на найбільшій життєздатній топології, що визначається 100% надходженням пінгів.

Інший експеримент на випадковій деревоподібній топології з 50 вузлів без обмеження пропускної здатності показав різницю у використанні процесору для вузлів, які передають та приймають дані, а також вузлів, які не беруть участі в передачі. Трафік генерувався протягом 10 секунд між двома віддаленими вузлами за допомогою утиліти iperf3. Нарешті, проведення експерименту на лінійній топології з 750 вузлами із недосконалими повільними з'єднаннями показав стійкість Yggdrasil до таких умов. Застосовано наступні параметри з'єднань: пропускна здатність дорівнювала 10 Мбіт/с із затримкою 10±5 мс. Трафік генерувався між першим і останнім вузлом за допомогою стандартної утиліти ping. Останні два тести проводилися вручну. Процедуру бенчмаркінгу автоматизовано за допомогою модифікованої версії скрипту Python «benchmark1», включеного в meshnet-lab.

Результати порівняльного аналізу пам'яті за топологією випадкового дерева підтверджують припущення про те, що на використання пам'яті вузлом позитивно впливає кількість його з'єднань. Незважаючи на збільшення загального розміру мережі, використання пам'яті для абсолютної більшості вузлів залишається приблизно у тому ж діапазоні. Надходження пакетів для топології випадкового дерева залишається весь час на рівні 100%.

Оскільки в лінійній топології кожен вузол, крім першого та останнього, має лише два зв'язки, відтінок і розмір точок представляють положення вузлів у лінії. Довжина мережі позитивно впливає на використання пам'яті, що підтверджується піковим її використанням, наведеним у таблиці 1. Ця тенденція більш помітна на лінійних топологіях, хоча для мереж із понад 300 вузлів вона перестає бути настільки постійною.

Іншу динаміку для лінійної топології можна спостерігати при середньому використанні процесору. Коли вузли розташовані в лінію, використання процесору масштабується відповідно до її довжини і тим більше, чим ближче до середини лінії конкретний вузол. Крім того, кореневий вузол завжди ділить лінію на дві частини, для яких довжина та середні точки розглядаються

Таблиця 1

**Пікове використання пам'яті**

| Розмір мережі | Пікове використання пам'яті, КБ | | Розмір мережі | Пікове використання пам'яті, КБ | | Розмір мережі | Пікове використання пам'яті, КБ | |
|---|---|---|---|---|---|---|---|---|
| | Дерево | Лінія | | Дерево | Лінія | | Дерево | Лінія |
| 50 | 28640 | 26640 | 300 | 29300 | 30300 | 550 | 28664 | 30740 |
| 100 | 27980 | 27740 | 350 | 28812 | 31840 | 600 | 28624 | 32712 |
| 150 | 29036 | 29496 | 400 | 29004 | 30512 | 650 | 30092 | 32116 |
| 200 | 27432 | 28104 | 450 | 27736 | 30812 | 700 | 29620 | 30912 |
| 250 | 28660 | 29912 | 500 | 29428 | 31168 | 750 | 29564 | 30656 |

окремо. Мережі з 700 і 750 вузлами ніколи не досягають 100% надходження пакетів. Подальше дослідження виявило, що це відбувається через повернення утилітою ping «змішаного» результату середовищу meshnet-lab. У довгих топологіях потрібно чекати значно довше, поки прийде початковий пакет. Це призводить до того, що вузол надсилає другий пакет, що успішно надходить і тому результат становить «50%».

Маючи цю інформацію, ознакою конвергентної мережі можна вважати надходження пакетів вище 95% та, таким чином, дізнатися приблизний час конвергенції (табл. 2). Вимірювання записані з інтервалом принаймні 60 секунд, тому числа не слід приймати за номіналом. Однак їх можна використати для побудови графіка часу конвергенції, що передбачає лінійну залежність між довжиною мережі та часом конвергенції.

Експеримент проводився на mesh мережі розміром у 19 вузлів. Вузол n1 посилав пінг на віддалений вузол n20 великими пакетами, через що середовище визначало ці вузли, як вузли із значним об'ємом трафіку (рис. 7).

Звертаючи увагу на обсяги трафіку можна помітити, як після вузла n2 трафік нібито розподіляється між вузлами n3 і n5 знижуючи швидкість (з 750 Кбіт/с до 375 Кбіт/с). Подальше дослідження надало змогу визначити, що це є наслідком жадібної маршрутизації, через яку потоки трафіку n1 → n20 і n20 → n1 передаються різними шляхами. Коли трафік передається від n1 до n20, використовується короткий шлях n4 – n5, оскільки він дозволяє пропустити вузол n3. Таким же чином для трафіку, що передається від вузла n20 до n1, використовується з'єднання між вузлами n7 – n3, оскільки воно дозволяє пропустити вузол n5. Ця поведінка чітко простежується в топології, яка наведена на рисунку 8. Для вузлів використовуються кабельні з'єднання із односпрямованими потоками трафіку між вузлами n1 і n2. З'єднання поза деревом між вузлами n12 – n14 та n13 – n15 дозволяють трафіку пропускати два вузли і тому вважаються ефективними. Ця особливість використання різних шляхів для потоків трафіку між двома вузлами є корисною для збільшення пропускної здатності та зменшення заторів при передачі.

Ця особливість використання різних шляхів для потоків трафіку між двома вузлами є корисною для збільшення пропускної здатності та зменшення заторів при передачі.

**Висновки.** У роботі проведено дослідження масштабованості експериментальної схеми маршрутизації Yggdrasil щодо логіки, ліміту переходів, використання процесора та пам'яті. Експерименти з демоном маршрутизації проведено на різного розміру топологіях із використанням середовищ meshnet-lab і coreemu-lab. Виявлено певні особливості використання даної схеми маршру-

Таблиця 2

**Час конвергенції на лінійній топології**

| Довжина мережі | Час конвергенції, с | Довжина мережі | Час конвергенції, с | Довжина мережі | Час конвергенції, с |
|---|---|---|---|---|---|
| 50 | 60 | 300 | 206 | 550 | 462 |
| 100 | 65 | 350 | 220 | 600 | 401 |
| 150 | 129 | 400 | 224 | 650 | 326 |
| 200 | 135 | 450 | 252 | 700 | 502 |
| 250 | 190 | 500 | 310 | 750 | 361 |

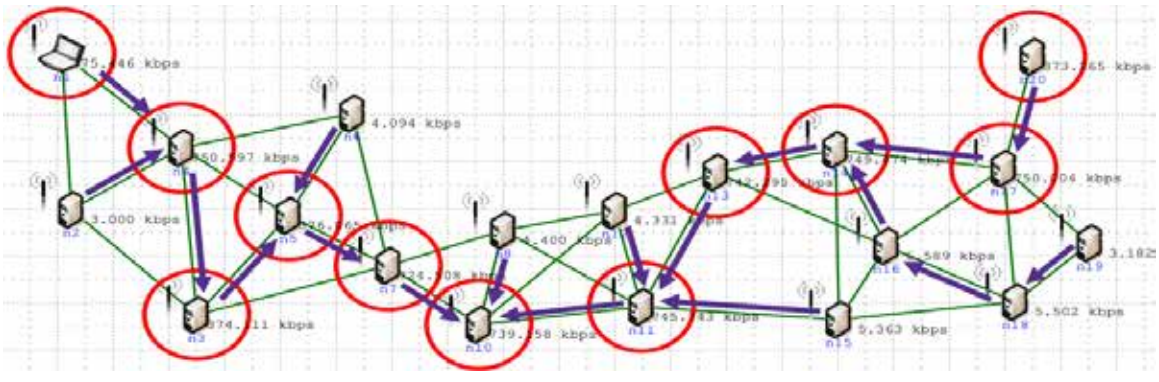

**Рис. 7. Бездротова mesh мережа на 19 вузлів, трафік між n1 та n20**

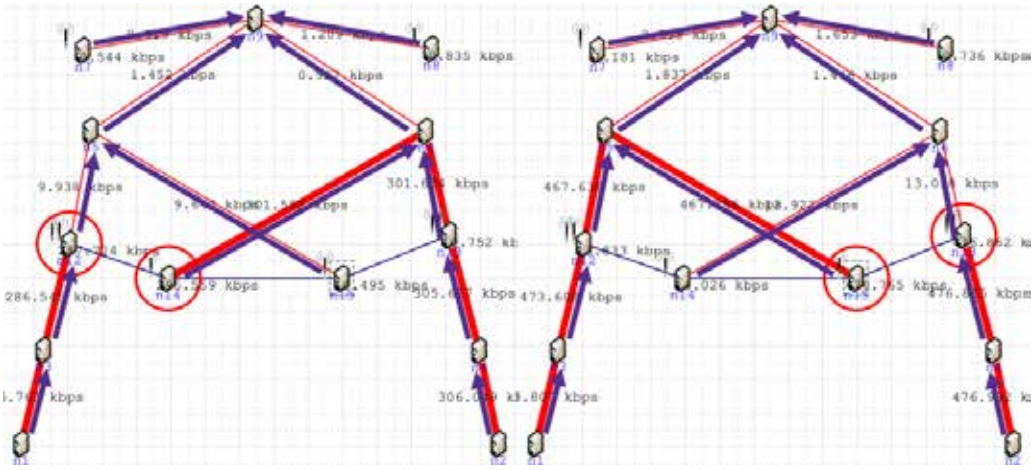

**Рис. 8. Мережа на 15 вузлів, потоки трафіку n1 → n2 та n2 → n1**

тизації як основи масштабної децентралізованої mesh мережі. Результати показали, що Yggdrasil значно перевершує інші протоколи mesh маршрутизації в обмеженні переходів, визначаючи параметри використання системних ресурсів. Експерименти демонструють типовий час конвергенції, а також тенденції використання пам'яті та процесора Yggdrasil і підтверджують їх масштабування з глибиною мережі, а не з її загальним розміром. Yggdrasil визнано перспективним методом розгортання масштабних децентралізованих сітчастих мереж.

**Список літератури:**


1. Greig J. A. Wireless Mesh Networks as Community Hubs: Analysis of Small-Scale Wireless Mesh Networks and Community-Centered Technology Training. Journal of Information Policy. 2018. 8(1). P. 232–266.
2. Рудьковський О.Р., Киричек Г.Г. Програмний комплекс з підтримки розподіленої взаємодії мережевих пристроїв та додатків. *Вчені записки ТНУ ім. В.І. Вернадського. Серія «Технічні науки».* 2021. Вип.32(71). № 2. С. 229–234.
3. Yggdrasil Network. URL: https://yggdrasil-network.github.io/ (дата звернення: 06.01.2025).
4. Brendel J., Cremers C., Jackson D., Zhao, M. The Provable Security of Ed25519: Theory and Practice. IEEE Symposium on SP. 2021. P. 1659-1676.
5. Khaledi M., Rovira-Sugranes A., Afghah F., Razi A. On greedy routing в dynamic uav networks. IEEE International Conference on Sensing, Communication and Networking (SECON Workshops), 2018. P. 1-5.
6. Messan PN, Krupinski S., Maurelli F., Vallicrosa G., Ridao, P. Evaluation of computer networking methods for interaction with remote robotic systems. IEEE AFRICON. 2021. P. 1-6.
7. Reich B. Wifi-Ad-hoc Mesh Networks for mobile Systems (Doctoral dissertation, Hochschule für Angewandte Wissenschaften Hamburg). 2024.
8. Киричек Г.Г., Щетінін М.О. Конфігурація серверів з використанням Ansible. Publishing House "Baltija Publishing". 2021. С. 15–17.
9. Wu D., Liebeherr J. A Low-Cost Low-Power LoRa Mesh Network for Large-Scale Environmental Sensing. IEEE Internet of Things Journal. 2023. P. 1.
10. Yamamoto R., Yamazaki T., Ohzahata S. VORTEX: Network-Driven Opportunistic Routing for Ad Hoc Networks. Sensors. 2023. 23(6). P. 2893.
11. Prabu S., Maheswari M., Jothi B., Banupriya J., Bindu G. Efficient Bloom Filter-Based Routing Protocol for Scalable Mobile Networks. Engineering Proceedings. 2023. 59(1). P. 75.
12. Schubert D., Jaeger B., Helm M. Network Emulation using Linux Network Namespaces. Network. 2019. 57.
13. Massaron L., Mueller J. P. Python for Data Science For Dummies. Hoboken, New Jersey, USA: John Wiley & Sons, Inc., 2015. 418 p.

**Kyrychek H.H., Pestov O.D., Tiahunova M.Yu. SCALE DECENTRALIZED MESH NETWORK**

*The paper investigates the scalability of the experimental Yggdrasil routing scheme in terms of logic, hop limit, CPU usage, and memory. Experiments with the routing daemon were conducted on different sized topologies using the meshnet-lab and coreemu-lab environments. Certain features of using this routing scheme as the basis of a large-scale decentralized mesh network have been identified. The purpose of the work is to conduct research, determine the feasibility of using the Yggdrasil routing scheme as the basis of a large-scale decentralized mesh network. The object of the research is the process of designing a large-scale decentralized mesh network based on the Yggdrasil routing scheme. The subject is models, methods and software tools for implementing a large-scale decentralized mesh network based on Yggdrasil. The research proposes new approaches to deploying a large-scale decentralized mesh network as an alternative to hierarchical ISP networks. The authors identify increased CPU usage on long branches of the spanning tree and traffic anomalies introduced by a wandering root node, and propose predictive mining of small keys for deterministic placement of root nodes. They also present a system model and justify the use of the IEEE 802.11s standard with routing disabled as the basic link-layer technology. They also obtain an approximate structure of a large-scale decentralized mesh network and technical requirements for its nodes. The results show that Yggdrasil significantly outperforms other mesh routing protocols such as OLSR and Babel in limiting hops by determining system resource usage parameters. The experiments demonstrate the typical convergence time, as well as the memory and CPU usage trends of Yggdrasil and confirm that they scale with network depth rather than with its overall size. Therefore, Yggdrasil is a promising method for deploying large-scale decentralized mesh networks of independent nodes and an alternative to centralized hierarchical networks.*

***Key words:*** *scalability, mesh network, Yggdrasil, routing, IEEE 802.11s.*