DOI: https://doi.org/10.36910/6775-2524-0560-2025-59-25
УДК 004.451
Мельник Василь Михайлович[1], к. ф-м. н, доцент
http://orcid.org/0000-0001-8282-6639
Багнюк Наталія Володимирівна [2], к.т.н., доцент
https://orcid.org/0000-0002-7120-5455
Ройко Олександр Юрійович[1], к.т.н.
https://orcid.org/0000-0001-8642-7707
Кізим Світлана Олександрівна[1], заступник директора з навчально-виробничої роботи
https://orcid.org/0009-0003-1205-2086
[1]Волинський фаховий коледж національного університету харчових технологій, Луцьк, Україна
[2] Луцький національний технічний університет, Луцьк, Україна

МЕРЕЖЕВІ СОКЕТИ НА ОСНОВІ ПСЕВДОНІМІВ З ВІДДАЛЕНОЮ ПЕРЕДАЧЕЮ АДРЕС

Мельник В. М., Багнюк Н. В., Ройко О. Ю., Кізим С. О. Мережеві сокети на основі псевдонімів з віддаленою передачею адрес. На сьогодні арсенал мережевих адрес версії IPv-4 уже вичерпаний, проте багато користувацьких вузлів все ще використовують їх для спілкування за допомогою традиційних протоколів Internet. Поряд з цим діє технологія передачі мережевих адрес (ПМА) великого масштабу (ПВМ), призначена для включення адрес версії IPv-4 в мережу, навіть у випадку повної їх занятості. Хоча механізм ПМА володіє перевагами, в ньому є і прояв негативних ефектів, який у конкретних умовах спричиняє недоступність серверних додатків до глобальної мережі, спричиняючи сповільнення чи блокування їх роботи. Для обходу подібних ситуацій в роботі запропоновано мережеву сокетну бібліотеку на основі псевдонімів, призначену для роботи серверних додатків та їх комунікації на базі автоматизованого розподілу каталогів (АРК). Бібліотека інтегрує в собі механізм обходу ПМА, реалізований на застосуванні програмних функцій та АРК. В роботі подано також опис елементів та умови використання розробленого програмного забезпечення для автоматизованого розподілу каталогів сокетної бібліотеки з метою мережевої комунікації додатків в операційній системі Linux.
Ключові слова: сокети-псевдоніми, віддалена передача адрес, автоматизований розподіл каталогів, UDP-протокол, комунікація додатків.

Melnyk V., Bahniuk N., Royko O., Kizym S. Network sockets based on aliases with remote address transfer. Today, the arsenal of IPv4 network addresses is exhausted, but many user nodes still use them to communicate using traditional Internet protocols. Along with this, there is a large-scale network address transfer (NAT) technology (LSN), designed to include IPv4 addresses in the network, even in case of their full occupancy. Although the LSN mechanism has advantages, it also has negative effects, which in specific conditions causes the inaccessibility of server applications to the global network, causing their work to slow down or block. To circumvent such situations, a network socket library based on aliases is proposed in the article, designed for the server applications operation and their communication based on automated directory distribution. The library integrates a NAT bypass mechanism using program functions and automated catalog distribution. The paper also describes the elements and conditions to use the developed software for automated catalogue distribution of socket library for the purpose of network applications communication in the Linux operating system.
Keywords: socket-aliases, remote address transfer, automated directory distribution, Protocol UDP, application communication.

Постановка наукової проблеми та аналіз останніх досліджень і публікацій На сьогодні весь арсенал мережевих адрес версії IPv4 уже повністю вичерпаний, проте перед багатьма користувачами все ще стоїть необхідність у їх використанні мережевими вузлами для спілкування за допомогою традиційних Internet-протоколів. Швидке переведення протоколів версії IPv4 на одному і тому ж мережевому рівні на версію IPv6 є довготривалою задачею. Також вже діє технологія передачі мережевих адрес (ПМА) великого масштабу (ПВМ), призначена для підтримки і включення в мережу адрес версії IPv4 [1, 2], навіть у випадку повного їх розподілу всесвітньою асоціацією IANA (Internet Assigned Numbers Authority). У випадку підключення локальних чи мереж малого бізнесу за допомогою ПВМ є призначена лише єдина IPv-4 адреса на зовнішній стороні мережевого інтерфейсу. Не дивлячись на те, що механізм ПМА володіє багатьма перевагами, в ньому має місце і прояв негативного ефекту, який у деяких конкретних умовах спричиняє недоступність серверних додатків до глобальної мережі, спричиняючи сповільнення чи блокування їх роботи та незручності для самих користувачів.

Для вирішення цієї задачі і можливості обходу подібних ситуацій в даній роботі запропоновано мережеву сокетну бібліотеку на основі псевдонімів для роботи серверних додатків з метою комунікації на базі автоматизованого розподілу каталогів (АРК). Така бібліотека повинна інтегрувати в собі механізм обходу ПМА, реалізований на застосуванні програмно розроблених функцій та АРК. Подібну мережеву бібліотеку було скомпоновано на основі визначених комунікаційних псевдонімів. Також подано опис елементів та умови використання розробленого програмного забезпечення для автоматизованого розподілу каталогів сокетної бібліотеки з метою мережевої комунікації в операційній системі Linux.

В основу механізму комунікації бралося припущення про те, що кожна програма управляє IP-адресою, а мережевий сокет збоку взаємодіючих операційних систем має можливість її використовувати. Відомо, що для більшості мережевих додатків використовуються імена хостів, кожен з яких пов'язаний з IP-адресою комп'ютера, а всяка IP-адреса в необробленому форматі не є сприятливою до читання для більшості мережевих користувачів. В роботі [3] запропонована архітектура мережевих сокетів, яка базується на іменах-псевдонімах, а замість IP-адреси хоста сокетною константою для них є AF_NIKNAME, що використовується для комунікації мережевих додатків та виклику сокетного інтерфейсу. Сокет, заснований на іменах, надає уніфікований інтерфейс для мережевих розробників, бажаючих створити і використовувати зв'язок із віддаленим хостом на основі «повних доменних імен» (ПДІ). Система доменних імен вже стала фактичним стандартом для надавання імен Інтернет-службам, проте в деяких випадках вона не зовсім підходить для використання кінцевими користувачами.

Переваги в роботі з іменами серверів чи доменів мають місце тоді, коли за допомогою доменних імен надаються асоційовані з ними імена хостів. Оцінити обсяг затрат на них буває навіть і складно. Якщо вони і вирішують проблеми використання системи доменних імен, то середовище комунікації для них, в основному, забезпечують Інтернет-провайдери. Кожен раз в ході їх підключення до Інтернет-мережі може проявитися деякий недолік, який полягає в призначенні провайдером IP-адреси, що відрізнятиметься від попередньо діючої. Це говорить про неможливість використання одних і тих же імен хостів постійно, так як система доменних імен абсолютно непридатна для частої реєстрації запитів на ресурси за порівняно короткий проміжок часу. З іншого боку отримання доменних імен кінцевими користувачами для запитів є важчим завданням, так як в системах доменних імен діє власний простір імен.

Інколи навпаки, система доменних імен слугує корисною інфраструктурою для підтримки підсистеми сервісу в мережі, якщо описані вище проблеми вже наперед вирішені. У випадку, коли динамічно будуються окремі мережеві віртуальні об'єднання, вузли віртуальних об'єднань забезпечуються мережевим підключенням через Інтернет-провайдери після віддаленої передачі мережевих адрес. Під час спілкування таких вузлів необхідно для кожного віртуального об'єднання застосовувати унікальне ім'я.

Метою в даній роботі є створення системи сервісу на використанні імен та служб розподілених каталогів в одноранговій мережі, використання якої в мережевих середовищах з ПМА є досить складним завданням. Для даної системи розроблене програмне забезпечення з використанням імен, що називається мережевим автоматизованим розподілом каталогів (МАРК) [4], яке містить механізм обходу трансферу мережевих адрес. Для організації іменної комунікації на базі Linux програмно створено сокетну бібліотеку, яка застосовується для визначення псевдоніма сокета в одноранговій мережі (ВПОС) і використовує функції МАРК під час встановлення з'єднання з віртуальними групами. Таким чином, розроблені програми є призначені для спілкування віртуальних груп за допомогою МАРК в одноранговій мережі, поєднаній зі звичайною ПМА-мережею.

Основна частина дослідження.

Модель однорангових мереж збоку архітектури не відрізняє систем, призначених для розподілених автономних мереж накладання збоку клієнта, від таких же систем зі сторони сервера. Кожен вузол однорангової мережі може відігравати ролі маршрутизаторів, клієнтів чи серверів, утворюючи єдину сукупну мережу, і не потрібно, щоб кожен з них виконував функції продуктивної робочої станції.

Всі однорангові мережі поділяються на структуровані та неструктуровані. Неструктуровані мережі використовуються у додатках однорангової комунікації більш ранніх версій. До них можна віднести такі системи, як Gnutella, Freenet та інші, які не застосовують ніяких адресних структур у своїх мережевих з'єднаннях, а обмін повідомленнями здійснюються за алгоритмом так званого наводнення, який на сьогодні є неефективним через непристосованість його до широких масштабних мереж, значним утрудненням мережевого трафіку та збільшенням кількості вузлів у одноранговій мережі.

З іншого боку, реалізація пошуку і надсилання запитів у структурованих однорангових мережах є дуже ефективною, оскільки вони на відміну від неструктурованих однорангових мереж використовують адресну структуризацію. Існує кілька алгоритмів однорангових структурованих мереж, таких як Kademlia [5], Ackord [6], Symphony [7] та інші, які здатні використовувати структуру даних в якості таблиці розподіленого хешу (ТРХ), яка володіє розподіленими та децентралізованими функціями. ТРХ подібна до хеш-таблиці, що поєднує ключ зі значенням, застосовуючи таку пару для організації подальшої взаємодії. Для цього всі вузли однорангових мереж повинні бути відповідно структуровані і представляти собою «попарні накопичення» для розподілених хешових таблиць. Це означає, що дані цих таблиць повинні розміщуватися на накопиченнях (buckets) [5], які в дійсності і є вузлами однорангових мереж. Це

досягається завдяки використанню кожним вузлом однакової хеш-функції. Внаслідок цього діючий простір даних ТРХ лінійно розширюється, так як розмір структурованої однорангової мережі також має тенденцію збільшуватися. Проте трафік повідомлень логарифмічно розширюється, так як отримання їх логарифмічно зростає за залежністю log(N), де N – це кількість вузлів у діючій системі.

Спочатку розглядалася можливість реалізації безкореневої сервісної системи на базі імен та залучення механізму ТРХ, яка могла б слугувати сервісним вузлом для реалізації авторозподілу директорій у віртуальних об'єднаннях через Інтернет, які використовують ТРХ-функції. Механізм автоматизованого розподілу каталогів в одноранговій мережі комунікації володіє базою даних, яка містить індекси ключових значень, використовуючи ТРХ на внутрішній стороні взаємодії. Це надає можливість доступу та отримання інформації про ресурси зберігання у всіх вузлах автоматизованого розподілу каталогів – можливість визначити їх псевдоніми, які ставляться у відповідність для реалізації з'єднання між активною мережевою інформацію та базою даних з індексами ключів.

Досить часто неможливо реалізувати побудову мережі так, щоб алгоритми Kademlia чи інші відомі із них [5-7], використовуючі ТРХ, включали б вузли трансферу мережевих адрес всередині власних мереж. Тому розглянемо додавання механізму обходу трансферу мережевих адрес в системі з автоматизованим розподілом каталогів.

Наглядна схема накладання механізму обходу передачі мережевих адрес на мережевий механізм з накопиченням каталогів для групового сервісу зображена на рис. 1. З її аналізу видно, що існує два типові види мереж накладання на таблиці розподіленого хешу. Перша мережа, під назвою «верхня», яка здійснює розподілений UDP-обхід через мережу з трансфером мережевих адрес (UDP Distributed Traversal through NAT – UDTN), а «нижня» – виконує сервіс для ТРХ. Верхня мережа UDTN у своєму складі містить лише вузли, які володіють глобальною IP-адресою, а нижня – є сервісною для ТРХ, яка включає в себе всі вузли, однорангової мережі з авторозподілом каталогів. Для кожного вузла в верхній мережі зберігаються IP-адреса та номер порта і асоціюються з ідентифікатором вузла з нижньої мережі. З іншого боку, нижня сервісна мережа забезпечує сервіс індексів ключових значень таблиць розподіленого хешу для користувачів з автоматизованим розподілом каталогів.

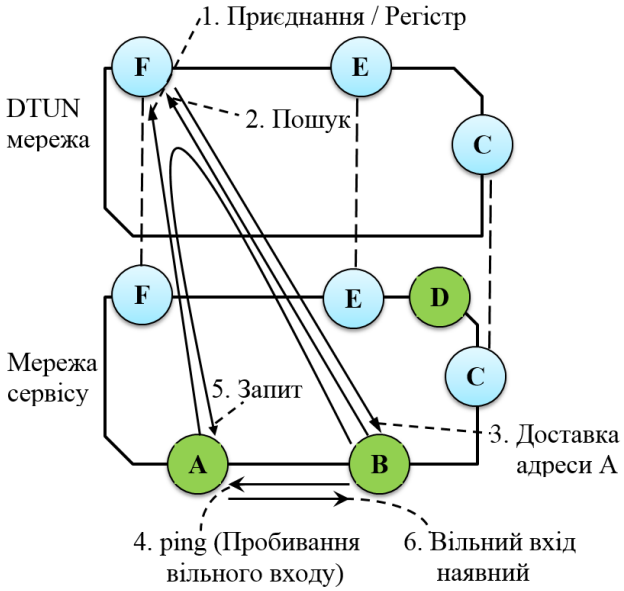


Рис. 1. Обхід трансферу мережевих адрес за допомогою накопичення каталогів для групового сервісу

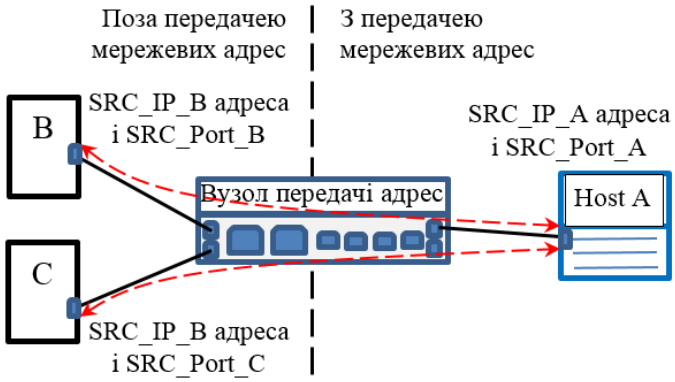
Вузли A, B і D (рис. 1) знаходяться всередині мережі з ПМА, а вузли C, E і F володіють глобальними IP-адресами. Обхід передачі мережевих адрес виконується в декілька кроків. Якщо вузол A приєднується до однорангової мережі з накладанням автоматизованого розподілу каталогів, то він реєструє свої IP-адресу та номер порта в мережі розподіленого UDP-обходу трансферу мережевих адрес, тобто в мережі UDTN. В той же момент часу тільки вона може бути однозначно визначена між підлеглими мережами трансферу адрес або залишатися взагалі не визначеною.

Усі вузли з авторозподілом каталогів в одноранговій мережі використовують в якості таблиць маршрутизації розподілений обхід UDP трансферу мережевих адрес та службову мережу для таблиці

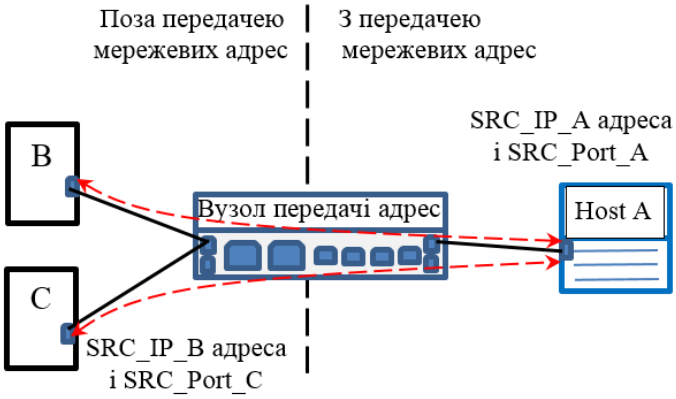
розподіленого хешу. Іншими словами, коли вузол B з'єднується з вузлом A, йому необхідно отримати інформацію про вузол A з верхньої мережі, що містить тільки вузли з глобальною IP-адресою UDTN. Для цього вузол B відокремлює і використовує позначений вузол F, який надає власну таблицю маршрутизації верхньої мережі UDTN для виконання такого обходу. Зазначений вузол F відноситься до верхньої мережі UDTN і надає відповідь на інформаційний запит, який включає в себе IP адресу та номер порта вузла A. Після цього вузол B надсилає спеціальне пінг-повідомлення, або повідомлення Kademlia [5], і «пробиває диру-зв'язок» для можливості доступу до цього вузла. Однак вузол B вимагає від вузла F пробивання отвору з'єднання до вузла A, що і виконується.

Після отримання вузлом A відповіді від вузла B на запит пробивання дірки доступу через проміжний вузол F, вузол A надсилає вузлу B спеціальне пінг-повідомлення. За допомогою цієї процедури на цей момент уже можуть бути встановленими та отриманими дані ключів-значень від таблиці розподіленого хешу (нижньої мережі), якщо мережа накладання, що здійснює розподілений UDP-обхід передачі мережевих адрес, включає визначену кількість вузлів всередині звичайної мережі з ПМА. Верхня мережа з автоматизованим розподілом каталогів для з'єднання між вузлами використовує протокол транспортного рівня UDP.

Існують два види передачі мережевих адрес (рис. 2), які відрізняються власними реалізаціями [8]. Використовуючи їх можна провести відслідковування процедури трансферу мережевих адрес через UDP протокол. Симетрична система передачі мережевих адрес доступна для виконання запитів з одної і тієї ж внутрішньої IP-адреси та порта до конкретно призначених IP-адреси та порта, які відображаються на одну і ту ж саму пару зовнішньої адреси IP та порта. Коли один і той же хост виконує надсилання пакета повідомлення з однаковими адресою джерела і значенням порта, призначеного різним кінцевим точкам в різних напрямках, то в таких випадках використовується інше відображення. Однак, лиш зовнішні хости, які отримали пакет, мають можливість надсилати UDP-пакет в зворотну сторону до внутрішнього хоста.



Механізм симетричної передачі мережевих



Механізм конічної передачі мережевих адрес

Рис. 2. Типи трансферу мережевих адрес

Конусна система трансферу мережевих адрес доступна для створення з'єднань всіх запитів з однієї внутрішньої IP-адреси та порта, асоційованого з однією зовнішньою IP-адресою та портом. В цьому

випадку будь-які пакети повідомлень від зовнішніх хостів доставляються до внутрішнього хоста і якщо точка отримання пакета, визначена IP-адресою та номера порта, вже відображена, то IP-адреса та номер порта джерела відправки зовсім не мають впливу на процес доставки. Процедура обходу передачі мережевих адрес, наведена вище, властива тільки для конусної мережі з трансфером адрес. Мережа з обходом передачі адрес протоколом UDP виконує послугу ретрансляції повідомлень і є подібною до TURN [9], призначеної для вузлів, підпорядкованих симетричній мережі з ПМА.

Мережа з автоматизованим розподілом каталогів для групової комунікації і обходом ПМА застосовує два типи процедур: пробивання дірок-доступів для зв'язку і передавання повідомлень або їх ретрансляцію. Останній тип доступний для обходу обох типів ПМА, і являється прикладкою до ресурсів. Таким чином, тип пробивання зв'язків є більш переважаючим, ніж ретрансляція повідомлень.

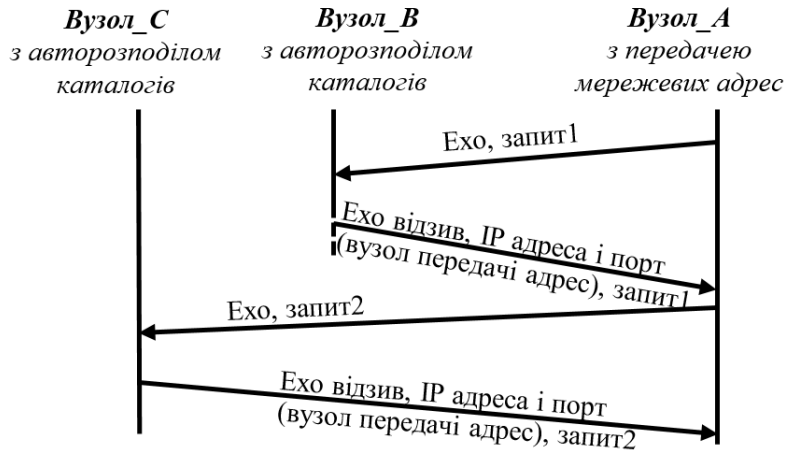


Рис. 3. Послідовність типів підтвердження

Послідовність утворення з'єднання для типів ПМА схематично зображена на рис. 3, з якого видно, що якщо вузол А, керований мережею ПМА, приєднується до мережі з автоматизованим розподілом каталогів для групового сервісу, то він відмежовується від мереж з конусною або симетричною ПМА, використовуючи комунікаційний зв'язок між двома вузлами, характерний для мережі з розподіленим UDP-обходом мережевої передачі адрес (UDTN). В цьому випадку вузол А надсилає ехо-повідомлення до вузла В в UDTN-мережі з запитом 1 (рис. 3), який містить ідентифікатор згенерованого випадкового числа. В свою чергу вузол В надсилає повідомлення-відповідь, що містить IP-адресу джерела і номер порта, на даний вузла А з ехо-повідомленням. Далі вузол А і вузол С обмінюються своїми аналогічними повідомленнями, як і раніше. Повідомлення з вузла А також передає два параметри власної інформації: зовнішню IP-адресу та номер зовнішнього порта зі сторони ПМА-мережі, попередньо надісланих з повідомлень-відповідей від вузлів В та С.

Таким чином, типи ПМА можуть бути обґрунтовані і встановлені цими надісланими повідомленнями. Якщо наявні обидва порти з однаковими номерами, то це говорить про дію конічної ПМА, а в іншому випадку – діє симетрична ПМА. З цього моменту мережа з розподіленим UDP-обходом передачі адрес має можливість для вузла А призначити процедуру для механізму обходу ПМА. Якщо діє конічний тип ПМА, то вузлу А надається пробивання дірок-з'єднань, а якщо це симетричний ПМА, то вузлу А присвоюється тип, асоційований з повідомленням.

3. Надійність передачі даних та присвоєння імені

Мережа з автоматизованим розподілом каталогів для обміну даними чи їх приєднання використовує протокол UDP, який не використовує попередньо встановленого з'єднання і забезпечує звичайний обмін, орієнтований на транзакції, не маючи власних функцій для ведення контролю та відновлення помилок. Однак в мережевих додатках потрібна надійність під час використання UDP, на основі якої протокол вищого рівня чи додаток могли б забезпечувати контроль помилок в процесі передачі даних в режимі реального часу. Тому з використанням протоколу UDP мережеві додатки важко підтримують процедуру об'єднання кількох повідомлень, особливо якщо розмір повідомлення перевищує

характерний розмір максимального блоку передачі (МОП). Наприклад, система доменних імен (СДІ), яка є надійною для UDP-обміну, накладає обмеження для запитів на повідомлення до 512 байтів.

Існує надійний протокол даних (НПД), орієнтований на зв'язок на транспортному рівні [10], який призначений для ефективної підтримки об'ємної передачі даних програмами моніторингу та управління на хостах, зокрема, їх завантаженню, налагодженню та дампінгу. Ці програми вимагають точної інформації та чітко визначених сеансів. Тому даний протокол забезпечує надійну передачу даних зі збереженням асоціації з'єднання. Оскільки НПД був створений давніше, то сьогодні діючі операційні системи не можуть реалізувати його у власний мережевий стек. Однак, на його основі розроблено мережевий *протокол для надійної передачі* пакетів *даних* (ПНПД) засобами *автоматизованого розподілу каталогів для групового сервісу* (АРКГС), поведінка якого досить подібна на звичайний надійний протокол передачі даних [8]. ПНПД утворює сеанс зв'язку після отримання шляху проходження через мережу з ПМА, який є також типовим до пробивання з'єднань або ретрансляції повідомлень.

Однак сеанс з'єднання утримує ПМА на подальшому шляху передавання пакетів аж до кінця з'єднання. Тому, використовуючи ТРХ, реалізується мережевий сервіс, який асоційовано працює з ПНПД і надає можливість надійного обміну повідомленнями, навіть які перевищують характерний розмір максимального блоку передачі. З використанням ПНПД можна легко ідентифікувати кожен сеанс комунікації в мережі з АРКГС. Якщо дана функція у операційній системі виконує сервіс процесів між собою як мережевий сокет, то вона є доступною не лише для ТРХ в мережі з АРКГС, але і для інших діючих мережевих програм, дозволяючи їм комунікувати між собою через незадіяну мережу з ПМА.

АРКГС утворює службу на основі імен, використовуючи ТРХ, яка уже згадувалася, для програм, комунікуючих на базі ПНПД. ТРХ також захоплює таблицю маршрутизації встановленої мережі накладання з чіткою асоціацією ідентифікатора кожного вузла та його IP-адреси, а АРКГС утримує таблицю маршрутизації мережі накладання, в якій містяться ідентифікатори та IP-адреси у парі для кожного з вузлів. Усі АРКГС-комунікації ідентифікуються однозначно і відрізняються між собою, а базові додатки ПНПД різняться номером ПНПД-порта. Для дискретизації кожен ідентифікатор насправді виглядає як хеш, що володіє 160-бітовою структурою і для звичайних мережевих користувачів являється кодовою інформацією, не придатною для читання. Крім того, в мережі з АРКГС можна присвоювати псевдонім із власним ідентифікатором, вкладаючи йому в якості даних асоційовану пару: значення – ключ в ТРХ, де в розумінні ключа фігуруватиме псевдонім, а значення міститиме ідентифікатор.

Рисунок 4 демонструє рівневу побудову АРКГС та схему іменування для двох комунікуючих додатків, заснованих на ПНПД. Вузол комунікації на стороні сервера очікує з'єднання від клієнта за допомогою утвореного доступу. Зі схеми видно, що комунікуючі додатки мають можливість з'єднуватися, використовуючи власні ідентифікатори і клієнтський вузол може взаємодіяти з вузлом сервера через визначення його псевдоніма, якщо все-таки серверному вузлу надано його.

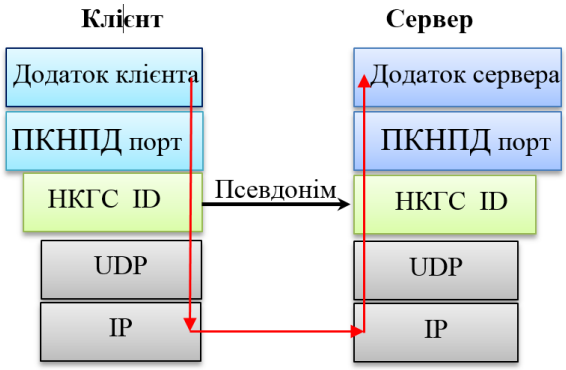


Рис. 4. Мережеві рівні, характерні для моделі накопичення каталогів для групового сервісу

Також послуги ТРХ володіють механізмом оновлення даних, а збережені дані, що передаються протягом тривалого часу, можуть зникати із сегментів. Порожні сегменти можуть використовуватися іншими вузлами для збереження нових пар даних типу ключ-значення, однак таке збереження їх залежить і від конкретних реалізацій ТРХ. Можна також заміняти у вільних сегментах дані типу ключ-значення, виконуючи їх перевстановлення. В роботі пропонується можливість реалізації з'єднання на узгодженому використанні двох механізмів: допустимих сегментів та розписаних даних ключ-значення в реалізації служби іменування АРКГС. Збережені дані типу ключ-значення вже не можна перезаписати (змінити) в

зайнятих сегментах зберігання. Управління цими сегментами виконується через ідентифікатори кожного із вузлів і тільки ті вузли, які на початку передають в сегменти дані типу ключ-значення, мають за правом доступу продовжувати використовувати їх, навіть і тоді, коли вузли покидають з'єднання з АРКГС.

АРКГС також може використовувати перевірку цифрових підписів для реалізації «досить хорошої або PGP-конфіденційності» [11] та підвищення надійності роботи з псевдонімами. Служба іменування для мережі з АРКГС залежить від ТРХ і має можливість використовувати псевдонім у всіх АРКГС-вузлах, якщо псевдонім ще ніким не задіяний. Для спроби отримання псевдоніма вузлами, служба видає спеціальний запит для вводу даних типу ключ-значення, який дотримується PGP-конфіденційності.

Рисунок 5 демонструє схему зв'язку за допомогою пари ключ-значення. Тут зображено три фізичні вузли, утворюючі групове віртуальне з'єднання у мережі накладання з АРКГС. Передбачається, що віртуальні вузли в даній групі виконують між собою обмін файлами з відкритими ключами, застосовуючи механізм PGP-конфіденційності. Якщо вузол А виконує розміщення пари даних типу ключ-значення для передавання у вузол В, то ці дані підписуються віртуальним вузлом А. Якщо віртуальний вузол С отримує дані, які намагається розмістити вузол А, то вузол С має можливість виконувати перевірку підпису, використовуючи свою довірчу базу даних покладання, де уже збережена інформація про вузол А і його відкритий ключ. Також застосування PGP-конфіденційності підвищує групову мережу на рівні довіри, а для підвищення надійності з'єднання між фактичним користувачем вузла та файлом відкритого PGP-ключа [12] можна використати цифровий підпис збоку користувача третьої сторони. Завдяки даному підходу можна з довірою використовувати з'єднання, яке не може бути самостійно підписаним до іншого файлу з відкритим ключем, що належить невідомому користувачу.

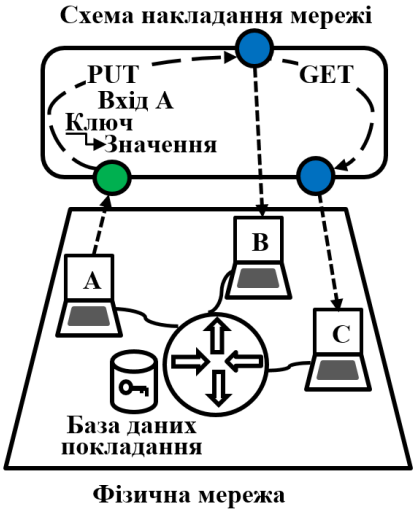


Рис. 5. Організація обміну даними за допомогою використання пари типу ключ-значення

4. Інтерфейс визначення псевдоніма для АРКГС

Для забезпечення сервісної системи, орієнтованої на псевдоніми, для віртуальної комунікуючої групи пропонується метод визначення псевдоніма для однорівневого сокета (ВПОС), реалізований програмним забезпеченням у вигляді бібліотечного пакета. Для реалізації АРКГС система включає в себе розроблену сокетну бібліотеку на основі псевдонімів, яка надає користувачам/додаткам відповідний інтерфейс функцій. Автоматизований розподіл каталогів для групового сервісу – це доменне програмне забезпечення, яке реалізує сервіс директорії, використовуючи ТРХ на базі Kademlia [13], і механізм обходу ПМА. Доменне програмне забезпечення здатне надавати незалежний сервіс для бази з даними типу ключ-значення, так як нею оперує програмний механізм визначення псевдоніма сокета в однорівневій мережі та самі користувачі. Для цього було створено інтерактивний інтерфейс командного рядка (ІКР) для АРКГС, який виконує перевірку достовірності дій процесу ВПОС та користувачів.

Опишемо командний інтерфейс, розроблений і наданий для АРКГС, який містить наступні функції: new – створення нового вузла, delete – видалення існуючого вузла, set_id – встановлення власного ідентифікатора АРКГС, get_id – отримання ідентифікатора АРКГС, join – приєднання до мережі з АРКГС, put – встановлення значення за відомим ключем, get – отримання значення за явним ключем, rdp_listen – створення бічного прослуховуючого сокета для ПНПД, rdp_connect – створення бічного сокета для підключення ПНПД і інші. В момент запуску АРКГС необхідно з'єднатися з мережею накладання та

виконати резервування псевдоніму для кожного вузла. Також в межах одного процесу на АРКГС можна створювати зразу кілька вузлів і коли вузол створюється, команда *create* вимагає від АРКГС-процесу його PID, який в подальшому буде використовуватися іншими розробленими командами для здійснення управління. В цьому випадку у межах АРКГС-процесу функція мультиплексування вузлів використовує багатокористувацьке середовище в якості хмарного комп'ютера чи приватного віртуального сервера.

Однак на практиці не обов'язковим для одного користувача створювати багато вузлів. До того ж, стоїть завдання приєднання до мережі накладання командою *join* від ІКР, яка для реалізації приєднання вимагає IP-адресу та номер порта одного із вузлів мережі з розподіленим обходом UDP трансферу мережевих адрес. В цей момент вузол АРКГС використовує автоматично визначений АРКГС-ідентифікатор з хешованим рядком про мережеву інформацію для кожної фізичної машини, тобто її IP-адресу та номер порта.

Керування доступними сегментами пам'яті АРКГС виконується також за допомогою ідентифікаторів. Якщо фізичні вузли ПК, які використовують АРКГС, переміщаються в інше місце мережевого розташування, то для них генеруються різні ідентифікатори. Відповідно, якщо такий вузол через короткі проміжки часу знову приєднується до мережі накладання, то він вже не зможе використовувати той самий (попередній) псевдонім, а функцією, яка надає права доступу до сегментів, збереженому сегменту присвоюється ідентифікатор попереднього вузла. Для встановлення статичних ідентифікаторів в реалізовано команду *set_id*, використовуючи звичайний інтерфейс командного рядка (ІКР). Дана команда використовує символьний рядок типу *string* для генерації хешу, асоційованого з ідентифікатором АРКГС, який використовується перед командою *join* для виконання приєднання. У ВПОС заданий символьний рядок *string* використовує хешоване значення файлу відкритого ключа для виконання умов PGP-конфіденційності та пароль користувача. Таким чином, це дає можливість постійно виконувати призначення псевдоніма.

Надалі, після приєднання вузол АРКГС повинен призначати псевдонім, асоційований із власними ідентифікаторами, які і необхідно отримати. Для цього розроблена команда *get_id*, значенням повернення якої є ідентифікатор вузла, і для повернутого ідентифікатора вона виконує прив'язку псевдоніма, використовуючи команду *put*, що передає відомі нам дані типу ключ-значення в ТРХ. Пара даних ключ-значення використовує інформацію про ресурси псевдоніма, а ВПОС надає бібліотечні інтерфейси, які покривають виконання згаданих дій та команд.

Для створення сокета псевдоніма використовується сокет ПНПД з визначеним псевдонімом від ТРХ. Для того, щоб забезпечити дію даної функції для процесів операційної системи, їм потрібно мати можливість комунікації між внутрішніми процесами. ВПОС в цей момент з'єднується з АРКГС через контакт, обумовлений ІКР, використовуючи сокет Unix Domain. Проте сокет, який використовує ІКР між ВПОС і АРКГС, не є доступний для передачі необхідного байтового потоку даних, тому що ІКР утворює базову лінію зв'язку. Отже, для надання сокета псевдоніма для інших процесів, необхідно ще раз з'єднатися з сокетом Unix domain.

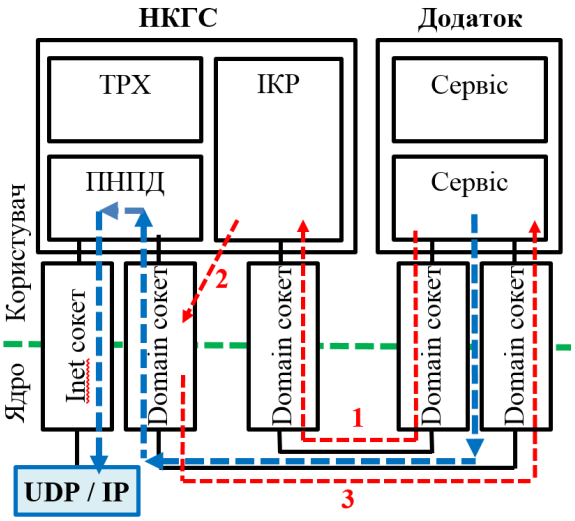


Рис. 6. Схема сокета псевдоніма

На рисунку 6 подана схема взаємодії ВПОС та АРКГС, на якій також наведено схему побудови шляху сокета псевдоніма в діючій операційній системі. В дослідженні розроблена прикладна програма, яка використовує сокет псевдоніма у ВПОС. Першим кроком додаток надсилає команду ІКР в момент, коли він починає відокремлювати сокет псевдоніму для використання за допомогою ВПОС. То ж якщо додаток являє собою серверне програмне забезпечення, він використовує команду *rdp_listen*, а якщо він є програмним забезпеченням з боку клієнта, то ним використовується команда *rdp_connect*. На даний момент ВПОС здійснює налаштування сокета UNIX Domain до прийому з'єднання від ПНПД. Далі ІКР отримує повідомлення і аналізує його. Якщо прийняте повідомлення є від *rdp_listen*, то АРКГС створює сокет прослуховування з ПНПД. У разі, коли повідомлення є від *rdp_connect*, то він створює сокет з'єднання з намаганням встановити сеанс зв'язку із серверною програмою, використовуючи ПНПД.

На завершення, АРКГС підключається до ВПОС через сокет Unix Domain. У випадку, коли створити сеанс не вдається, АРКГС повертає повідомлення про помилку через ІКР. Для виконання команд *rdp_listen* та *rdp_connect* потрібен АРКГС-ідентифікатор та номер ПНПД-порта. В результаті ВПОС буде створений і призначений для дії визначення псевдоніма перед тим, як зазначені команди виконають надсилання. Звідси слідує, що аргументом передавання ідентифікатора вузла є АРКГС-ідентифікатор, який також призначений для виконання проходження по мережі і представляє заздалегідь визначений псевдонім перед виконанням зазначених операцій.

5. Результати реалізації та обговорення

В роботі застосовано програмне забезпечення АРКГС, створене в засобами C++, та ВПОС – засобами C Objective. Вони використовують відомі для цих програмних середовищ бібліотеки *libevent*, *boost*, *libcrypto*, *libpgpme*, *libresolv*, а також *Framework Foundation*, *AppKit*, *Security* та інші засоби. Розроблене програмне забезпечення випробувано в операційній системі Linux Ubuntu, а інші характерні показники машинних вузлів були наступними: CPU – Xeon 2.4GHz, пам'ять – 8 GB, диск – SATA 160 GB, мережа – Gigabit Ethernet.

Бібліотека сокета псевдоніма подається як програмний об'єкт, клас якого об'єднує в собі наступні функції: *create()*, *sockfd()*, *sendto()*, *recvfrom()*, *listen()*, *connect()* та *close()*. Об'єкт класу за допомогою функції *create()* створює сокет псевдоніма. Якщо він використовується серверною програмою, то об'єкт використовує фкнуцію *listen()*, якій необхідно надати в якості аргументу лише номер ПНПД-порта. Якщо об'єкт використовується програмою клієнта, то він використовує функцію *connect()*, для якої потрібно надати два аргументи: АРКГС-ідентифікатор або псевдонім та номер ПНПД-порта. Встановлений сокет використовує методи *sendto()* і *recvfrom()*. Коли сеанс спілкування закінчується, його закриває класова функція *close()*. Використовуючи сокет UNIX-domain, ВПОС надає псевдонім, API якого подібний на звичайний сокет, однак для реалізації зв'язку він не використовує традиційний мережевий протокол. Створений об'єкт сокета UNIX-domain є звичайним компонентом в операційних системах стандарту POSIX, який може бути використаний для виводу дескриптора сокета визначеного екземпляра класу за допомогою функції *sockfd()*. Дескриптор сокета надається деяким системним викликам, призначеним для реалізації сокетних подій *select()*, *poll()*, *kevent()* і інших.

Описуючи ефективність роботи ВПОС в єдності з АРКГС, розглянемо рис. 7, який ілюструє масштабованість вузлів в накладній мережі. Горизонтальна вісь позначає кількість діючих вузлів, а вертикальна вісь відмічає середній час реагування. Пошук псевдоніма виконано 50 разів, а під час фіксувався час реагування і обчислене його середнє значення подавалося для побудови графіка. Вимірювання проведено в точках, позначених червоним кольором на графіку. Оцінка вимірювання роздільної здатності псевдоніма виконана в робочій мережі з АРКГС між 10-ма фізичними робочими станціями у тій же локальній мережі із характеристиками, наведеними вище. Число вузлів АРКГС варіювалося від 40 до 3000 на 10 робочих станціях шляхом балансування навантаження. Час вимірювання роздільної здатності псевдоніма в експерименті з використанням ВПОС становив різницю часу відправки запиту та часу отримання відповіді за допомогою функції *getSearchTime()*.

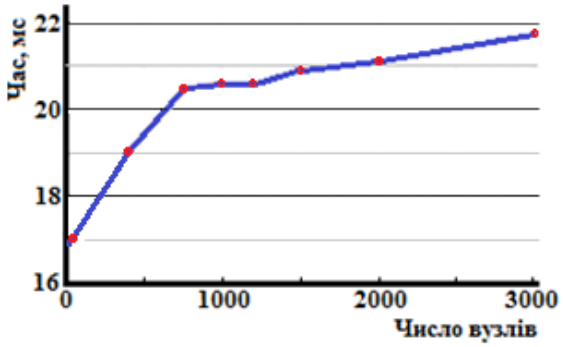


Рис. 7 – Залежність часу пошуку псевдоніма від кількості вузлів

За результатами проведених вимірювань часу пошуку для 40 вузлів він приблизно становить 16,9 мс (мікросекунд). Зі зростанням кількості вузлів час пошуку зростає не лінійно і, наприклад, для 3000 вузлів він досягає приблизно 21 мс, а розбіжність в кожній точці в ході виконання вимірювання становила в середньому ~0,5 мс. Порівняння отриманих усереднених значень говорить про те, що у діючій системі ВПОС та АРКГС час розбіжності є порівняно досить незначний. АРКГС також володіє масштабованістю, так як зростання часу відклику не було серйозно обумовлене збільшенням числа вузлів пошуку. Проте експериментальні умови в ході проведення вимірювань для випадків зростання кількості вузлів майже не відрізнялися від експериментальних вимірювань для невеликої кількості вузлів, де час відгуку становив 17 мс.

Надалі було б добре провести аналіз величини пропускної здатності чи оцінки продуктивності роботи такої системи в реальному мережевому середовищі. Для отримання даних реального аналізу ефективності роботи експериментальної тестової мережі необхідні широкомасштабні тестові стенди, ПМА-бокси та врахування складніших умов аналітичного оцінювання.

Висновки. Представлено розробку і застосування ВПОС, РКГС з розробленим програмним забезпеченням для мережевого сокета-псевдоніма з діючими функціями для реалізації механізму обходу ПМА та визначення конкретних псевдонімів з правом іменної комунікації. Так як реєстрація імен користувачів домашньої мережі для віртуальних об'єднань з боку Інтернет-провайдерів є складною через симплексність простору імен СДІ та недоступність його до іменування в кожному віртуальному комунікаційному об'єднанні, в роботі було розроблено службу іменування під назвою АРКГС, яка виконує сервіс розподілених каталогів в однорівневій мережі з обходом ПМА. За її допомогою реалізуються призначення та виявлення псевдонімів для кожного комунікуючого вузла. Якщо віртуальні об'єднання або вузли кінцевих хостів мають можливість використовувати службу іменування, то вона не може з'єднуватися з внутрішніми вузлами мереж з ПМА.

Запропонований в даній роботі протокол комунікації може бути реконструйований і перенесений для роботи у версію IPv6, однак таке дослідження вимагатиме значних зусиль та затрат. На відміну від провайдерських технологій LSN або CGN для розширення ПМА-мережі, в даній роботі було розроблено сокетну бібліотеку псевдонімів ВПОС і програмні функції її використання з метою здійснення комунікації з мережевими вузлами, навіть при їх знаходженні всередині мереж з ПМА. Коректну роботу розробленої системи було підтверджено на базі систем Linux Ubuntu з варіюванням кількості комунікуючих вузлів, а також механізму автентифікації пакетів повідомлень за допомогою цифрових підписів.

Список літератури

1. Yamagata I., Miyakawa S. Рішення для обміну адресами IPv4: плюси та мінуси. URL: http://tools.ietf.org/html/draft-nishitani-cgn-05. Інтернет драфт, серпень 2024. URL: https://surl.li/tcevwt (дата звернення: 15.09.2024).
2. Mikko D. Що таке NAT операторського рівня (CGN/CGNAT). Глосарій термінів. Інтернет драфт, 2024. URL: https://surl.li/rtjtik (дата звернення: 15.09.2024).
3. Tianlong Li, Tian Song, Yating YangAuthors Info & Claims. iStack: загальний іменований стек протоколів для іменованих мереж даних. NSDI'24: Матеріали 21-го симпозіуму USENIX з проектування та впровадження мережевих систем. Вид. січ., 2024. № статті 16. с. 267-280.
4. Sylverans I. Автоматизована каталогізація – бібліотечно-інформаційний менеджмент. 2024. URL: https://surl.lu/ktmnsi (дата звернення: 25.10.2024).
5. Tower K Одноранговa (P2P) архітектура. Останнє оновлення : 23 Трав, 2024. URL: https://surli.cc/jovcvv (дата звернення: 15.10.2024).
6. Ion Stoica, Robert Moris, David Carger & Frans Kaashock. Chord: Масштабована одноранговa служба пошуку для інтернет-додатків. Відеоресурс. 2024. Каліфорнійський університет. URL: https://surli.cc/jyjiwc (дата звернення: 25.10.2024).

7. Tower K. Хешування в розподілених системах. 2022. Електронний ресурс. URL: https://surl.li/kruvlu (дата звернення: 05.11.2024).
8. Prasaname R. Перехід до ключових моментів проходження протоколу датаграм користувача (UDP) через транслятори мережевих адрес. 2023. URL: https://surl.li/rroygb (дата звернення: 08.11.2024).
9. Rosenberg J. Перехід до ключових моментів проходження за допомогою реле навколо NAT. 2023. URL: https://surl.lu/jktupw (дата звернення: 05.11.2024).
10. . E. Krings. Протоколи потокової передачі для прямого мовлення. Все, що вам потрібно знати. Опубліковано 26 липня 2024 р. Електронний ресурс. URL: https://surl.li/dhwjhq (дата звернення: 05.11.2024).
11. Hana H. Підпис коду – налаштуйте SignServer і OpenPGP для підпису коду та пакетів. 2023. URL: https://surl.lu/bcrplu (дата звернення: 10.01.2025).
12. Brad Wyro. Шифрування, дешифрування та керування ключами на стороні сервера за допомогою OpenPGP. Блог MDaemon Technologies. 2024. URL: https://surl.li/comilc (дата звернення: 10.01.2025).
13. Christopher Skalnik. Розподілені хеш-таблиці з Kademlia. 2021. URL: https://surl.li/exitth (дата звернення: 10.01.2025).

References

1. Yamagata I., Miyakawa S. IPv4 Address Sharing Solutions: Pros and Cons. http://tools.ietf.org/html/draft-nishitani-cgn-05. Internet Draft, August 2024. URL: https://surl.li/tcevwt (date of access: 15.09.2024).
2. Mikko D. What is Carrier-grade NAT (CGN/CGNAT). Glossary of Terms. Internet Draft, 2024. URL: https://surl.li/rtjtik (date of access: 15.09.2024).
3. Tianlong Li, Tian Song, Yating YangAuthors Info & Claims. iStack: a general and stateful name-based protocol stack for named data networking. NSDI'24: Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation. Pub. Jan., 2024. Article No.: 16. P. 267-280.
4. Sylverans I. Automated cataloging – library and information management. 2024. URL: https://surl.lu/ktmnsi (date of access: 25.10.2024).
5. Tower K Peer-to-Peer (P2P) Architecture. Last Updated : 23 May, 2024. URL: https://surli.cc/jovcvv (date of access: 15.10.2024).
6. Ion Stoica, Robert Moris, David Carger & Frans Kaashock. Chord: A scalable peer-to-peer lookup service for internet applications. Videoresource. 2024. California university. URL: https://surli.cc/jyjiwc (date of access: 25.10.2024).
7. Tower K. Hashing in Distributed Systems. Dec., 2022. URL: https://surl.li/kruvlu (date of access: 05.11.2024).
8. Prasaname R. Jump to key moments of Traversal of User Datagram Protocol (UDP) Through Network Address Translators. Appril, 2023. URL: https://surl.li/rroygb (date of access: 05.11.2024).
9. Rosenberg J. Jump to key moments of Traversal Using Relays around NAT. Jun 2023. URL: https://surl.lu/jktupw (date of access: 08.11.2024).
10. E. Krings. Streaming Protocols for Live Broadcasting. Everything You Need to Know. Posted on July 26, 2024. URL: https://surl.li/dhwjhq (date of access: 05.11.2024).
11. Hana H. Code Signing – Set up SignServer and OpenPGP to Sign Code and Packages. Sept., 2023. URL: https://surl.lu/bcrplu (date of access: 10.03.2025).
12. Brad Wyro. Server-side Encryption, Decryption & Key Management with OpenPGP. MDaemon Technologies Blog. 2024. URL: https://surl.li/comilc (date of access: 10.01.2025).
13. Christopher Skalnik. Distributed Hash Tables with Kademlia. 2021. URL: https://surl.li/exitth (date of access: 10.01.2025).