ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Дослідження гіперболічної геометрії хмарних комп'ютерних мереж»

на здобуття освітнього ступеня магістра

зі спеціальності 122 Комп'ютерні науки
(код, найменування спеціальності)

освітньо-професійної програми Комп'ютерні науки
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

________________ Владислав ЯМКОВИЙ
(підпис) *(Ім'я, ПРІЗВИЩЕ здобувача)*

Виконав: Владислав ЯМКОВИЙ
здобувач вищої освіти
група КНДМ-61

Керівник: Віктор ВИШНІВСЬКИЙ
науковий ступінь, вчене звання д.т.н., професор

Рецензент: ________________
науковий ступінь, вчене звання ________________

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерних наук

Ступінь вищої освіти Магістр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедрою Комп'ютерних наук

______________ Віктор ВИШНІВСЬКИЙ

«___» ___________ 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Ямковому Владиславу Валерійовичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Дослідження гіперболічної геометрії хмарних комп'ютерних мереж

керівник кваліфікаційної роботи Віктор ВИШНІВСЬКИЙ, д.т.н., професор,

затверджений наказом Державного університету інформаційно-комунікаційних технологій від «19» жовтня 2023 р. № 145.

2. Строк подання кваліфікаційної роботи: «29» грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література з питань організації комп'ютерних мереж, посібники з аналітичної геометрії та теорії графів, науково-технічні роботи стосовно складних хмарних мереж.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

   1. Дослідження принципів побудови хмарних комп'ютерних мереж;
   2. Геометричний аналіз комп'ютерних мереж зі складною топологією;
   3. Розробка ефективної моделі маршрутизації даних в комп'ютерних мережах.

5. Перелік ілюстративного матеріалу: *презентація*.

   1. Розподіл функціональних зон у хмарних інформаційних системах;
   2. Загальна архітектура хмарних автоматизованих систем управління;
   3. Порівняння протоколів маршрутизації;

4. Зважені бінарні дерева та їх декомпозиція;
5. Гіперболічні вкладення.

6. Дата видачі завдання: «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапу кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури та джерел	19.10-05.11.23	
2	Вивчення теоретичних матеріалів для аналізу хмарних комп'ютерних мереж	05.11-12.11.23	
3	Дослідження геометричних репрезентацій хмарних комп'ютерних мереж	13.11-19.11.23	
4	Аналіз геометричної моделі складної хмарної мережі	20.11-25.11.23	
5	Застосування гіперболічної геометрії для мережевої маршрутизації	27.11-03.12.23	
6	Оформлення роботи. Вступ, висновки, реферат	04.12-10.12.23	
7	Розробка обов'язкових демонстраційних матеріалів	11.12-20.12.23	
8	Попередній захист роботи	21.12-29.12.23	

Здобувач вищої освіти ____________ Владислав ЯМКОВИЙ
(підпис) (ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи ____________ Віктор ВИШНІВСЬКИЙ
(підпис) (ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 40 стор., 11 рис., 1 табл., 13 формул, 13 джерел.

Наукове завдання – розробка геометричної моделі комп'ютерних мереж з метою вирішення задачі маршрутизації у складних хмарних мережах.

Мета роботи – підвищити ефективність маршрутизації мережевого трафіку.

Об'єкт дослідження – хмарні комп'ютерні мережі.

Предмет дослідження – технології маршрутизації трафіку в географічно розподілених комп'ютерних мережах.

Короткий зміст роботи:

Проведено аналіз комп'ютерних мереж, що виникають в географічно розподілених сценаріях хмарних веб-сервісів. Розглянуто типові моделі ієрархічних комп'ютерних мереж та способів їх утворення.

Розглянуто методи математичного моделювання мереж зі складною топологією. Визначений новий підхід для вирішення задачі маршрутизації з використанням геометричної маршрутизації.

Визначені переваги алгоритмів що використовують стисле геометричне моделювання мережевих графів для виконання евристичного алгоритму маршрутизації.

Застосування розробленого підходу до моделювання комп'ютерних мереж зі складною ієрархією дозволяє подальшу розробку нових алгоритмів маршрутизації.

КЛЮЧОВІ СЛОВА: ГЕОГРАФІЧНО РОЗПОДІЛЕНІ СЕРВІСИ, МАРШРУТИЗАЦІЯ ТРАФІКУ, КОМП'ЮТЕРНІ МЕРЕЖІ, МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, НЕЕВКЛІДОВА ГЕОМЕТРІЯ

ABSTRACT

Text part of the master's qualification work: 40 pages, 11 pictures, 1 table, 13 equations, 13 sources.

Scientific task – development of geometric model of a computer network with the purpose of solving routing problem in complex cloud networks.

Purpose of this work – increase effectiveness of network traffic routing.

Object of research – cloud computer networks.

Subject of research – network traffic routing technologies in geographically distributed computer networks.

Summary of the work:

The work performs analysis of geographically distributed scenarios of cloud web-services. Typical models of hierarchical computer networks and methods of their formation are reviewed.

Methods of mathematical modeling of networks with complex topology are reviewed. Proposed new approach to solving routing problems by using geometrical routing.

Identified advantages of algorithms, which use succinct geometrical modeling of network graphs to execute a heuristic routing algorithm.

Application of proposed modeling method for computer networks with complex hierarchy allows further development of improved routing algorithms.

KEYWORDS: GEOGRAPHICALLY DISTRIBUTED SERVICES, TRAFFIC ROUTING, COMPUTER NETWORKS, MATHEMATICAL MODELING, NON-EUCLIDEAN GEOMETRY

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

DNS – Domain Name System, система доменних імен.

SDN – Software Defined Network, програмно визначена мережа.

SLA – Service Level Agreement, угода про рівень послуг.

OCI – Open Container Initiative (назва організації).

OSI – The Open System Interconnection model, мережева модель.

VPS – Virtual Private Server, віртуальний приватний сервер.

АСУ – Автоматизована Система Управління.

ІС – Інформаційна система.

МОЗД – Модифіковані обмежено зважені дерева (запропонована структура даних).

ЦОД – Центр обробки даних (дата-центр).

ВСТУП

Хмарні обчислення – це парадигма побудови інформаційних систем, що в комерційному контексті дозволяє надавати послуги та продавати продукцію через Інтернет на міжнародному рівні, завдяки географічно розподіленим комп'ютерним мережам. Такий підхід делегує відповідальність щодо доступності та безпеки окремих точок входу, що фіксується юридично у вигляді договору про рівень послуг (SLA). Однак, це так само створює нові складнощі, зокрема в частині мережевої швидкодії, безпеки та координації.

Географічно розподілені комп'ютерні, що лежать в основі хмарних обчислень, значно залежать від ефективності маршрутизації мережевого трафіку – процесу автоматизованого пошуку оптимальних шляхів для доставки пакетів даних. В умовах сценаріїв із складними, гетерогенними мережами з динамічною топологією, із мінливим та невизначеним споживанням мережевих ресурсів, традиційні алгоритми маршрутизації, які спираються на критерії найкоротшого шляху чи мінімальної вартості, можуть не справлятись із багатовимірною та нелінійною природою проблеми пошуку ефективних маршрутів складних хмарних мереж.

В цій роботі обґрунтовується необхідність розробки геометричних алгоритмів маршрутизації, які могли б використовувати інформацію про геометрію масштабних комунікаційних мереж та користуватись перевагами неевклідової геометрії графів. Пропонується метод вкладення графів в гіперболічний простір, що гарантує успішність евристичного пошуку мережевих шляхів для пакетів даних.

В роботі показано існування вкладень графів у метричні простори, що можуть представлятись стисло та дозволяють ефективне порівняння дистанцій. Зокрема, в цій роботі розглядаються методи вкладення мережевих графів, що не допускають появи аномальних сегментів, які роблять евристичні алгоритми пошуку найкоротшого шляху неможливими, та що такі вкладення в неевклідові простори можуть бути представлені асимптотично оптимальною кількістю інформаційної ентропії. Пропонується стислий метод вкладення, що спирається на автократичні дерева обмеженого збалансування, вкладені в гіперболічний простір для вирішення недоліків класичних дистанційно-векторних алгоритмів.

Таким чином, ***об'єктом дослідження*** є процес моделювання хмарних комп'ютерних мереж.

Предмет дослідження – технології маршрутизації в географічно розподілених комп'ютерних мережах.

Мета дослідження – удосконалення методів репрезентації складних комп'ютерних мереж із застосуванням геометричної маршрутизації з метою покращення алгоритмів пошуку найкоротшого шляху, що не використовують глобальну інформацію про мережу, та підтримка SLA хмарних сервісів, що залучають декількох постачальників мережевих послуг.

Наукова новизна дослідження – запропоновано неевклідовий метод геометризації комп'ютерних мереж, що дозволяє просторово ефективну репрезентацію мережі з метою пошуку найкоротших шляхів для маршрутизації пакетів даних з невідомою глобальною топологією.

Запропонований підхід дозволяє побудову динамічних мереж, стійкіших до частої зміни маршрутів, що актуально як для веб-застосунків – зокрема таких, що використовують більше одного надавача послуг – так і в мобільних сценаріях, ad-hoc мережах спеціального призначення тощо.

1 ПРИНЦИПИ ПОБУДОВИ ХМАРНИХ КОМП'ЮТЕРНИХ МЕРЕЖ

1.1 Введення до хмарних обчислень та грід-систем

Хмарні обчислення засновуються на автоматизованому постачанні обчислювальних ресурсів через мережу Інтернет, з метою побудови гнучких, ефективних, безпечних та фінансово передбачуваних рішень для різноманітних потреб бізнесів та держави. Зінченко О.В., Іщеряков С.М., Прокопов С.В., Сєрих С.О., Василенко В.В. в їх спільному навчальному посібнику «Хмарні технології» [1] визначають хмару як великий пул легко використовуваних і легкодоступних віртуалізованих ресурсів, таких як апаратні комплекси, сервіси тощо, що можуть бути динамічно масштабовані в залежності від потреб користувачів.

Особливістю хмарних обчислень є можливість працювати з хмарними wпослугами з будь-якої точки і будь-якого пристрою, що має доступ в Інтернет, а також динамічно реагувати на зміни в потребах ринку та бізнес-завданнях підприємств. Така технологія розподіленої обробки даних, в яких потужності надаються як Інтернет-послуги, спираються на географічно розподілені фізичні ресурси – ЦОД, що об'єднані мережею Інтернет.

В наукових працях початку століття, часто можна зустріти поняття «грід-систем» (Grid-systems) в контексті, в якому сьогодні застосовуються «хмарні обчислення» – Шимчук Г.В., Маєвський О.В., Назаревич О.Б., Стадник М.А. в конспектах «Грід-системи та технології хмарних обчислень» роблять справедливе зауваження, що на практиці межі між цими (grid і cloud) типами обчислень досить розмиті [2] – але розглянемо уважніше ці поняття.

Грід-системи передбачають використання географічно розподілених ресурсів з метою рішення певної задачі. Це поняття з'явилось доволі давно в академічній літературі, та описує в загальному розподілену систему, що відрізняється від кластерних систем тим, що кожен вузол такої системи виконує окрему задачу, а не частину однієї спільної – хоча така задача може виконуватись із метою, спільною з іншими вузлами.

Хмарні системи з'явились, скоріше, як комерційне поняття – найчастіше, це системи для обслуговування користувачів та автоматизованого керування ресурсами, що побудовані на базі грід-систем із керованим публічним доступом. Такі автономні системи виділяють апаратні ресурси в необхідних значеннях по запиту користувачів та збирають телеметрію щодо використання ресурсів для автоматизованого та передбачуваного ціноутворення. Користувачі, в свою чергу – інші бізнеси – можуть будувати сервіси таким чином, щоб вони автоматично масштабувались в залежності від потреб, та навіть взаємодіяти між собою в рамках окремих хмар та їх специфічних функцій, мінімізуючи собівартість інфраструктури в довгостроковій перспективі.

Позитивною стороною залучення хмарних обчислень є вже зазначена у визначенні масштабованість, делегація підтримки апаратної та елементів програмної частини – як і всіх ризиків пов'язаних з їх підтримкою – на компанію, що спеціалізується на цьому. Такі системи також можуть підтримувати конфігурації з підвищеною стійкістю до відмов та балансуванням навантаження завдяки динамічній алокації задач доступним вузла, а також підтримувати реплікацію даних між багатьма локаціями. Рисунок 1.1 зображує загальний розподіл функціональних зон між споживачами та надавачами хмарних послуг.

Рисунок 1.1 – Розподіл функціональних зон у хмарних ІС.

Однак, хмарні сервіси мають так само подолати ряд складнощів, що властиві комп'ютерним мережам в цілому:

- Виникають безпекові складнощі, оскільки хмари загального користування можуть передбачати спільний доступ до чутливих даних та ресурсів між недовіреними або невідомими вузлами. Окрім того, такі ресурси можуть знаходитись в юрисдикції більш ніж однієї країни, що додає юридичних ризиків.
- Координація стає складнішою, оскільки хмари за природою гетерогенні та географічно розпорошені, тоді як контроль за доступністю вузлів у грід-системах, як правило, передбачає центрального координатора. Одним із проявів нетривіальності задачі полягає в проблемі балансування навантаження та його однорідності в мережі. Кожен вузол може мати різні спроможності та завантаженість.
- Маршрутизація постає як істотна проблема, оскільки з'являється потреба адресації вузлів у глобальній мережі, пошуку ефективних та швидких шляхів для доставки повідомлень та потоків даних між географічно розпорошеними вузлами, що при цьому є частиною цільної ІС. Враховуючи потенційні масштаби таких мереж, алгоритми маршрутизації повинні ефективно знаходити оптимальні шляхи для пакетів даних, спираючись на дані про мережеву завантаженість, затримку та фізичну дистанцію між вузлами.

Вищезазначені проблеми характерні для будь-яких достатньо великих ІС, однак фактор географічної розподіленості особливо сильно впливає на роботу хмарних сервісів. Тому питання маршрутизації та функціювання мережевого рівня хмарних систем стає найістотніших серед перелічених – та непереможним.

1.2 Хмарні ІС та їх архітектура

Хмарні обчислення маніфестують себе через відповідні хмарні ІС, що залучають ресурси, надані користувачеві у вигляді послуг. Справжні хмарні застосунки можуть містити велику кількість технічних деталей, що приховуються за договорами про певний рівень послуг із надавачами хмарних послуг. Всесвітньо відомі корпорації, такі як Amazon, Google, Microsoft, пропонують свої рішення для задач збереження та обробки даних, за посередництвом автоматизованих систем управління (АСУ) що дозволяють реалізувати самообслуговування споживачів через мережу Інтернет.

Розглядаючи безпосередньо хмарні ресурси, для організації обчислювальних ресурсів та сховища у вигляд придатний для автоматизованого ціноутворення та підтримки гарантій за контрактами зі споживачами, такі ресурси потребують можливостей вільно об'єднувати та розділяти дійсні програмно-апаратні ресурси надавачів послуг, щоб нарощувати й звільняти потужності відповідно до потреби за запитом АСУ. Облік спожитих ресурсів і оплата, як правило, йде за погодинним фактом використання.

Така технологічність та реактивність АСУ неможлива без технічного рішення, що дозволяє координацію ресурсів та розподіл задач відповідно до спроможності ЦОД. Відповідно, незалежно від форми надання хмарних послуг, має місце певний вигляд *віртуалізації*, що б дозволила також й ізолювати ресурси на логічному рівні, зберігаючи оптимальне розгортання задач на фізичні ресурси ЦОД.

Віртуалізацію можна розглядати як перехід від фізичного уявлення про обчислювальні ресурси та сховище до їх логічної репрезентації; приховування справжньої природи таких ресурсів та їх представлення абстрактне, квантифіковане та еластичне. Такий погляд дещо відрізняється від мислення про віртуалізацію як засіб компартменталізації, розділення, що є скоріше безпековою перспективою, хоча й також має місце. Іншими словами, віртуалізація це процес сегрегації форми та сутності, представлення та деталей.

Фундаментальним поняттям для віртуалізованих систем є *гіпервізор*, з яким взаємодіє користувацька АСУ для виділення та обліку ресурсів – програмне або мікропрограмне (вбудоване) забезпечення, що дозволяє об'єднувати та розподіляти ресурси між операційними системами та застосунками. Зазвичай, прийнято говорити про гіпервізори двох типів:

- Гіпервізори 1-го типу, що працюють на найнижчому (найближчому до апаратного забезпечення) рівні, як правило – нижче основної операційної системи комп'ютера. Прикладом цьому буде Microsoft Hyper-V, що працює в кожній копії Windows 11 та виконує безпекові функції, або ж рідний гіпервізор для операційних систем сімейства Linux – KVM. Такі гіпервізори можуть давати безпосередній доступ до апаратних ресурсів та керувати ними.

- Гіпервізори 2-го типу, що працюють як програмне забезпечення в рамках операційної системи та, як правило, *паравіртуалізують* апаратне забезпечення – емулюють його роботу на програмному рівні. Нині такі гіпервізори мають більш обмежене застосування.

На Рисунку 1.2 зображена різниця між типами гіпервізорів наглядно.

Рисунок 1.2 – Типи гіпервізорів.

Віртуалізації так само підлягають і мережеві ресурси. Віртуальні ресурси розміщуються на окремих фізичних серверах та для взаємодії потрібно організувати зв'язок між віртуальними машинами. Для цього використовується віртуальна комутація vSwitch що дозволяє об'єднувати як фізичні, так і віртуальні мережеві адаптери. Загальна схема використання віртуальної комутації наводиться як Рисунок 1.3.

Таким чином, застосовуючи автоматизацію навколо гіпервізорів, ЦОДи можуть істотно розширити асортимент своїх послуг та залучити цілі нові категорії користувачів. Адже об'єктивно, що невикористані апаратні ресурси – витрачений процесорний час, пуста оперативна пам'ять – це втрачені марно ресурси.

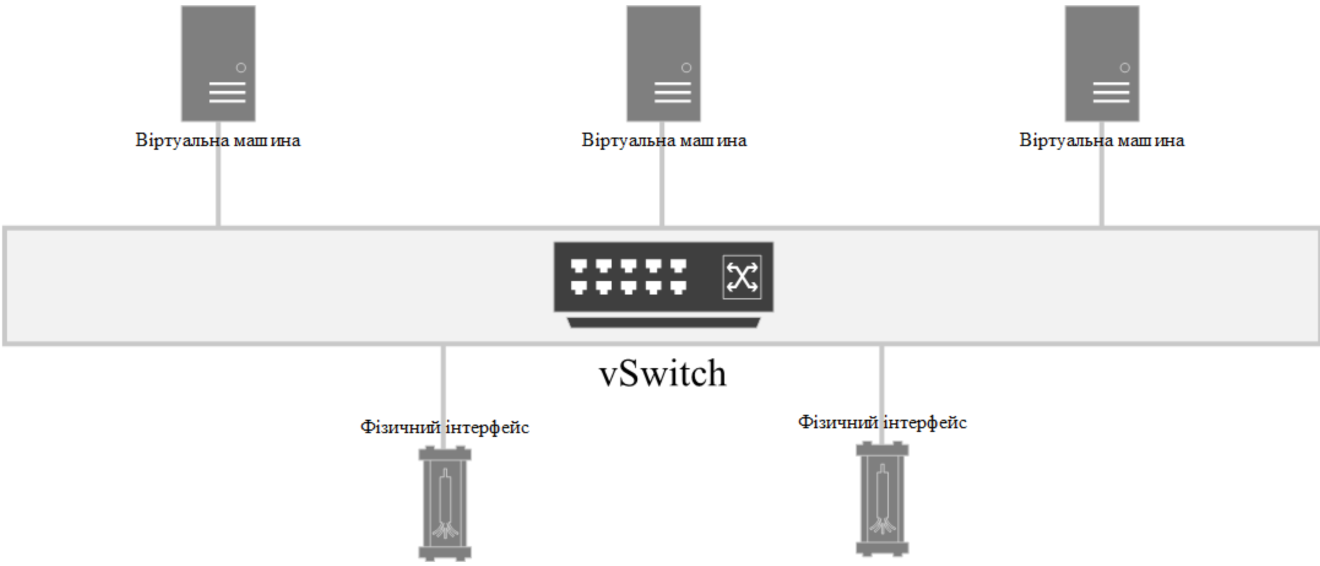


Рисунок 1.3 – Схема використання vSwitch.

На практиці ж, таке просте зображення може приховувати досить масштабну топологію. Представимо, що кожна віртуальна машина це VPS, в якому також подібна картина. Це може бути не лише тоді, коли використовується повноцінна віртуалізація, а й при залученні рішень для контейнеризації хмарних застосунків, таких як Docker чи інші реалізації специфікації OCI [3].

Варто зупинитись детальніше на зазначених технологіях. Сьогодні, спостерігається тенденція значного відходження від монолітної архітектури грід-систем із залученням класичних SDN рішень. Із розвитком програмного забезпечення та підходів до дизайну ІС, рішення на базі масштабних суцільних віртуальних машин накшталт OpenStack почали відходити в минуле, навіть для наукових застосувань. Інструменти накшталт Docker Swarm чи Kubernetes дозволяють розгортати цілі інфраструктури, описані декларативним чином за концепцією Infrastructure-as-Code (інфраструктура як код). Провідні компанії сфери, такі як Mirantis [4], що раніше спеціалізувались на підтримці рішень на базі OpenStack, зараз адаптують Kubernetes та навіть використовують їх в комбінації.

Підхід до опису інфраструктури ІС у вигляді декларативного контракту на манір програмного коду робить ІС відтворюваною, а тому – стійкою до втрати інформації та налаштувань на окремих вузлах. Однак, контейнеризація та ізоляція, що залучається для розгортання таких інфрастуктур, значно ускладнюють мережеву топологію. Рисунок 1.4 зображує загальну архітектуру хмарних сервісів та показує як контейнеризація застосунків додає цілий мережевий шар до типової мережі ЦОД.

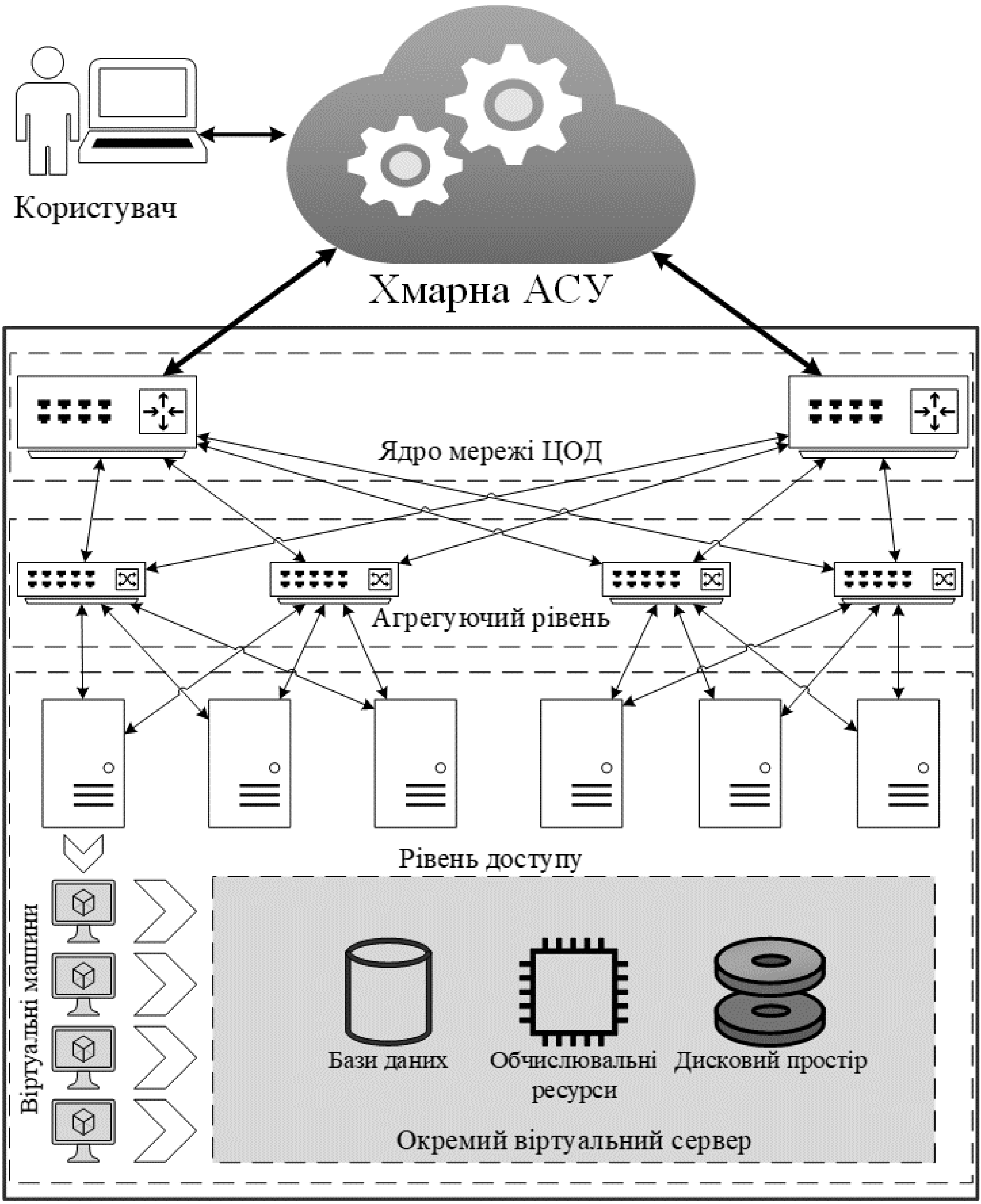


Рисунок 1.4 – Загальна архітектура хмарних сервісів.

В середовищах хмарних сервісів основою є географічно розподілені хмарні комп'ютерні мережі, що надає істотні переваги для збережності інформації та доступності послуг. До прикладу, система мережевих імен DNS таких великих надавачів послуг як CloudFlare [5] побудована на величезній мережі географічно розподілених ресурсів.

В свою чергу, надавачі хмарних послуг, як правило, надають послуги в рамках регіональних одиниць в різних країнах світу. Відповідальність за об'єднання хмарних ресурсів у єдиний інформаційний простір для належної роботи ІС, правильність використання наданих обчислювальних та мережевих послуг все ще покладається на користувача.

Тож яким чином можливо перейти від таких регіонально розпорошених ресурсів до чогось такого ж гармонічного, як CloudFlare? Швидке реагування та актуальність даних таких сервісів досягається завдяки інтелектуальній автоматизації навколо маршрутизації, що враховує географію клієнтів та кінцевих користувачів. Враховуючи широту мереж в таких сервісах, алгоритми маршрутизації повинні ефективно визначати оптимальні шляхи для пакетів даних між вузлами. Складнощі вирішення питання маршрутизації мають багато форм:

- **Обмеження по затримках та пропускній здатності.** Географічні дистанції неуникно збільшують затримку, а для ad-hoc рішень (мережі спеціального призначення) це може мати істотний характер, що унеможливлює щільну координацію для організації хмар чи розгортання грід-систем.
- **Динамічні мережеві умови.** В таких грід-системах умови вкрай динамічні. Обриви каналів, їх переповнення та флуктуація шаблонів використання, а також можливі будь-які складнощі з реальними маршрутами через проблеми у мережах-посередниках, вимагають адаптивних стратегій маршрутизації, що могли б реагувати на зміни умов.
- **Складнощі з масштабуванням.** По мірі того, як мережа розширюється, механізм маршрутизації повинен масштабуватись відповідно. Це значить потребу в підтримці швидко зростаючої кількості вузлів при підтримці ефективності та швидкодії з ростом ІС.

- **Оптимізація шляхів.** Знаходження найкоротших або найефективніших шляхів у мережі з тисячами вузлів та зв'язків може бути обчислювально дорогоб задачею. Викликом стає правильне врахування факторів, таких як надійність зв'язків, пріоритетність інформації, а також можливість при цьому проводити аналіз трафіку в реальному часі.

В середовищах, що залучають більше одного постачальника хмарних послуг, топологія природньо динамічна та географічно розпорошена. Такий динамізм виникає через змінні мережеві умови, включаючи ненадійність з'єднань з окремими постачальниками та флуктуації пропускної здатності та (або) затримки мережевих каналів. Це стає викликом для протоколів маршрутизації.

1.3 Огляд протоколів маршрутизації

Розглянемо такі два класи протоколів (алгоритмів) маршрутизації:

(1) **Протоколи з аналізом стану каналів**, які створюють маршрути на базі глобальних знань, таких як пропускна спроможність трафіку та інші метрики;

(2) **Дистанційно-векторні протоколи**, що будують маршрути відштовхуючись від певної метрики, що відображає дистанцію до місця призначення.

Як правило, дистанційно-векторні протоколи застосовуються в розподілених мережах з величезною кількістю вузлів для збереження доступності, швидкої адаптації до змін в мережі – зокрема, для взаємодії між операторами автономних систем мережі Інтернет. Протоколи з аналізом стану каналів, в свою чергу, мають сенс в мережах контрольованого розміру, оскільки передбачають, що кожен маршрутизуючий вузол зберігає глобальну модель мережі та незалежно обчислює найефективніші шляхи до кожного іншого вузла в мережі. Обґрунтуємо таке твердження порівнявши аналітичну вартість двох сімейств алгоритмів, що лежать в основі наведеної дихотомії протоколів.

Підручник Гніденка М.П., Вишнівського В.В., Сєриха С.О., Зінченко О.В., Прокопова С.В «Конвергентна мережна інфраструктура» [6] детальніше розглядає внутрішню маршрутизацію. Загальні відмінності протоколів наведені в Таблиці 1.1. Далі в роботі порівнюється аналітична ефективність алгоритмів.

Таблиця 1.1 – Порівняння класів протоколів маршрутизації.

	Клас маршрутизаційних протоколів	
Характеристика	**протоколи з аналізом стану каналів**	**дистанційно-векторні протоколи**
Повнота інформації	Будуються на алгоритмі Дейкстри та обраховують всі шляхи; потребують знання глобальної топології для обрахунку всіх шляхів.	Будуються на евристиці Беллмана-Форда; кожен маршрутизатор містить лише інформацію про сусідні вузли.
Стан	Кожен маршрутизатор повинен підтримувати повну топологію та використовувати таку інформацію для рішень щодо маршрутизації.	Вузли відповідають лише за моніторинг стану прилеглих ребер, передають інформацію про зміни оточуючим вузлам.
Витратність	Висока, оскільки такий протокол передбачає обробку значної кількості інформації про топологію та стан мережевих каналів.	Низька, оскільки евристичні алгоритми не вимагають знання глобальної топологічної конфігурації мережі.
Адаптивність	Добре для відносно статичних мереж, але передбачає значні витрати при зміні станів та не є просторово ефективними.	Мережа довше досягає стабільного стану, можуть відбуватись зациклення через неконсистентні дані.
Приклади протоколів	OSPF, IS-IS	RIP, BGP, Babel [7]

Розглянемо випадок протоколів з аналізом стану каналів, що шукають найкоротший шлях за відомим алгоритмом Дейкстри, що знаходить найкоротші шляхи до всіх вершин графу. Представимо мережу як зважений граф $N=(V,E)$, де V – множина вузлів (ком'ютерів, серверів, маршрутизаторів тощо), а E – множина впорядкованих пар вершин, що описує ребра графу – зв'язки мережі. Кожне ребро $e \in E$ також має невід'ємну вагу $w(e)$. Нехай $v_0 \in V$ це початковий вузол, а для будь-якої дистанції $v \in V$ визначимо $d(v)$ як найкоротшу дистанцію від обраного v_0 до v. **Алгоритм Дейкстри** знаходить $d(v)$ наступним чином:

1. Розпочнемо з $d(v_0) = 0$. Для кожної вершини $v \neq v_0$ будемо вважати $d(v) = \infty$.

2. Розглянемо кортеж (1.1):

$$V_0 = V \setminus \{v_0\} = (v_1, v_2, \dots) \tag{1.1}$$

Кожному $v \in V_0$ відповідає дистанція $d(v)$ – кортеж слід підтримувати впорядкованим за дистанцією.

3. Для кожного вузла v та для кожного його сусіда $u \in V_0$, порівняємо поточне значення $d(u)$ із сумою всіх дистанцій на шляху (1.2)

$$\sum_{i=1}^{n} d(v_i) \tag{1.2}$$

де n – кількість вузлів на шляху, враховуючи вагу ребра (v, u). Оновимо дистанцію $d(v)$, якщо останнє значення менше.

4. Зменшимо пріоритет v у черзі з пріоритетами, щоб відобразити нову дистанцію.

Алгоритм припиняє роботу, коли вершин у V_0 не залишилось, а остаточні значення $d(v)$ відповідають найкоротшим дистанціям від v_0 до v.

Визначимо обчислювальну складність алгоритму Дейкстри, для чого проаналізуємо серію операцій, які виконуються, та асимптотичний час кожного кроку алгоритма:

- Присвоєння початкових дистанцій: $O(|V|)$;
- Вставлення вершин в кортеж-чергу з пріоритетами: $O(|V| \log |V|)$, оскільки кожне вставлення має складність $O(\log |V|)$ а виконати потрібно n вставок.

- Оцінка граней: $O(|E| \log |V|)$, оскільки визначення пріоритетності сусідніх вершин має складність $O(\log |N|)$ а всього може бути щонайбільше |E| оцінок. Таким чином, можемо визначити загальну складність алгоритму Дейкстри (1.3)

$$O(n \log n + m \log n) = O\big((n + m) \log n\big) \tag{1.3}$$

Або ж, якщо врахувати, що кількість ребер з'єднаного графу завжди більша чи дорівнює кількості вузлів ($m \geq n$) маємо складність (1.4):

$$O(m \log n) \tag{1.4}$$

Однак вищезазначена складність має прихований недолік, що полягає в прихованій передумові: протоколи з аналізом стану каналів потребують знання про повну топологію мережі. Такі алгоритми не підходять для нещільних графів, оскільки «відвідують» всі вершини, навіть якщо в дійсності вони недоступні з джерела, та є непрактичними для динамічних графів, оскільки найкоротші шляхи обчислюються з нуля кожного разу, коли граф змінюється. Крім цього, алгоритм Дейкстри доказовно не працює у випадку, коли в мережі є ребра з від'ємною вагою.

Дистанційно-векторні протоколи, в свою чергу, потребують лише локальної інформації від безпосередніх сусідів та значно простіший за своєю сутністю. Фактично, вся робота надалі присвячена саме виконанню дистанційно-векторного пошуку оптимальних маршрутів у мережевих графах. Візьмемо за приклад алгоритм Беллмана-Форда.

1. Знову розпочнемо з $d(v_0) = 0$ та будемо вважати для кожної вершини $v \neq v_0$ що $d(v) = \infty$.

2. Для кожного v окрім v_0 проходимось по всім вершинам $(u, v) \in E$, оновлюємо $d(v)$ якщо (1.5)

$$d(u) + w(u, v) < d(v) \tag{1.5}$$

де w(u, v) – вага ребра (u, v).

3. Робимо зворотню циклічну перевірку. Після останньої ітерації над вершинами, потрібно зробити ще один прохід для кожного ребра в E. Якщо будь-яке значення $d(v)$ все ще може бути зменшене, сигналізувати про існування циклу з від'ємною вагою.

Тепер проаналізуємо обчислювальну складність алгоритму Беллмана-Форда. Врахуємо факт, що 2-й етап буде виконано $|E|$ раз за ітерацію, в загальному – |V| – 1 ітерацій. Таким чином, загальна часова складність складає (1.6)

$$O(|V| \cdot |E|) \tag{1.6}$$

Дійсно, це гірший аналітичний показник, ніж у (1.4). В той же час, просторова складність передбачується (1.7)

$$O(|V|) \tag{1.7}$$

Таким чином, можна заключити, що алгоритм Беллмана-Форда може проявляти вищу загальну обчислювальну складність порівняно з алгоритмом Дейкстри, але його спроможність виявляти цикли з від'ємними вагами, знижені вимоги до доступності актуальної інформації про мережеву топологію, та адаптивність в динамічних мережевих середовищах дозволяє йому бути ефективним вибором в окремих мережевих сценаріях.

Такий розподіл особливо релевантний в контексті евристичної теореми Брюера, що постулює неможливість забезпечити одночасно всі три бажані вимоги до розподілених ІС: узгодженість даних (consistency), доступність (availability) та стійкість до розділення (partition tolerance), де баланс між задоволенням зазначених вимог диктує вибір протоколу маршрутизації в дизайні та операціях хмарних ІС. В такому випадку, дистанційно-векторні протоколи пропонують підвищену доступність та стійкість, оскільки можуть продовжувати працювати з частковою інформацією, таким чином впевнюючись в продовженні функціювання мережі навіть в умовах неоднорідного стану. Це можна протиставити високій узгодженості протоколів з аналізом станів каналів, що іноді досягається за рахунок доступності та стійкості до розділення, особливо під регулярними та (або) значними змінами топології.

1.4 Постановка проблеми евристичної географічної маршрутизації

Як вже визначено, в умовах відсутності добре стандартизованого, попередньо обчисленого адресного простору та обмеженості ресурсів окремих вузлів, прийнято розглядати техніки *географічної маршрутизації*, що зменшує кількість інформації якою оперують окремі вузли. Присвоюючи кожному вузлу мережі координати спираючись на їх розташуванні у фізичному просторі, вузли потім використовують ці координати для пересилки повідомлень відповідно до шляху, що наближує до цільової координати та координат сусідів.

Найпростіші підходи до географічної маршрутизації є *евристичними*, в тому розумінні що вузли завжди перенаправляють пакети даних до сусіда який найближче до цілі. Евристична маршрутизація може бути вразлива до нескінченних циклів, коли існує вузол ближчий до цілі за всіх сусідів. Тоді як ефективність сучасних маршрутизаторів експериментально себе вже підтвердила, схоже, що академічна література має недостатнє теоретичне обґрунтування, що б ідентифікувало обставини в яких стратегії гарантовано працюють, а в яких – ні. Як факт, відомо, що евристичні алгоритми можуть провалювати пошук маршруту хоча б для певного відсотку мережевих конфігурацій.

Припустимо, нас цікавить наступне питання: нехай є неорієнтований граф $N=(V,E)$, *вкладенням* якого є функція над метричним просторі (X,d) така що $f: V \to X$ із наступною властивістю: для кожної окремої пари вершин $(u,v) \in E$ існує така вершина s що (1.8)

$$d\big(f(s),f(v)\big) < d(f(u),f(v)) \qquad (1.8)$$

Визначення (1.8) описує поняття *скорочуючого шляху* від s до v що існує для кожної пари вершин що нарисована в метричному просторі. Нажаль, не кожне вкладення може вважатись евристичним – не є незвичним для геометричного вкладення графу в евклідовий простір, навіть не перетинаючись, мати «пустоти» які унеможливлюють наївний пошук маршруту. Найелементарніший приклад графу, що демонструє описану проблему, наведено як Рисунок 1.5 – для маршрутизації повідомлень із C до A за гранями графу, такі повідомлення повинні пройти через вершину B, що знаходиться далі від пункту призначення.

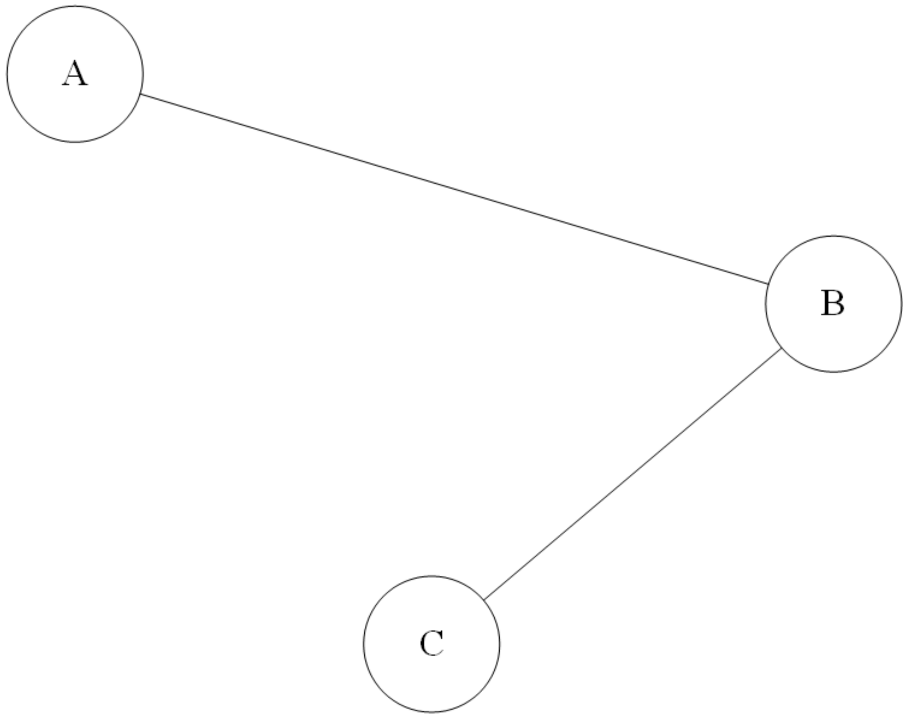


Рисунок 1.5 – Неевристичне вкладення графу з трьома вершинами.

В дійсності, мережа із зіркоподібною топологією (дерево із одним некрайнім вузлом) з достатньою кількістю гілок не може бути вкладений таким чином щоб усі його шляхи залишались евристичними. Число Ньютона (контактне число) для простору є максимальною кількістю одиничних кілець/сфер, дотичних до спільного одиничного кола; контактні числа в двох та трьох вимірах, відповідно, 6 та 12, та загальне контактне число обмежується степеневою функцією виміру. В зіркоподібній топології з кількістю гілок що перебільшує контактне число, якісь із двох гілок сформують кут менший за $\pi/3$ біля центрального вузла. Коли утворюється такий щільний кут, вкладення не може бути евристичним: якщо, як на Рисунку (1.5), кут щільний, при цьому $|BC| \leq |AB|$, тоді маршрут від C до A через B не може бути евристичним. Тоді, для того щоб знайти евристичне вкладення для довільного з'єднаного графу, буде недостатньо використовувати евклідовий простір обмеженої вимірності; потрібно розглядати неевклідові простори.

Одною з ранніх та вкрай цитованих робіт, що закладає засади геометричної маршрутизації, є робота Бозе [8] яка проводить екстракцію площинного підграфу, вкладає його та маршрутизує повідомлення складнішим евристичним алгоритмом ніж розглянутий в цій роботі: їх алгоритм проходить між регіонами відсіченими сегментом лінії між вершинами використовуючи суброздільний алгоритм проходження. Нажаль, запропонований алгоритм має прихований недолік, через що погано підходить для мотивації до застосування геометричної маршрутизації – зокрема, кожне розглянуте евристичне вкладення використовує координати вершин що вимагають $\Omega(|V| \log |V|)$ біт в гіршому випадку. Таким чином, такий евристичний підхід має такі ж просторові вимоги як і традиційні підходи із таблицею маршрутизації. Більше того, зазначена схема вкладення має погіршені вимоги до пропускної спроможності, оскільки повідомлення передбачені такою схемою також потребують зазначену кількість інформаційної ентропії для репрезентації, тоді як традиційні таблиці маршрутизації потребують лише $\Theta(\log|V|)$ біт.

Оскільки мета евристичних вкладень полягає в покращенні та спрощенні традиційних схем маршрутизації, якщо вкладення й мають бути корисними для рішення задачі геометричної маршрутизації, вони мають бути стислими – тобто представляти першини таким чином, щоб представлення використовували полілогарифмічну кількість біт та дозволяли ефективне порівняння дистанцій з використанням таких репрезентацій.

Далі в роботі пропонується метод геометричного аналізу хмарних комп’ютерних мереж, що використовує гіперболічне вкладення мережевого графу для стислого представлення мережі та гарантованого пошуку шляхів з використанням евристичного алгоритму.

2 ГЕОМЕТРИЧНИЙ АНАЛІЗ КОМП'ЮТЕРНИХ МЕРЕЖ ЗІ СКЛАДНОЮ ТОПОЛОГІЄЮ

2.1 Використаний метод моделювання складних мережевих топологій

Топологічні моделі комп'ютерних мереж можна вважати найвищим рівнем абстракції над формою та структурою мережевої архітектури. Розглянемо метод, який використовується в цій роботі для топологічного моделювання хмарних комп'ютерних мереж. Для моделювання мереж фундаментальним поняттям є граф, що містить вершини та ребра. Граф є моделлю топологічної структури мережі як фізичного, так і логічного рівня – підручник П. П. Воробієнко, Л. А. Нікітюка, П. І. Резніченко «Телекомунікаційні та інформаційні мережі» [9] відводить розділ застосуванню графів для моделювання топологічної структури мережі. В контексті цієї роботи, розглядається топологія логічних зв'язків мережевого рівня OSI – сформулюємо проблему маршрутизації. Скористаємось математичною мовою теорії графів – для ознайомлення, підручник Балоги С. І. «Дискретна математика» [10] містить досить стисле введення в необхідну теорію.

Розглянемо загальний випадок неорієнтованого графу N (2.1) з довільною кількістю вершин та ребер, що були б достатні для формування відображення з'єднаної мережі:

$$N(V,E): V(n), E(m), n \in (1;\infty), m \in [n-1; \tfrac{n^2-n}{2}] \qquad (2.1)$$

Істотна умова для вибору m полягає в тому, щоб граф містив достатньо ребер, щоб з'єднати всі вузли – задача маршрутизації не має рішення для вузлів, що не мають жодного ребра, яке б з'єднало їх із кожним іншим вузлом, безпосередньо або опосередковано іншими вузлами. Нижня межа n-1 дозволяє розглядати мережеві топології незамкнені, такі як зірка або лінія. Верхня межа m випливає з леми про рукостискання – у кожній мережі буде парне число непарних вузлів, а кожен вузол гіпотетично може мати ребро з кожним із n-1 інших вузлів у мережі.

Вкладенням графа *N(V,E)* називається ін'єктивне відображення N для кожної вершини V на площину невід'ємних чисел, де така функція відображає кожну грань на сегмент лінії, що з'єднує точки; вкладення площинне, якщо такі сегменти перетинаються тільки і тільки в точках, що відображають вузли *V*. Відповідно, маючи вкладення графу можемо розглядати дистанцію між відображеними вузлами, що визначається відповідно до властивості, наведеної в (1.8) попереднього розділу. Розглядаючи евристичну маршрутизацію на базі такої скорочуючої дистанції, маємо справу з наступним твердженням:

Твердження 1. Будь-який зв'язаний граф може бути вкладений в площину таким чином, що між будь-якими двома вершинами n_1 та n_2 існує скорочуючий шлях що починається з n_1 та закінчується n_2.

Під евристичними алгоритмами маршрутизації, в цій роботі надалі розглядаються алгоритми маршрутизації, які для вкладеного в площину графа та обраних вузлів v_0 та *v*, знаходять шлях $v_1, v_2, \ldots$ наступним чином:

```
Нехай i := 0;
допоки vi ≠ v:
  якщо існує такий k прилеглий до vi, що d(vi,v)<d(k,v):
    нехай i=i+1, vi=k;
  інакше: кінець.
```

Припустимо, що вкладення можна назвати *евристичним*, якщо цей алгоритм завжди завершується на $v_i = v$. Для тощо, щоб перефразувати цю властивість у геометричну форму, розглянемо вкладення графу, вузол *v* такого вкладення, а також всі сусідні вузли $v_1, v_2, \ldots$. Сотою *v* буде множина всіх точок на площині, що ближчі до v ніж до будь-якого із сусідів (див. Рисунок 2.1). Сота вузла не обов'язково є обмеженою.

Теорема 1. Наступні твердження щодо вкладення графа є тотожними:

(1) Вкладення евристичне.

(2) Між будь-якими двома вершинами існує скорочуючий шлях (Твердження 1).

(3) Для будь-яких двох вузлів v_0 та v ($v_0 \neq v$) існує сусід k такий що $d(k, v) < d(v_0, v)$.

(4) Сота будь-якого вузла *v* не може містити ніяких вузлів окрім *v*.

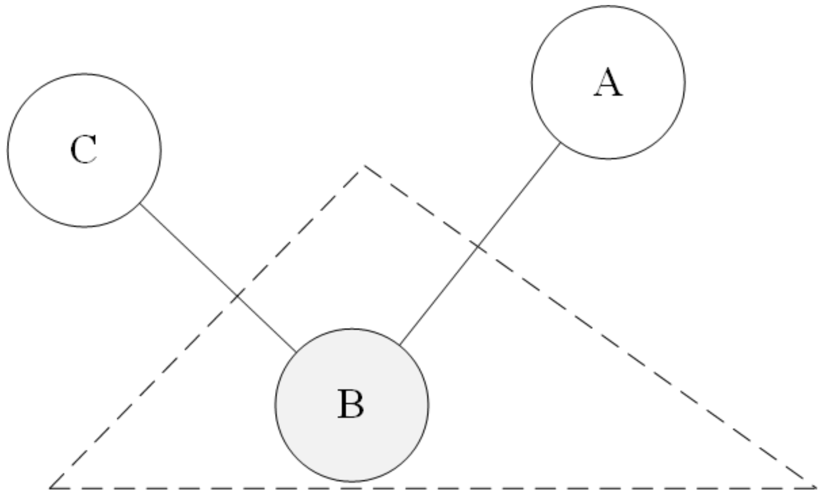


Рисунок 2.1 – Сота вузла включає всі точки наближені до вузла.

Доведення. Пов’яжемо вже відомі факти для встановлення тотожності тверджень Теореми 1:

1. (1 ⇒ 2) Евристичний алгоритм дає шлях для будь-яких v_0 та v.
2. (2 ⇒ 3) Перший пошук скорочуючого шляху дає такого сусіда.
3. (3 ⇒ 1) Евристичний прохід із v_0 до v має завершитись в певній точці v', оскільки дистанція до v зменшується з кожним кроком. Якщо $v' \neq v$, тоді v' не має сусідів ближчих до v, що суперечить (3).
4. (3 ⇒ 4) Якщо сота деякого v містить вузол k, тоді k не є сусідом v та будь-який сусід для v має дистанцію більшу до k ніж до v, що суперечить (3).
5. (4 ⇒ 3) Якщо (3) не справджується, тоді ближче v до v_0 ніж до будь-якого із сусідів v_0, тобто v знаходиться в соті v_0. ■

Як зазначалось у попередньому розділі, нажаль, не кожне вкладення є придатним для евристичного пошуку оптимальних шляхів. Сформуємо та доведемо такі твердження – одночасно цим самим визначимо цікаву властивість, що є достатньою, щоб вважати вкладення евристичним.

Твердження 2. Якщо опукле вкладення не має кутів що більше чи рівні $2\pi/3$ між послідовними ребрами в рамках регіону простору, висіченого гранями площинного графу, що не перетинаються, тоді таке вкладення є евристичним.

Доведення. Доведемо вищезазначену теорему через зворотнє твердження. Будемо вважати, що вкладення не є евристичним; будемо розглядати кут більший за $2\pi/3$. В такому випадку, існують два вузли a та b такі, що жоден сусід a не може бути ближчим до b ніж сам вузол a.

Розглянемо послідовних сусідів для вузлів a, c та d таких, що формують регіон b, сформований кутом cad; відповідно до вищезазначеної передумови, c та d не лежать у внутрішньому колі з центром b та радіусом $|bc|$ (див. Рисунок 2.2). Розглянемо також вузол b' в трикутнику cbd що був би найближчим до сторони cd (це може бути й сам вузол b). Внутрішня частина чотирикутника $acb'd$ не містить вершину, оскільки жоден вузол не лежить в трикутнику cad через його опуклість, та такий вузол лежав би ближче до cd ніж до b'. Тому, існує кут сформований послідовними гранями, що мають спільний вузол в b' такий, що був би щонайменше рівний $cb'd$ – що, в свою чергу, щонайменше такого ж розміру, як і dbc.

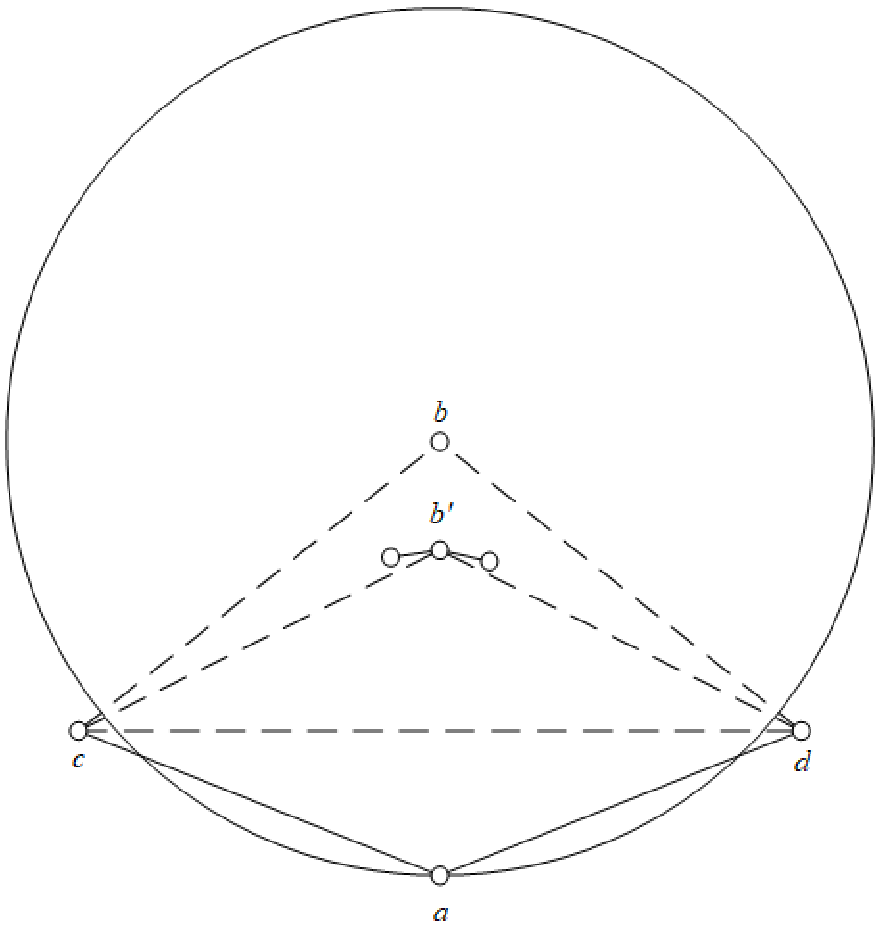


Рисунок 2.2 – Неевристичне вкладення з великим кутом.

Заключимо, що існує такий кут у вкладенні, що був би щонайменше такий же великий, як і кут ctd. Однак, c та d не знаходяться всередині кола, з чого слідує (2.2).

$$2 \cdot cad + cbd \geq 2\pi \tag{2.2}$$

В свою чергу, наслідком (2.2) є те, що один із двох кутів (cad або $cb'd$) не може бути меншим за $2\pi/3$. ■

Варто зазначити, що таке твердження має обмежену застосовність, оскільки будь-який регіон із 6 чи більше сторін повинен містити кут $2\pi/3$. В цілому, навіть триангульовані графи – графи, отримані шляхом заповнення простору певними паттернами – можуть вимагати кути більші за $2\pi/3$. Тобто залишаються графи, вкладення яких не підлягає евристичному аналізу.

На практиці, коли евристична маршрутизація неможлива, існуючі алгоритми маршрутизації застосовують *регіональну маршрутизацію*: вони обходять регіон (чи те, що виглядає з точки зору реалізації протоколу як регіон) допоки евристичний алгоритм не відновлює свою функцію. Регіональна маршрутизація та відповідно названий прикладний протокол FACE розглядається також у раніше згаданій роботі Бозе [8] та показує, що таке рішення може ставати доволі складним, або не гарантованим. Розглянемо тепер особливий випадок, коли всі регіони є трикутними.

Дано площинний граф із опуклим вкладенням такий, що кожен вузол зберігає власну позицію, позицію сусіда та інформацію про два регіони що мають спільне ребро cd. В процесі маршрутизації, відбувається переміщення в бік кінцевої точки v евристичним методом. Якщо досягнуто вузла $k \neq v$ без сусіда ближчого до v, розпочинається фаза *регіональної маршрутизації*, що полягає в ідентифікації двох послідовних (за годинниковою стрілкою) сусідів i, j таких, що ikj створює кут, та регіон, в який потрапляє промінь kv (регіон, що припадає на дві грані ki та kj). Потім із k продовжуємо навігацію по регіону, наприклад, по часовій стрілці, поки одна з наступних не відбудеться одна з двох подій:

1. Досягається вершина b' така, що $d(b', v) < d(b, v)$, де під b мається на увазі вузол в якому регіональна маршрутизація розпочинається; в такому випадку, евристична маршрутизація продовжується.

2. Досягається вершина *r*, грань *rt* якої в регіоні ближче до *v* ніж до *d(r, v)* та *d(t, v)*. Регіональна маршрутизація продовжується на іншому регіоні, що проходить за *rt*.

Щоб побачити, що відбудеться одна з вищенаведених подій, пригадаємо, що дистанція із *v* до точок опуклого багатокутника (регіону), не включаючи *v* та його нутрощі, досягається або у вершині, або на границі регіону – див. Рисунок 2.3.

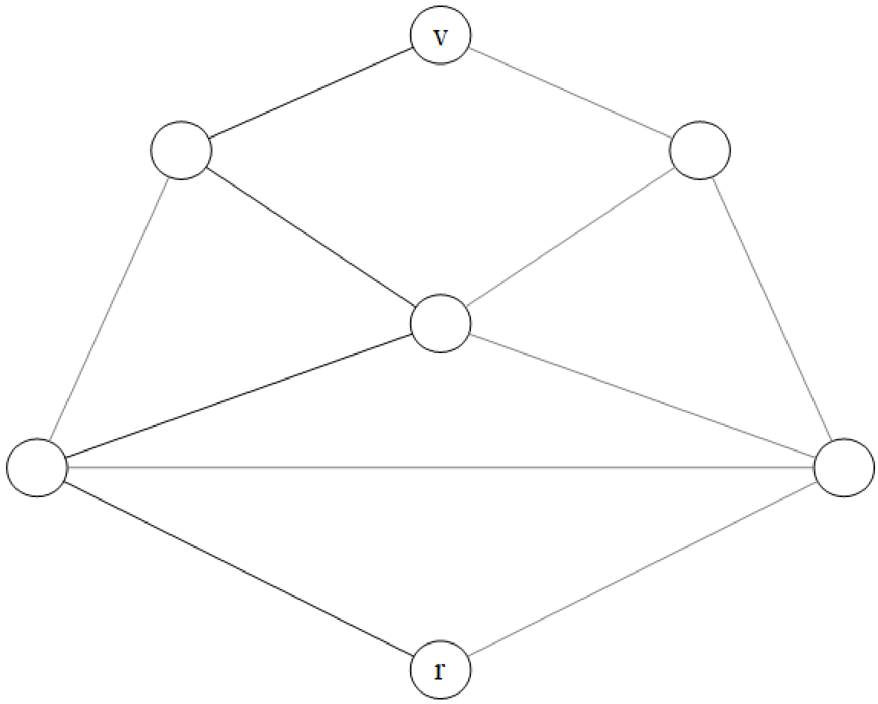


Рисунок 2.3 – Регіональна маршрутизація на опуклому вкладенні.

Теорема 2. Дано площинний граф та його опукле вкладення, вищеописаний алгоритм маршрутизації завжди завершується в *v*.

Доведення. Визначимо дистанцію від регіону *f* до *v*, *d(f, v)* визначимо як мінімум |*vp*| для всіх точок *p* в *f*. Якщо мінімум зустрічається на грані *pd* регіону *f*, можемо сказати що *pd реалізує d(f, v)*.

Потрібно показати, що під час регіональної маршрутизації, жоден регіон не обходиться більше, ніж один раз. В кожному обході регіону *f*, або евристичний алгоритм відтворюється, у випадку чого нічого вже не потрібно доводити, або ж досягається грань та реалізується *d(f, v)*. Коли регіональна маршрутизація ініціюється в точці *r*, ми знаємо, що *d(f, v)* < *d(r, v)*, оскільки *r* буде вершиною *f* і промінь *rv* лежить в *f*. Далі, коли регіональна маршрутизація продовжується із регіону *f* у регіон *f'*, знову *d(f', v)* < *d(f, v)*, оскільки це відбувається коли спільна між *f* та *f′* грань реалізує *d(f, v)*. З цього слідує, що дистанція до *v* (що в розширеній версії також включає регіони) послідовно зменшується під час виконання алгоритму. ■

Отже, наразі вже достатньо обґрунтовано, чому не кожне вкладення може бути евристичним. Не є незвичним для геометричних вкладень графів у евклідові простори – в тому числі для таких, ребра яких не перетинаються – мати «пустоти» що роблять евристичні алгоритми маршрутизації ненадійними. Наведене Твердження 1 неможливо довести методом індуктивним, виходячи з леми, що в евристичному вкладенні, будь-який обраний кінцевий вузол повинен мати грань до найближчого вузла-сусіда у вкладенні, інакше не має сусідів ближчих до себе окрім самого себе. Але безпосередньо аналітичне доведення – навіть для спеціального випадку, коли вкладення максимально площинне (Рисунок 2.3) – виявилось нетривіальним, навіть якщо інтуїтивно можна погодитись щодо справедливості Твердження 1. Тоді, як твердження пропонує *достатні* умови для евристичного вкладення, в жодному випадку воно не є *необхідним*. Вірна характеристика графів із евристичним вкладенням в площину, таким чином, залишається відкритим питанням.

2.2 Модифіковані обмежено збалансовані дерева

Ідея використання віртуальних координат в гіперболічній площині для виконання евристичної маршрутизації не є новою і також вже досліджувалась – неопосередкований (але й не найефективніший) підхід використовується у часто цитованій роботі Клейнберга з Берклі [11] що так і називається, «географічна маршрутизація з використанням гіперболічного простору». Наводиться така теорема:

Теорема 3. Будь-який скінченний з’єднаний граф має евристичне вкладення в гіперболічний простір.

Робота також пропонує вкладення в гіперболічний простір, але упускає важливу умову, щоб таке вкладення підходило для застосування в геометричній маршрутизації – кожне з евристичних вкладень згаданих вище використовує представлення координат вершин, що використовує $\Omega(n \log n)$ біт в гіршому випадку. Тобто, такі евристичні підходи мають аналогічні просторові вимоги, що й традиційні підходи на таблицях маршрутизації. Навіть гірше, зазначене евристичне вкладення має більші вимоги до пропускної здатності, оскільки використовують відповідний простір для заголовків мережевих повідомлень, тоді як традиційні таблиці маршрутизації використовують заголовки розміром $\Theta(\log n)$ біт.

Якщо вкладення й можуть бути корисними, то вони повинні бути *стислими*. Далі ми будемо розглядати модифікований варіант обмежено зважених дерев, що є формою представлення бінарного дерева як структури даних, що формує ключовий компонент знахідок роботи. Бінарні дерева вперше були запропоновані ще в 1972 році [12] і є фундаментальними для реалізації динамічних множин, списків та послідовностей. Розглянемо *модифіковані обмежено зважені дерева* (МОЗД), що є новою структурою даних у вигляді бінарного дерева, яка складає ключовий елемент цієї роботи. В першу чергу, відповідно, це обмежено зважені дерева, які зберігають зважені елементи як крайні елементи, таким чином, щоб глибина кожного елементу з вагою w_i відповідає (2.3), де W – сума всіх вагових значень.

$$O\left(\frac{\log W}{w_i}\right) \tag{2.3}$$

Твердження 3. Властивістю МОЗД є те, що дистанція з будь-якого крайнього елементу до будь-якого іншого крайнього елементу, завжди строго більша за дистанцію від кореня до такого ж кінцевого вузла.

При цьому, дистанції на дереві вимірюються довжину найпростішого шляху. Звісно, ця властивість передбачає, що такі бінарні дерева не є правильними, в розумінні що дозволяє мати деяким внутрішнім вузлам в таких деревах лише один дочірній елемент. Перевага МОЗД, в контексті евристичних вкладень, в тому, що під час руху від будь-якого крайнього вузла до будь-якого іншого крайнього вузла, корінь завжди ближче до пункту призначення, аніж до джерела. Складність, звісно, полягає в тому, щоб побудувати структуру, яка одночасно відповідала б Твердженню 3 та була добре збалансована.

Виявляється, існує відносно простий метод перетворення будь-якого обмежено зваженого дерева в таке, що відповідає Твердженню 3. Тому припустимо, що ми маємо впорядкований набір із n елементів з вагами $\{w_1, w_2, \ldots\}$, таких що $w_i \geq 1$. Як виведено вище, якщо ми зберігаємо ці елементи в крайніх вузлах бінарного дерева T, ми можемо говорити, що таке бінарне дерево T *обмежено зважене*, якщо глибина кожного елементу i відповідає (2.3). Так само, $W = \sum_i w_i$.

Припустимо, тоді, що T це впорядковане обмежено зважене дерево, та нехай r буде позначати корінь такого дерева. Для того, щоб перетворити дерево T в таке, що відповідає Твердженню 3 дерево T', ми замінимо ребро, що з'єднує кожен крайній вузол на його батьківський, встановлюючи шлях довжиною (2.4). Функція par знаходить

$$1 + d_T(r, par(v)) \tag{2.4}$$

В (2.4), $d_T(v, w)$ буде означати довжину шляху від v до w в дереві T. Тобто, ми вставляємо «пусті» вузли між кожним крайнім вузлом та його попередником, що відповідає глибині його попередника (див. Рисунок 2.4).

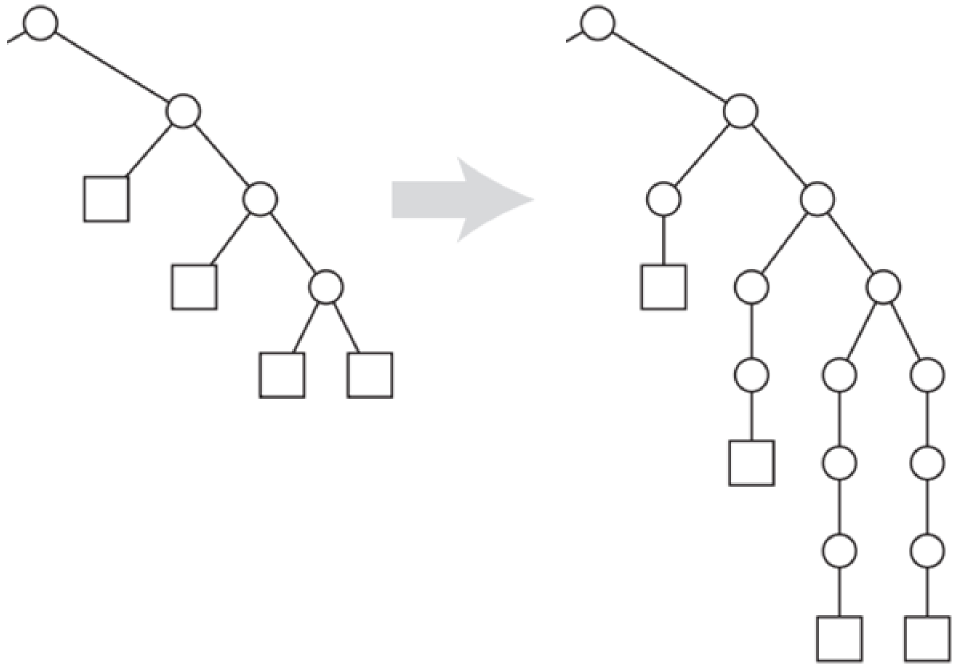


Рисунок 2.4 – Перетворення обмежено збалансованого бінарного дерева в МОЗД

Перетворення збільшує глибину кожного крайнього вузла в дереві T на значення, що є меншим за множник 2 і зберігає глибину всіх інших вузлів у T незмінною. Таким чином, якщо глибина крайнього вузла, що зберігає елемент i в T раніше могла бути щонайбільше $c \log W / w_i$, для деякої константи c, тоді для глибини відповідного крайнього вузла в T' є меншою ніж $2c \log W / w_i$, що все ще відповідає (2.3). Враховуючи те, що T обмежено збалансоване, виходить, що T' також обмежено збалансоване. Це буде нашою Теоремою 3.

Теорема 3. Вищезазначена трансформація обмежено збалансованого дерева T створює МОЗД T'.

Таким чином, є шлях для побудови будь-якої впорядкованої множини елементів як МОЗД для такої множини. Розширене дерево має дещо більше ніж лінійну кількість вузлів, але евристичне вкладення не буде мати потребу будувати МОЗД явним чином.

2.3 Декомпозиція завантажених шляхів

Нехай T буде впорядкованим деревом довільного порядку та глибини із n вузлів. Опишемо схему, що назвемо декомпозицією завантажених шляхів, для декомпозиції T в ієрархічний набір шляхів. Для кожного вузла v в T, нехай $n(v)$ визначає число нащадків в піддереві з коренем в v, включаючи сам вузол v. Для кожної грані між предком та нащадком, $e = (v, w)$ в T, назвемо таке e *завантаженим* ребром якщо $n(v) > n(w)/2$. Інакше, позначимо e як *вільне* ребро. Поєднані компоненти завантажених ребер формують шляхи, що називаються завантаженими шляхами, які можуть мати суміжні вільні шляхи.

Існує дегенеративний випадок, коли можна розглядати шлях з нульовою дистанцією, що складається з єдиного вузла в T дотичного тільки до вільних ребер як важких шляхів.

Варто вернути увагу, що розмір піддерева подвоюється кожного разу коли ми проходимо вільну грань від нащадка до предка, як це зображено на Рисунку 3. Таким чином, якщо стиснути кожен завантажений шлях в T у один «супервузол», зберігаючи відносний порядок вузлів, тоді ми визначаємо дерево Y, глибини $O(\log n)$. Вузли можуть мати будь-який порядок в Y. Тому, для задачі структурування даних, можемо замінити кожен вузол v в Y у якого є d нащадків $v_1, v_2, \ldots, v_d$ використовуючи обмежено збалансоване дерево, що використовує $n(v_i)$ значень як ваги.

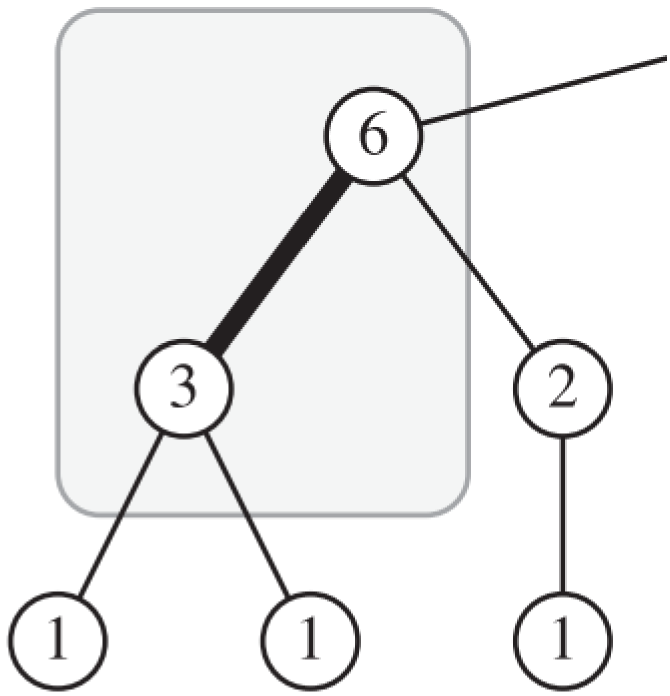


Рисунок 2.5 – Декомпозиція завантаженого шляху дерева.

Корисною властивістю такої заміни є те, що будь-який шлях від крайнього вузла до кореня в кінцевому бінарному дереві буде мати довжину *O(log n)*, оскільки розміри підшляхів в такому обмежено збалансованому дереві сходиться до такого значення. Тобто, така сума може бути описана як значення пропорційне до чогось накшталт (2.5), де кожен n_i це значення *n(vᵢ)*.

$$\begin{aligned}\mathrm{Log}\, n_0 + \log\frac{n_1}{n_0} + \log\frac{n_2}{n_1} + \cdots + \log\frac{n}{n_k} \\ = \log n_0 + \log n_1 - \log n_0 + \cdots + \log n - \log n_k \\ = \log n \qquad (2.5)\end{aligned}$$

3 РОЗРОБКА ЕФЕКТИВНОЇ МОДЕЛІ МАРШРУТИЗАЦІЇ ДАНИХ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

В цьому розділі показано, як створити комбінаторне евристичне вкладення в метричний простір бінарного дерева, побудованого на двійково-раціональних числах. Такий простір називається *метричним простором бінарного дерева*.

Розглянемо нескінченне бінарне дерево, що є абстрактним метричним простором, в якому дистанції між будь-якими двома вузлами дерева це просто число ребер на найкоротшому шляху між ними. Але існує інша природня метрика, що може бути сформована на такому ж дереві, завдяки вкладенню в простір *двійково-раціональних чисел* (Рисунок 3.1). Тобто, раціональні номери зі знаменниками, які є степенями двійки.

Нехай f буде відображенням з такого дерева B на відкритий інтервал (0, 1) що з'єднує корінь дерева з 1/2, та що з'єднує нащадків вузла х на рівні і дерева на $f(x) \pm 2^{-i-2}$; тобто, нащадки кореня відображаються на раціональні числа 1/4 та 3/4, нащадки нащадків – на 1/8, 3/8, 5/8, 7/8, і так далі. Ми визначаємо бінарну метрику на B як метрику в якій дистанція між двома вузлами дерева x та y складає $|f(x) - f(y)|$. Зверніть увагу, що усі дистанції в такому просторі менші за 1.

Такі графи можуть мати евристичне вкладення у відокремленому метричному просторі що суміщає ці дві метрики; ми назвемо його *метричним простором бінарних дерев*. Точка в такому просторі представляється парою (x, y) де x та y є вузлами нескінченного бінарного дерева B, та в якому x має бути нащадком y (або бути самим y).

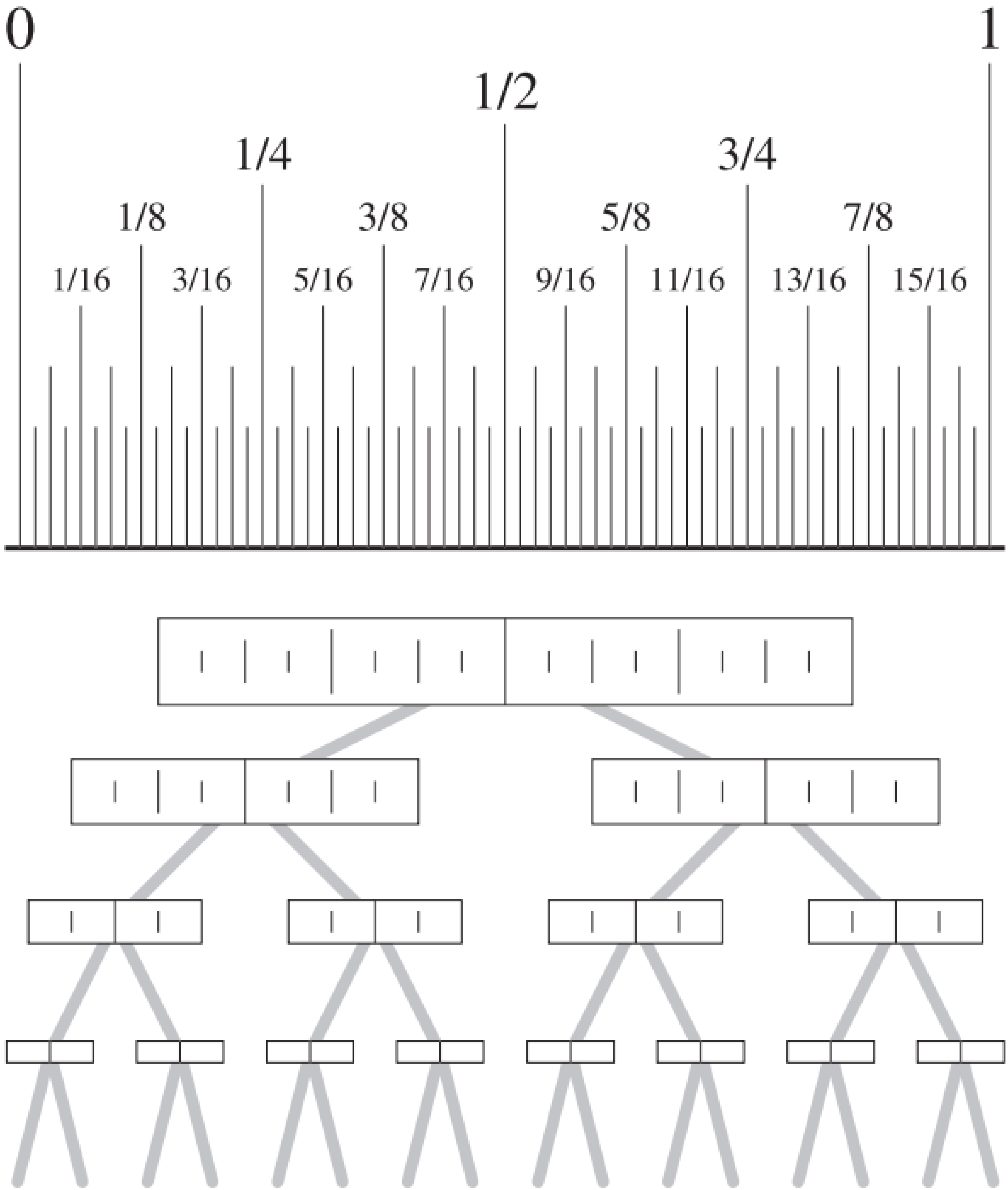


Рисунок 3.1 – Двійково-раціональні числа (зверху)

та схема бінарного простору (знизу)

ВИСНОВКИ

Робота проводить аналіз хмарних комп'ютерних мереж, що знаходяться в географічно розподілених сценаріях хмарних веб-застосунків та частин сервісів. Розглянуто типові моделі ієрархічних комп'ютерних мереж та те, як топологічно складна комп'ютерна мережа може виникнути вже в рамках одного ЦОД. Проведено порівняння двох основних сімейств алгоритмів маршрутизації – алгоритмів з аналізом станів каланів, та дистанційно-векторних алгоритмів. Визначено істотні недоліки перших в географічно розподілених мережах, із урахуванням потенційної динаміки таких мереж.

Розглянуто математичне формулювання для мереж та формально визначено вищезазначені алгоритми маршрутизації над такими математичними моделями мереж. Обґрунтовано, чому виключно дистанційно-векторні протоколи можуть використовуватись у масштабних мережах, та як аналітична ефективність протоколів з аналізом станів каналів приховує неготовність алгоритму до роботи в масштабних мережах та просторову неефективність алгоритмів.

Визначено основну слабкість дистанційно-векторних протоколів та оглянуто її геометричну природу. Оглянуто наявні рішення – в тому числі такі, що так само як і ця робота, використовують гіперболічний простір. Наведено критику для таких існуючих робіт, що полягає у низькій просторовій ефективності запропонованих рішень. Визначено новий підхід до вирішення задачі маршрутизації з використанням гіперболічного вкладення в самоподібні (фрактальні) дерева. Запропоновано нову структуру даних – модифіковані обмежено зважені дерева (МОЗД).

Пропонується стислий спосіб представлення координат вузлів графів, вкладених як МОЗД. Стверджується, що будь-який скінченний граф, за таким представленням, буде знаходитись в нескінченному самоподібному дереві. Застосування розробленого підходу дозволяє розробку ефективних програмних маршрутизаторів, що гарантують наявність оптимальних шляхів між вузлами динамічної та потенційно масштабної мережі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Зінченко О.В., Іщеряков С.М., Прокопов С.В., Сєрих С.О., Василенко В.В.. «Хмарні технології». – 2020. – 74 с.
2. Шимчук Г.В., Маєвський О.В., Назаревич О.Б., Стадник М.А.. «Грід-системи та технології хмарних обчислень». – 2016. – 340 с.
3. Open Container Initiative [Електронний ресурс]. – Режим доступу: https://opencontainers.org/
4. How Kubernetes Saved OpenStack [Електронний ресурс]. – Режим доступу: https://www.mirantis.com/resources/how-kubernetes-saved-openstack/
5. CloudFlare [Електронний ресурс]. – Режим доступу: https://www.cloudflare.com/application-services/products/load-balancing/
6. Гніденко М.П., Вишнівський В.В., Зінченко О.В., Сєрих С.О., Прокопов С.В.. «Конвергентна мережна інфраструктура». – 2019. – 173 с.
7. Chroboczek, J., D. Schinazi, «The Babel Routing Protocol», RFC 8966. – 2021 – Режим доступу: https://www.rfc-editor.org/info/rfc8966
8. Bose P., Morin P., Stojmenović I., Urrutia J., «Routing with Guaranteed Delivery in Ad Hoc Wireless Networks». – 2001. – Режим доступу: https://link.springer.com/article/10.1023/A:1012319418150
9. П.П. Воробієнко, Л.А. Нікітюк, П.І. Резніченко. – САММІТ-Книга, 2010. – 708 с.:
10. Балога С. І.. «Дискретна математика». – м. Ужгород, 2012 – 90 с.
11. R. D. Kleinberg, «Geographic Routing Using Hyperbolic Space». – 2007. – Режим доступу: https://www.cs.cornell.edu/~rdk/papers/pgr.pdf
12. J. Nievergelt, E. Reingold, «Binary search trees of bounded balance» – 1972. – Режим доступу: https://dl.acm.org/doi/pdf/10.1145/800152.804906
13. D. Comer, «A Note on Median Split Trees» – 1980. – Режим доступу: https://dl.acm.org/doi/pdf/10.1145/357084.357092

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

ДИПЛОМНА РОБОТА

на ступінь вищої освіти магістр

із спеціальності 122 Комп'ютерні технології

Дослідження гіперболічної геометрії хмарних комп’ютерних мереж

Виконав: студент 6 курсу, групи КНДМ-61

Ямковий Владислав Валерійович

Керівник: завідувач кафедри Комп'ютерних наук

д.т.н., професор Вишнівський В.В.

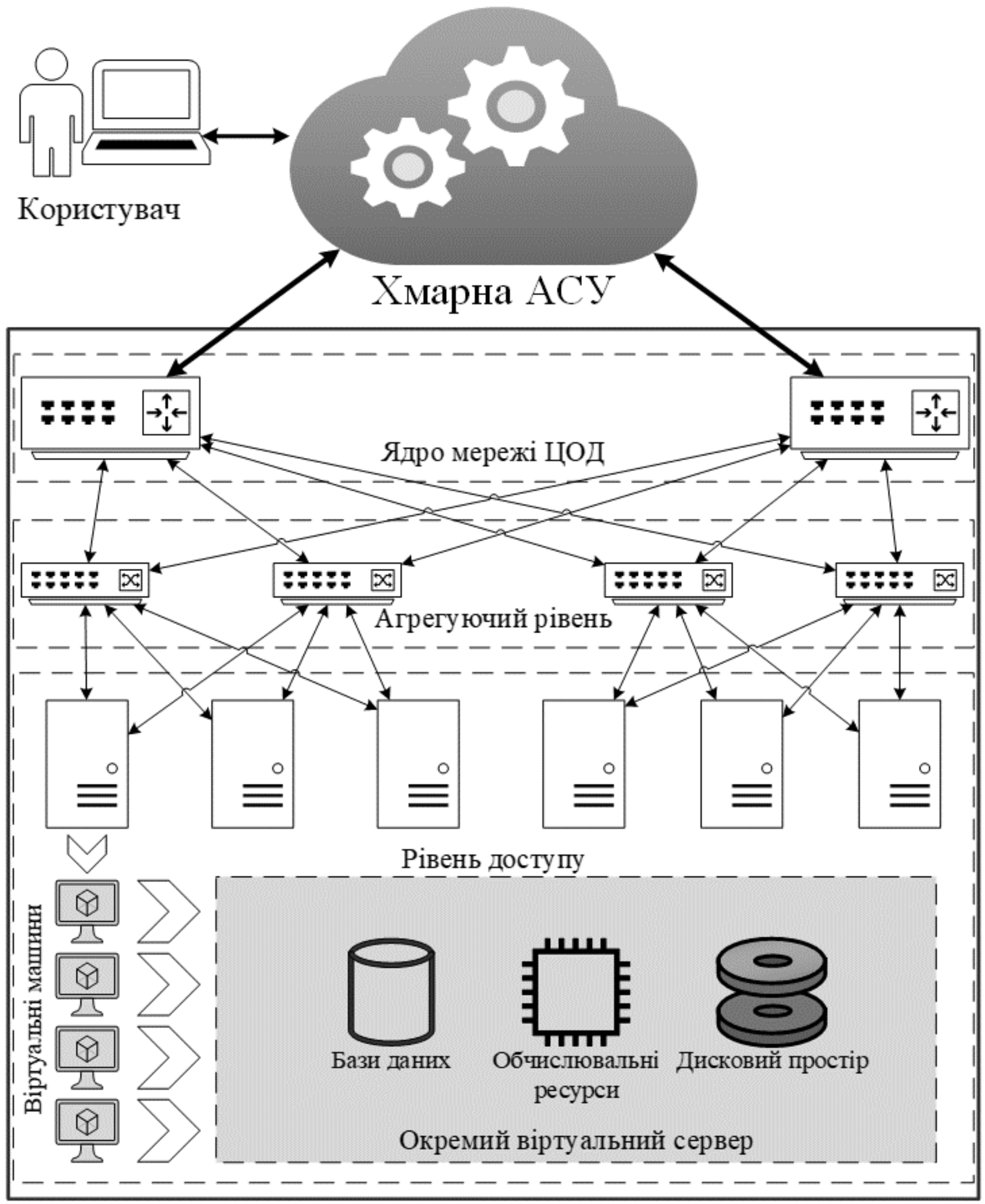
Київ - 2023

ЗАГАЛЬНА ХАРАКТЕРИСТИКА ДИПЛОМНОЇ РОБОТИ

Тема	Дослідження гіперболічної геометрії хмарних комп’ютерних мереж
Мета дослідження	Удосконалення математичних методів маршрутизації даних в хмарних комп’ютерних мережах
Наукове завдання	розробка геометричної моделі комп’ютерних мереж з метою вирішення задачі маршрутизації
Об’єкт дослідження	хмарні комп’ютерні мережі
Предмет дослідження	технології маршрутизації трафіку в географічно розподілених комп’ютерних мережах

Розподіл функціональних зон у хмарних інформаційних системах

Загальна архітектура хмарних АСУ



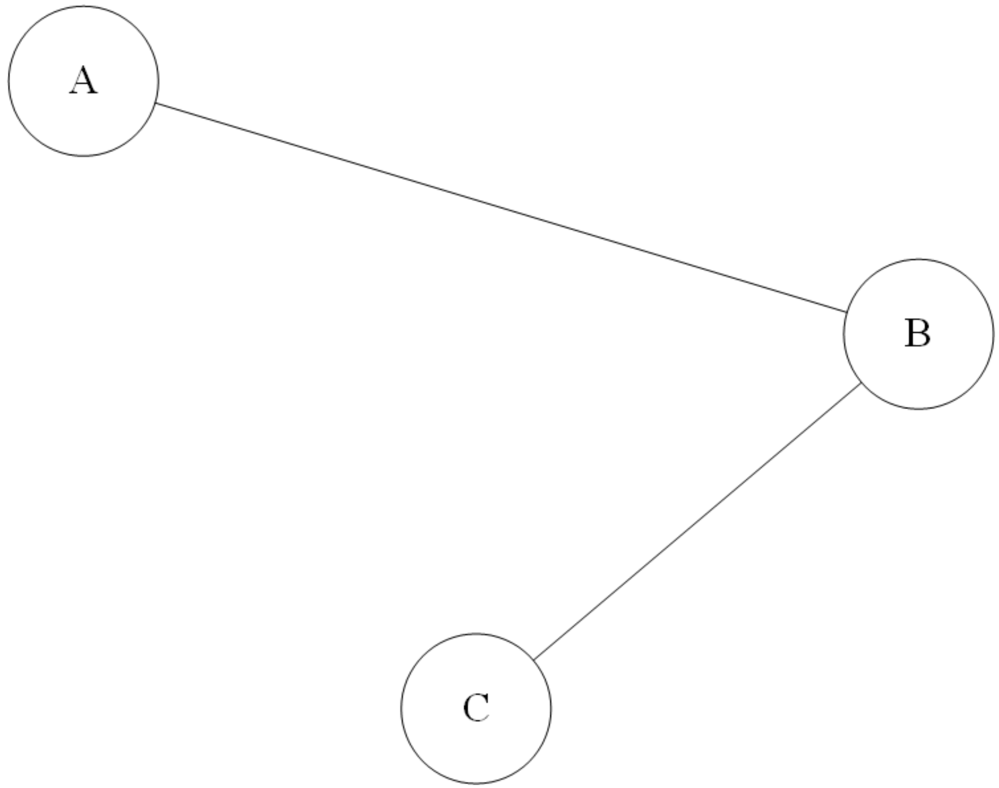
Порівняння протоколів маршрутизації

	протоколи з аналізом стану каналів	дистанційно-векторні протоколи
Повнота інформації	Будуються на алгоритмі Дейкстри та обраховують всі шляхи; потребують знання глобальної топології для обрахунку всіх шляхів.	Будуються на евристиці Беллмана-Форда; кожен маршрутизатор містить лише інформацію про сусідні вузли.
Стан	Кожен маршрутизатор повинен підтримувати повну топологію та використовувати таку інформацію для рішень щодо маршрутизації.	Вузли відповідають лише за моніторинг стану прилеглих ребер, передають інформацію про зміни оточуючим вузлам.
Витратність	Висока, оскільки такий протокол передбачає обробку значної кількості інформації про топологію та стан мережевих каналів.	Низька, оскільки евристичні алгоритми не вимагають знання глобальної топологічної конфігурації мережі.
Складність	$O(\|E\|\log\|V\|)$	$O(\|V\|\cdot\|E\|)$

Евристичний пошук шляхів

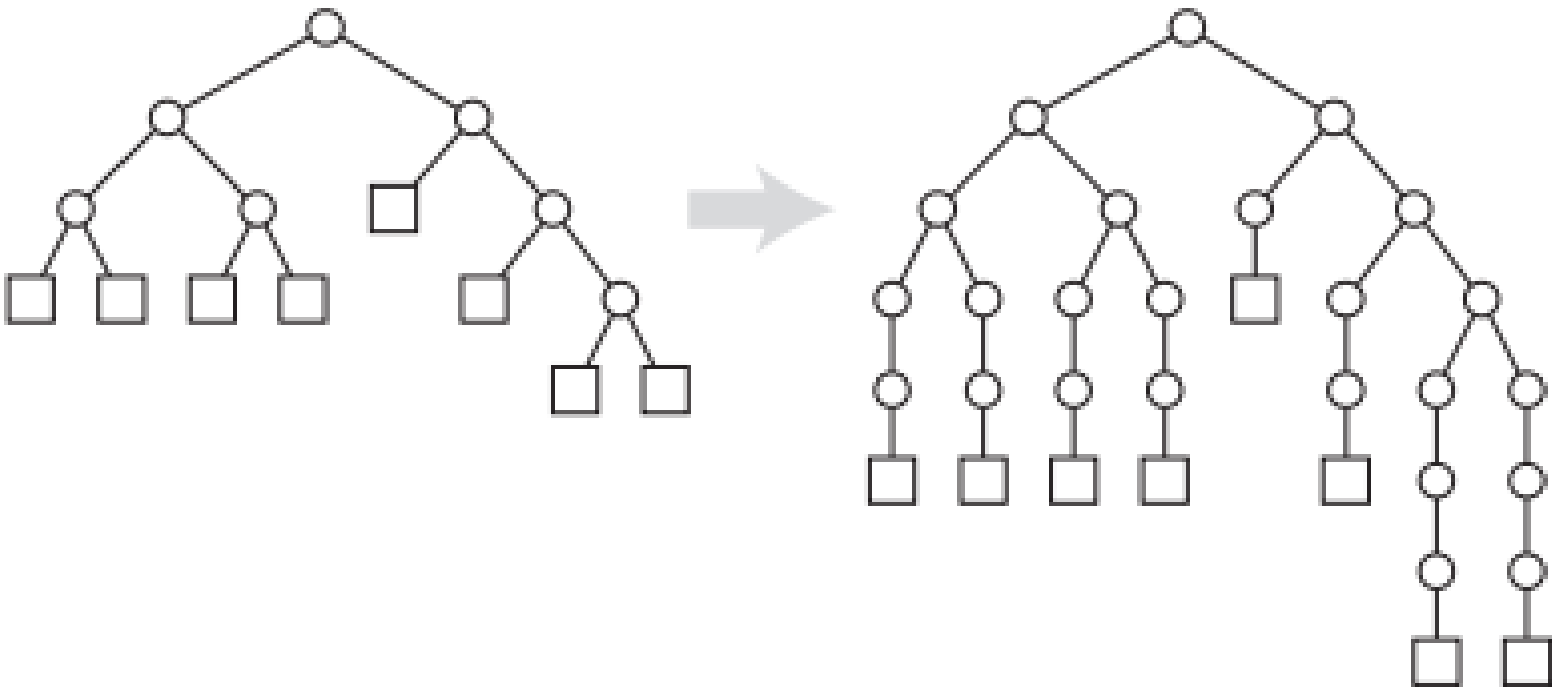
```
Нехай i := 0, j₀ := p₁;
допоки jᵢ ≠ p₂:
  якщо існує такий k прилеглий до jᵢ, що d(jᵢ,p₂)<d(k,p2):
        нехай i=i+1, jᵢ=k;
        інакше: кінець.
```

Проблема евристичного пошуку шляхів

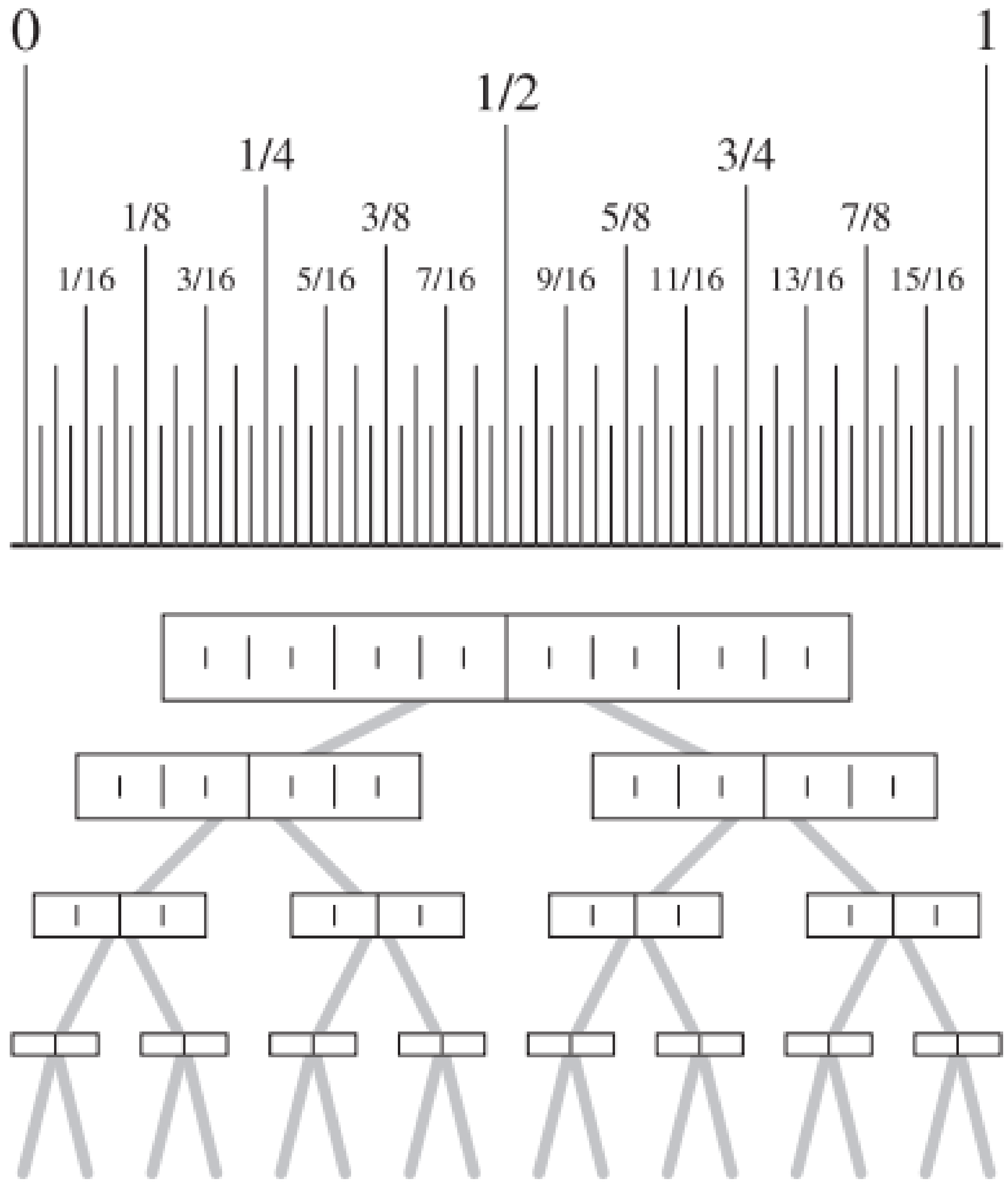


$$d(A, C) < d(A, B)$$

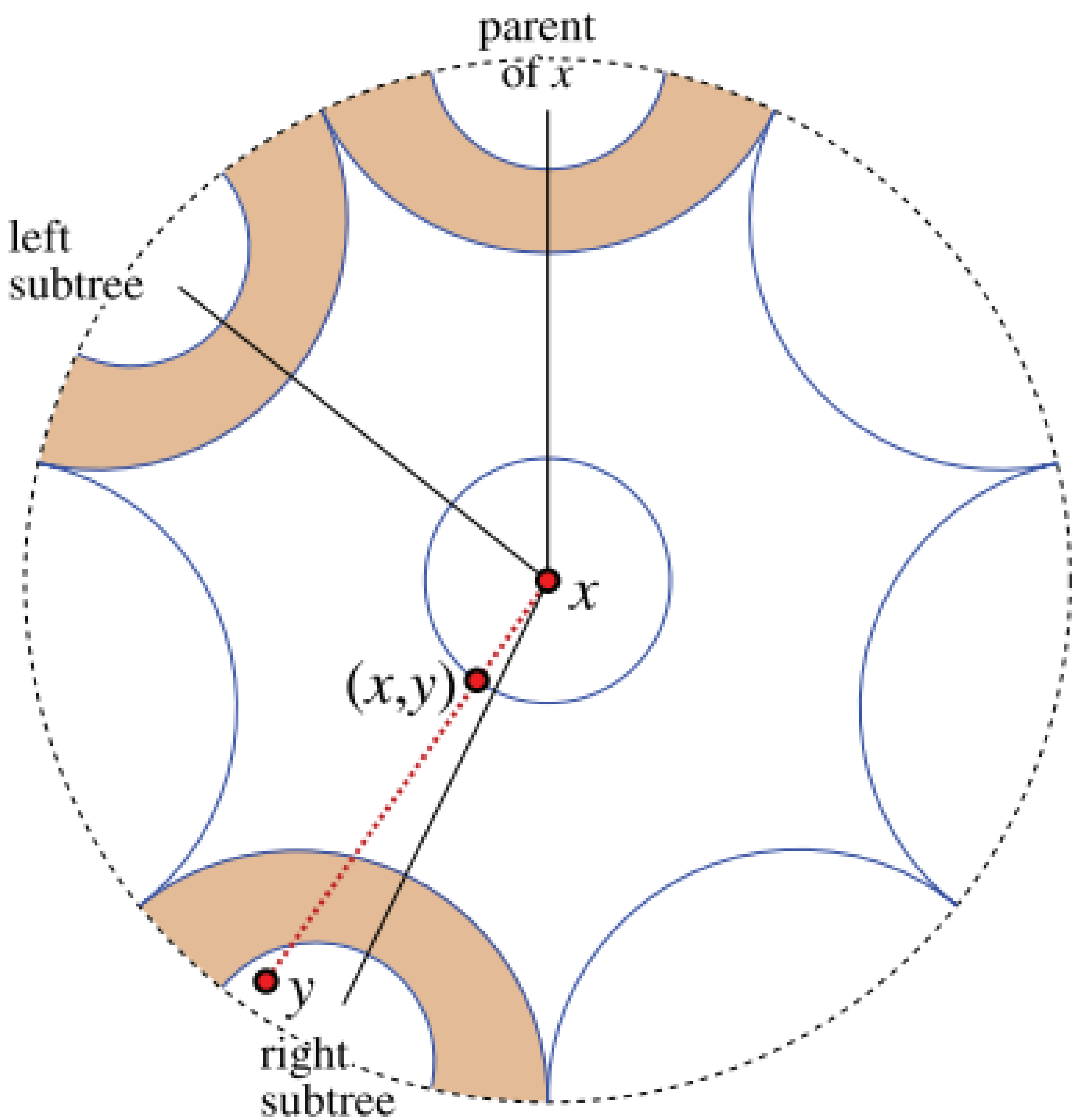
Зважені бінарні дерева



Декомпозиція диадичних дерев



Гіперболічне вкладення



ВИСНОВКИ

1. В роботі проведено аналіз комп'ютерних мереж, що знаходяться в географічно розподілених сценаріях хмарних веб-сервісів. Розглянуто типові моделі ієрархічних комп'ютерних мереж та способів їх утворення.
2. Розглянуто методи математичного моделювання мереж зі складною топологією. Визначений новий підхід для вирішення задачі маршрутизації з використанням геометричної маршрутизації.
3. Визначені переваги алгоритмів що використовують стисле геометричне моделювання мережевих графів для виконання евристичного алгоритму маршрутизації.
4. Застосування розробленого підходу до моделювання комп'ютерних мереж зі складною ієрархією дозволяє підвищити ефективність розробки нових алгоритмів маршрутизації.