Національний університет “Львівська політехніка”

На правах рукопису

Шпур Ольга Миколаївна

УДК 621.391

Підвищення якості надання композитних сервісів у мережах із сервісно-орієнтованою архітектурою

05.12.02 – *телекомунікаційні системи та мережі*

Дисертація на здобуття наукового ступеня
кандидата технічних наук

Науковий керівник -
доктор технічних наук,
доцент **Стрихалюк Б.М.**

Ідентичність всіх примірників дисертації
ЗАСВІДЧУЮ:
Вчений секретар спеціалізованої
вченої ради **/І.В. Демидов/**

Львів – 2017

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API - Інтерфейси програмування застосувань
CoS - Клас сервісу
EDA - Підсистема розводки по обладнанню
ETSI - Європейський інститут стандартизації телекомунікацій
FTP - Протокол передачі файлів
HDA - Горизонтальна розподільча підсистема
HTTP - Протокол передачі гіпер-текстових документів
IaaS - Інфраструктура як послуга
IETF - Відкрите міжнародне співтовариство проектувальників
IoT - Інтернет речей
IP - Інтернет протокол
ITU-T - Міжнародний союз телекомунікацій
LBVM - Локальне балансування навантаження між віртуальними машинами
MDA - Головна розподільна підсистема
MPLS - Багатопротокольна комутація на основі міток
NVF - Віртуальні мережеві функції
OSI - Модель взаємодії відкритих систем
PaaS - Платформа як послуга
QoS - Якість обслуговування, якість надання послуг, якість сервісу
RSVP - Протокол резервування ресурсів
SaaS - Програмне забезпечення як послуга
SLA - Угода про рівень забезпечення якості надання послуг
SMTP - Простий протокол пересилання пошти
SOAP - Протокол обміну структурованими повідомленнями в розподілених обчислювальних системах
ToS - Рівень пріоритету IP, вид послуги
VLAN - Віртуальна локальна мережа
VoIP - Телефонія на основі протоколу IP
VM - Віртуальна машина
ZDA - Зонові сегменти з вузлами консолідації
РМ - Фізична машина
ПЗ - Програмне забезпечення
ТКС - Телекомунікаційна система
ЦОД - Центр обробки даних

ВСТУП

Актуальність теми. Взаємопроникнення інформаційних технологій, сервісної інженерії та телекомунікаційних систем ставить нові вимоги перед проектантами, котрі повинні враховувати існування теоретично необмеженої множини сервісів, що складно підпорядковуються існуючій класифікації інформаційно-телекомунікаційних послуг. Відомі методи передавання інформаційних потоків не здатні забезпечити підтримку процесів надання композитних сервісів, у яких мета обслуговування досягається взаємодією декількох елементарних складових з параметрами, що можуть суттєво відрізнятися. Розробники таких сервісів враховують можливості виникнення помилок на нижчих рівнях Еталонної моделі взаємодії відкритих систем, проте передбачають надання сервісу навіть у випадку суттєвих мережевих збоїв, окрім критичних. Однак, це призводить до зниження якості обслуговування, що має незворотні наслідки з точки зору кінцевого користувача.

Поєднання різних телекомунікаційних технологій, які створюють передумови для побудови гнучких та високопродуктивних сервісно-орієнтованих систем, можуть використовуватись для розв`язання задач у різних галузях. Однак, на сьогодні, одним із стримуючих факторів щодо впровадження та подальшого розвитку таких мереж є теоретичне недоопрацювання моделей надання сервісів, що зумовлені відсутністю вичерпних відомостей про структуру мереж центрів оброблення даних (ЦОД). Тому, у процесі надання композитного сервісу слід враховувати параметри інформаційно-телекомунікаційної інфраструктури з метою оптимального вибору елементарних складових цього сервісу або їх міграції у її віртуалізованій реалізації.

Дослідженням задач підвищення якості надання композитних сервісів активно займаються в першу чергу закордонні фахівці. Однак, в Україні інтерес до задач цього класу невпинно зростає. Зокрема, ці питання у своїх роботах розглядали представники наукових шкіл професора Поповського В.В., професора Беркман Л.Н., професора Глоби Л.С., професора Ложковського А.Г.

Серед іноземних дослідників слід відзначити роботи Кривінської Н.В., Лунтовського А.О., Schill A., Wolf A., Soares J., Dias M., Carapinha J. та інших.

Основна частина робіт згаданих авторів спрямована на покращення якості надання композитних сервісів шляхом удосконаленого управління мережними ресурсами та їх планування. Проте, не враховано процес формування композитних сервісів, який відкриває нові можливості щодо покращення їх надання без необхідності ускладнення алгоритмів ресурсного управління. Для досягнення цієї мети слід розв’язати протиріччя між якістю надання композитного сервісу та тривалістю його формування з елементарних сервісних компонентів. Це потребує розроблення нових моделей розгортання та надання веб-сервісів, а також удосконалення методів маршрутизації інформаційних потоків з урахуванням стійкості структури центрів оброблення даних для зменшення кількості переоцінок маршрутів обміну даними між елементарними сервісами.

Таким чином, покращення часових параметрів надання композитних сервісів з одночасним підвищенням стійкості віртуальних топологій ЦОД, які утворюються дистанційно-векторними методами в умовах різкого зростання різноманітності потоків у сучасних гетерогенних мережах для задоволення потреб користувачів у інформаційно-комунікаційних застосуваннях реального часу є актуальним науковим завданням.

Зв’язок роботи з науковими програмами, планами, темами. Тематика дисертаційної роботи безпосередньо пов’язана з пріоритетними напрямками розвитку науки і техніки в рамках державних програм розвитку та інформатизації Кабінету Міністрів України, координаційних планів науково-дослідних робіт Міністерства освіти і науки України "Перспективні інформаційні технології, прилади комплексної автоматизації, систем зв’язку" та "Прикладні дослідження з найважливіших проблем природничих, суспільних і гуманітарних наук". Дисертація виконувалась в рамках держбюджетних науково–дослідних робіт "Моделі та структури конвергентних телекомунікаційних мереж на основі CLOUD – технологій" ("ДБ/CLOUD"),

(2013–2014 рр.), № держреєстрації 0113U003184 та «Методи побудови та моделі інформаційно – телекомунікаційної інфраструктури на основі SDN – технологій для систем електронного урядування" ("ДБ/SDN"), (2015–2016 рр.), № держреєстрації 0115U000444.

Мета і завдання дослідження. Мета дисертаційної роботи полягає у покращенні часових параметрів надання композитних сервісів з одночасним підвищенням стійкості віртуальних топологій центрів обробки даних у мережах із сервісно-орієнтованою архітектурою.

Для досягнення поставленої мети необхідно розв’язати такі завдання:

1. Формалізувати параметри формування та надання композитних сервісів;
2. Провести аналіз методів оцінки ефективності надання композитних сервісів з урахуванням структури дата-центру;
3. Удосконалити метод пошуку маршруту з урахуванням стійкості структури віртуалізованого центру обробки даних сервісно-орієнтованої мережі;
4. Розробити метод балансування навантаження з урахуванням доступності сервісних компонентів;
5. Розробити модель розподілу мережних ресурсів між дата–центрами;
6. Розробити модель надання композитного сервісу в сервісно-орієнтованих мережах із урахуванням структури центру обробки даних та процесу міграції віртуальних машин, які підтримують сервісні компоненти.
7. Оцінити ефективність запропонованих методів та алгоритмів.

Об’єкт дослідження – процес надання послуг у мережах із сервісно-орієнтованою архітектурою.

Предмет дослідження – методи покращення параметрів якості надання композитних сервісів у мережах із сервісно-орієнтованою архітектурою.

Методи дослідження. Дослідження виконано на основі використання положень теорії ймовірності та математичної статистики, теорії випадкових графів, методів лабораторного експерименту, аналітичного та імітаційного моделювання.

Наукова новизна роботи полягає у тому, що:

1. Вперше запропоновано метод балансування навантаження з урахуванням доступності фізичних ресурсів та з використанням віртуалізації мережних функцій (NFV – Network Function Virtualization), який реалізований на основі модифікованої сервісно–орієнтованої архітектури управління мережею, що дало змогу підвищити ефективність використання апаратних ресурсів центру обробки даних та зменшити затримку надання сервісів кінцевому користувачу.
2. Набула подальшого розвитку модель надання композитного сервісу в сервісно–орієнтованих мережах, шляхом урахування структури центру обробки даних та процесу міграції компонентів сервісу, що дало змогу покращити часові параметри якості надання послуг кінцевим користувачам;
3. Набула подальшого розвитку модель розподілу мережних ресурсів між дата–центрами на основі об’єднання та перегрупування потоків запитів, що дозволило можливість покращити часові параметри процесу передавання даних.
4. Удосконалено метод пошуку маршруту з урахуванням стійкості структури мережі віртуалізованого центру обробки даних, шляхом врахування параметрів віртуальної топологічної структури у метриці маршрутизації, що дало змогу зменшити затримку у процесі пошуку маршруту у віртуалізованій частині центру обробки даних сервісно–орієнтованої мережі, а також мінімізувати затримку надання атомарного сервісу.

Практичне значення одержаних результатів полягає в тому, що:

1. Удосконалено метод пошуку маршруту з урахуванням стійкості структури віртуалізованого центру обробки даних мережі, що дало змогу зменшити на 12% затримку у процесі пошуку маршруту у динамічно-змінній структурі ЦОД за рахунок врахування особливостей її топології.

2. Розроблено алгоритм балансування навантаження, який дав змогу зменшити тривалість обслуговування запитів у 3 рази та зменшити затримку передавання пакетів з кінця в кінець у 2,75 рази за рахунок врахування доступності фізичних ресурсів та використання віртуалізації мережних функцій.
3. На основі моделі розподілу мережних ресурсів між дата–центрами розроблено алгоритм, який дозволив зменшити завантаженість граничних маршрутизаторів мереж центрів обробки даних у 1,5 рази за рахунок більш ефективного використання ресурсів оптичної мережі між дата-центрами.
4. Розроблено алгоритм прокладання наскрізних тунелів між дата-центрами, що дало змогу зменшити затримку передавання пакетів з кінця в кінець у 2,92 рази за рахунок максимізації завантаженості оптичної несучої в оптичній телекомунікаційній системі зі спектральним ущільненням каналів.
5. Розроблено програмно-апаратний комплекс для надання композитних сервісів із гарантованим рівнем якості обслуговування, що забезпечило зменшення затримки надання сервісів на 70%.

Отримані в рамках дисертаційного дослідження результати дають змогу на етапі проектування мережі заздалегідь передбачити впровадження хмарних сервісів і адаптуватися до різноманітності і ускладнення їх структури з перспективою надання користувачам інфраструктури в якості сервісу (IaaS).

Основні результати дисертаційної роботи використані та впроваджені:

- у Львівській філії ПАТ "Укртелеком" для підвищення функціональності корпоративної мережі шляхом впровадження методів ефективного розподілу мережних ресурсів між дата–центрами;
- у ПП "Цифрові технології" для підвищення якості обслуговування абонентів у процесі надання розподілених хмарних сервісів; для підвищення коефіцієнта доступності композитних додатків на основі веб–сервісів;

- у навчальному процесі кафедри телекомунікацій Національного університету «Львівська політехніка» для модернізації курсів лекцій з дисциплін «Телекомунікаційні та інформаційні мережі, ч.1», «Розподілені сервісні системи та Cloud–технології» та «Системне програмування інфокомунікацій».

Апробація результатів дисертації. Основні наукові результати і положення дисертації представлені, доповідались та всебічно обговорені на 16-ти міжнародних та всеукраїнських науково-технічних конференціях, наукових семінарах та симпозіумах: Міжнародній конференції «Сучасні проблеми радіоелектроніки, телекомунікацій, комп'ютерної інженерії» TCSET'2016, 2014 (м. Львів-Славське, 2016, 2014 рр.); Науково-технічній конференції «Проблеми телекомунікацій» (м. Київ, 2014, 2015, 2016 рр.); Міжнародній науково-технічній конференції «Сучасні інформаційно-телекомунікаційні технології» (м. Київ, 2015 р.); Міжнародній науково-технічній конференції «The experience of designing and application of CAD Systems in microelectronics» CADSM'2015 (Поляна-Свалява, 2015р.); International Scientific-Practical Conference «Problems of Infocommunications, Science and Technology» PICS&T'2015 (м. Харків, 2015 р.); Міжнародній науково-практичній конференції «Нові досягнення в галузі інформаційно-комунікаційних технологій» AICT-2015 (м. Львів, 2015 р.); Міжнародній науково-практичній конференції «Фізико-технологічні проблеми радіотехнічних пристроїв, засобів телекомунікацій, нано- та мікроелектроніки», (м. Чернівці, 2014 р.); Міжнародній науково-технічній конференції «Вимірювальна та обчислювальна техніка в технологічних процесах» (м. Одеса, 2014 р.); 69-ій науково-технічній конференції професорсько-викладацького складу, науковців, аспірантів та студентів (м. Одеса, 2014 р.); Науково-практичній та науково-методичній конференціях «Сучасні проблеми телекомунікацій і підготовка фахівців в галузі телекомунікацій» (м. Львів, 2014, 2013 рр.). Крім цього, дисертаційна робота представлена на науковому семінарі кафедри телекомунікацій Національного університету «Львівська політехніка».

Публікації. За результатами досліджень, які викладені у дисертаційній роботі, опубліковано 27 наукових праць, серед них 2 статті [1-2] у закордонних фахових виданнях, що індексуються міжнародними науково-метричними базами, 5 статей у фахових виданнях України [3-7], які індексуються міжнародними науково-метричними базами, 4 статті у фахових виданнях України за переліком МОН [8-11] та 16 публікацій [12-27] у збірниках праць міжнародних і всеукраїнських конференцій.

Особистий внесок здобувача. Всі результати наукових, теоретичних і практичних досліджень, які викладені в дисертації, одержані автором особисто. У працях, опублікованих у співавторстві, дисертантові належать: у роботах [1, 14, 22, 25] – розроблення методу балансування навантаження на основі інтегрованої системи управління ресурсами з використанням функції NVF; [2, 24] – створення імітаційної моделі надання хмарних послуг із оптимізацією часу надання сервісу з врахуванням структури ЦОД; [3, 10, 23] – розроблення методу оцінки доступності програмних компонент у системах із сервісно-орієнтованою архітектурою; [4, 12, 15, 20] – розроблення методу підвищення ефективності використання мережних ресурсів інформаційно-телекомунікаційних систем; [5, 11, 17, 21, 27] – розроблення алгоритму пошуку шляху за критерієм мінімальної затримки; [6, 7, 19] - створення тестової платформи мультисервісної телекомунікаційної мережі; [8, 9, 13, 16, 18, 26] – створення моделі надання сервісу з урахуванням особливостей віртуалізації центру обробки даних.

Структура та обсяг роботи. Робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатку. Загальний обсяг роботи складає 149 сторінок друкарського тексту, із них 7 сторінок вступу, 120 сторінок основного тексту, 60 рисунків, 10 таблиць на 12 сторінках, список використаних джерел зі 126 найменувань, додаток на 3 сторінках.

РОЗДІЛ 1.
АНАЛІЗ МЕТОДІВ ПОБУДОВИ ІНФРАСТРУКТУРИ ТА МОДЕЛЕЙ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ОБСЛУГОВУВАННЯ В МЕРЕЖАХ ІЗ СЕРВІСНО-ОРІЄНТОВАНОЮ АРХІТЕКТУРОЮ

У першому розділі проведено аналіз та показано важливість задач забезпечення якості обслуговування для розвитку сучасних сервісно-орієнтованих телекомунікаційних мереж. Доведено, що важливим аспектом при покращенні основних показників QoS є зниження затримки наскрізного передавання інформації з одночасним підвищенням стійкості віртуальних топологій ЦОД, які утворюються дистанційо-векторними методами за умов різкого зростання динаміки потоків у сучасних гетерогенних мережах. Розв’язання даного протиріччя можливе шляхом підвищення ефективності балансування навантаження в мережах на основі сервісно-орієнтованих архітектур.

1.1. Моделі надання сервісів у мережах із сервісно-орієнтованою архітектурою

З появою комп’ютерів та обчислювальних пристроїв почалася нова ера інформатизації суспільства. Перші операційні системи та програми були монолітними та мали обмежену функціональність. Програми були повністю залежними від апаратної архітектури, для якої вони були створені, а тому не могли функціонувати ні на одному комп’ютері, що мав відмінну архітектуру.

У 1980-х роках компанія IBM створила перший суперкомп’ютер, який володів великою потужністю та високою продуктивністю. Оскільки такий комп’ютер був набагато потужнішим від стаціонарних комп’ютерів та міг значно швидше проводити складні обчислення, його обчислювальні ресурси почали надавати в оренду користувачам за допомогою методів віддаленого доступу. У цьому випадку клієнтські термінали відповідали за встановлення та розрив з’єднання з суперкомп’ютером, обмін інформацією, представлення результатів обчислення у зручному та зрозумілому для користувача форматі.

При цьому всі обчислення проводилися на суперкомп'ютері. Така архітектура отримала назву клієнт-сервер (рис. 1.1) і лягла в основу майже всіх розподілених систем, які існують на сьогодні.

Розподілена система – це колекція незалежних компонентів, які для користувача постають, як єдина цілісна система [28, 49, 74, 120]. Розподілена система складається з компонентів, які є повністю автономними. Користувачі, що взаємодіють з такою системою, сприймають її, як єдине ціле. Це означає, що компоненти розподіленої системи певним шляхом повинні спілкуватися та взаємодіяти між собою. А отже, одним з найважливіших завдань створення розподіленої системи є забезпечення взаємодії незалежних компонентів.

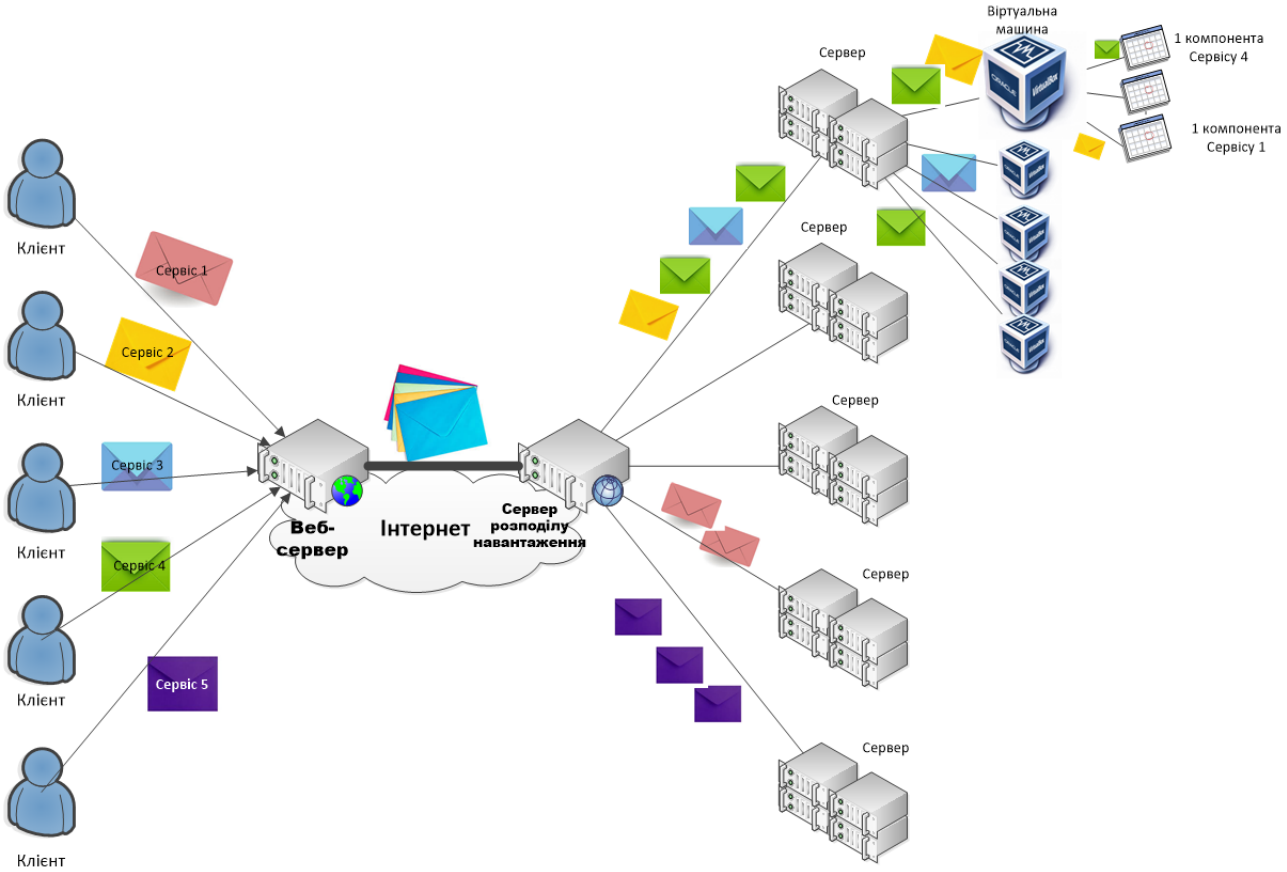


Рис. 1.1. Архітектура розподілених систем – «клієнт-сервер».

Одною з основних характеристик розподіленої системи є відмінність параметрів та характеристик компонентів розподіленої системи та принцип за яким вони спілкуються, приховується від користувача. Це саме стосується і внутрішньої організації розподіленої системи. Іншою важливою характеристикою є те, що користувачі та додатки можуть взаємодіяти з

розподіленою системою, відповідно до узгодженого та уніфікованого способу, незалежно від того, де і коли така взаємодія має місце.

Разом з тим архітектура клієнт-сервер не вирішує питання монолітності та тісної прив'язаності програм до архітектури апаратного забезпечення. Програми, створені для різних платформ та на різних мовах програмування, не мали уніфікованих та стандартизованих засобів, які б дозволяли їм спілкуватися між собою та взаємодіяти на належному рівні. З цією метою було створене програмне забезпечення та бібліотеки, які відносяться до логічного рівня Middleware. Цей рівень дав змогу відділити програмне забезпечення від апаратного завдяки використанню стандартизованих протоколів спілкування (Sockets, RPC, RMI, CORBA). Програмне забезпечення проміжного рівня надавало програмам вищого рівня стандартний інтерфейс для спілкування з іншими програмами по мережі, при цьому архітектура апаратного забезпечення та операційної системи приховувалася від цих програм. Це забезпечило функціонально прозору взаємодію програм відносно архітектури комп'ютера та технології мережі, тобто підвищилась їх інтероперабельність.

З розвитком інформаційних технологій змінювалися і підходи до створення програм. Створювалися програми за принципом компонентно-орієнтованого програмування. Компоненти характеризувалися принципом модульності, що дозволяло їх модифікувати, заміняти чи видаляти з системи без значних зусиль та фінансових витрат. Проте одним з основних недоліків компонентів є те, що вони є надто великими і все ще володіють багатьма функціями. А це означає, що для заміни чи модифікації однієї функції необхідно замінити весь компонент. Це привело до появи таких програмних компонентів як веб-сервіси, які і стали основою сучасної сервісно-орієнтованої архітектури.

Веб-сервіс – це функціональний компонент, можливості якого доступні для використання через Інтернет. Перевага веб-сервісів над іншими технологіями полягає в тому, що вони не прив'язані ні до однієї апаратної платформи, операційної системи чи мови програмування [29, 54, 68, 123-126]. У той час, як традиційні інформаційні ресурси зорієнтовані на пряму взаємодію з

людиною, веб-сервіси переважно спілкуються з людиною за допомогою спеціалізованих клієнтських додатків.

Для спілкування з клієнтськими додатками та між собою веб-сервіси використовують текстові повідомлення на основі технології XML. Веб-сервіси дають змогу створювати складні апаратно розподілені програмні комплекси для вирішення задач різного виду, потребуючи від розробника мінімум часу та зусиль.

Веб-сервіс є компонентом сервісно-орієнтованої архітектури. Сервісно-орієнтована архітектура – це підхід до створення програм, який базується на використанні розподілених, слабо пов’язаних між собою компонентів, що взаємодіють між собою за допомогою стандартизованих протоколів та інтерфейсів. Програмні комплекси такого роду, зазвичай, представлені набором веб-сервісів, що спілкуються між собою за допомогою протоколу SOAP. Цей протокол використовується для передачі повідомлень в форматі XML і може працювати поверх будь-яких протоколів прикладного рівня наприклад: SMTP, FTP, HTTP, HTTPS та інші.

Кожний веб-сервіс призначений для виконання однієї елементарної функції. Завдяки цьому забезпечується принцип модульності, що є дуже важливим при побудові розподілених систем, оскільки модифікація такої функції чи її заміна не вимагатимуть значних затрат зусиль та коштів та не вплинуть на роботу всієї системи. Програмування декількох веб-сервісів, кожний з яких реалізує певну функцію, дає змогу створити розподілену програму, яка володіє потрібною функціональністю. Більше того, така розподілена програма характеризується гнучкістю та масштабованістю завдяки можливості динамічного додавання функцій (веб-сервісів) до її логіки та можливості встановлення різних критеріїв вибору тої чи іншої функції при заданих умовах виконання. Такі розподілені програми називають композитними додатками.

Для розробки та надання будь якого типу композитного додатку доцільно використовувати розподілені обчислювальні системи на основі cloud-

технології. Такі системи володіють потужними обчислювальними ресурсами. Вони реалізовані у формі центрів обробки даних (ЦОД).

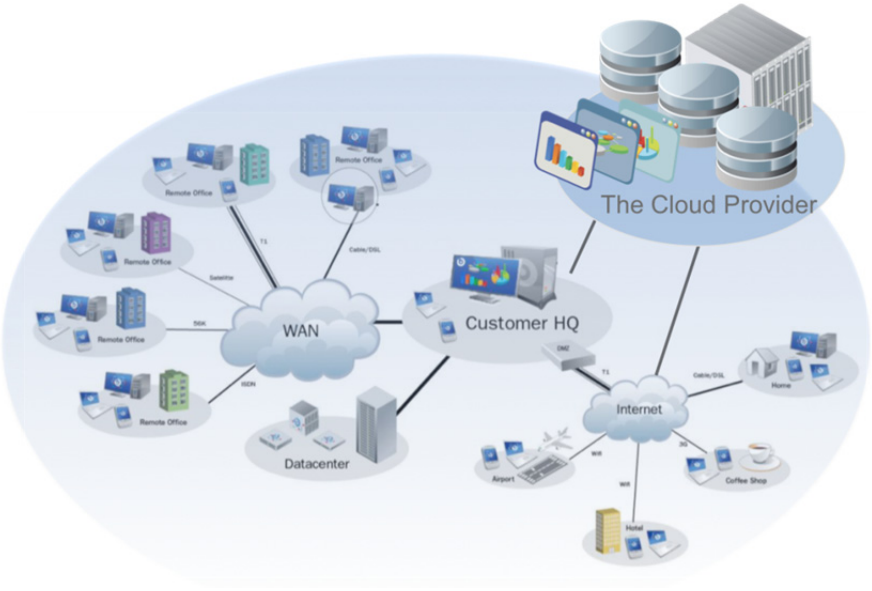


Рис. 1.2. Концепція сервісів Cloud-системи

Центр обробки даних – це цілий комплекс інженерних та IT-систем, який є невід'ємною частиною множини телекомунікаційних структур, він повинен забезпечити єдиний інформаційний ресурс з гарантованими рівнями достовірності, доступності та безпеки даних. В хмарних мережах центри обробки даних містять не тільки сервери зберігання даних, але й фізичні сервери, що здійснюють обробку запитів та надання сервісів. ЦОД забезпечують створення необхідної інфраструктури, ключові технології віртуалізації і спільне використання ресурсів (multi-tenancy). Віртуалізація дає можливість надавати доступ до мережевих ресурсів, як до віртуальних сегментів. Це означає, що пристрої або їх компоненти (наприклад, системи зберігання) надаються за запитом, незалежно від свого фізичного місця розташування і способу фізичного підключення до мережі.

На кожному такому сервері може міститися від однієї до кількох десятків віртуальних машин, які здатні обробляти та задовільняти відповідними компонентами чи додатками запити на надання сервісу. Однак логічна топологічна структура таких центрів обробки даних не завжди є стійкою і може змінюватися динамічно. Особливо, при міграції віртуальних машин з одного сервера на інший або й навіть на інший ЦОД. Під міграцією розуміється

можливість «вимкнути» VM в одному фізичному сервері, потім виконати, якщо цього не було зроблено заздалегідь, підготовчі операції, пов'язані з перенесенням набору даних, що відповідає цій VM, в інший фізичний сервер, і «ввімкнути» VM на іншому фізичному сервері, тобто здійснити її ініціалізацію з присвоєнням, найчастіше, іншої IP – адреси [30, 61, 75]. Перенаправлення запитів на інші логічні, а часом і фізичні канали, буде впливати на загальний час надання сервісу. Основною перевагою таких центрів є висока продуктивність, надійність, масштабованість, безпека, доступність, а також прозорість реалізації. На основі таких центрів обробки даних створюють різні моделі cloud систем, що дають змогу реалізувати розподілений сервіс будь-якої складності.

Існує декілька найпоширеніших моделей хмарних систем:

- *Платформа як послуга (PaaS)* - це надання інтегрованої платформи для розробки, тестування, розгортання і підтримки веб-додатків, як послуги, організована на основі концепції хмарних обчислень. Послуга доступна через мережу Інтернет. Наприклад, Google Apps надає застосунки для бізнесу в режимі онлайн, доступ до яких відбувається за допомогою Інтернет-браузера, тоді як ПЗ і дані зберігаються на серверах Google.
- *Програмне забезпечення як послуга (SaaS)* – це надання програмних сервісів (наприклад, сервіси Gmail), що працюють на основі обчислювальної хмари в оренду. Причому, користувачі використовують тільки ті функції, які їм потрібні (і, відповідно, сплачують за їх використання).
- *Інфраструктура як послуга (IaaS)* - це надання комп'ютерної інфраструктури (як правило у формі віртуалізації) як послуги на основі концепції хмарних обчислень. Найбільшими гравцями на ринку інфраструктури як послуги є Amazon, Microsoft, VMWare, Rackspace та Red Hat. Хоча деякі з них пропонують більше, ніж просто інфраструктуру, їх об’єднує мета продавати базові обчислювальні ресурси. IaaS складається з трьох основних компонентів:

- Апаратні засоби (сервери, системи зберігання даних, клієнтські системи, мережеве обладнання);
- Операційні системи та системне ПЗ (засоби віртуалізації, автоматизації, основні засоби управління ресурсами);
- Проміжне ПО (наприклад, для управління системами).

IaaS володіє рядом ключових характеристик. Серед яких:

- *Технології віртуалізації:* технології віртуалізації дозволяють вам взяти обладнання і розділити його обчислювальні потужності на частини, які відповідають поточним потребам, тим самим збільшуючи утилізацію наявних потужностей. В результаті можна перейти від придбання, управління і амортизації апаратних активів до придбання процесорного часу, дискового простору, мережевої пропускної здатності, необхідної для виконання задач;

- *Інтегровані системи управління:* у минулому для управління різними типами устаткування потрібно різне ПО управління. Віртуалізація дозволяє реалізувати весь набір функцій управління в одній інтегрованій платформі;

- *Можливість використання кращих архітектур і фреймворків:* для реалізації необхідної інфраструктури використовують готові інфраструктури, які реалізовані з врахуванням необхідного набору функцій.

Infrastructure as a Service (IaaS) дозволяє відмовитися від підтримки складних інфраструктур центрів обробки даних, клієнтських і мережевих інфраструктур, а також дозволяє зменшити пов'язані з цим капітальні витрати та поточні витрати. Робота такої моделі надання сервісів створює не тільки ряд переваг. Використання функцій віртуалізації та реплікації сервісів впливає на погіршення якості надання сервісів. Важливим аспектом у наданні cloud послуг на основі інфраструктури як сервісу є швидкість надання цих сервісів, наявність вільних каналів для їх надання та необхідної смуги пропускання для задоволення потреб користувачів. Під швидкістю надання сервісів розуміють забезпечення найменшого можливого часу надання сервісу, тобто зменшення часу обслуговування (обробки) запитів, які надходять на обслуговування до центру обробки даних. Стрімкий розвиток цих інфокомунікаційних мереж та

зміна мережі на рівні IaaS ставить все більші вимоги до якості надання послуг. Тому є актуальними питання щодо пришвидшення надання сервісів.

1.2. Аналіз моделей забезпечення якості обслуговування інформаційних потоків у телекомунікаційній мережі

Основною метою усіх телекомунікаційних мереж, в кінцевому результаті, є забезпечення заданних показників якості обслуговування (Quality of service, QoS). Підтримка якості обслуговування в сучасних мережах є досить трудомістким завданням і вимагає узгодженого розв'язання цілого комплексу задач управління та ефективного розподілу мережних ресурсів. У процесі забезпечення необхідних показників якості обслуговування та підвищення продуктивності телекомунікаційних систем (ТКС) необхідно:

- ведення та постійне оновлення єдиної бази даних щодо стану ТКС – її топології, завантаженості вузлів, трактів передачі та ін.;
- забезпечення високого рівня відмовостійкості мережі;
- реорганізації доступу до використання та збалансованого завантаження доступних мережних ресурсів;
- автоматизованого контролю параметрів трафіка користувачів у відповідності до укладеної угоди щодо якості обслуговування (Service Level Agreement, SLA);
- раціональної організації та адаптивної зміни стратегій маршрутизації трафіка;
- реконфігурації режимів роботи мережного обладнання, в тому числі настроювання механізмів пріоритетної обробки пакетів на всіх або частині мережних вузлів.

Відповідно до рекомендацій ITU-T E.800 якість обслуговування (QoS) – це певна інтегральна оцінка, яка визначає ступінь задоволеності користувача наданої йому постачальником послуг [31]. Це визначення уточнене в рекомендації E.860: "Якість обслуговування – ступінь відповідності

обслуговування, яке надається користувачеві постачальником та прописаний в угоді між ними" [32].

Фахівці компанії Cisco дали своє визначення терміну "якість обслуговування" – "Здатність мережі забезпечити необхідний сервіс заданому трафіку в певних технологічних рамках (Frame Relay, ATM, Ethernet й 802.1 мережі, SONET і IP мережі)". Відповідно до змісту RFC 2475 під сервісом варто розуміти набір характеристик передачі пакетів в одному напрямку одним або декількома мережними маршрутами. [33]

Будь який сервіс описується параметрами якості обслуговування серед яких:

1. Смуга пропускання (bandwidth) - описує номінальну пропускну спроможність середовища передачі інформації та визначає ширину каналу.
2. Затримка при передачі пакета (delay) - є сумарною величиною, що об’єднує в собі різновиди затримок: затримка вхідного інтерфейсу, затримка поширення, час очікування в черзі, затримка комутації (час передачі й обробки пакета), затримка формування трафіку, мережна затримка. Деталі виникнення кожної з перерахованих типів затримки наведено в табл.1.1

Таблиця 1.1

Типи затримок

Тип затримки	**Причина виникнення затримки**
Затримка вхідного інтерфейсу (serialization delay).	Час, необхідний для передачі пакету у фізичне середовище. Виникає на виході будь-якого фізичного інтерфейсу.
Затримка поширення (propagation delay).	Час, необхідний для передачі інформації на інший кінець каналу. Залежить від середовища поширення і відстані.

Продовження таблиці 1.1

Тип затримки	Причина виникнення затримки
Затримка в черзі (queuing delay). Величина непостійна	Час, витрачений пакетом на перебуванні в черзі в очікуванні подальшої передачі (вихідна черга) або в очікуванні можливості перетнути комутаційне поле (вхідна черга) (в очікуванні комутації)
Час пересилання або обробки (forwarding or processing delay)	Час, необхідний для прийняття вхідного пакета і його обробки, поки пакет не буде поставлений у чергу для подальшої передачі
Затримка, пов'язана з формуванням трафіку (shaping delay).	За умови здійснення формування трафіку, це час, на який пакети, що підлягають передачі, затримуються, щоб уникнути втрат пакетів у середовищі
Мережна затримка (network delay)	Затримка, внесена компонентами мережі провайдера послуг

3. Варіація затримки при передачі пакетів (jitter) - різниця у величині сумарної затримки при передачі різних пакетів того ж самого потоку. Параметр визначає максимальну затримку при наскрізному передаванні пакетів з кінця в кінець.
4. Втрати пакетів (packet loss) - визначає кількість пакетів, що відкидаються мережею під час передачі. Основними причинами втрат пакетів є перевантаження мережі й ушкодження пакетів під час передавання лінією зв'язку. Найчастіше відкидання пакетів відбувається з першої причини – у місцях перевантаження, де кількість пакетів, що надходять, набагато перевищує верхню межу розміру вихідної черги. Крім того, відкидання пакетів може викликатися недостатнім розміром вхідного буфера.

Вимірювання цих параметрів здійснюється протягом певного часового інтервалу, причому, чим цей інтервал менший, тим більш жорсткі вимоги ставляться до всіх елементів мережі. Забезпечення QoS при передачі "з кінця в кінець" вимагає взаємодії всіх вузлів на шляху пакетів трафіка й визначається

надійністю, функціональністю й продуктивністю "слабкої ланки". Для мереж із сервісно-орієнтованою архітектурою, особливо таких як cloud, ці параметри є надзвичайно важливими, оскільки надання віддаленого сервісу тісно пов’язано із взаємодією віртуалізованих частин системи при передачі усіх компонентів сервісу. Гранично допустимі значення параметрів QoS, згідно рекомендацій ITU-T, для різних типів хмарних сервісів наведені в таблиці 1.2

Таблиця 1.2

Гранично допустимі значення параметрів QoS

Тип сервісу	Параметри QoS				
	Час встановлення з'єднання, с	Смуга пропускання каналу, Мбіт/с	Ймовірність розриву з'єднання	Затримка, мс	Джитер, мс
IP-телефонія	0,5..1	до 085	10^{-3}	< 400	< 150
Відеодзвінки	0,5..1	0,512	10^{-3}	30..100	<30
Мережеве "радіо"	0,5..1	0,256	10^{-3}	< 1000	-
Відео за запитом	0,5..1	2..20	10^{-3}	30..100	<30
Передача данних	0,5..1	0,128..100	10^{-6}	50..1000	-
IP телебачення	0,5..1	0,512..5	10^{-6}	< 1000	-

QoS можна розглядати, як міру якості передачі і доступності сервісу в мережі. Виділяють три основні моделі забезпечення якості обслуговування [50, 70]:

1. Модель з негарантованою доставкою (Best Effort Service) - полягає в забезпеченні зв’язності вузлів мережі без гарантії доставки пакету адресату; при цьому відкидання пакету може відбутись при переповненні буферу вхідної або вихідної черги будь-якого комунікаційного вузла;
2. Модель інтегрованого обслуговування (Integrated Service) - реалізує метод резервування ресурсів та забезпечує наскрізну (End-to-End) якість обслуговування. В основі архітектури Int-Serv лежить протокол резервування ресурсів – RSVP (Resource ReSerVation Protocol);

3. Модель диференційованого обслуговування (Differentiated Service) - базується на методі пріоритезації навантаження. Клієнт може обрати потрібний рівень якості надання послуги шляхом встановлення відповідного значення поля коду диференційованої послуги (Differentiated Services Code Point – DSCP). Ідея архітектури з диференціацією сервісів полягає в мінімізації службового навантаження з метою виключення затримок.

Перевагою моделі IntServ є забезпечення чітко визначеної і гарантованої пропускної здатності. Однак збільшення часу встановлення з'єднання, неефективне резервування смуги пропускання, що заважає широкому використанню RSVP в пакетних мережах роблять таку модель неефективною. Проте, найбільший недолік IntServ пов'язаний з масштабованістю RSVP, особливо у високошвидкісних магістральних мережах, де обсяг ресурсів, які необхідні маршрутизатору для обробки й зберігання інформації RSVP, збільшується пропорційно кількості потоків QoS.

Перевагами моделі DiffServ є простота пріоритеризації трафіка, можливість масштабування, підвищена надійність. Все це визначає гнучкість та універсальність технології. Проте технологія має певні суттєві недоліки. Зокрема при передачі однорідного трафіка його пріоритизація стає не ефективною, адже при цьому мережа починає працювати в режимі Best Effort, а через вибіркове відкидання пакетів в періоди сплесків існує велика ймовірність відмови в обслуговуванні з'єднань з низьким пріоритетом.

Очевидно, що взаємна робота IntServ та DiffServ є оптимальним варіантом для надання необхідної якості QoS із кінця в кінець. У таких мережах, як cloud, використовується гарантована доставка сервісів користувачу з підтримкою необхідного рівня QoS, тому основним показником якості надання послуги для кінцевого користувача є затримка. Вибір параметра ґрунтується на рекомендаціях ITU-T Y.1540 [31]. При цьому він виділяється не лише, як основний критерій передачі трафіка реального часу, а як параметр, що найбільш повно відображає функціонування мережі. При цьому слабкі місця

однієї моделі будуть компенсуватися відповідними рішеннями іншої [34, 55, 69].

Час затримки передачі кожної із моделей QoS визначатиметься по-різному. Зокрема при роботі механізму IntServ враховується його поетапність, де процес передачі даних включає час передачі сигнального повідомлення Path - t_{path}; час передачі сигнального повідомлення Resv - t_{RESV}; час передачі блоку абонентських даних - t_{data}:

$$T_{IntServ} = t_{path} + t_{RESV} + t_{data} \tag{1.1}$$

Кожна складова часу може бути подана як сумарний час затримки на вузлах мережі - $t_{обр\_path}$, $t_{обр\_RESV}$, $t_{обр\_data}$ та час затримки передачі по лінії зв'язку повідомлень Path, Resv та даних відповідно - t_{line_path}, t_{line_RESV}, t_{line_data}:

$$t_{path} = t_{line\_path} + t_{обр\_path} \tag{1.2}$$

$$t_{RESV} = t_{line\_RESV} + t_{обр\_RESV} \tag{1.3}$$

$$t_{\text{data}} = t_{line\_data} + t_{обр\_data} \tag{1.4}$$

При використанні моделі DiffServ враховується, що процес передачі даних мережею $T_{DiffServ}$ включає проведення попередньої обробки (класифікація та маркування) трафіка на граничних вузлах мережі - t_{class_diff}, затримки в буферах вузлів - t_{delay_diff} та затримки в лінії - t_{line_diff}:

$$T_{DiffServ} = t_{line\_diff} + t_{delay\_diff} + t_{class\_diff} \tag{1.5}$$

Час обробки даних у вузлах мережі визначається затримками пакетів в буферах. Для розрахунку часу затримки на вузлах можна використати теорію дифузійної апроксимації [35, 73]:

$$t_{delay\_diff} = P \cdot \frac{t_s \cdot C_a^2 \cdot C_s^2}{2m \cdot (1-\rho)} \tag{1.6}$$

де m – розмір буфера вузла мережі;

ρ – навантаження системи трафіком;

P– ймовірність відмови в обслуговуванні через зайнятість приладів;

t_s – середній час обслуговування пакету мережевим пристроєм;

C_a^2, C_s^2 – квадратичні коефіцієнти варіації розподілу вхідного потоку та часу обслуговування відповідно.

Згідно з теорією телетрафіка, навантаження системи визначається:

$$\rho = \frac{\lambda}{\mu} \tag{1.7}$$

де λ – інтенсивність поступання абонентського трафіка;

μ – інтенсивність обслуговування трафіка пристроєм.

Ймовірність відмови в обслуговуванні залежить від завантаження вузлів мережі і може бути обчислена за другою формулою Ерланга. Квадратичні коефіцієнти варіації, відповідно до розподілу Парето, будуть визначатися [36, 62, 72]:

$$C_x^2 = \frac{(1-\alpha)^2\left(L^\alpha - k^\alpha\right)}{\alpha\left(L\cdot k^\alpha - L^\alpha \cdot k\right)^2} \cdot \left(\frac{L^2 \cdot k^\alpha - L^\alpha \cdot k^2}{(2-\alpha)} - \frac{\alpha\left(L\cdot k^\alpha - L^\alpha \cdot k\right)^2}{(1-\alpha)^2\left(L^\alpha - k^\alpha\right)} \right) \tag{1.8}$$

де α – коефіцієнт ваги розподілу;

L – максимальний розмір блоку даних;

k – мінімальний розмір блоку даних.

При цьому під розміром блоку даних розуміють розмір пакета, що генерується додатком або ж розміри груп пакетів, що виникають в результаті роботи додатку чи проходження пакетів по мережі.

1.3. Забезпечення параметрів QoS у мережах із сервісно-орієнтованою архітектурою

Зростання попиту на телекомунікаційні мережі висуває нові вимоги до мережевих технологій, які повинні забезпечити захищений, надійний, і, найважливіше, якісний доступ до інфокомунікаційних послуг, які надаються користувачеві. Відповідно до стрімкого зростання чисельності користувачів та значного розширення спектру надаваних послуг оператору зв'язку необхідно модернізувати власну мережу.

Розподілені обчислення є новою гілкою розвитку в IT сфері. Вони поєднують між собою набір технологій доступу та систем, які покликані забезпечити необхідні сервіси кінцевим користувачам. Основною складовою при розподілених обчисленнях є cloud системи, які за допомогою реплікації компонентів сервісу та великих обчислювальних ресурсів забезпечують виконання будь яких запитів на надання сервісу у найбільш короткі строки. Різноманітність таких запитів та їх обслуговування було б неможливим без функції віртуалізації. У хмарних системах віртуалізація є технологією, що дозволяє абстрагуватися від апаратних засобів до точки, де програмні стеки можуть бути розгорнуті і вони будуть задіяні, не будучи прив'язаними до конкретного фізичного сервера. Віртуалізація дозволяє створити динамічний центр обробки даних, де сервери забезпечують пул ресурсів, які будуть використані в разі потреби, і де додатки для обчислення, зберігання і мережеві ресурси, змінюватимуться динамічно з метою задоволення потреб. Користувач має доступ до власних даних, але не може управляти і не повинен піклуватися про інфраструктуру, операційну систему і власне програмне забезпечення, з яким він працює. Проте важливим аспектом у наданні cloud послуг є швидкість надання цих сервісів, наявність вільних каналів для їх надання та необхідної смуги пропускання для задоволення потреб користувачів. Під швидкістю надання сервісів розуміють забезпечення мінімального часу надання сервісу, тобто зменшення часу обслуговування (обробки) запитів, які надходять на обслуговування до центру обробки даних [37, 53, 67].

1.3.1. Особливості передавання інформаційних потоків у мережевих вузлах центрів обробки даних та їх вплив на параметри QoS

Звичайний центр обробки даних умовно має три основні рівні (згідно TIA/EIA-942)[38, 58]:

1. MDA /Main Distribution Area — головна розподільна підсистема, забезпечує інтерфейс доступу до ЦОД і розподіляє трафік головної магістралі по внутрішніх магістралях. Вона включає кінцеве обладнання операторів зв'язку, маршрутизатори, магістральні комутатори тощо;

2. HDA /Horizontal Distribution Area- горизонтальна розподільча підсистема, направляє трафіки внутрішніх магістралей по локальних лініях (довжиною не більше 100 м), що виходять в апаратні зони (стійки);
3. EDA/Equipment Distribution Area — підсистема розводки по обладнанню, що доставляє трафік в робочі області до серверів. Для обслуговування областей, де потрібні часті переконфігурації можуть використовуватися зонові сегменти з вузлами консолідації (ZDA).

Центри обробки даних таких мереж, як cloud дещо відрізняються. У їх структурі ці рівні є дещо не чіткими і взаємопоглинаючими. Характерними ознаками хмарного ЦОД є консолідація і віртуалізація серверів, наявність функціонального гіпервізора (оркестратора) і високий рівень автоматизації управління обчислювальної інфраструктурою [39, 64].

Архітектура центру обробки даних, як праивло будується за загальною тришаровою моделлю топології мережі доступу, агрегації і ядра мереж з можливими елементами мережі (комутатори і маршрутизатори). Розглянемо топологію, яку показано на рис.1.3. Сервери можуть бути підключені через 1 Гбіт лінії до початку стійки (Top of Rack TOR) комутатора, який, в свою чергу, підключається через одну або декілька 10 Гбіт ліній до агрегаційного кінця рядка (End of Row EOR) комутатора. Комутатор EOR використовується для підключення міжсерверів через стійки. Агрегаційні комутатори під'єднанні до комутатора ядра для підключення зовнішніх центрів обробки даних.

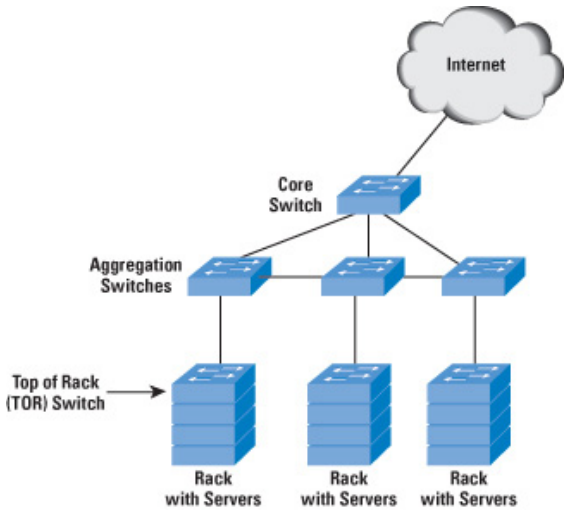


Рис. 1.3 Приклад архітектури мережевого комутатора центру обробки даних

Хмарний ЦОД логічно складається з п’яти основних рівнів (рис.1.4):

- рівня агрегації;
- рівня доступу;
- рівня додатків;
- рівня зберігання даних;
- рівня отпичних каналів

З точки зору логічної топології, «зовнішні» сервери головної розподільчої підсистеми логічно відокремлені від серверів додатків підсистеми HDA, які, в свою чергу відокремлені, від серверів підсистеми EDA [40, 51]. Трафік передається спочатку від клієнта до «зовнішнього» сервера, потім від «зовнішнього» сервера до сервера додатків і , нарешті, від сервера додатків до сервера бази даних. Логічне розділення має на увазі, що кожен рівень є особливою функціональною зоною і має свої логічні канали.

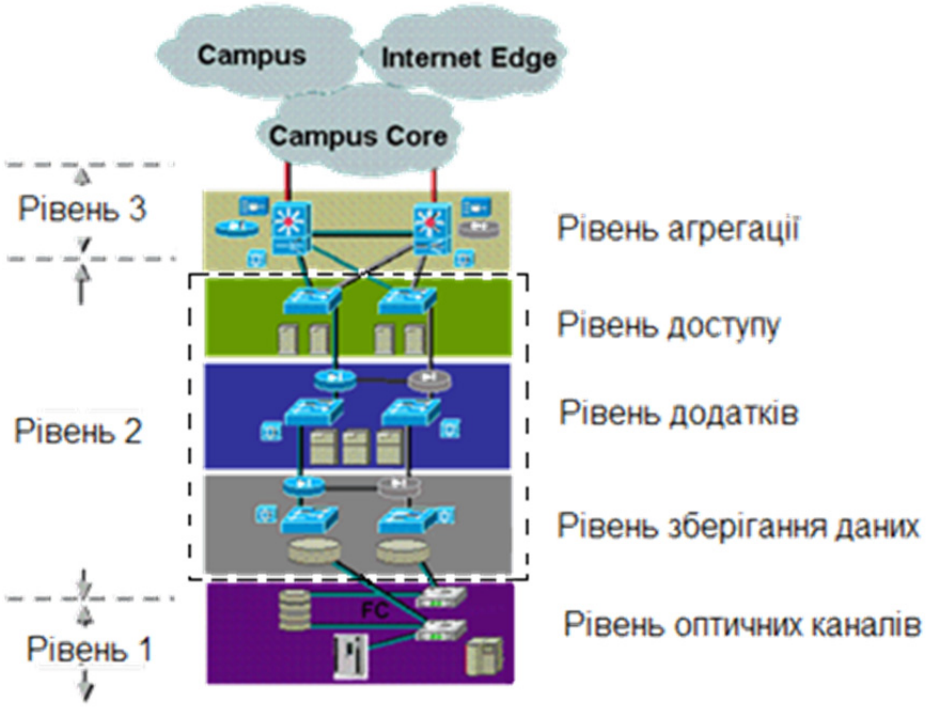


Рис. 1.4 Логічна топологія мережі ЦОД

Запит на надання сервісу по фізичним каналам передається на рівень агрегації, де система управління здійснює пошук і виділення необхідних ресурсів для його обслуговування. Під ресурсами розуміється наявність вільних фізичних серверів з необхідним користувачеві ПЗ. На рівні доступу на основі

даних про вільні фізичні та логічні канали система виділення ресурсів знаходить необхідні ресурси і здійснюється опрацювання запитів на доступ до необхідного сервісу. Між рівнем доступу та рівнем додатків за допомогою алгоритму маршрутизації здійснюється виділення і передача по фізичних каналах, а також запускається алгоритм пошуку логічних каналів для доступу до віртуальних машин. Слід зазначити, що на одній фізичній машині може міститися до декількох десятків віртуальних машин, які мають здатність виконувати лише один тип програмного комплексу. Доступ та пошук оптимального як фізичного, так і логічного шляху передачі до програмних комплексів на віртуальних машинах для надання сервісу здійснюється за допомогою алгоритму мінімального зв’язного дерева. Однак такий алгоритм здійснює автоматичне блокування надлишкових в цей час зв'язків для повної зв'язності портів, і, як наслідок, не може забезпечити належної якості сервісу. Крім того, таке блокування зв’язків може призвести до великої кількості втрачених запитів, а велика надлишковість службової інформації призводить не лише до завантаженості каналів, а й до збільшення часу надання сервісу. Вибір маршрутів повинен враховувати виділений ресурс у фізичній структурі та співставляти його з вимогами потоків, тобто контролювати завантаженість. Якщо даної вимоги не дотримуватись, то виникає неефективне використання доступного ресурсу, або погіршення якості сервісу для потоків. Метрика даного алгоритму враховує завантаженість лише до «центральних» (на які здійснюється найбільша кількість запитів) фізичних машин і не аналізує структуру інших з’єднань, тобто не має здатності щоразу аналізувати логічну топологію між віртуальними машинами. Такий аналіз стає особливо необхідним при міграції віртуалізованих частин центру оброки даних з одного сервера на інший. Перенаправлення запитів на інші логічні, а й часом і фізичні канали, які впливатимуть на загальний час надання сервісу. Тому така «нестійка» структура ЦОД вносить затримки при обслуговуванні запитів на надання сервісу, і, як наслідок, призведе до погіршення якості обслуговування. Отож, постає задача зменшення часу обробки запитів, які надходять на обслуговування до центру обробки даних, з врахуванням топологічної

структури такого центру. Однак врахування структури ЦОД недостатньо: важливим фактором при цьому виступає живучість такої структури, адже чим стійкіша структура, тим система швидше виконує та перенаправляє користувачу необхідний сервіс.

Дослідженням даної проблеми займалися багато вчених. Так у роботі [30] проведена оцінка і аналіз живучості «типових» структур (зірка, кільце), в яких процес появи нових вузлів не здійснює значного впливу на процес надання сервісу кінцевому користувачу. У роботі [36] запропонована модель для оцінки ефективності високо віртуалізованого хмарного центру з Пуасонівським розподілом надходження задач та звичайним розподілом розміру задач. Модель базується на двоступеневій техніці апроксимації, де основний немарківський процес спочатку моделюється, як вбудований напів марківський процес, який, в свою чергу, потім моделюється, як апроксимований процес Маркова, але тільки в моменти надходження надзадач. Однак, дана модель не передбачає зміни положення віртуальних машин і, як наслідок, їх вплив на час надання сервісу кінцевому користувачу. Навіть при великому різноманітті моделей надання сервісів, доступність до компонентів сервісу в залежності від стійкості топологічної структури мережі все одно залишається пріоритетною задачею. Зокрема, дослідження доступності фізичних серверів в умовах динамічного розгортання віртуальних машин та зміни їх місця розташування проводиться вченими Hamzeh Khazaei, Jelena Mišić, Vojislav B. Mišić та Nasim Beigi-Mohammadi [37]. Ще одна робота, авторами якої є Adam Grzech та Paweł Świątek, присвячена дослідженню доступності складних програм на основі сервісно-орієнтованої архітектури. В цій роботі проведено дослідження структури складних програм та способу її оптимізації для підвищення доступності [41]. Однак, навіть при такій кількості досліджень в жодній із робіт не проведено паралелей і взаємозв’язку між стійкістю динамічно змінних топологій та якістю надання сервісів користувачам.

1.3.2. Переваги та недоліки існуючих методів балансування навантаження

Головною характеристикою мереж із сервісно-орієнтованою архітектурою є їх здатність до розподілу навантаження по великій кількості пулів ресурсів. Всі види додатків мають можливість отримати достатню кількість обчислювальної потужності, дискового простору та інформаційних послуг [1, 52, 66]. Проте, спільне використання ресурсів призводить до цілої низки проблем: велика кількість одночасних запитів на обслуговування до сервера може призвести до його зупинки, в той час як інші сервери досі простоюють. Внаслідок такого розбалансування системи збільшується час надання сервісів кінцевим користувачам, і як наслідок, погіршується рівень якості обслуговування.

Балансування навантаження сприяє підвищенню продуктивності розподіленої системи в аспекті розподілу інформаційних потоків між множиною взаємодіючих хостів [11, 60, 63]. Така система або прагне рівномірно розподілити навантаження на кожен хост і мати дуже малі відхилення від робочого навантаження на всіх інших фізичних хостах, або забезпечує уникнення перевантажень і блокувань на окремих серверах.

Проблема оцінки доступних параметрів в центрах обробки даних досліджується не тільки вітчизняними, а й іноземними дослідниками. Однак, навіть при великому різноманітті моделей надання сервісів, доступність до компонентів сервісу все одно залишається пріоритетною задачею. Над проблемою доступності компонентів на різних рівнях cloud системи працюють багато вчених.

Так у роботі [44] запропонований алгоритм LBVM дозволяє розділяти навантаження на віртуальні машини між фізичними вузлами по заздалегідь визначених кластерах. Навантаження на кожному сервері періодично реєструється, при чому з різними мітками, і кожні кілька хвилин проводиться запуск централізованого алгоритму, який відслідковує чи відбулася міграція VM чи ні. Коли кількість фізичних хостів збільшується, цей алгоритм буде вузьким місцем системи.

Ще одна робота, авторами якої є Fel Ma, Feng Liu та Zhen Liu присвячена дослідженню доступності складних програм на основі сервісно-орієнтованої архітектури. В цій роботі розроблена модель балансування навантаження з розподілом віртуальних машин в центрі обробки даних з використанням методу TOPSIS. Результати показують, що система може досягти більшого балансування навантаження в великомасштабному середовищі хмарних обчислень з меншою кількістю міграцій віртуальних машин. Автори роботи [2] запропонували свій підхід до розподілу навантаження у великомасштабних розподілених файлових системах (DFS). У цій роботі, алгоритм балансування працює так, щоб найбільш завантажений вузол перевіряв наявність репліки в найменш завантаженому вузлі та здійснював перенаправлення частини запитів до такого вузла. [45]. Автори роботи [46] запропонували генетичний алгоритм (GA) по стратегії балансування навантаження. Цей метод оптимізації балансує навантаження cloud середовища, мінімізуючи робочий цикл сервісу (сумарну тривалість обслуговування запиту всіма компонентами сервісу). Таким чином GA забезпечує необхідний рівень QoS та ефективно використовує ресурси кожного фізичного сервера. Dhinesh Babu L.D у своїй роботі [47] запропонував новий алгоритм балансування навантаження різнотипних запитів. Запропонована методика досягає справедливого балансу навантаження між віртуальними машинами, таким чином, що практично досягається максимальна пропускна здатність. Причому, обслуговування запитів проводиться згідно їх пріоритетів так, що час очікування обробленого запиту зведений до мінімуму. Автор використовує різні алгоритми планування надходження задач, щоб отримати найкращу продуктивність. Такий алгоритм балансування навантаження на основі планування надходження задач та їх пріоритезації дозволяє зменшити час відгуку системи на обробку запитів. Проте, запропоновані методи базуються на засадах дистанційно-векторної маршрутизації і не враховують динамічність зміни структур дата центрів.

Збалансований розподіл навантаження є основним завданням в області хмарних обчислень. Він дозволяє забезпечити оптимальне використання ресурсів системи та здійснювати аналіз функціональної доступності кожного

окремого елемента. Дослідження доступності фізичних серверів в умовах динамічного розгортання віртуальних машин проводиться вченими Joao Soares, Miguel Dias, Jorge Carapinha, Bruno Parreira, Susana Sargento Carapinha [48]. Їх робота присвячена розробці платформи Cloud4NFV з використанням функцій VNF. Ця платформа включає в себе моделювання VNF, як інфраструктурних ресурсів. Однак дана платформа та алгоритм на основі якої вона працює, не враховує доступних вільних апаратних чи програмних ресурсів кожної фізичної машини. Робота [49] пропонує свого роду реалізацію адаптивної динамічної міграції віртуальних машин. Пропонується алгоритм розподіленого балансування навантаження на вибірковій основі, що дозволяє відслідковувати міграцію між VLAN для спрощення площини управління та зменшення часу реконфігурації мережі ЦОД. Реалізується проста модель, яка зменшує час міграції віртуальних машин між центрами обробки даних і виконує переміщення віртуальних машин шляхом перетворення їх в Red Hat Cluster послугу. Однак реалізація такого алгоритму балансування навантаження потребує значних затрат апаратних ресурсів та не оцінює стану інших фізичних машин.

Саме тому необхідна розробка підходу, який дозволить оцінити показники доступних ресурсів та дасть змогу підвищити ефективність балансування навантаження, що призведе до зменшення затримки при наданні сервісів та забезпечить необхідний рівень QoS.

1.4. Висновки до 1-го розділу

1. Проведено аналіз моделей надання сервісів у мережах із сервісно-орієнтованою архітектурою. Встановлено, що одним із ключових способів побудови інфраструктури та взаємодії у таких мережах є «клієнт-серверна» модель, що дозволяє не здійснювати прив’язки сервісних програм, які надаються у вигляді композитного веб-сервісу, до архітектури апаратного забезпечення. Кожний веб-сервіс призначений для виконання однієї елементарної функції. Завдяки цьому забезпечується принцип модульності, що є дуже важливим при побудові розподілених систем, оскільки модифікація

такої функції чи її заміна не вимагатимуть значних затрат зусиль та коштів та не вплинуть на роботу всієї системи. Логічно, що для розробки та надання будь якого типу композитного додатку доцільно використовувати розподілені обчислювальні системи на основі cloud-технології. В результаті аналізу обґрунтовано, що найважливішим компонентом у cloud та сервісно-орієнтованих мережах є центр обробки даних, який забезпечує обробку та створення необхідної інфраструктури, ключових технологій віртуалізації і спільне використання ресурсів (multi-tenancy). Від його роботи напряму залежить підтримка та забезпечення відповідного рівня якості обслуговування. Використання функцій віртуалізації та реплікації сервісів впливає на погіршення якості надання сервісів. Важливим аспектом у наданні cloud послуг на основі інфраструктури як сервісу є швидкість надання цих сервісів, наявність вільних каналів для їх надання та необхідної смуги пропускання для задоволення потреб користувачів.

2. В роботі проаналізовано моделі забезпечення якості обслуговування інформаційних потоків у телекомунікаційній мережі. Досліджено основні параметри, що характеризують QoS в мережах із сервісно-орієнтованою архітектурою, які є базовими при наданні сервісів користувачам cloud мережі та технологій за допомогою яких в сучасних пакетних мережах реалізуються методи забезпечення гарантованої якості обслуговування. Встановлено, що взаємна робота IntServ та DiffServ моделей є оптимальним варіантом для надання необхідної якості QoS при передачі запитів із кінця в кінець. Така «гібридна» модель обслуговування дозволить забезпечити гарантовану доставку сервісів з підтримкою необхідного рівня QoS. За основний критерій якості обслуговування обрано параметр часу затримки передачі, що ґрунтується на рекомендаціях ITU-T Y.1540. При цьому, він виділяється не лише, як основний критерій передачі трафіка реального часу, а як параметр, що найбільш повно відображає функціонування мережі.

3. Розглянуто основні недоліки та переваги при застосуванні існуючих методів забезпечення параметрів QoS у cloud мережах. Сформульовано основні невирішені задачі в галузі забезпечення якості обслуговування в таких

телекомунікаційних мережах. Зростання потреби у інформайно-комунікцаійних застосуваннях реального часу, зокрема в рамках реалізації концепції IoT, формує вимоги щодо зниження затримки наскрізного передавання інформації з одночасним підвищенням стійкості віртуальних топологій ЦОД, які утворюються дистанційо-векторними методами за умов різкого зростання динаміки потоків у сучасних гетерогенних мережах. Розв’язання даного протиріччя можливе шляхом підвищення ефективності балансування навантаження в мережах на основі сервісно-орієнтованих архітектур.

РОЗДІЛ 2.
МЕТОДИ ПОКРАЩЕННЯ ПАРАМЕТРІВ ЯКОСТІ НАДАННЯ ПОСЛУГ В МЕРЕЖАХ ІЗ СЕРВІСНО-ОРІЄНТОВАНОЮ АРХІТЕКТУРОЮ

В розділі запропоновано моделі, методи та алгоритми підвищення параметрів якості надання послуг в мережах із сервісно-орієнтованою архітектурою. Розвинуто спосіб пошуку шляху передачі за критерієм мінімальної затримки для центрів обробки даних. Запропоновано модель надання сервісу на основі методу пошуку маршруту з урахуванням стійкості структури віртуалізованого центру обробки даних . Розроблено інтегровану архітектуру системи управління ресурсами з використанням функцій NVF. Розроблено метод балансування навантаження на основі формалізації ї ресурсів ЦОД. Наведені у розділі результати опубліковано у працях [2-4, 7, 8, 12-13, 15, 17, 19-20, 24].

2.1. Топологічно-динамічний пошук шляху за критерієм мінімальної затримки для центрів обробки даних

Нехай сервіс є об’єктом для надання послуги клієнтам сервісно-орієнтованої мережі, основною характеристикою якого є тривалість обробки запиту $t_{обр}$. Розглянемо атомарний сервіс, тобто сервіс, що реалізований однією програмою, яка встановлена на одній віртуальній машині. Припустимо, що користувач надсилає запит на надання сервісу до хмарного центру обробки даних [3, 13].

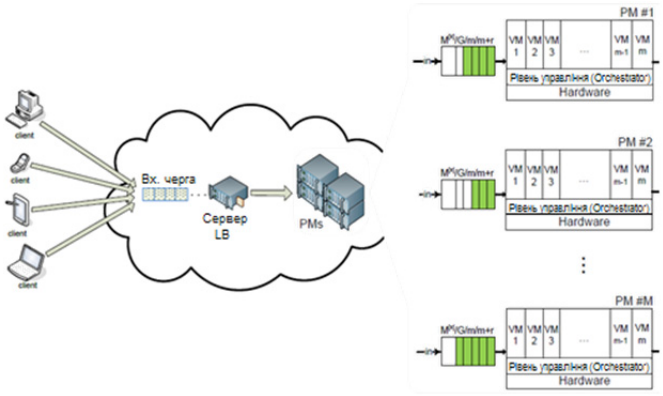


Рис. 2.1. Принцип надходження запитів до хмарного центру обробки даних.

ЦОД ініціалізує виділення фізичного сервера (PM), який містить необхідний тип сервісу або додатку. При надходженні декількох запитів на одну і ту ж PM продуктивність таких сервісів зменшується, оскільки доступ сервісів до ресурсів фізичної машини відбувається за методом часового розділення. Як наслідок, при збільшенні кількості сервісів на одній PM та при збільшенні інтенсивності надходження запитів на дану PM, а також при збільшенні інтенсивності надходження запитів на кожен сервіс збільшується тривалість обслуговування запитів кожним сервісом. У випадку зменшення продуктивності сервісу, система керування переносить цей сервіс на іншу VM чи PM. Логічна топологія не змінюється, але дані проходять по інших фізичних каналах. В такому випадку загальний час передачі сервісу від користувача до ЦОД і в зворотньому напрямку розраховуватиметься:

$$t_{пер} = \sum_{1}^{n} t_{комут.} + \sum_{1}^{n-1} t_{п.к.з.} + t_{обр.} \qquad (2.1)$$

де n – кількість запитів; $t_{комут.}$ - час проходження запиту через систему комутації; $t_{п.к.з.}$ – час пошуку каналів, по яких буде здійснюватися передача; $t_{обр.}$ – час обробки запиту, який є сумою часів обробки запиту сервісом, який складається *з* k атомарних сервісів:

$$t_{обр.} = \sum_{1}^{k} t_{а.с.} \qquad (2.2)$$

Оскільки оптимальний шлях передачі змінюється, то це призводить до збільшення часу пошуку каналів, по яких буде здійснюватися передача - $t_{п.к.з.}$, що в свою чергу призведе до збільшення загального часу передачі і виникнення затримки (рис.2.2, 2.4).

Для вирішення даної проблеми пропонуємо здійснювати пошук шляху за критерієм мінімального часу проходження [4, 7]. В основі цього методу лежить спосіб розрахунку оптимального шляху передачі на основі даних про поширення інформації та зміни в топології мережі. Ці дані складають єдину метрику маршруту, яка виявляє компроміс між вибором оптимального маршруту та властивостями трафіку, оскільки, імовірність одночасного

існування потоків із максимальним сервісом на маршрутах із спільною лінією є малою. Оптимальність шляху забезпечується найкращим показником загальної метрики, при цьому потенційний об'єм доступних ресурсів є набагато більший від необхідного [56, 65, 117-119].

Метрика представляється у вигляді модифікованої метрики протоколу EIGRP, яка враховує поточну затримку на передачу кожної компоненти сервісу на інтерфейсах мережевих вузлів:

$$M = \left(K_1 * \min C + \left(\frac{K_2 * \min C}{256 - L} \right) + K_3 * D \right) * \left(\frac{K_5}{R + K_4} \right) * 256 \tag{2.3}$$

де K_i – коефіцієнти, які задає адміністратор мережі для коригування композитної метрики; *min C* – мінімальне значення пропускної здатності на шляху проходження оптимального маршруту, по якому будуть надсилатися дані; L – завантаженість кожної ланки в мережі; D – сумарна затримка на інтерфейсах; R – надійність шляху.

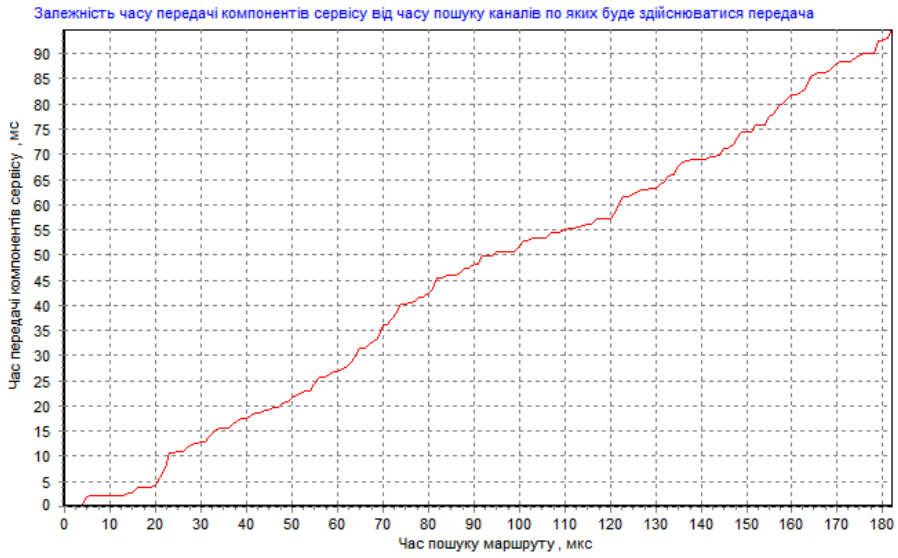


Рис. 2.2 Залежність часу надання сервісу від часу пошуку каналу, по якому буде здійснюватися передавання

Дана метрика не передається між маршрутизаторами. Вона вираховується локально на маршрутизаторі, і існує тільки на ньому. Далі маршрутизатор передає тільки змінені параметри та зберігає дані про структуру зв'язків зі всіма своїми сусідами в таблиці, що дозволяє швидко знаходити альтернативні маршрути в разі відключення основних. Якщо відповідний маршрут не

знайдений, маршрутизатор опитує сусідів про наявність альтернативних маршрутів.

Припустимо, що деякий інтерфейс на виході, при передачі інформації від фізичного сервера А до віртуальної машини, має значення затримки t, при пороговому значенні затримки T, тобто $t < T$ [12, 17]. Віртуальна машина з цілим програмним комплексом в результаті впливу різних факторів мігрує на фізичний сервер В (рис. 2.3).

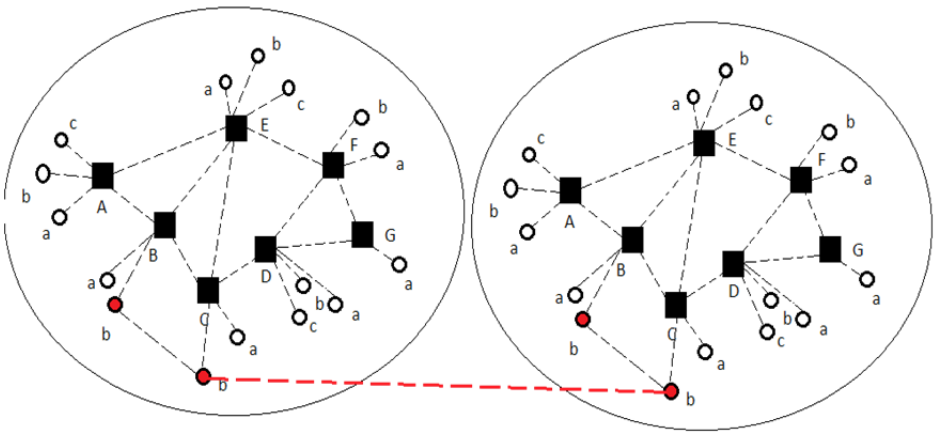


Рис.2.3.Принцип міграції компонентів сервісу

При передаванні запиту на надання сервісу в динамічно-змінній структурі аналізується наявність логічного маршруту, для якого значення затримки на передачу кожної компоненти сервісу на інтерфейсах мережевих вузлів t не перевищує встановлене порогове значення T. Якщо, внаслідок порушення стійкості структури такий маршрут відсутній - відбувається перерахунок комбінованої метрики, відповідно до значень затримки та встановлюється вже новий маршрут для передавання запитів. Це дозволяє більш суворо контролювати тривалість пошуку каналів для передавання та, відповідно, підтримувати належний рівень QoS. Даний метод дозволяє зменшити час на пошуки маршруту, не втрачаючи при цьому запитів та не збільшуючи загального часу їх передавання, оскільки адаптивно враховується зміна логічної топологічної структури мережної платформи.

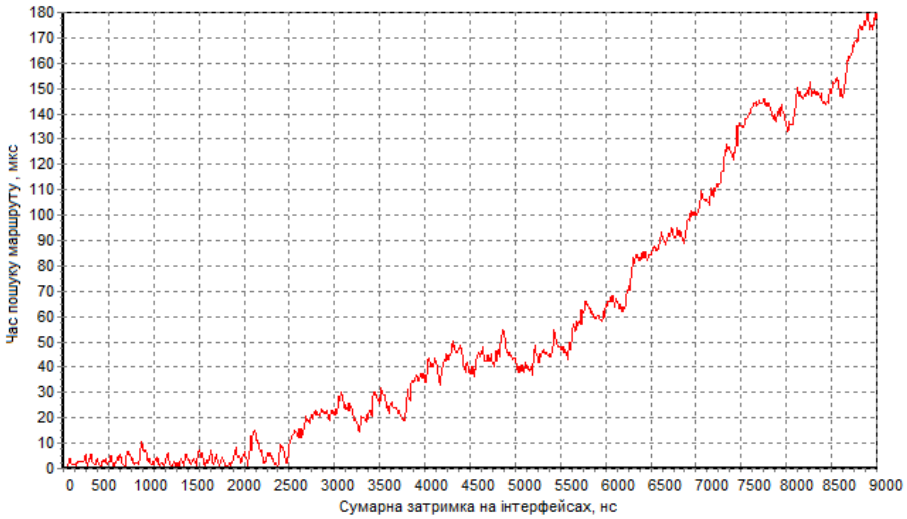


Рис. 2.4 Залежність часу пошуку оптимального маршруту від сумарної затримки на інтерфейсах

2.2. Модель надання сервісу на основі методу пошуку маршруту з урахуванням стійкості структури віртуалізованого центру обробки даних

Для повноцінної роботи та забезпечення необхідного рівня якості обслуговування інформаційна система повинна мати цілком певний запас стійкості до зовнішніх дестабілізуючих впливів із зовнішнього середовища [59, 60, 63]. Як на всю інформаційну систему, так і на її окремі елементи можуть впливати різні дестабілізуючі фактори такі як атаки, видалення окремих матеріалів з веб-сайтів мережі Інтернет, знищення або відключення інформаційних серверів, публікація об’єктів (сервісів, додатків, компонентів), які в деяким чином конфігурують вихідну інформаційну систему, або народження нової інформаційної системи, яка може знизити актуальність або просто знищити вихідну систему.

Кількісні показники живучості істотно залежать від параметрів, що визначають умови працездатності інформаційної системи. Поточний рівень працездатності визначає кількість, якість і зміст функцій, які узагальнюються поняттям «мета функціонування системи». Для забезпечення мети функціонування системи можна застосувати одну із стратегій [76, 84, 90, 121]:

- F-стратегії - стратегію забезпечення відмовостійкості (Fault-tolerance);
- S-стратегії - стратегію забезпечення живучості (survivability).

У процесі формування F-стратегії необхідно визначити множину станів системи $S^{(f)} = \{s_v^{(f)}\}$, при яких необхідно протидіяти загрозам порушення працездатності, задавати варіанти розподілу функцій між працездатними компонентами інформаційної системами серед множини станів *S (f)*.

Стратегія забезпечення відмовостійкості орієнтована на повну компенсацію передбачених функціональних відмов і забезпечення показників ефективності функціонування систем.

У процесі формування S -стратегії для кожного стану із множини станів *S (f)* необхідно додатково напрацювати рішення, які стосуються функцій системи: звужувати або ні множину функцій, які разом складають мету функціонування; як це зробити; спрощувати або ні алгоритм реалізації функцій і т.д.

У процесі аналізу і оцінки функціональної живучості інформаційної системи допускається, що можна забезпечити необхідні зв'язки між окремими функціональними компонентами. Що стосується оцінки стійкості у таких інформаційних мережах, як мережі центрів обробки даних, то тут необхідно забезпечити якомога більшу стійкість зв’язків між кожним із компонентів сервісу. Під стійкістю зв’язків розуміється стійкість топологічних структур при передачі чи міграції компонентів, тобто щоб існував як фізичний, так і логічний маршрут при реплікації компонентів [13, 77, 85, 98, 122].

Позначимо множину компонентів сервісу (інформаційних функцій), що надаються системою ЦОД через $F = \bigcup_{k \in I} F_k = \{f_1, f_2, ..., f_n\}$, причому кожна компонента F_k потенційно може виконати множину функцій $\varphi_n : \{1,2,...,p\} \to P(F)$, де $P(F)$ - множина всіх підмножин *F*. Якщо $\varphi_n(k) = \{f_{k_1}, f_{k_2}, ..., f_{k_j}\}, 1 \le k_m \le n, 1 \le m \le j$, то функціональна компонента F_k може виконати функції $f_{k_1}, f_{k_2}, ..., f_{k_j}$.

В кожен конкретний момент часу F_k виконує певну множину функцій, що визначаються $\varphi_t : \{1,2,...,p\} \to P(F)$. Якщо $\varphi_t(k) = \{f_{k_1}, f_{k_2}, ..., f_{k_j}\}, 1 \le k_m \le n$, то

компонента F_k може виконувати функції $f_{k_1}, f_{k_2},..., f_{k_j}$. Коли $\varphi_t(k) = \varnothing$, компонента F_k є недоступною.

Кожна $f_{k_m} \in F, m = \overline{1, j}$ характеризується ефективністю виконання або кількістю опрацьованих запитів на компоненту c_{k_m}. Ефективність виконання визначається як $\varphi_{ef} : F \times \{1,2,...., p\} \times P(F) \to C$, де C – деяка числова множина $\varphi_{ef}(f_{k_m}, k, \varphi_t(k)) = c_{k_m}$ означає, що функціональна компонента F_k виконує функції $\varphi_t(k) = \{f_{k_1}, f_{k_2},..., f_{k_j}\}$, тоді ефективність виконання $f_{k_m} \in \{f_{k_1}, f_{k_2},..., f_{k_j}\}$ рівна c_{k_m}. Тоді умова доступності надання компоненти (тобто ефективного опрацювання запитів на компоненту) матиме вигляд:

$$\begin{gathered} \bigcup_{k=1}^{P} \varphi_n(k) \supseteq F \\ \varphi_t(k) \subseteq \varphi_n(k) \ \ \forall k = \overline{1, p} \\ \sum_{k=1}^{p} \varphi_{ef}(f_{k_m}, k, \varphi_t(k)) \geq c_{k_m}, \ \ \forall k_m = \overline{1, n} \end{gathered} \tag{2.4}$$

Якщо компоненти F_k є функціонально однорідними, а мета забезпечення відповідних параметрів QoS, що забезпечується завдяки наявності відповідної кількості доступних функціональних компонент:

$$R(F_k, t) \geq R^* = const$$

де $R(F_k, t)$ - середня кількість доступних компонент сервісу в момент часу $t \geq 0$, R^* - мінімально допустима кількість доступних компонент, при якій працездатність системи і рівень якості обслуговування не менше певного порогового мінімуму, то оцінка функціональної живучості системи буде функцією:

$$N(F_k, t) = \frac{\overline{\Omega}(F_k, t)}{N_\omega} \tag{2.5}$$

де $\overline{\Omega}(F_k, t)$ - математичне очікування працездатності системи в момент часу $t \geq 0$; N_ω - сумарна доступність всіх компонентів сервісу.

Структурна живучість розглядається, як можливість реконфігурацій мережі, які дозволять створити структуру, що забезпечує виконання критичного рівня підмножини функцій мережею.

При розгляді структурної живучості враховується топологія мережі, міжкомпонентні зв'язкі і доступність компонент. Завдання, пов'язані з аналізом структурної живучості, можна звести до завдань надійності, стійкості та зв'язності топологічних структур [83, 86, 94].

Аналіз структурної живучості вимагає визначення:

- Структури для надання сервісів в деякий момент часу, коли виникають небажані впливи на систему;

- Вимог до окремих видів ресурсів системи і їх взаємозв'язку;

- Вимог до функціональних можливостей компонент сервісу та системи в цілому.

Структурну живучість системи можна оцінювати при деяких припущеннях, які дозволяють спростити задачу оцінки та звести її до задачі аналізу зв'язності графів, оцінки ймовірності формування стійкості структури в разі небажаних впливів і т.п.

При розгляді інформаційної системи необхідно також враховувати кількість структур в цій системі, які можуть виконувати критичну кількість інформаційних функцій. Для підрахунку цієї кількості необхідно виділяти такі структури, тобто переходячи на мову теорій графів і складних мереж - знаходити в мережевих структурах взаємопов’язаність компонент, яка повинна базуватися на оцінці складності побудови мереж, їх максимальних і мінімальних перетинах, потоках в цих мережах, доступних ресурсах і т.д.

При дослідженні структурної живучості за допомогою графових моделей сукупність компонентів сервісів у системі $G = (V, R)$ представляють у вигляді вершин графа $v \in V$, а ребра графа $r \in R$ відповідають зв'язкам між ними. Систему, яка моделюється за допомогою графа, вважають зруйнованою, якщо в разі видалення вершин або ребра граф буде виконуватися хоча б одна із умов:

- Граф складається мінімум з двох компонентів;

- Не існує шляхів для певних множин вершин;

- Кількість вершин в найбільшій компоненті графа $G = (V, R)$ менше деякого заздалегідь заданого числа;

- Найкоротший шлях перевищує деяку задану величину.

Відповідно, система вважається живучою, якщо ці умови не виконуються. Структурну живучість будь яких систем зазвичай характеризують різними показниками зв'язності. Розрахунок таких показників, як, наприклад, ймовірність зв'язності за умов випадкового існування ребер графа, на практиці обмежується обчислювальною складністю таких завдань. Одночасно, використовуючи зв’язки графа, що моделюють систему, можна отримати досить прості граничні оцінки необхідних показників.

Структуру системи хмарних обчислень можна представити у вигляді графа. Проте, з врахуванням відключення або міграції віртуальних машин такий граф буде не детермінований, і математично його можна описати лише за допомогою теорії випадкових графів (моделями Балобаші-Альберта чи Ердеша-Реньї [78]).

Нехай структура центру обробки даних представлена у вигляді графа $G = (V, R)$, який має випадкову кількість вузлів (під вузлами розуміється VM) та ребер R (фізичних чи логічних каналів), причому $R = \{(r, s)\}$ - множина логічних каналів зв'язку між компонентами сервісів. Будь який канал зв’язку (r, s) характеризується своєю довжиною - l_{rs} (км), пропускною спроможністю - μ_{rs} (біт / с, пакет / с), вартістю - $C_{rs}(l_{rs}, \mu_{rs})$ та потоками f_{rs}, причому $f_{rs} \le \mu_{rs}$ $\forall (r, s) \in E$. Кожній вершині V характерні значення $w(V)$ і $\overline{w}(V)$, що відображають поточне і граничне завантаження елемента системи, які залежать від інтенсивності потоків запитів λ. У разі, коли поточне завантаження $w(V)$ елемента системи досягає граничного значення $\overline{w}(V)$ елементи системи виходять з ладу, або ж відбувається їх міграція на менш завантажену фізичну машину (PM). Потоки запитів на надання компонентів сервісу, що проходять через цю вершину перерозподіляються по «сусіднім» елементам системи. Вихід

з ладу елемента системи в теоретико-графовій термінології відповідає видаленню з графа системи вершини з інцидентними їй ребрами.

Матриця якості потоків, які передаються між довільними вершинами *i* та *j* буде представлена:

$$H = \left\| h_{ij} \right\|, i, j = 1, n \text{ (пакетів / с)}, \quad (2.6)$$

де h_{ij} - інтенсивність потоку, який необхідно передати від і до j.

В результаті передачі потоків запитів на компоненти сервісів у мережі дата-центру будуть виникати:

1) затримка між суміжною парою вузлів t_{rs};

2) T_{ij} - середня затримка між заданою парою вузлів;

3) D - загальна середня затримка в системі

Для визначення середньої затримки на надання сервісу введемо такі умови та обмеження.

- кількість функціонально-незалежних, незамінних компонентів програми становить k ;
- у випадку обслуговування запиту компонент буде доступним для всіх інших запитів, тобто компонент перейде до обслуговування наступного запиту паралельно з обробкою поточного запиту;
- на одній віртуальній машині можна встановити тільки один компонент чи його екземпляр, незалежно від функціональності, тому для розгортання окремого компоненту використовується нова віртуальна машина;
- на всіх фізичних серверах можуть одночасно виконуватися не більше ніж N віртуальних машин одночасно;
- кількість компонентів для всіх функцій програми є однаковою;
- вхідні потоки запитів на компоненти надходять за ступеневим законом розподілу;
- затримкою у вузлах нехтуємо, вважаємо, що вони мають буфер необмеженої ємності;

- часи обслуговування одного і того ж пакету в різних каналах є статистично незалежними випадковими величинами.

Якщо потік запитів на компоненту йде по декількох маршрутах від *i* до *j*, то λ_{ij} розділяється так що:

$$\lambda_{ij} = \left\{\lambda_{ij}^1,...,\lambda_{ij}^k\right\}$$

$$\lambda_{ij} = \sum_{q=1}^{k} \lambda_{ij}^q \tag{2.7}$$

де λ_{ij}^q - частка потоку, що протікає по *q*-му маршруту.

$$T_{ij} = \sum_{q=1}^{k} p_{ij}^q T_{ij}^q \tag{2.8}$$

де p_{ij}^q - імовірність вибору *q*-го маршруту M_{ij}^q; T_{ij}^q - затримка на *q*-му маршруті; $T_{ij} = \frac{1}{\lambda_{ij}} \sum_{q=1}^{k} \lambda_{ij}^q T_{ij}^q$. Враховуючи, що $\lambda_{rs} = \lambda_{ij}^q,\ \forall (r,s) \in \mu_{ij}^q$ і за умови, що маршрути не перетинаються, отримаємо:

$$T_{ij} = \frac{1}{\lambda_{ij}} \sum_{q=1}^{k} \sum_{(r,s)\in M_{ij}^q} \frac{\lambda_{ij}^q}{\mu_{rs} - \lambda_{ij}^q} \tag{2.9}$$

В результаті, загальна середня сумарна затримка на надання компонентів сервісу в системі визначатиметься:

$$D = \frac{1}{h_\Sigma} \sum_{(r,s)\in E} \frac{f_{rs}}{\mu_{rs} - f_{rs}} \tag{2.10}$$

де $h_\Sigma = \sum_i \sum_j h_{ij}$ - сумарна величина вхідного потоку; f_{rs} - сумарний потік, що протікає по каналах (r,s).

Якщо враховувати стійкість кожної вершини, тобто працездатність кожної віртуальної машини центру обробки даних, то під матрицею якості можна розуміти ймовірність стійкості структури системи, при якій забезпечується необхідний рівень параметрів QoS. Необхідний рівень

параметрів якості залежить від завантаженості кожної вершини в момент часу *t,* на рівень якої впливає сумарний потік, що протікає по каналах (r,s), та пропускної спроможності каналів.

$$D = \frac{1}{P_{cm}} \sum_{Deg.v} \frac{w_t(v)}{C - w_t(v)} \tag{2.11}$$

де P_{cm} – ймовірність стійкості структури системи, $w_t(v)$ – завантаженість вершин протягом часу *t*, *C* – пропускна здатність каналу зв'язку, *Deg v* – кількість ребер (каналів) приєднаних до вершини *v*.

Фактично проблема зменшення часу обслуговування (обробки) запитів, які надходять на обслуговування до центру обробки даних, з врахуванням топологічної структури такого центру зводиться до зменшення середньої сумарної затримки *D*. Однак така затримка повинна задовольняти умови :

$$D = \frac{1}{P_{cm}} \sum_{Deg.v} \frac{w_T(n)}{C - w_T(n)} \le D_{зад} \tag{2.12}$$

Для пошуку шляху за критерієм мінімального часу проходження максимальна затримка, при швидкості 100 Мбіт\с, рівна 100 мс. Затримка може бути зменшена тільки у тому випадку, коли структура центру обробки даних буде стійкою напротязі часу *T*. Стійкість структури (живучість) пропонуємо оцінити на основі методу, що включає в себе: оцінку структури мережі (моделі мережі), завантаженості кожної віртуальної машини в момент часу *t*, а також взаємопов'язаність вузлів між собою.

У момент часу t =0 необхідно провести перевірку по всіх вершинах $v \in V$, і сформувати множину з $\overline{V_1}$ вершин, для яких справедлива нерівність $w_0(\overline{v_j}) \ge \overline{w}(\overline{v_j})$. У всі наступні моменти часу $t = 1,2,...,T$ використовуватиметься правило:

$$w_{t+1}(v_{i_j}^j) = w_t(v_{i_j}^j) + \varepsilon_j \overline{w}(\overline{v_j}),\ i_j = 1,2,...,\left|\xi(\overline{v_j^t})\right|,\ j = 1,2,...,\left|\overline{V_t}\right| \tag{2.13}$$

де ε_j - параметр розподілу завантаження. Він може залежати від різних факторів, в найпростішому випадку він рівномірно розподіляє граничне завантаження по сусіднім вершинам. Для кожної вершини $\overline{v_j}$ значення параметру ε_j визначається як $\varepsilon_j = \frac{1}{\deg \overline{v_j^t}}$

Якщо $w_{t+1}(v_{i_j}^j) \geq \overline{w}(v_{i_j}^j)$, то це означає, що вершина $v_{i_j}^j$ припинила свою роботу і видаляється з графа, тобто відбувається реконфігурація мережі. Відповідно, сійкість структури системи порушується.

Стійкість до руйнування графа пропонуємо оцінити, як одиниця мінус суму ймовірності взаємопов'язаності вузлів між собою та завантаженості вузлів в момент часу *t*, який не перевищує максимально можливого завантаження системи.

$$P_{ст} = 1 - \sum_{i=1}^{n} w_i(t) + P_{зв_i} \tag{2.14}$$

де $w_i(t)$ – завантаженість вузлів в момент часу t; $P_{зв_s}$ – ймовірність зв'язності вузлів.

Ймовірність зв'язності між парою вузлів можна оцінити, як:

$$P_{зв}(A,B) = 1 - \prod_{i=1}^{n} \prod_{i=1}^{A,B} p_{найкор.шляху_i} p_{ребер.з.\max k_i} \tag{2.15}$$

де $p_{найкор.шляху_i}$ - ймовірність існування *i-го* найкоротшого маршруту; $p_{ребер.з.\max k_i}$ - ймовірність прокладання *i-го* найкоротшого маршруту по ребрах з максимальним ваговим коефіцієнтом *k*

Якщо завантаженість вузлів в момент часу *t* менша чи рівна максимально можливому завантаженню системи, то це призведе до збільшення стійкості структури системи, що в свою чергу призведе до зменшення затримки. Використання запропонованого методу дозволить зменшити тривалість пошуку каналів, по яких здійснюватиметься передача, оскільки, зменшиться затримка, яка закладена у метрику алгоритму пошуку оптимального шляху, що враховує

стійкість структури. Це забезпечить швидшу передачу компонентів сервісу та, відповідно, пришвидшить його надання.

Для оцінки ефективності методу необхідно випадковим чином сформувати зв'язки (відповідно до умов моделі мережі) між вузлами та оцінити ймовірність взаємопов'язаності спочатку довільних пар вузлів між собою, а потім загальну взаємопов'язаність мережі, яка визначатиметься, як сума пов'язаності усіх пар вузлів. Оцінка завантаженості кожної віртуальної машини, на протязі часу T, буде проводитися з використанням методу Нороса, з врахуванням, що кожна VM, відповідно до теорії систем масового обслуговування, буде системою типу G\G\1. Для спрощення вважатиметься, що коефіцієнт варіації інтервалів між запитами та тривалості їх обслуговування будуть сталими величинами. Максимальна завантаженість повинна визначатися по максимальній кількості запитів на обслуговування.

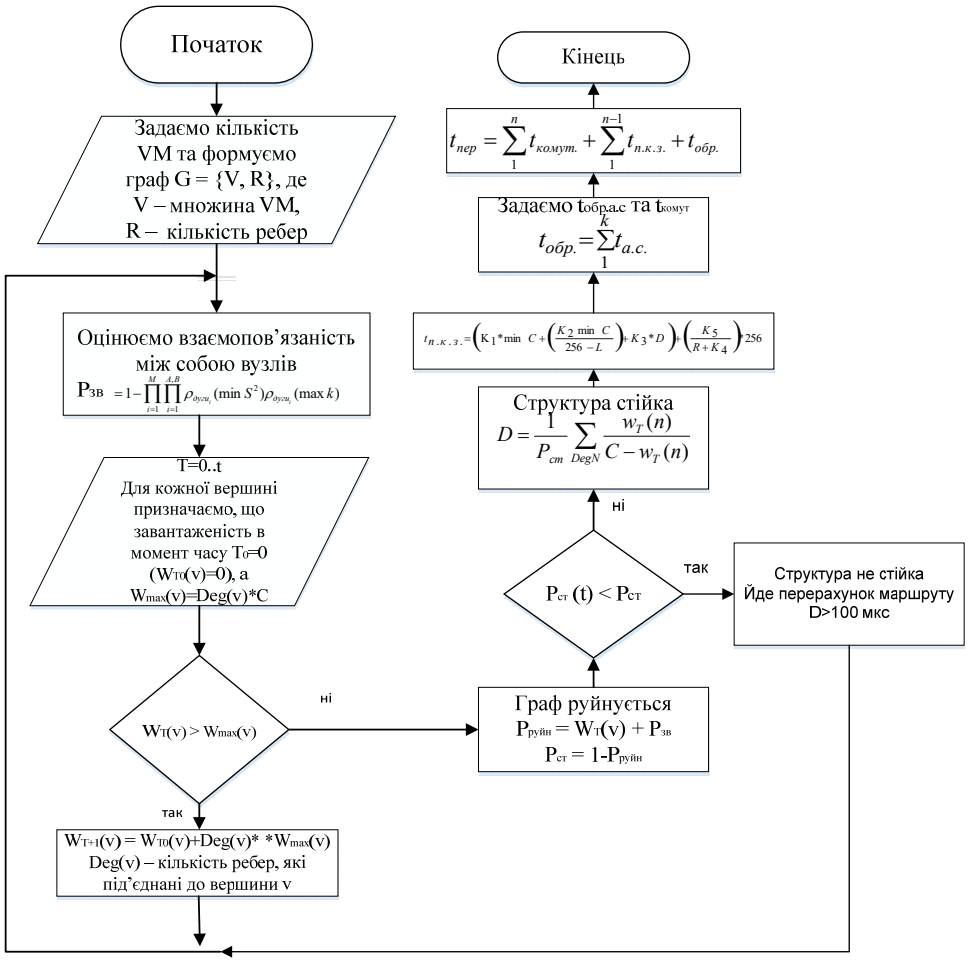


Рис.2.5 Блок-схема алгоритму роботи методу пошуку маршруту з урахуванням стійкості структури віртуалізованого центру обробки даних

Для оцінки стійкості структури ЦОД необхідно врахувати: якщо ж в моменти часу $T = 1..t$ ймовірність стійкості структури була меншою, то структура є не стійкою і йде перерахунок оптимального маршруту, що веде до збільшення часу надання сервісу кінцевому користувачу. Оцінка буде проводитися до моменту поки граф буде існувати.

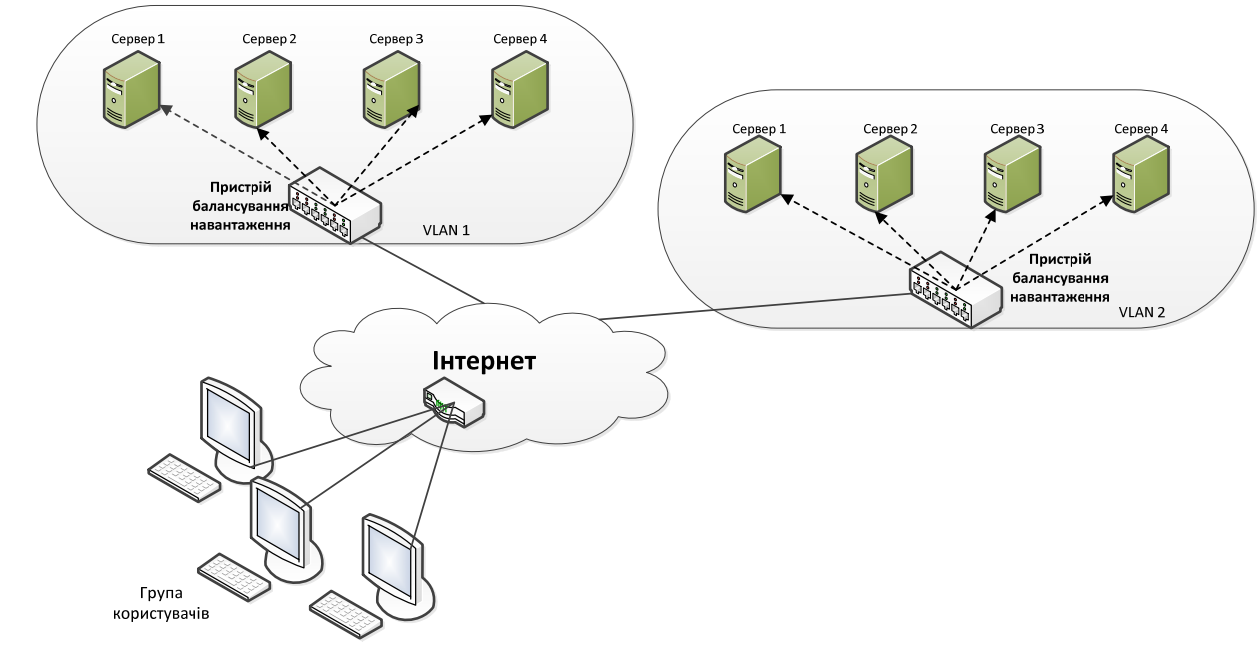
2.3. Підвищення якості надання композитних сервісів з використанням механізмів балансування навантаження

Одна з найважливіших ідей, що лежать в основі хмарних обчислень - це масштабованість, а ключовою технологією, яка забезпечує цю властивість, є віртуалізація [2, 78, 93, 100]. Віртуалізація дає змогу більш ефективно використовувати сервери, так як не потребує нарощення апаратної частини. Віртуалізація також забезпечує міграцію у реальному часі, особливо, коли сервер перевантажений, і екземпляр операційної системи з її додатками може бути перенесений на новий менш завантажений сервер.

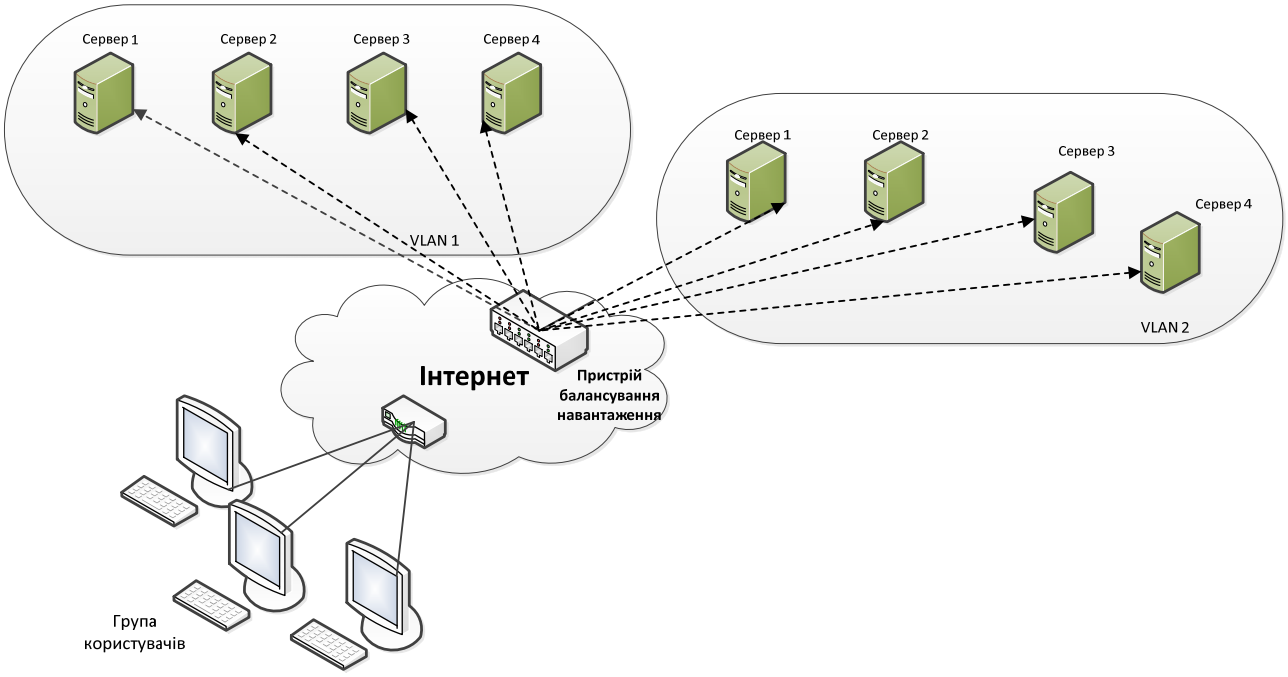
Балансування навантаження сприяє підвищенню продуктивності розподіленої системи в аспекті розподілу інформаційних потоків між множиною взаємодіючих хостів [11, 15, 80, 97]. Така система або прагне рівномірно розподілити навантаження на кожен хост і мати дуже малі відхилення від робочого навантаження на всіх інших фізичних хостах, або забезпечує уникнення перевантажень і блокувань на окремих серверах. Балансування навантаження являє собою стратегію розподілу навантаження між двома або більше системами для забезпечення надмірності і високої доступності.

Розрізняють балансування навантаження на двох різних рівнях:

- балансування навантаження локального рівня - серверне балансування навантаження (SLB) по локальній мережі (LAN);
- балансування навантаження глобального рівня - балансування навантаження на сервері (GSLB) по Wide Area Network (WAN) мережі.



а)



б)

Рис.2.6 Приклад балансування навантаження між серверами за допомогою пристрою балансування навантаження а) на локальному рівні; б) на глобальному рівні

Балансування навантаження локального рівня відноситься до стратегій розподілу навантаження в локальних мережах центрів обробки даних. Його можуть здійснювати маршрутизатори або комутатори рівня додатків, які здатні розподілити навантаження між двома або більше серверами, віртуальними

машинами або системами. Розподіл здійснюється або по IP адресі або по адресі канального рівня (зазвичай Ethernet MAC).

Ідея балансування навантаження на глобальній мережі полягає в тому, щоб зменшити навантаження на одному центрі обробки даних шляхом глобального розподілу навантаження на кілька центрів обробки даних, розташованих в інших країнах світу. GSLB може базуватися на двох незалежних системах- система доменних імен (DNS) і Border Gateway Protocol (BGP). Балансування навантаження DNS є, мабуть, найстарішим методом балансування навантаження і існувала задовго до звичайного SLB. DNS - один з методів розподілу навантаження, або відмовостійкості за рахунок надмірності кількості серверів, яка забезпечується шляхом управління відповідями DNS-сервера, відповідно до якоїсь статистичної моделі. У найпростішому випадку DNS працює, відповідаючи на запити не тільки однією IP-адресою, а списком з декількох адрес серверів, що надають ідентичний сервіс, які повторюються циклічно. Як правило, прості клієнти намагаються встановлювати з'єднання з першою адресою зі списку, таким чином різним клієнтам будуть видані адреси різних серверів, що розподілить загальне навантаження між серверами.

Немає стандартної процедури для визначення того, які адреси будуть використовуватися додатком - деякі сервери намагаються змінити порядок адрес в списку. Деякі настільні клієнти намагаються отримати альтернативні адреси після того, як не вдалося встановити з'єднання протягом 30-45 секунд.

Система DNS часто використовується для розподілу навантаження територіально розподілених веб-серверів. Наприклад, у компанії є один домен і три ідентичних веб-сайти, розташованих на трьох серверах з трьома різними адресами. Коли один користувач отримує доступ до головної сторінки, він буде направлений на першу адресу IP. Другий користувач, який звертається до головної сторінки, буде відправлений на наступну адресу IP, а третій користувач буде відправлений на третій адресу IP. У кожному разі, коли IP-адреса видається, вона відправляється в кінець списку. Четвертий користувач, відповідно, буде відправлений знову на першу адресу IP, і так далі.

Балансування навантаження на основі BGP системи ефективне тоді, якщо існують декілька маршрутів з однаковими адміністративними відстаннями і вартостями. Маршрутизатор рівня додатків повинен вирішити, який маршрут використовувати, враховуючи, що його кожен наступний вноситиме свої зміни в таблиці маршрутизації та зможе вибирати різні маршрути. Така стратегія балансування навантаження використовується у Netscape. Для підключення до доступних веб-сайтів прописується в браузері Netscape список можливих і доступних IP-адрес на які можна перерозподіляти потоки трафіку [79, 88, 95]. Проте, цей підхід ефективний лише для вихідного трафіку на сайт Netscapes і не підходить для тієї величезної кількості доступних браузерів, веб-сайтів та хмарних сервісів, які існують сьогодні.

Характеристики обслуговування запитів веб-сервісами (компонентами) розподіленого сервісу мають відповідати певним вимогам, що дадуть змогу забезпечити задовільний рівень обслуговування запитів. Зазвичай, у випадку складних процесів кількість веб-сервісів, які будуть задіяні в процесі обслуговування запиту, їхня різноманітність, цільове призначення, доступність та характеристики можуть по різному впливати на якість обслуговування користувача. Кожний веб-сервіс у процесі обслуговування може по різному впливати на якість обслуговування конкретного запиту залежно від свого поточного завантаження, стану завантаження апаратних ресурсів сервера, продуктивності, ефективності та доступності необхідних ресурсів для виконання операцій. Таким чином, сумарний вплив всіх компонент на якість обслуговування запиту можна виразити формулою:

$$QoS_S = \frac{\sum_{i=1}^{n} QoS_i}{n}, \tag{2.16}$$

де QoS_S - середнє значення якості обслуговування запиту під час проходження по конкретному маршруту, що складається з визначеної наперед кількості веб-сервісів n;

QoS_i – середнє значення якості обслуговування на визначеному проміжку

часу для i-ого веб-сервісу.

Це значення можна знайти за допомогою такої формули:

$$QoS = \sum_{i=1}^{n} QoS_i = \frac{\sum_{j=1}^{m} QoS_j(t)}{m}, \quad t \in (t_1; t_2), \tag{2.17}$$

де $QoS_j(t)$ – якість обслуговування конкретного запиту, який надійшов на обслуговування на i-ий веб-сервіс у момент часу t, що належить проміжку часу від t_1 до t_2.

Під час аналізу якості обслуговування запиту для кожного можливого маршруту можна спостерігати той факт, що якість обслуговування для кожного з них є різною. А тому правильна організація композитного додатку, що виконує функції конкретного сервісу та рівномірний перерозподіл запитів між його компонентами, має критичне значення для надання послуг абоненту з високою якістю обслуговування.

Однією з найголовніших причин збільшення затримок при передачі є великі обсяги повідомлень, що передаються між веб-сервісами. Це спричинено технологією XML, яка передбачає використання тегової конструкції опису елементів повідомлення. Опис елементів здійснюється за допомогою тегів-слів, реалізується за допомогою вилучення з композитних додатків веб-сервісів, що негативно впливають на процес обслуговування, та заміни їх на більш продуктивні, з вищими показниками якості обслуговування. Це дає змогу динамічно налаштовувати продуктивність композитного додатку при різних умовах завантаження як мережі, так і серверного обладнання, а, відповідно, покращувати якість обслуговування запитів. Управління цим процесом може здійснюватися за допомогою методів оркестрування чи хореографії залежно від того, який з методів використовували для створення сервісу. У випадку застосування методу оркестрування координатор самостійно визначить той веб-сервіс, продуктивність якого негативно впливає на роботу сервісу загалом, і замість нього використає веб-сервіс, що має таку саму функціональність і вищу продуктивність.

У випадку застосування методу хореографії веб-сервіси за допомогою спеціалізованих протоколів спілкування визначать можливу заміну для веб-сервісу, продуктивність якого надто низька.

Проте в умовах динамічного середовища, в якому перебувають композитні додатки та сервіси, фізична і логічна адреса веб-сервісу, який використовується для доступу до композитного додатку, може змінюватися. Це здійснюється з метою покращення ефективності системи за допомогою міграції різних компонентів системи з одного фізичного пристрою на інший.

Якщо говорити про хмарні сервіси та їх компоненти, то при балансуванні навантаження між серверами локального або глобального рівня необхідно врахувати і можливість їх міграції. Зазвичай, рішення про міграцію віртуальної машини приймається на основі даних про завантаженість серверів: наявність вільної пропускної здатності логічного і фізичного каналу, наявність ресурсів віртуальних машин (оперативної пам’яті, процесора і т.п.) [15, 20, 92]. У даній роботі пропонується новий підхід для оцінки цих показників, що дасть змогу підвищити ефективність балансування навантаження за допомогою реалізації інтегрованої архітектури управління з використанням технології NVF (Network Function Virtualization).

2.4. Формування інтегрованої архітектури системи управління ресурсами з використанням методу балансування навантаження та функцій мережевої віртуалізації

Принцип побудови архітектури управління з використанням функцій NVF наведений на рис. 2.7 [2, 15]. Запропонована архітектура управління включає в себе:

- Віртуальний менеджер інфраструктури (VIM), який відповідає за управління NFV ресурсами;
- Менеджер віртуалізації мережних функцій (NFV) - відповідає за управління життєвим циклом примірників NFV (примірника, конфігурації, оновлення, масштабування вгору/вниз). Менеджер NFV несе відповідальність за весь життєвий цикл NFV і через Оркестратор’а

може зажадати ресурсів інфраструктури та взаємодіяти з ними, наприклад, встановити програмний компонент або налаштувати VNF.

- Менеджер аналізу ресурсів (RD) – відповідає за аналіз стану ресурсів системи. В його обов'язки входить оцінка наявних фізичних, віртуальних, апаратних та телекомунікаційних ресурсів системи. На основі даних про достатню вільну пропускну здатність каналів та доступність компонентів того чи іншого сервісу, приймається рішення або про надання сервісу, або про міграцію його компонентів на менш завантажений сервер. Він надсилає запити до Оркестратора, який приймає рішення про необхідність міграції віртуальних машин з метою балансування навантаження та оптимального використання ресурсів;
- Менеджер управління міграціями – відповідає за процес міграції віртуальних машин та надсилає команду VDE (Virtual Distributed Ethernet) менеджеру про зміну місця перебування VM (номера порта, за яким вона була доступна, VLAN, в який вона переноситься і т.п.);
- VDE менеджер – підтримує і керує роботою VDE комутаторів, які дають можливість об'єднання віртуальних машин у VLAN'и, що полегшує їх адміністрування.

VDE комутатор є віртуальним аналогом Ethernet-комутаторів. Всі віртуальні пристрої, підключені до VDE мають можливість бачити один одного, як наче вони присутні у реальній мережі Ethernet. Вони мають здатність швидко здійснювати переконфігурацію мережі без будь-яких впливів на вхідну чергу запитів та незалежно від кількості портів. Ця характеристика має важливе значення для мінімізації затримки при передачі трафіку з кінця в кінець. Такий комутатор дозволяє управляти потоками, регулюючи число пакетів між електронними буферами вхідних і вихідних портів.

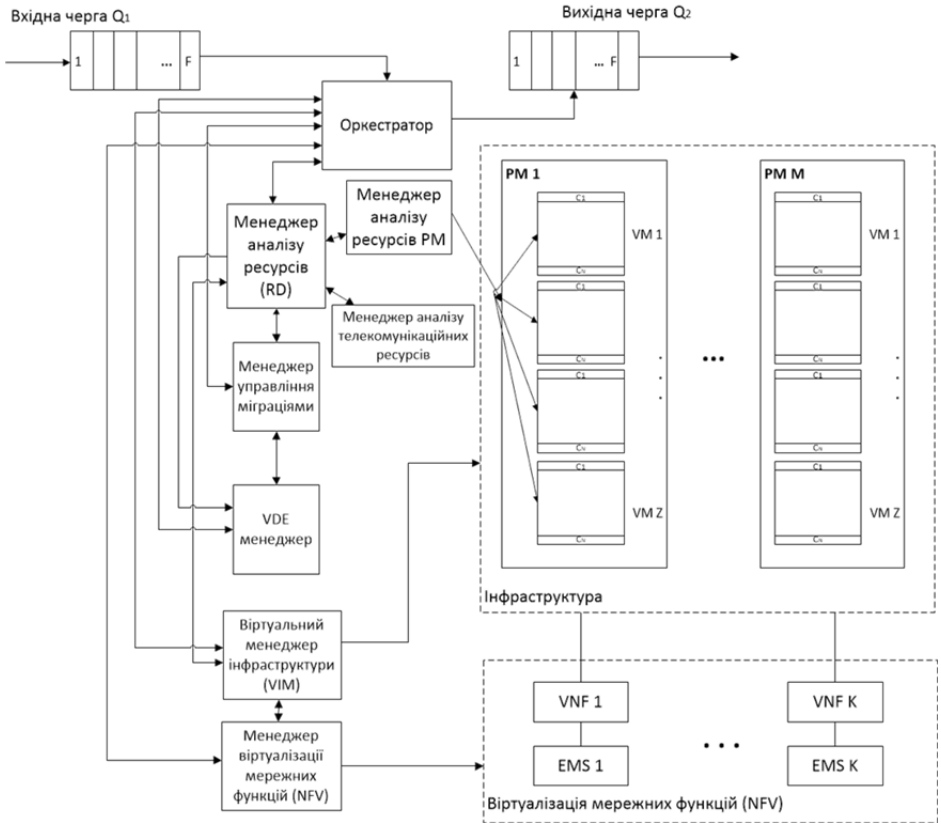


Рис. 2.7 Концептуальна модель системи управління сервісно-орієнтованої мережі

- Оркестратор - відповідає за оркестрування та управління ресурсами NFV і надає мережеві послуги на NFVI (Network Function Virtualization Infrastructure). Оркестратор містить загальний інтерфейс, який слідкує за розгортанням та наданням послуг; обробляє інформацію служб моніторингу та конфігурації; на основі даних від RD приймає рішення про міграцію та надсилає вказівку Менеджеру управління міграціями про перенесення віртуальних машин та VDE менеджеру - про зміну місця їх перебування. З іншого боку, він має інтерфейс до різних NFV менеджерів, а також до VIM'ів. Крім того, він має доступ до сервісу та інфраструктури модуля NFV, який містить інформацію про пріоритетність послуг.

Нехай у сервісно-орієнтовану мережу надходить запит на надання сервісу. Інтенсивність надходження таких запитів підкоряється експоненційному закону розподілу дискретної випадкової величини. Запит потрапляє у вхідну чергу. Оркестратор дає вказівку RD перевірити, чи всі

компоненти необхідного сервісу доступні, тобто чи вистачить обчислювальних потужностей у фізичних, а як наслідок і у віртуальних машин на обробку та надання усіх компонентів. Допоки PM (Phisical mashine) RD перевіряє на доступність вільних ресурсів, стан фізичних та логічних каналів, та визначає оптимальний шлях передачі для обміну компонентами по найменш завантажених каналах. У випадку, якщо RD повідомляє про недостатню обчислювальну потужність чи відсутність необхідної пропускної здатності, Оркестратору і надсилає інформацію про вільні і доступні ресурси до Менеджера управління міграціями, який здійснює вибір PM та переносить недоступний компонент сервісу (проводить міграцію VM), залучаючи до цього VDE менеджера. VDE менеджер визначає, в якій VLAN знаходилась перезавантажена запитами VM та в якій VLAN знаходиться недовантажена VM, і за допомогою VDE комутатора здійснює переключення логічної частини компонента сервісу та проводить реконфігурацію системи у відповідності до проведених змін (тобто здійснює оновлення таблиць маршрутизації). Увесь процес міграції та робота системи ведеться під наглядом Оркестратора та VIM. У випадку, якщо кількість запитів на той чи інший компонент буде невпинно зростати, то VIM може запустити виконання функцій NFV. Технологія віртуалізації мережевих функцій дає змогу створити програмний аналог будь-якого фізичного мережевого пристрою та запустити на ньому обробку запитів на надання сервісу.

2.4.1. Формалізація надання ресурсів ЦОД з інтегрованою системою управління

Як було описано у розділі, RD відповідає за аналіз стану ресурсів системи. В його обов’язки входить оцінка наявних фізичних, віртуальних, апаратних та телекомунікаційних ресурсів системи. Такий аналіз буде здійснюватися на основі максимального інтегрального показника доступних ресурсів кожної фізичної машини. Для його визначення необхідно контролювати наявність незайнятих апаратних ресурсів та їх доступність. Під доступністю маємо на увазі наявність вільних ресурсів віртуальних машин та

каналів зв'язку. Параметри незадіяних потужностей кожної віртуальної машини, на якій розташовано M_i компонент, розраховуються за співвідношеннями:

$$CPU_{pr} = \frac{\sum_{i=1}^{k} M_i * CPU_i}{\sum_{i=1}^{k} M_i} \tag{2.18}$$

$$RAM_{pr} = \frac{\sum_{i=1}^{k} M_i * RAM_i}{\sum_{i=1}^{k} M_i} \tag{2.19}$$

де M_i – кількість додатків (компонентів) у i -тій VM (Virtual mashine); CPU_i – тактова частота процесора VM, яку використовує i-тий компонент; RAM_i – об'єм оперативної пам'яті VM, яку використовує i-тий компонент; k – кількість VM однієї PM.

Для оцінки стану вільних ресурсів пропонуємо наведений на рис. 2.8 алгоритм. Він забезпечує пришвидшення обробки запитів та аналізує ступінь завантаженості кожної машини на базі інформації про обчислювальні потужності кожного вузла (в нашому випадку вузлом є VM). Нехай $z = \overline{1, Z}$ - віртуальна машина, на який знаходиться $i = \overline{1, K}$ компонентів сервісу $j = \overline{1, S}$, а $m = \overline{1, M}$ – фізична машина, на якій знаходиться $Z_m = \overline{1, Z}$ віртуальних машин та $P = \{P_1, \ldots, P_n, \ldots P_N\}, n = \overline{1, N}, P_n = \{e_1, \ldots e_l, \ldots e_L\}, l = \overline{1, L}$ – множини шляхів, де N – кількість шляхів між компонентами i та $i+1$, L – кількість ребер у шляху n, які з'єднують VM_z та VM_{z+1}.

Спочатку перевіряється вхідна черга запитів на компоненти сервісу. Якщо там міститься $f = \overline{1, F}$ запитів на ту чи іншу компоненту сервісу j, запускається аналізатор, який визначає, які ресурси необхідні для його обслуговування (таким чином формується таблиця значень CPU, RAM, вільної пропускної здатності (C)), які забезпечать надання доступу до компонента за час $t < t_{кр}$ та тип VM для кожного компонента складеного сервісу. Одночасно

перевіряється, чи компонент i сервісу j доступний. Кожний компонент сервісу використовує для своєї роботи апаратні потужності (CPU, RAM) фізичної та віртуальної машин, на яких він розташований. Resource discoverer перевіряє, чи z-та VM володіє достатніми апаратними ресурсами, щоб надати користувачеві ще один екземпляр компонента i. Якщо ресурсів достатньо, то перевіряємо, чи вистачить ресурсів каналу для того, щоб встановити з'єднання компонента з кінцевим користувачем.

$$C_{P_n} > R_{i,j}, \tag{2.20}$$

де $R_{i,j}$ - швидкість інформаційного потоку, що генерує *i*-тий компонент сервісу *j*. Всі параметри, які будуть доступні у момент перевірки стану мережі Resource discoverer'ом будуть записані у масив $\overline{A}$, що буде формувати чергу на обслуговування. Туди вноситиметься пропускна здатність каналу, якої вистачить для надання компонента. Resource discoverer повідомляє про це Оркестратору і компонент переноситься у вихідну чергу запитів Q_2.

Менеджер телекомунікаційних ресурсів перевіряє стан каналів, та визначає оптимальний шлях передачі та обміну компонентами по найменш завантажених каналах. Оскільки в систему надходять запити на компоненти сервісів, які передаються різними маршрутами, то можна визначити частку зайняття пропускної здатності каналу кожним компонентом, відносно загальної кількості маршрутів, що проходять по даному каналу.

Для початку визначимо ваговий коефіцієнт швидкості інформаційного потоку *i*-ого компоненту *j*-ого сервісу відносно їх загальної кількості, трафік яких передається по *l*-ому каналу зв'язку:

$$k_{i,j|l} = \frac{R_{i,j|l}}{\sum\limits_{i,j} R_{i,j|l}} \tag{2.21}$$

Тоді, частка незайнятої пропускної здатності *l*-ого каналу становитиме:

$$k_{0|l} = 1 - \frac{\sum\limits_{i,j} R_{i,j|l}}{C_l} \tag{2.22}$$

де C_l – смуга пропускання *l*-ого каналу.

Значення частки зайняття пропускної здатності $\mathrm{Pr}_{i,j|l}$ виражається пропорційно до $k_{i,l}$:

$$\mathrm{Pr}_{i,j|l} = k_{i,j|l} \cdot \left(1 - k_{0|l}\right) \cdot 100\% \tag{2.23}$$

Будемо вважати, що $\mathrm{Pr}_{i,j|l}$ відповідає пріоритету компоненти по відношенню до інших, трафік яких передається по *l*-ому каналу зв'язку. Це слід розуміти таким чином, що завершення передавання цієї компоненти призведе до вивільнення найбільшої пропускної здатності каналу, що, у свою чергу, дасть змогу обслуговувати чергу запитів із найбільшою ефективністю.

Вважатимемо, що певна смуга пропускання маршруту використовується неефективно, коли $\min\left(k_{0|l} \mid \mathrm{P}_n\right)$ максимальне. Тоді, за допомогою VDE switcher manager'а та Migration manager'а здійснюється процес завантаження частини смуги, що не використовується, та відбувається її перерозподіл між компонентами із більшою потребою.

Якщо при оцінці вільних потужностей виявиться, що ресурсів для надання ще одного екземпляру компонента чи наступного компонента не достатньо, тобто виконуються умови

$$CPU_{available}(z) > CPU(i,j) \tag{2.24}$$

$$RAM_{available}(z) > RAM(i,j) \tag{2.25}$$

$$C_{P_n} \min\left(k_{0|l} \mid \mathrm{P}_n\right) > R_{i,j} \mid P_n, \tag{2.26}$$

то здійснюється реплікація *z*-ої віртуальної машини на іншу фізичну машину PM_{m}. Інформація про необхідність міграції компоненту надсилається Оркестратору та Migration manager'у, який здійснює переміщення перевантаженої VM_{z} у співпраці із VDE switcher manager'ом. Якщо міграція неможлива, наступить відмова в обслуговуванні, і RD починає весь аналіз ресурсів спочатку.

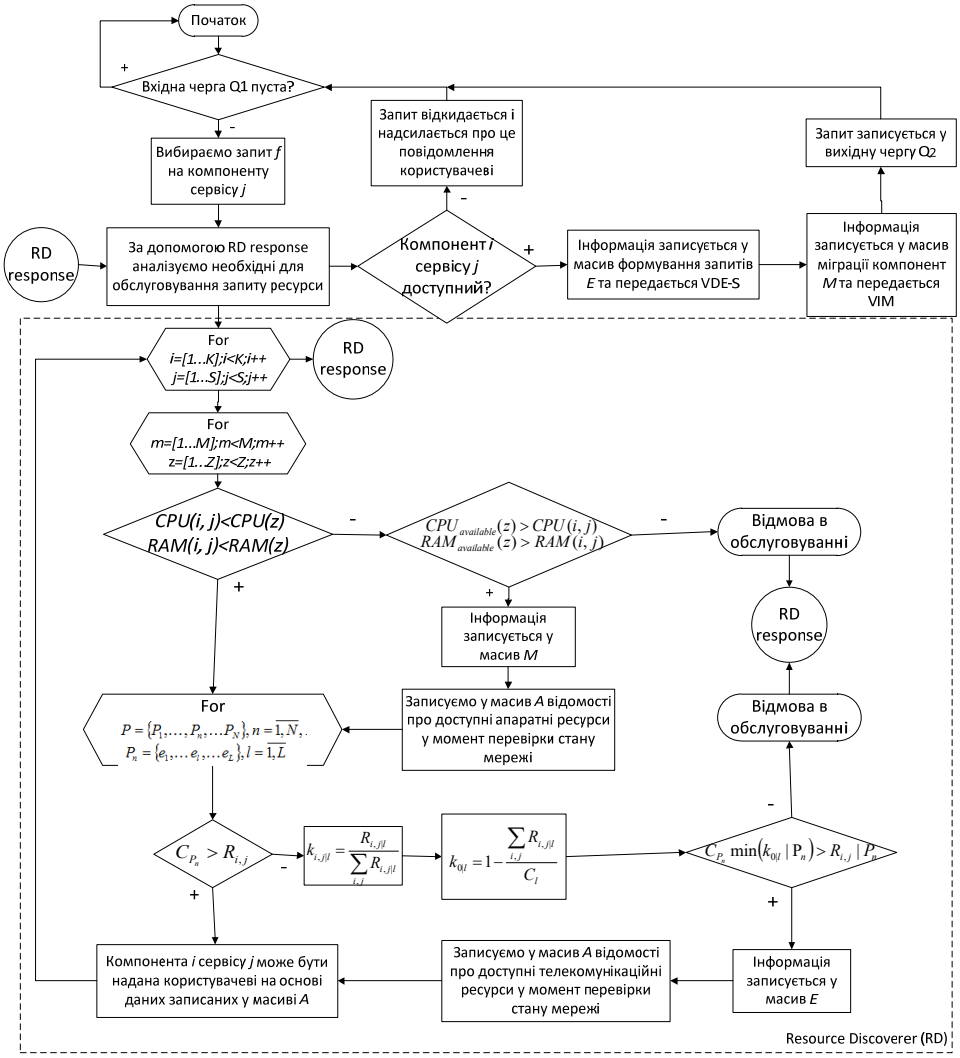


Рис. 2.8 Блок-схема алгоритму роботи методу балансування навантаження з врахуванням доступності фізичних ресурсів та з використанням функцій мережевої віртуалізації

Максимальне значення показника вільних віртуальних та телекомунікаційних ресурсів буде передаватися Orchestrator'у, який, в разі потреби, здійснюватиме міграцію додатків на менш завантажені PM.

2.5. Висновки до 2-го розділу

1. Основним параметром, що впливає на погіршення якості надання сервісу користувачам сервісно-орієнтованої мережі є затримка. Збільшення загальної тривалості передачі сервісу на пряму залежить від часу пошуку оптимального маршруту між його компонентами. Оскільки оптимальний шлях передачі змінюється, внаслідок перевантажень та міграції віртуальних машин, то це призводить до збільшення часу пошуку каналів, по яких буде

здійснюватися передача - $t_{п.к.з.}$, що в свою чергу призведе до збільшення загального часу передачі і виникнення затримки. Для вирішення даної задачі у роботі запропоновано здійснювати пошук шляху за критерієм мінімального часу проходження, в основі якого лежить спосіб розрахунку оптимального шляху передачі на основі даних про поширення інформації та зміни в топології мережі. Ці дані складають єдину метрику маршруту, яка виявляє компроміс між вибором оптимального маршруту та властивостями трафіку, оскільки, імовірність одночасного існування потоків із максимальним сервісом на маршрутах із спільною лінією є малою

2. Для оцінки стійкості у таких інформаційних мережах, як мережі центрів обробки даних, необхідно забезпечити якомога більшу стійкість зв'язків між кожним із компонентів сервісу. Під стійкістю зв'язків розуміється стійкість топологічних структур при передачі чи міграції компонентів, тобто щоб існував як фізичний, так і логічний маршрут при реплікації компонентів. У роботі запропоновано модель надання сервісу на основі методу пошуку маршруту з урахуванням стійкості структури віртуалізованого центру обробки даних . Перевагою даної моделі є вибір стратегії, яка базується на методі пошуку маршруту з урахуванням стійкості структури ЦОД, який врахувує інтенсивність запитів, що надходять до VM, що дасть змогу на підставі даних про стійкість структури в конкретні моменти часу не здійснювати повторний перерахунок оптимального шляху передачі та забезпечити зменшення затримки при наданні сервісу та підтримку чи підвищення рівня QoS.

3. Однією з найголовніших причин збільшення затримок при передачі є великі обсяги повідомлень, що передаються між веб-сервісами та не ефективний розподіл фізичних ресурсів на їх обслуговування. У роботі пропонується новий метод для ефективного розподілу фізичних ресурсів, що дасть змогу підвищити ефективність балансування навантаження за допомогою реалізації інтегрованої архітектури управління з використанням технології NVF. Суть методу полягає в оцінці максимального інтегрального показника доступних ресурсів фізичної машини. Максимальне значення показника вільних віртуальних та телекомунікаційних ресурсів будуть передаватися

Оркестратору, який в разі потреби здійснюватиме міграцію додатків на менш завантажені сервери. У випадку якщо кількість запитів на той чи інший компонент буде невпинно зростати, то Віртуальний менеджер інфраструктури може запустити виконання функцій NFV, що дасть змогу створити програмний аналог будь-якого фізичного мережевого пристрою та запустити на ньому обробку запитів на надання сервісу. Унікальність методу полягає у можливості балансування навантаження як на глобальному так і на локальному рівнях роботи ЦОД. Це дозволить зменшити не тільки тривалість обслуговування запитів, а й частку втрачених запитів. Завдяки інтегральній оцінці телекомунікаційних та програмно-апаратних ресурсів, запропонований метод дасть змогу розвантажити найбільш завантажених сервер чи ЦОД та зменшити час затримки надання сервісу користувачам, і, відповідно, підвищити якість надання послуг.

РОЗДІЛ 3.
МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ РОЗПОДІЛУ ІНФОРМАЦІЙНИХ ПОТОКІВ В ЦЕНТРАХ ОБРОБКИ ДАНИХ

У даному розділі проведено моделювання та дослідження розподілу інформаційних потоків в центрах обробки даних. Розроблено імітаційну модель формування структури центру обробки даних з використанням засобів Matlab. Досліджено ефективність застосування методу пошуку маршруту з урахуванням стійкості структури віртуалізованого центру обробки даних на основі розробленої імітаційної моделі. В результаті моделювання було встановлено, що при збільшенні стійкості структури час пошуку каналів, по яких буде здійснюватися передача запитів, зменшується, що в результаті призведе і до зменшення загального часу передачі сервісу до центру обробки даних. Проведено моделювання інтегрованої системи управління з використанням функції NVF з використанням засобів C++. На основі отриманих залежностей доведено ефективність методу балансування навантаження з урахуванням доступності фізичних ресурсів і з використанням функцій мережевої віртуалізації та спостерігається зниження затримки на надання компонентів сервісу та здійснюється контроль за ресурсами кожного серверу. Наведені у розділі результати опубліковано у працях [1, 5, 6, 9, 11, 14, 16, 18, 22, 25, 27].

3.1. Розроблення імітаційної моделі структури ЦОД

Імітаційне моделювання процесу обслуговування інформаційних потоків завжди вимагало від розробника імітаційної моделі перевірки адекватності створеної моделі процесам, що відбуваються в реальній системі масового обслуговування. Найпростіший спосіб визначити характеристики системи обслуговування полягає в отримані експериментальних даних щодо процесу обслуговування. Аналіз цих даних дає змогу визначити, які параметри системи обслуговування необхідно змінити для того, щоб підвищити якість обслуговування, тобто оптимізувати процес [81, 90].

Сучасні системи масового обслуговування містять велику кількість компонентів, кожний з яких є складною системою, яка також має свої параметри та характеристики. Загалом всі ці компоненти впливають на характеристики якості обслуговування системи. А тому для створення адекватної імітаційної моделі та адекватного оцінювання результатів моделювання необхідно врахувати всі компоненти, що беруть участь в процесі обслуговування.

Велика кількість абонентів, програмних додатків та сесій, що генеруються цими додатками, та їхня різноманітність значно впливають на характеристики трафіку, що надходить у систему обслуговування. Тому, щоб змоделювати такий трафік, необхідно застосувати потужний математичний апарат, який би дав змогу більш або менш точно описати характеристики такого трафіку. Зрозуміло, що найефективнішим способом моделювання в такій ситуації є розроблення спеціального програмного забезпечення.

Завдяки програмній реалізації імітаційної моделі можна не тільки повністю реалізувати всі необхідні функції моделі, але і забезпечити контроль над її роботою. Програмне забезпечення дає змогу за допомогою графічного інтерфейсу користувача динамічно змінювати параметри моделі, тим самим оцінити поведінку системи, що моделюється, в конкретній ситуації, яка може виникнути в реальній системі обслуговування. Крім того, програмне забезпечення за допомогою графічної оболонки дає змогу в реальному режимі часу давати оцінку всім параметрам моделі. Це можливо здійснювати за допомогою графіків, діаграм, списків та таблиць, які обновляються в реальному режимі часу.

Моделювання динамічно змінних структур вимагає від розробника імітаційної моделі великих зусиль. Необхідно врахувати усі вимоги та умови при яких повинна змінюватися структура мережі. Важливим при цьому є вибір характеристик і параметрів передаваного по такій структурі трафіку.

Особливо непросто здійснювати моделювання структур сервісно-орієнтованих мереж [89, 96]. Для розроблення імітаційної моделі динамічно змінної структури мережі центру обробки даних, який є основою сервісно-

орієнтованої мережі використано програмне середовище Matlab [1, 5, 9, 18]. Це середовище розробки програмного забезпечення володіє всіма необхідними засобами для створення повноцінних, ефективних, багатопотокових динамічно змінних структур. За допомогою створення масивів можна згенерувати граф мережі, який складатиметься з вузлів і ребер, який дозволить наглядно відобразити структуру мережі.

У якості вузлів, в даній імітаційній моделі, будуть виступати віртуальні машини, а в якості ребер – логічні канали, які з'єднують їх між собою. Запити до віртуальних машин надсилаються за логнормальним законом розподілу (міжпакетний інтервал). Для спрощення приймемо, що інтенсивність поступлення запитів на обслуговування від одного користувача в середині сегменту та його тривалість (4 зап/год, 4000 с.), одна віртуальна машина володіє лише одним сервісом. Введено, що є два діючих стани кожного вузла (активний(1) чи пасивний(0)).

Розробка імітаційної моделі відбувалася в декілька етапів:

- Створення динамічно-змінної структури. В програмному середовищі Matlab рандомізовано створювалася матриця суміжності, яка вказула на кількість вузлів та зв'язки між ними. Зв'язки між вузлами генерувалися випадково, проте їх кількість не перевищувала 50% від максимальної кількості (максимальною кількістю зв'язків вважалося повнодоступне з'єднання вузлів між собою, тобто «кожен з усіма»). Після формування матриці проводилася оцінка степені вершини, коефіцієнта кластеризації та посередництва, а також визначалася довжина найкоротших шляхів та їх центральні вузли. Оцінка цих параметрів дасть змогу визначити найбільш завантажені вузли, тобто віртуальні машини через які проходить найбільша кількість запитів. У зв'язку із обмеженістю апартних ресурсів комп'ютера, на якому проводилося моделювання кількість вузлів була рівною 20;
- Створення генератора трафіку. Створювався масив запитів, які надсилалися до вузлів графа по логнормальному закону розподілу і

інтенсивністю 4 зап/год. Розмір запитів коливався від 1500 біт до 4500 байт, а тривалість обслуговування 4000 с.;

- Оцінка взаємопов’язаності вузлів між собою. Задаємо, що кожне ребро володіє параметром ваговий коефіцієнт, який вказує на його максимальну пропускну здатність (від 1 – 10 Мбіт/с). Оскільки модель описана у п.2.2 працює на основі аналізу внутрішньої мережі центру обробки даних, то відстань між вузлами (віртуальними машинами) не більша 25 м.;
- Перевірка стійкості структури. Для кожної вершини визначаємо її завантаженість (на основі параметрів отриманих на першому та другому етапі моделювання та з використанням методу Нороса) та проводимо оцінку стійкості структури та затримки на надання сервісу. У випадку зміни структури визначаємо загальний час передачі, приймаючи, що час комутації на кожному вузлі не більший 0,03 мс

3.2. Моделювання структури центру обробки даних та її вплив на параметри QoS

Для оцінки ефективності моделювання структури центру обробки даних необхідно випадковим чином сформувати зв’язки (відповідно до умов моделі мережі) між вузлами та оцінити ймовірність взаємопов’язаності спочатку довільних пар вузлів між собою, а потім загальну взаємопов’язаність мережі, яка визначатиметься, як сума пов’язаності усіх пар вузлів [6, 9, 14, 18]. Оцінка завантаженості кожної віртуальної машини, на протязі часу T, буде проводитися з використанням методу Нороса, з врахуванням, що кожна VM, відповідно до теорії систем масового обслуговування, буде системою типу G\G\1. Для спрощення вважатиметься, що коефіцієнт варіації інтервалів між запитами та тривалості їх обслуговування будуть сталими величинами.

У нашій моделі ми оцінюємо стійкість структури з точки зору параметру ймовірності того, що між двома сегментами в наступний момент часу існуватиме з’єднання, тобто існуватиме хоча б одне ребро, яке буде «ключовою ланкою» для з’єднання цих сегментів [11, 16]. Однак така ймовірність існування ребра (з’єднання) напряму буде залежати від ймовірності відмови

певного шляху в середині сегменту (тобто, що між віртуальними машинами існуватиме хоча б один маршрут), а також від ймовірності блокувань в середині кожного сегменту.

Після формування всіх необхідних умов, за допомогою аналітичних і статистичних методів оцінки ймовірності встановлення з'єднання та його надійності оцінюємо необхідний нам параметр.

Нехай зафіксуємо в момент часу T деякий стан мережі (кількість віртуальних машин і т.п.). Розглянемо зафіксований стан сегменту A (рис 3.1), в якому на даний момент часу T знаходиться 20 вузлів. Задана кількість вузлів є незначною, що пов'язано із наступними обмеженнями: складність розрахунків із збільшення кількості вузлів зростає, а також ПК на якому проводилось моделювання має невисоку продуктивність. Червоним кольором показані найбільш завантажені ребра, а в середині кожної вершини – її параметри (степені вершини, коефіцієнт посередництва і т.д).

Для оцінки зв'язності вузлів необхідно знайти множину незалежних маршрутів, які б задовольняли критерії мінімальної близькості та максимального вагового коефіцієнта. Для цього потрібно визначити ймовірність пов'язаності між двома випадково обраними вузлами A та B, причому кількість обраних пар є набагато більшою за кількість вузлів, що в подальшою дозволить судити про взаємопов'язаність загалом.

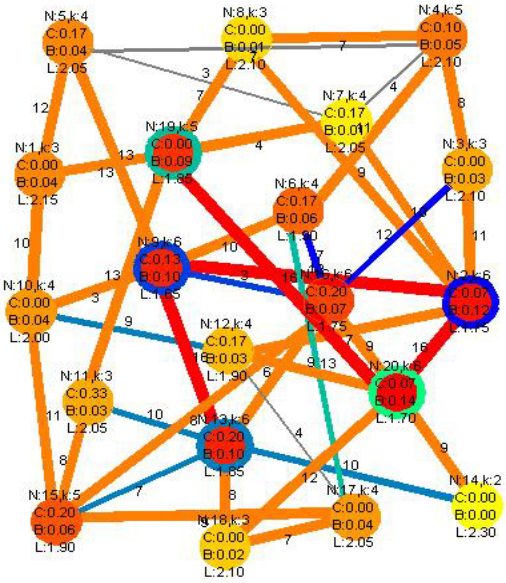


Рис.3.1 Структура змодельованої мережі центру обробки даних

Приймемо, що максимальна відстань між вузлами сегменту не перевищує 25 м, а максимальний ваговий коефіцієнт ребра не більша 10 Мбіт. Генеруємо матриці відстаней між вузлами (V) та вагових коефіцієнтів (K):

K=

0	1,4	8,2	0	4,1	5,2	1,2	3,0	3,4	6,2	5,1	8,5	3,5	9,8	9,9	0	4,7	9,7	7,8	7,5
4,4	0	9,5	0	5,7	0	5,4	7,1	9,3	6,7	4,0	9,9	5,6	7,4	8,1	8,0	1,4	0	5,0	4,8
6,6	1,4	0	0	4,4	0,7	0	0	9,3	3,8	0	5,0	5,9	0	2,6	5,5	5,5	6,8	8,6	8,0
0	0	0	0	0	6,4	0	8,7	0,4	0,3	0	9,0	3,2	0,2	7,2	2,4	0	0,5	0,6	7,6
7,9	9,2	8,0	0	0	0	6,8	0	4,5	0	8,1	0,7	6,3	0	0	7,3	3,5	0	6,4	6,2
1,5	0	6,5	8,9	0	0	0	0	0	0	0	7,7	6,9	2,0	0	9,6	0	1,5	3,1	0
0,4	0,3	0	0	6,4	0	0	8,5	5,2	0	1,8	4,7	5,6	0	2,4	3,4	0	8,1	8,8	9,9
4,0	5,7	0	2,3	0	0	1,5	0	8,1	0	0,3	7,9	0	9,6	5,0	9,5	5,1	2,8	8,0	0
0,6	8,5	7,7	5,9	3,4	0	5,5	8,5	0	2,2	0	2,9	0	1,7	9,3	1,3	0	5,4	0	2,8
9,5	7,3	2,7	7,9	0	0	0	0	0,3	0	0	6,0	1,9	5,1	7,5	0	7,0	0	0	7,0
5,2	5,3	0	0	3,3	0	6,1	1,7	0	0	0	6,4	0	2,8	7,4	0,8	0	0	6,4	0
5,8	6,5	5,8	5,5	2,2	1,5	6,7	5,3	9,9	2,0	9,3	0	8,6	0	0	2,4	0,7	0	2,5	0
1,1	2,3	0,2	2,2	4,1	6,7	5,8	0	0	0,8	0	0,5	0	3,2	1,5	0	2,2	2,3	2,6	1,4
4,8	3,3	0	2,5	0	5,2	0	1,7	2,1	1,9	9,3	0	3,7	0	0	4,8	4,4	0	9,3	4,2
6,7	7,4	8,1	0,6	0	0	6,5	5,0	4,3	4,4	8,3	0	1,7	0	0	7,3	1,2	2,6	4,6	1,4
0	6,8	2,8	8,5	0,4	9,2	7,1	0,7	8,4	0	5,5	4,4	0	3,4	7,7	0	5,9	0	0	8,0
3,5	2,1	7,2	0	7,7	0	0	3,0	0	9,3	0	8,4	0	2,6	9,8	3,0	0	8,5	1,0	0,4
0,1	0	8,3	1,2	0	3,2	8,6	9,0	9,8	0	0	0	0,3	0	7,4	0	4,2	0	5,8	0
7,0	9,1	2,9	9,0	7,0	7,1	9,0	5,3	0	0	5,0	7,7	9,2	1,3	6,1	0	2,8	0,6	0	0
7,4	1,4	7,4	0,6	5,3	0	1,4	0	2,6	3,9	0	0	2,5	7,9	6,3	1,5	2,7	0	0	0

V=

0	1,0	5,7	0	19,7	18,0	2,5	5,4	16,9	10,9	5,1	6,3	15,5	5,7	4,0	0	16,4	10,8	15,1	4,0
0,1	0	9,3	0	12,9	0	5,3	17,1	1,7	23,7	2,3	7,5	15,2	16,7	1,9	13,2	23,8	0	21,9	4,6
6,5	13,5	0	0	13,8	6,8	0	0	12,2	24,0	0	4,8	5,6	0	23,5	1,3	10,5	18,0	11,5	10,9
0	0	0	0	0	19,4	0	20,4	15,7	22,4	0	23,8	18,4	18,5	21,4	23,4	0	13,2	1,6	23,0
17,9	7,6	14,0	0	0	0	7,3	0	6,2	0	20,2	5,9	17,7	0	0	5,9	11,7	0	4,7	23,3
8,8	0	0,4	15,3	0	0	0	0	0	0	0	19,6	20	25,0	0	23,8	0	6,9	12,6	0
17,1	17,6	0	0	10,2	0	0	20,6	6,6	0	0,7	12,4	13,0	0	13,2	14,5	0	24,0	21,9	17,3
11,5	17,7	0	10	0	0	22,2	0	4,4	0	21,5	7,1	0	4,4	24,7	6,5	3,8	24,9	16,1	0
9,6	12,1	22,5	24,5	14,5	0	13,7	13,0	0	23,8	0	4,9	0	4,5	2,0	1,9	0	13,3	0	23,3
11,3	10,9	10,9	10,1	0	0	0	0	9,1	0	0	10,7	20,8	15,7	0,2	0	22,3	0	0	23,6
7,1	16,7	0	0	0,6	0	10,3	7,7	0	0	0	19,7	0	0	6,7	23,9	0	0	10,2	0
10,5	18,6	1,8	0,1	11,9	6,8	13,9	19,2	4,2	6,7	12,7	0	12,7	0	0	20,1	4,9	0	13,1	0
19,6	22,8	9,7	9,0	6,3	14,4	24,3	0	0	18,0	0	7,9	0	15,1	1,1	0	0,8	23,6	12,8	18,5
22,0	23,4	0	13,4	0	5,3	0	5,4	13,8	10,4	23,5	0	15,6	0	0	7,9	10,1	0	9,5	19,8
11,4	21,7	13,4	13,2	0	0	20,5	4,5	7,8	12,5	3,1	0	4,4	0	0	15,6	24,6	7,0	3,0	14,5
0	0,3	7,0	22,6	9,1	20,9	21,8	19,1	21,9	0	18,1	19,5	0	13,7	18,0	0	17,0	0	0	10,7
2,2	23,0	10,8	0	14,3	0	0	24,4	0	24,4	0	11,4	4,3	0,6	3,1	11,4	0	18,1	9,6	1,3
19,3	0	7,8	7,0	0	23,3	21,2	16,7	22,4	0	0	0	15,3	0	19,9	0	4,0	0	4,2	0
18,3	3,8	20,4	8,2	11,0	21,5	18,5	22,1	0	0	17,0	7,5	12,6	15,4	1,8	0	8,6	8,1	0	0
9,3	0	23,3	8,4	21,4	0	20	0	15,1	15,8	0	0	21,1	2,3	12,4	18,0	10,6	0	0	0

Вважаємо, що кількість ребер між вузлами становить не менше ніж 70% від кількості ребер повноз'вязної структури із заданою кількістю вузлів. Таку кількість ребер задано для того, щоб сегмент характеризувався високою зв'язністю та в процесі генерації графа до кожного із вузлів можна було доступитись через декілька шляхів.

Після генерування потрібних нам матриць та їх аналізу знаходимо всі можливі маршрути. Вони можуть складатися з одного ребра (це означатиме, що А і В з'єднані напряму) або здійснювати передачу через транзитні вузли (це означає, що між А і В є додаткова вершина С). У випадку коли існує безпосереднє з'єднання між А та В, то ймовірність того, що в такій системі дані два випадкові вузли будуть з'єднанні залежить від характеристик ребра та оцінюється:

Таблиця 3.1

Ймовірність безпосереднього з'єднання між довільною парою вузлів

0,39	0,227	0,18	0182	0,552	0,114	1	0,47	0,30	0,75	0,42
0,21	0,815	0,173	0,43	0,29	0,13	0,90	1	0,5	0,21	0,815

Дані вказують на те, що при такій кількості вузлів з великою ймовірністю хоча б 6 вузлів з'єднані напряму.

При кожному запуску моделі всі матриці, а відповідно і всі результати, будуть набувати різних значень, так як з'єднання між вузлами генеруються випадковим чином.

Оптимальних маршрутів може бути декілька, тому вибираємо лише той маршрут де сумарне значення вагових коефіцієнтів ребер і відстаней між вузлами для одного маршруту буде більшою від суми вагових коефіцієнтів ребер і відстаней між вузлами іншого маршруту, що можна визначити як:

$$\sum\left(\max\left[k_{1.1}+k_{1.2}\right],\min\left[S_{1.1}^{2}+S_{1.2}^{2}\right]\right)>\sum\left(\max\left[k_{2.1}+k_{2.2}\right],\min\left[S_{2.1}^{2}+S_{2.2}^{2}\right]\right) \qquad (3.1)$$

За такою умовою порівнюємо маршрути між собою і вибираємо найкращий маршрут для з'єднання. При такій кількості вузлів ймовірність того, що маршрут між довільною парою вузлів лежить через два ребра дуже велика. З цього слідує висновок, що від кількості вузлів у сегменті (а їх у нашому

випадку небагато) залежить наскільки буде складним маршрут (через скільки проміжних вузлів він буде проходити). Відповідно збільшення кількості ребер на маршруті зменшить ймовірність пов'язаності між собою вузлів. Для даної моделі ймовірність того що випадкові два вузли не взаємопов'язані між собою досить мала. За такою моделлю можна оцінити зв'язність довільної пари вузлів і говорити про зв'язність чи не зв'язність вузлів вцілому.

Для даної моделі ймовірність незв'язності вузлів оцінюється :

Таблиця 3.2

Ймовірність незв'язності вузлів мережі

1,159 E-06	2,26E-06	1,05E-10	2,71E-11	5,42E-11
2,48E-13	4,06E-19	4,93E-11	1,02E-13	3,13E-18
1,58E-08	1,01E-14	6,13E-26	1,81E-13	2,92E-11
6,54E-11	1,34E-06	3,34E-15	3,59E-11	4,27E-09

Звідси слідує висновок, що в такому маленькому сегменті майже всі вузли взаємопов'язані між собою. Проте зі зростом кількості вузлів не взаємопов'язаність вузлів зростає :

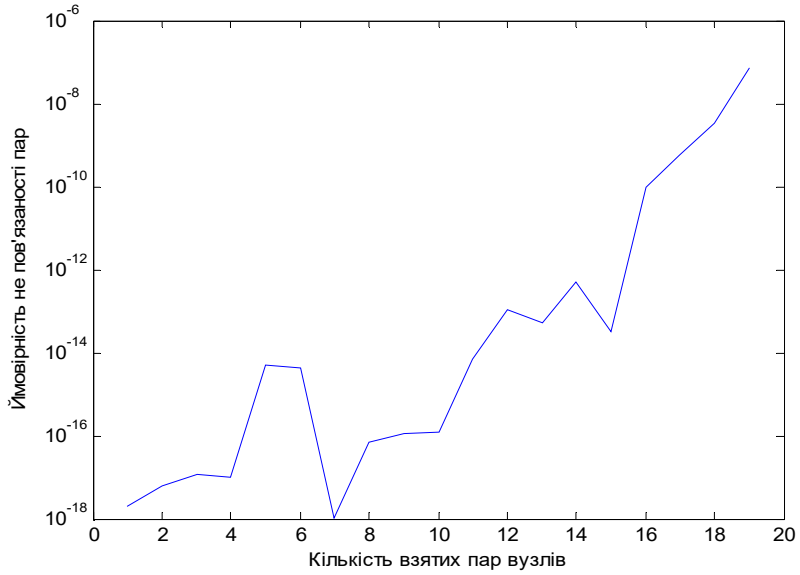


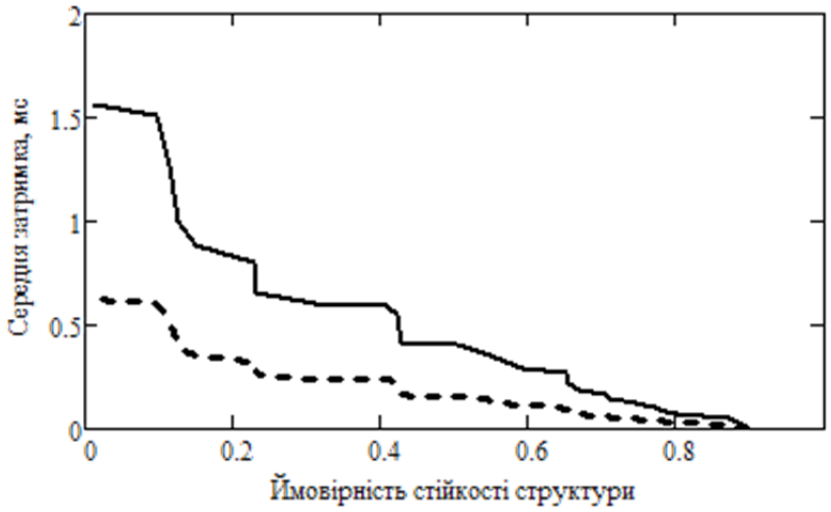
Рис. 3.2 Залежність ймовірності непов'язаності вузлів від їх кількості

З наведеного графіку та таблиці значень можна прослідкувати, що чим більша кількість вузлів тим ймовірність незв'язності зростає. Це свідчить про те, що у більших сегментах пов'язаність між собою вузлів зменшується.

Мережа стає більш розрідженою, а для оцінки маршруту з критерієм мінімальної близькості потребуватиметься більше часу.

3.3. Дослідження ефективності застосування методу пошуку маршруту з урахуванням стійкості структури віртуалізованого ЦОД на основі розробленої імітаційної моделі

При реалізації методів описаних у п. 2.1 та п. 2.2 в програмному середовищі Matlab було отримано залежності, які вказують на вплив оцінки стійкості структури мережі на затримку при обслуговуванні запитів та на час пошуку каналів, по яких буде здійснюватися передача. Для здійсненн моделювання було прийнято, що один атомарний сервіс обслуговується віртуальною машиної за $t_{a.c.}$= 0,005 мс. В результаті роботи моделі отримана залежність (рис.3.3), яка показує, що чим більш стійкіша структура мережі тим менша затримка, що дає змогу пришвидшити процес надання сервісу кінцевому користувачу і забезпечити необхідний рівень QoS.



—— *до впровадження методу пошуку маршруту з урахуванням стійкості структури ;*

- - - - *із застосуванням методу пошуку маршруту з урахуванням стійкості структури*

Рис. 3.3 Залежність ймовірності стійкості структури від середньої сумарної затримки

Після проведення моделювання із застосуванням методу пошуку маршруту з урахуванням стійкості структури віртуалізованого ЦОД залежність середньої сумарної затримки та ймовірності стійкості структури (рис.3.3),

показує зменшення часу затримки на 35% з 1,7 мс до 0,6 мс, що призводить до пришвидшення процесу надання сервісу кінцевому користувачу.

В результаті моделювання встановлено, що при зменшенні затримки час пошуку каналів, по яких буде здійснюватися передавання запитів, зменшується, що в результаті призведе і до зменшення загального часу передавання сервісу до ЦОД і назад.

В результаті отриманих залежностей можна говорити про загальне зменшення затримки та підвищення якості надання сервісів користувачам сервісно-орієнтованої мережі (рис. 3.4).

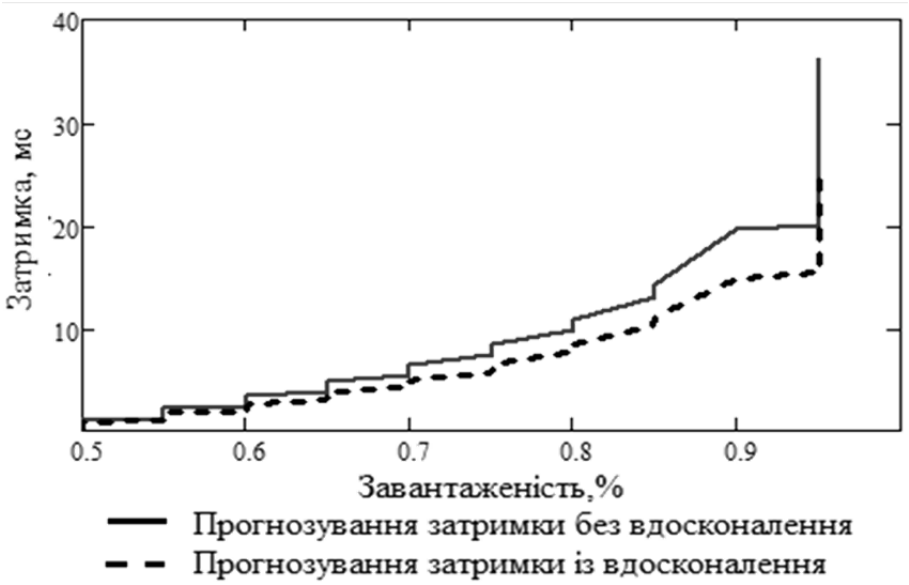
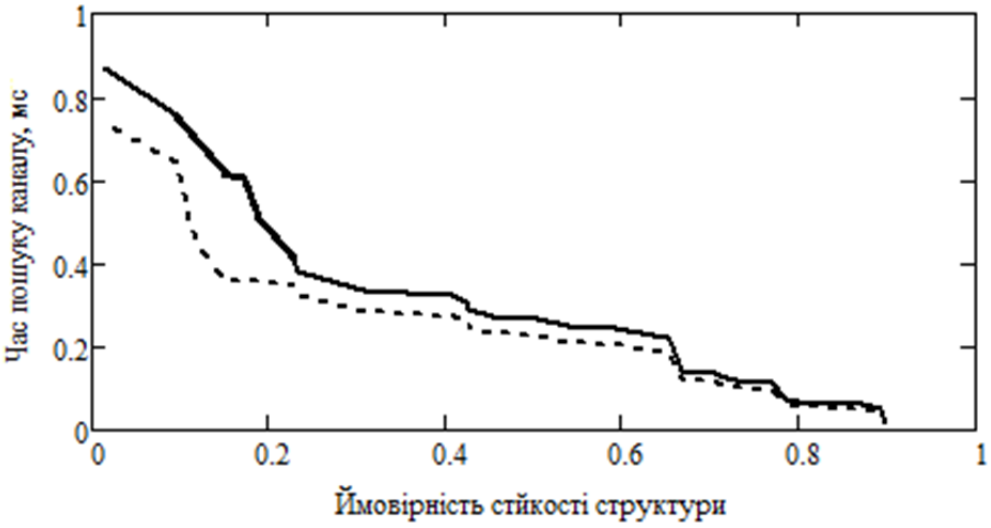


Рис. 3.4. Прогнозування тривалості затримки пакетів голосового сервісу дослідженої мережі для домашніх користувачів з використанням запропонованого методу

В результаті моделювання було встановлено, що при збільшенні стійкості структури час пошуку каналів, по яких буде здійснюватися передача запитів, зменшується, що в результаті призведе і до зменшення загального часу передачі сервісу до ЦОД і назад (рис. 3.5)



—— *до впровадження методу пошуку маршруту з урахуванням стійкості структури;*

- - - - *із застосуванням методу пошуку маршруту з урахуванням стійкості структури*

Рис. 3.5 Залежність часу пошуку каналів від ймовірності стійкості структури

3.4. Імітаційне моделювання інтегрованої системи управління з використанням функції NVF

Для дослідження ефективності запропонованих у п. 2.4 рішень, було розроблено імітаційну модель хмаринкової інфраструктури. Імітаційне моделювання процесу обслуговування завжди вимагало від розробника імітаційної моделі перевірки адекватності створеної моделі процесам, що відбуваються в реальній системі масового обслуговування. Найпростіший спосіб визначити характеристики системи обслуговування полягає в отримані експериментальних даних щодо процесу обслуговування. Аналіз цих даних дає змогу визначити, які параметри системи обслуговування необхідно змінити для того, щоб підвищити якість обслуговування, тобто оптимізувати процес [82, 99].

Сучасні системи масового обслуговування містять велику кількість компонентів, кожний з яких є складною системою, яка також має свої параметри та характеристики. Загалом всі ці компоненти впливають на характеристики якості обслуговування системи. А тому для створення адекватної імітаційної моделі та адекватного оцінювання результатів моделювання необхідно врахувати всі компоненти, що беруть участь в процесі обслуговування.

Велика кількість абонентів, програмних додатків та сесій, що генеруються цими додатками, та їхня різноманітність значно впливають на характеристики трафіку, що надходить у систему обслуговування. Тому, щоб змоделювати такий трафік, необхідно застосувати потужний математичний апарат, який би дав змогу більш або менш точно описати характеристики такого трафіку. Зрозуміло, що найефективнішим способом моделювання в такій ситуації є розроблення спеціального програмного забезпечення.

Завдяки програмній реалізації імітаційної моделі можна не тільки повністю реалізувати всі необхідні функції моделі, але і забезпечити контроль над її роботою. Програмне забезпечення дає змогу за допомогою графічного інтерфейсу користувача динамічно змінювати параметри моделі, тим самим оцінити поведінку системи, що моделюється, в конкретній ситуації, яка може виникнути в реальній системі обслуговування. Крім того, програмне забезпечення за допомогою графічної оболонки дає змогу в реальному режимі часу давати оцінку всім параметрам моделі. Це можливо здійснювати за допомогою графіків, діаграм, списків та таблиць, які обновляються в реальному режимі часу.

Основна частина симуляторів телекомунікаційних систем та мереж функціонує по принципу дискретних подій, що не завжди повною мірою відображає особливості процесів, які відбуваються в мережі [87, 91]. В основу розробки запропонованої архітектури покладено модель розгортання віртуальних машин на фізичних серверах та надання сервісу. Структурна схема моделі, розроблена з використанням засобів UML, відображена на рис.3.6

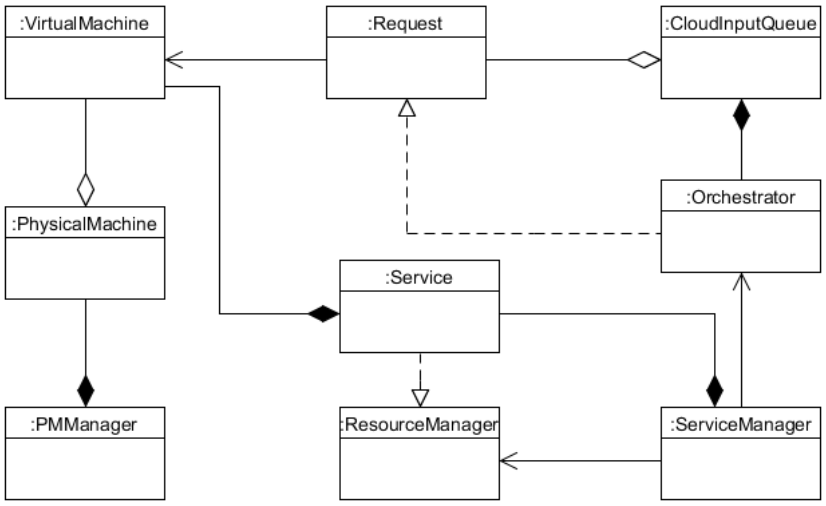


Рис. 3.6 Структурна схема імітаційної моделі

Концептуальна модель надання сервісу користувачеві із розгортанням інфраструктури та запропонованою системою управління і методом балансування відображена на рис. 3.7

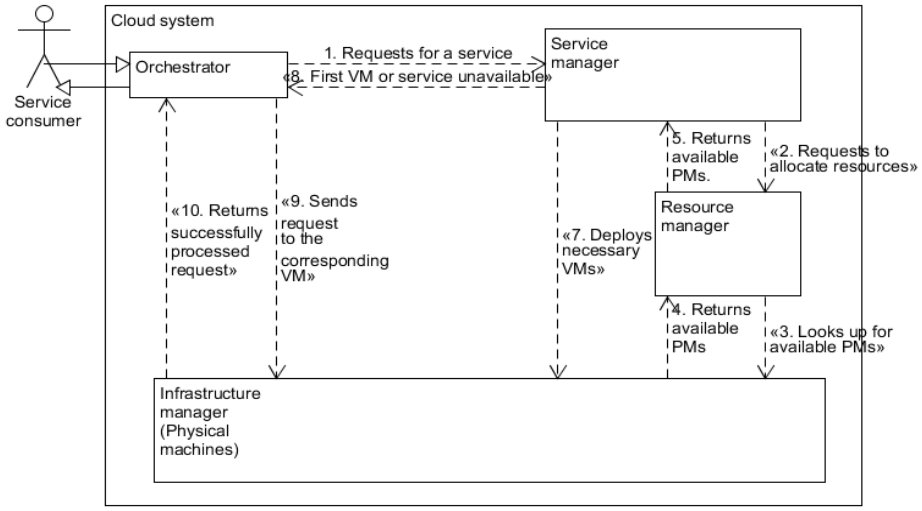


Рис. 3.7 Модель надання сервісу

На основі представленої імітаційної моделі у роботі розроблений програмний інструмент з використанням мови програмування C++ та середовища програмування Qt5.4 [2, 15, 18]. Для створення інфраструктури довільної конфігурації користувач має змогу задати необхідні параметри: кількість фізичних серверів, апаратні ресурси, які виділятимуться на розгортання кожної компоненти, кількість сервісів та їх тип. Всі елементи інфраструктури можуть бути незалежно налаштовані, хоча мають однакову конфігурацію по замовчуванню. З’єднання між ними формуються на основі

рандомізовано заповненої матриці суміжності. Основне вікно програми для налаштування конфігурації мережі та контролю над процесом моделювання відображено на рис. 3.8.

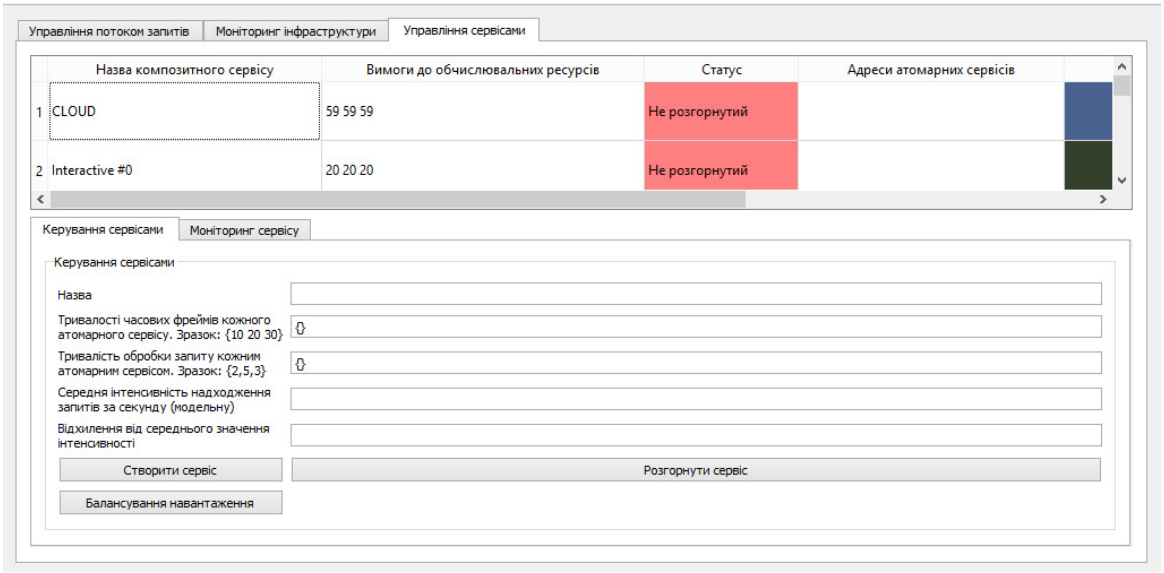


Рис. 3.8 Основне вікно програми для налаштування конфігурації мережі

3.5. Оцінка ефективності методу балансування навантаження на основі аналізу доступних компонентів сервісу

У мережі в ручному режимі створено набір сервісів, які розгортаються на створеній інфраструктурі. Інфраструктура системи представлена у вигляді матриці: по горизонталі – кількість фізичних серверів, по вертикалі – віртуальні машини, розгорнуті на кожному сервері. Параметри цих сервісів подано у таблиці 3.3, а їх розгортання на розробленій інфраструктурі на рис. 3.9

Таблиця 3.3

Параметри сервісів

Назва композитного сервісу	Колір	Вимоги до обчислювальних ресурсів	Адреси атомарних сервісів
1	Синій	{59, 59, 59}	Instance 1 {1001, 2001, 3001}
2	Чорний	{20, 20, 20}	Instance 2 {1002, 1003, 2002}
3	Фіолетовий	{20, 20, 20}	Instance 3 {2003, 3002, 3003}
4	Блакитний	{20, 20, 20}	Instance 4 {4001, 4002, 4003}
5	Зелений	{20, 20, 20}	Instance 5 {4004, 5001, 5002}
6	Бузковий	{20, 20, 20}	Instance 6 {5003, 5004, 6001}

	1	2	3	4	5	6
1	PM #1000 (99%)	VM #1001	VM #1002	VM #1003		
2	PM #2000 (99%)	VM #2001	VM #2002	VM #2003		
3	PM #3000 (99%)	VM #3001	VM #3002	VM #3003		
4	PM #4000 (80%)	VM #4001	VM #4002	VM #4003	VM #4004	
5	PM #5000 (80%)	VM #5001	VM #5002	VM #5003	VM #5004	
6	PM #6000 (20%)	VM #6001				
7	PM #7000 (0%)					
8	PM #8000 (0%)					
9	PM #9000 (0%)					
10	PM #10000 (0%)					

Рис. 3.9 Інфраструктура сервісно-орієнтованої системи

Для генерації трафіку використовуються генератори на основі логнормального (інтервал натходження запитів) та експоненціального (інтенсивність натходження) законів розподілу, які у поєднанні дають змогу отримати мультисервісний трафік з характеристиками, близькими до трафіку реальної сервісно-орієнтованої мережі.

Усі сервіси і генератори трафіку для кожного типу сервісу, активовано одночасно. Тривалість існування віртуальних машин не обмежена. Моделювання відбувається в три етапи.

На першому етапі проводиться аналіз роботи мережі, при функціонуванні відповідно до існуючої архітектури, та принципів сервісно-орієнтованої мережі. На цьому етапі аналізується використання фізичних ресурсів серверів, проводиться порівняльний аналіз їх завантаженості, затримки проходження пакетів з кінця в кінець, кількість опрацьованих та неопрацьованих запитів. Особлива увага у розробленому сценарії відводиться сервісу з найбільшою кількістю не опрацьованих запитів.

На другому етапі проводиться моніторинг інфраструктури системи та тривалості обслуговування запитів. Використання інтегрованої архітектури управління дасть змогу контролювати доступні апаратні та програмні ресурси.

На третьому етапі, вмикається алгоритм балансування навантаження для сервісів тривалість обслуговування запитів є найбільшою. Завдяки узгодженій роботі алгоритму балансування навантаження та інтегрованому управлінню інфраструктурою очікується зменшити тривалість обслуговування запитів, затримку передавання пакетів, що в цілому має підвищити продуктивність та якість мережі, та розвантажити сервер.

Моделювання проводилося у два етапи, без застосування запропонованих рішень та з їх застосуванням. Інтенсивність надходження запитів на кожен із сервісів наведено на рис. 3.10

На рис. 3.11 а, б, в, г, д, е відображено тривалість обслуговування запитів на надання кожного типу сервісу. У момент переходу з першого етапу на другий спостерігається зниження цього параметру та здійснюється контроль за ресурсами кожного серверу.

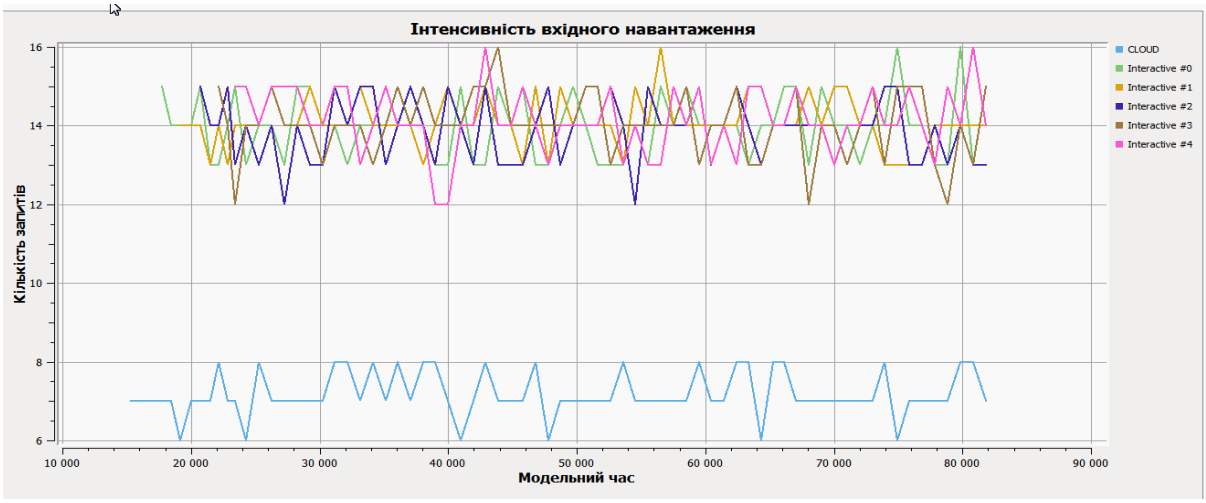
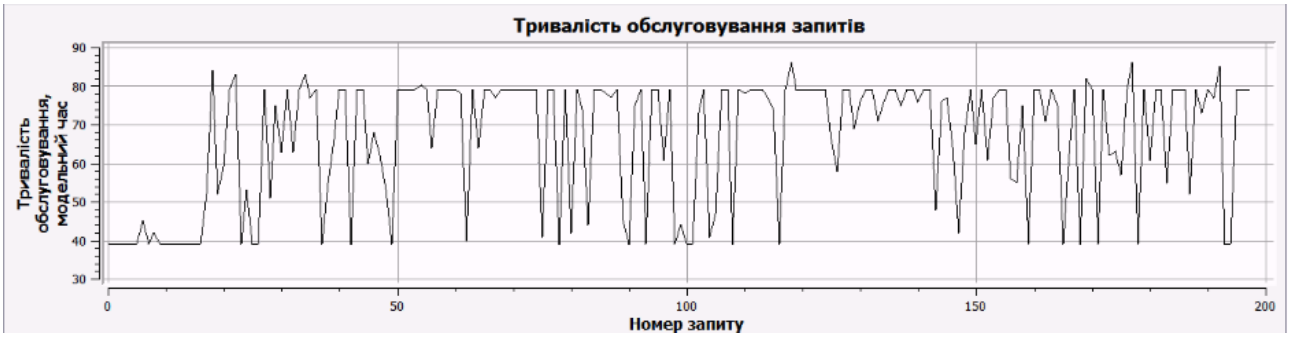


Рис. 3.10 Інтенсивність надходження запитів



а)

б)

в)

г)

д)

е)

Рис. 3.11 Тривалість обслуговування запитів на надання кожного типу сервісу

Параметри кожного із сервісів та тривалість обслуговування запитів на надання кожного з них наведені у таблиці 3.4

Таблиця 3.4

Тривалість обслуговування запитів на надання сервісів

Назва композит-ного сервісу	**Адреса атомарного сервісу**	**Тривалість надання сервісу**	**Кількість запитів на надання сервісу, що надійшли в систему**	**Кількість опрацьованих запитів**
1	Instance 1 {1001, 2001, 3001}	70 мод.сек.	212	212
2	Instance 2 {1002, 1003, 2002}	150 мод.сек.	394	391
3	Instance 3 {2003, 3002, 3003}	130 мод.сек.	367	365
4	Instance 4 {4001, 4002, 4003}	80 мод.сек..	405	404
5	Instance 5 {4004, 5001, 5002}	90 мод.сек.	374	372
6	Instance 6 {5003, 5004, 6001}	60 мод.сек.	332	332

З результатів представлених на рис. 3.11 та в таблиці видно, що найбільше запитів надходить на другий сервіс і тривалість їх обробки, всередньому, близько 150 секунд модельного часу. Одна секунда модельного часу рівна одній мікросекунді реального часу моделювання стану системи. Відповідно, апаратних та програмних ресурсів для надання ще однієї компоненти другого сервісу недостатньо. Буфери віртуальних машин переповнені запитами. У такому випадку Оркестратор приймає рішення про міграцію компонентів сервісу на інший фізичний сервер. На рис. 3.12 наведено вигляд інфраструктури мережі після міграції та ввімкнення алгоритму балансування навантаження.

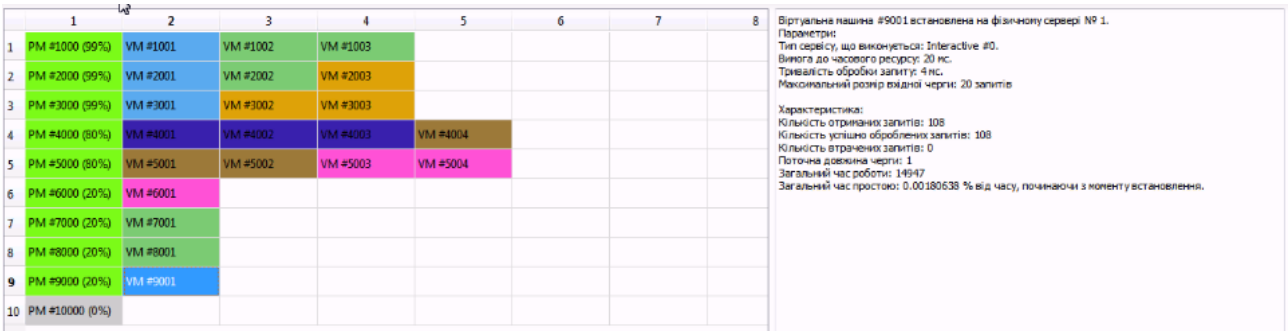


Рис. 3.12 Інфраструктури мережі після міграції та ввімкнення алгоритму балансування навантаження

Після ввімкнення алгоритму балансування навантаження тривалість обслуговування запитів зведені у таблиці 3.5

Таблиця 3.5

Тривалість обслуговування запитів на надання сервісів після ввімкнення алгоритму балансування навантаження

Назва композит-ного сервісу	**Адреса атомарного сервісу**	**Тривалість надання сервісу**	**Кількість запитів на надання сервісу, що надійшли в систему**	**Кількість опрацьованих запитів**
1	Instance 1 {1001, 2001, 3001}	70 мод.сек	407	407
2	Instance 2 {1002, 1003, 2002}	50 мод.сек	739	739
3	Instance 3 {2003, 3002, 3003}	130 мод.сек	711	706
4	Instance 4 {4001, 4002, 4003}	80 мод.сек	695	692
5	Instance 5 {4004, 5001, 5002}	90 мод.сек	732	728
6	Instance 6 {5003, 5004, 6001}	60 мод.сек	740	740

На рис. 3.13 відображено тривалість обслуговування запитів на надання другого типу сервісу після переходу на менш завантажену фізичну машину. Як видно, спостерігається зниження цього параметру та здійснюється контроль за ресурсами кожного серверу.

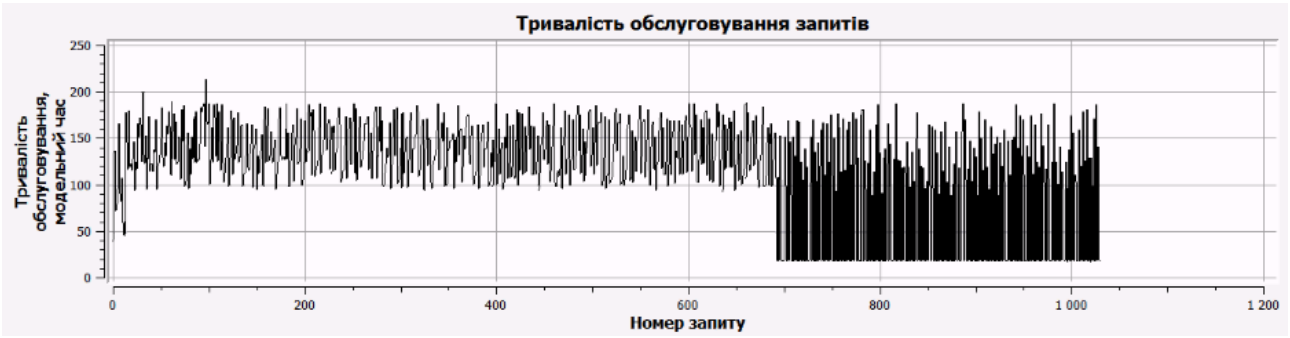


Рис. 3.13 Тривалість обслуговування запитів на надання другого типу сервісу із врахуванням запропонованих рішень

Аналізуючи отримані результати видно, що балансування навантаження за допомогою реалізації інтегрованої архітектури управління з використанням технології NVF дозволяє зменшити тривалість обслуговування запитів, приблизно у 3 рази. Завдяки інтегральній оцінці телекомунікаційних та програмно-апаратних ресурсів, запропонований метод дав змогу зменшити час затримки надання сервісу користувачам та розвантажити найбільш

завантажених сервер. Це особливо важливо в умовах, коли в мережі доступна велика кількість сервісів і передаються великі обсяги трафіку.

Моніторинг якості обслуговування (рис. 3.13) показує, що після перенесення компонентів сервісу середня тривалість обробки запитів та затримка пакетів з кінця в кінець зменшилася з 150 до 55 мс, тобто майже у три рази.

3.6. Висновки до 3-го розділу

1. Для оцінки ефективності моделі надання сервісу на основі методу пошуку маршруту з урахуванням стійкості структури віртуалізованого центру обробки даних розроблено імітаційну модель структури центру обробки даних. В результаті роботи моделі отримано залежність, аналіз якої свідчить про зменшення затримки при стабільності структури мережі, що дає змогу пришвидшити процес надання сервісу кінцевому користувачу і забезпечити необхідний рівень QoS. Після проведення моделювання із застосуванням методу пошуку маршруту з урахуванням стійкості структури віртуалізованого центру обробки даних ЦОД вдалося зменшити час затримки на 35%, що призведе до пришвидшення процесу надання сервісу кінцевому користувачу.

2. В результаті моделювання структури центру обробки даних встановлено, що при зменшенні затримки, час пошуку каналів, по яких буде здійснюватися передавання запитів, зменшується, що в результаті призведе і до зменшення загального часу передавання сервісу з кінця в кінець. В результаті отриманих залежностей можна говорити про загальне зменшення затримки та підвищення якості надання сервісів користувачам сервісно-орієнтованої мережі на 12%. Встановлено, що при збільшенні стійкості структури час пошуку каналів, по яких буде здійснюватися передача запитів, зменшується, що призведе і до зменшення загального часу передачі сервісу.

3. Проведено імітаційне моделювання сервісно-орієнтованої інфраструктури, в основу якої покладено модель розгортання віртуальних машин на фізичних серверах та надання сервісу із використанням методу балансування навантаження на основі аналізу доступних компонентів сервісу.

Аналізуючи отримані результати видно, що балансування навантаження за допомогою реалізації інтегрованої архітектури управління з використанням технології NVF дозволяє зменшити тривалість обслуговування запитів, приблизно у 3 рази. Завдяки інтегральній оцінці телекомунікаційних та програмно-апаратних ресурсів, запропонований метод дав змогу зменшити час затримки надання сервісу користувачам та розвантажити найбільш завантажених сервер. Це особливо важливо в умовах, коли в мережі доступна велика кількість сервісів і передаються великі обсяги трафіку. Моніторинг якості обслуговування показує, що після перенесення компонентів сервісу середня тривалість обробки запитів та затримка пакетів з кінця в кінець зменшилася з 150 до 55 мс.

РОЗДІЛ 4.
ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ НАДАННЯ КОМПОЗИТНИХ СЕРВІСІВ У РОЗПОДІЛЕНИХ ДАТА-ЦЕНТРАХ СЕРВІСНО-ОРІЄНТОВАНИХ МЕРЕЖ

В даному розділі розроблено програмно-апаратний комплекс надання композитних сервісів**,** який враховує управління оптичними ресурсами між ЦОД, що дає змогу проводити моніторинг та управління завантаженістю каналів, оптимального розподілу смуги пропускання кожної хвилі (логічного каналу) одного фізичного каналу, по якому здійснюється надання одного типу сервісу (з набором компонент, що працюють на окремих VM) та із використанням запропонованої математичної моделі надання сервісу, що базується на оцінці стійкості структури, і інтегрованої архітектури системи управління ресурсами, з використанням функції NVF дозволяє підтвердити адекватність отриманих результатів. Запропоновано використання даних методів та алгоритмів в телекомунікаційних структурах в якості основи для забезпечення якості сервісу при побудові сервісно-орієнтованих мереж. Результати, наведені в розділі, опубліковано у роботах [10, 21, 23, 26].

4.1. Модифікація режимів передавання потоків даних у транспортній системі розподілених ЦОД

Людство переживає швидке зростання обсягів використання відео та аудіо контенту, а кожен з користувачів очікує отримати високоякісні IP послуги. Хмарні сервіси можуть бути доступними лише при надійній взаємодії пристроїв у мережі передавання даних та достатній обчислювальній потужності кожного з них. Замість апаратної інфраструктури оператори все частіше обирають автоматизовані сервіси на основі cloud-технологій. Забезпечення такого роду послуг та використання IaaS вимагає великої пропускної здатності фізичних і віртуальних каналів між компонентами усіх рівнів системи передачі. Необхідно також врахувати і особливості маршрутизації потоків у хмарних мережах, особливо коли мова іде про передачу від провайдера послуг до cloud

середовища. Важливим аспектом при цьому постає моніторинг доступної пропускної здатності кожного фізичного каналу в мережі cloud, що дозволить забезпечити необхідні параметри якості надання сервісу кінцевим користувача.

Потік трафіку між центрами обробки даних підпорядковується законам самоподібності та надходить на обслуговування з нерівномірними інтервалами часу поступлення. Постає проблема ефективного розподілу ресурсів оптичного тракту відповідно до вимог трафіку. З одного боку, якщо ресурси каналу розподіляються відповідно до пікової швидкості, то це призведе до нерівномірного розподілу смуги пропускання, оскільки значна її частина буде використовуватися для передачі малогабаритних потоків і, у випадку надходження високо пріоритетного трафіку, не зможе бути звільненою. З іншого боку, не можливо забезпечити відповідний рівень параметрів QoS, якщо ресурси пропускної здатності розподіляються відповідно до середньої швидкісті поступлення пакетів. Тим паче, необхідно буде здійснювати прогнозування інтервалів поступлення запитів та, відповідно, їх обслуговування до центру обробки даних, що не дозволить у повінй мірі забезпечити динаміну гнучкість і роботу такої системи. В такому випадку, необхідно проводити агрегацію потоків трафіку перед його поступленням у буфер центру обробки даних, здійснювати перерахунок маршруту для кожного із агрегованих пакетів та, за необхідності, перерозподіляти ресурси каналів зв’язку відповіно до вимог трафіку. Як наслідок, це призведе до ускладнення процесів маршрутизації та виділення спектру, а також збільшить затримку на надання сервісу з кінця в кінець.

В сервісно-орієнтованої мережі кожен сервіс – це набір атомарних компонентів, які потребують достатньої вільної обчислювальної потужності та пропускної здатності каналів. У таких мережах процес міграції окремого компоненту потребує додаткових мережевих ресурсів. На фізичному рівні хмаринкової інфраструктури застосовують опто-волоконні системи зі спектральним ущільненням [10, 21, 98].

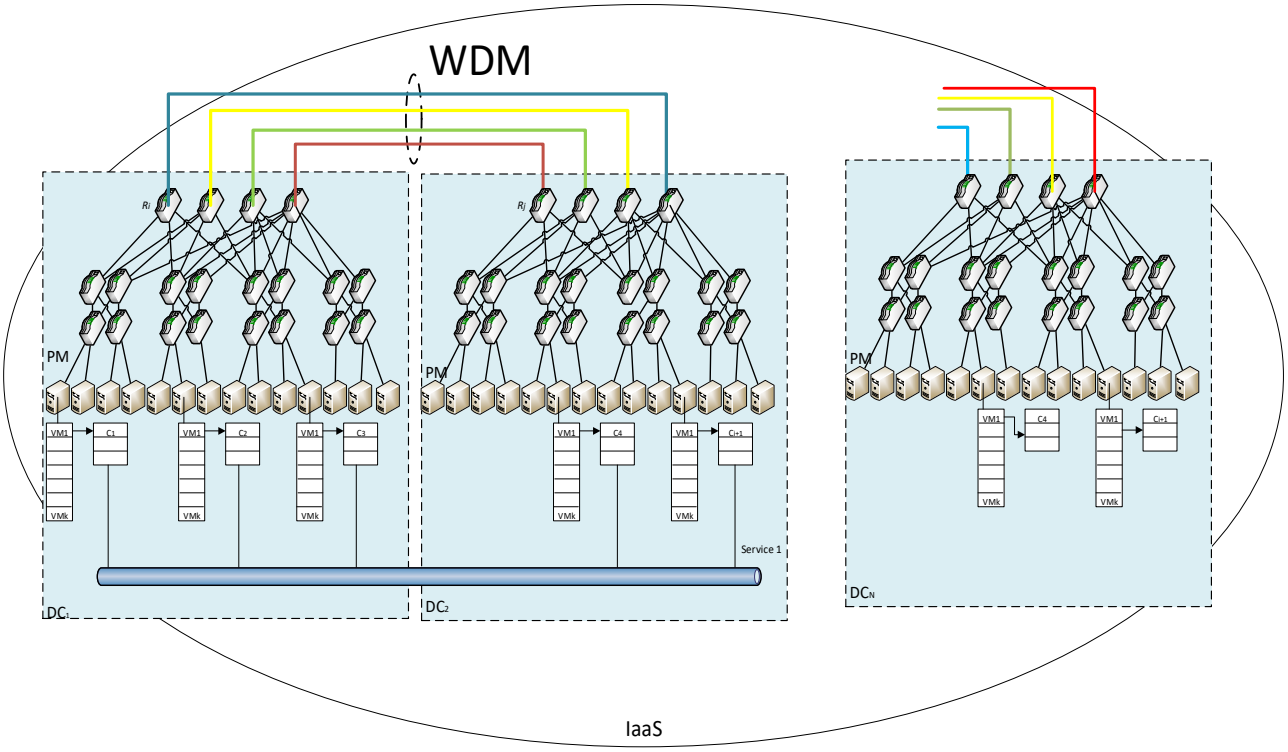


Рис.4.1 Архітектура сервісно-орієнтованої мережі

В процесі оптичної комутації пакети знаходяться в оптичному домені на всьому шляху передавання від першого до останнього мережевого вузла. Цей метод забезпечує високу продуктивність та швидкість передавання інформації, проте його реалізація вимагає вирішення певних проблем. Однією з таких проблем є блокування в оптичній мережі, що зазвичай вирішується встановленням хвильових конверторів та буферизацією сигналу в оптичному домені з використанням оптичних ліній затримки.

Електрична комутація (OTN, сигнал конвертується з оптичного в електричний, комутується, і конвертується в оптичний) - досі єдиний спосіб виконання комутації на суб-лямбда рівні. Майже 70% всього трафіку, який надходить у вузол транспортної мережі є транзитним (цей трафік призначений для іншого вузла) [99, 102, 110]. IP маршрутизатори змушені обробляти увесь транзитний трафік. Збільшення обсягу трафіку як власного (трафік адресований вузлу, на якому він знаходиться), так і транзитного вимагає підвищення потужності IP маршрутизатора. OTN комутація пропонує гнучку форму електричної комутації. Комутація по довжинах хвиль, що виконується оптичними крос-конекторами, не взмозі забезпечити комбінацію та розбиття

різних класів трафіку. Вона поєднує в собі швидкість комутації апаратного рівня з особливостями маршрутизації рівня IP, досягаючи високої продуктивності вузла, за рахунок комутації великої кількості інформації.

Враховуючи можливості комутації в оптичній площині, у роботі пропонується використання двох режимів передачі:

1) Стандартний. При даному режимі передачі пакети запитів на надання компонентів сервісу на граничному маршрутизаторі центру обробки даних групуються в блоки, взалежності від ЦОД призначення і передаються по оптичному тракту. На кожному проміжному вузлі оптичного тракту здійснюється оптоелектронне перетворення для блоку запитів.

2) Прозорий. Сигнальна інформація передається для певної групи блоків, відкриваючи наскрізний оптичний канал на деякий визначений інтервал часу. Часовий інтервал, протягом якого буде здійснюватися передача інформації залежить від інтенсивності надходження пакетів та параметрів QoS.

Слід зазначити, що весь службовий трафік (таблиці комутації, інформація про стан оптичних каналів і т.д) передається на окремо виділеній довжині хвилі $\lambda_{службова}$

4.1.1. Наскрізний режим передавання

Прозорий режим роботи являє собою передачу агрегований пакетів запитів на надання компонентів сервісу без здійснення опто-електронного перетворення на проміжних вузлах. В даному випадку передбачається відкриття наскрізного каналу для передавання даних між двома вузлами з резервуванням необхідної пропускної здатності. За допомогою протоколу CSPF (Constrained Shortest Path First) визначаються проміжні вузли, які формують оптимальний маршрут. Вузол-ініціатор відсилає запит PATH на встановлення з'єднання згідно протоколу RSVP-TE. Повідомлення PATH використовуватиметься протоколом RSVP для резервування спектральних і часових ресурсів кожного сегменту мережі. Вузол-одержувач, отримавши PATH, формує повідомлення RESV і відсилає його до вузла-ініціатора повідомляючи його про відкриття наскрізного каналу.

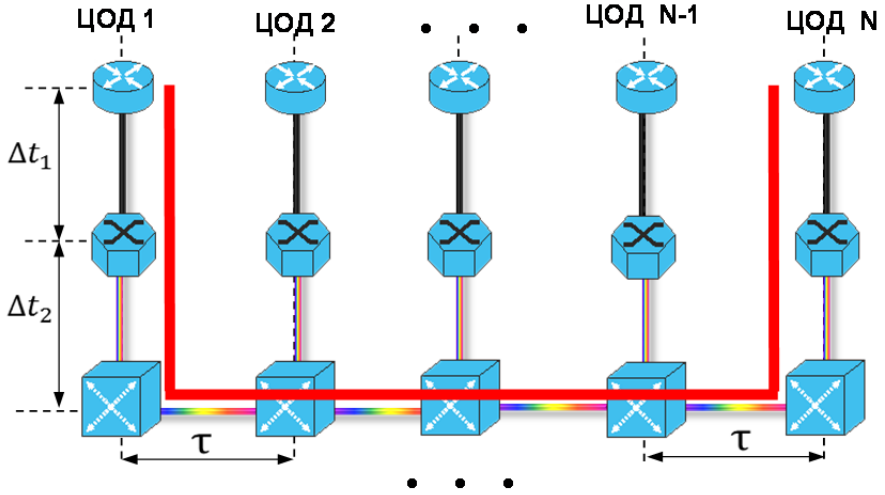


Рис. 4.2 Принцип наскрізної передачі потоків

Передача здійснюється на одній і тій же довжині хвилі або їх сукупності на кожному сегменті мережі. При цьому час передачі при використанні даного режиму передавання визначатиметься:

$$t_{передачі} = 2(\Delta t_1 + \Delta t_2) + (N-1)\tau \qquad (4.1)$$

У місцях, де необхідні довжини хвиль відсутні пропонується встановити хвильові конвертори. У випадку, коли хвильові конвертори не можуть вирішити проблему пошуку вільної хвилі, протокол CSPF вибирає інший маршрут.

В даному режимі вузли можуть виступати як в ролі кінцевих вузлів, так і в ролі проміжних. Прозорий режим передавання може встановлюватися між двома кінцевими вузлами або формуватися посеред маршруту передавання інформації (у випадку, коли інтенсивність поступлення блоків спричиняє збільшення ймовірності блокування роботи даного вузла). На рис. 4.3 наведено приклад мережі у формі графа, де через вузли цієї ж мережі проходять потоки запитів f_i.. Розглянемо потоки f_1 і f_2, які направлені від інших ЦОД до ЦОД 9. На вузлі *3* відбувається агрегація цих потоків і відкривається наскрізний канал до вузла *9* через проміжні вузли *4,6,8*. Такий самий принцип формування прозорого каналу для потоків f_5 і f_6. Вузол *13* відкриває наскрізний канал на замовлення до вузла *10* що відображається потоком f_4 [23, 100, 115]

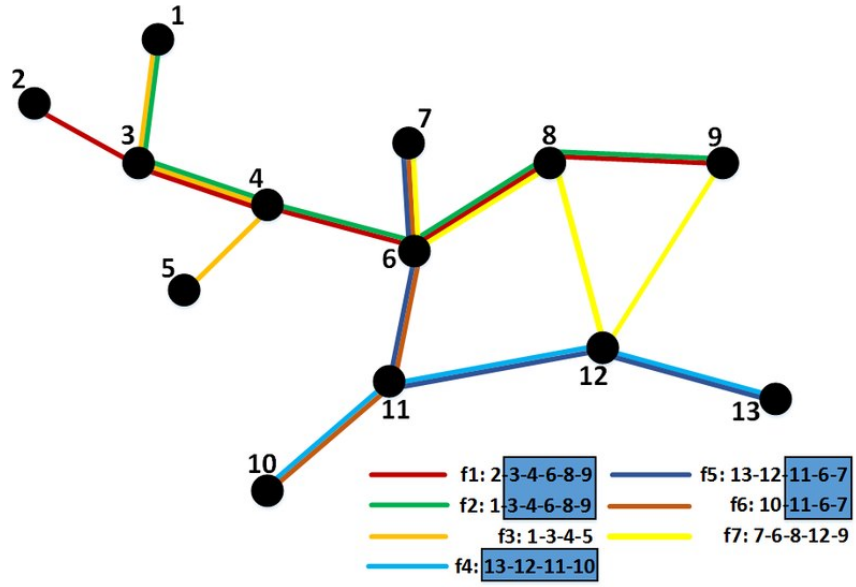


Рис. 4.3 Приклад графу транспортної мережі із потоками, які проходять через її вузли

4.1.2. Стандартний режим передавання

Відповідно до стандартного режиму передавання даних пакети запитів на надання компонентів сервісу на граничному маршрутизаторі центру обробки даних групуються в блоки, взалежності від ЦОД призначення і передаються по оптичному тракту (рис. 4.4). Оскільки стандартний режим передавання даних використовує опто-електронне перетворення, то проблема запису/зчитування полів заголовків кожного запиту відсутня – це реалізується програмним способом.

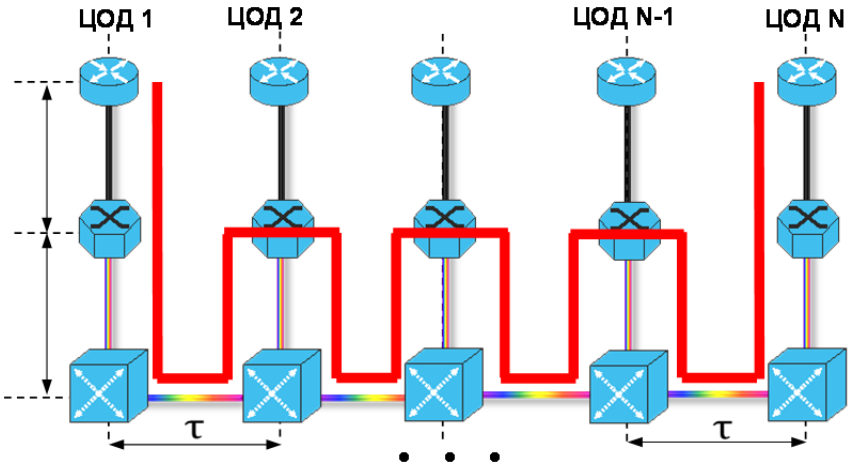


Рис. 4.4 Стандартний режим передавання даних

При цьому час передачі при використанні даного режиму передавання визначатиметься:

$$t_{передачі} = 2(\Delta t_1 + \Delta t_2) + (N-1)\tau + 2\Delta t_2 (N-2) \qquad (4.2)$$

Стандартний режим комутації відбувається на канальному рівні, що дозволяє чітко розділити процес обробки та комутації корисної інформації. Рис. 4.5 ілюструє, як комутатор, розташований між оптичним крос-конектором та пограничним маршрутизатором центру обробки даних, здійснює комутацію агрегованих блоків запитів на надання компонентів сервісу на основі інформації про вузол-призначення.

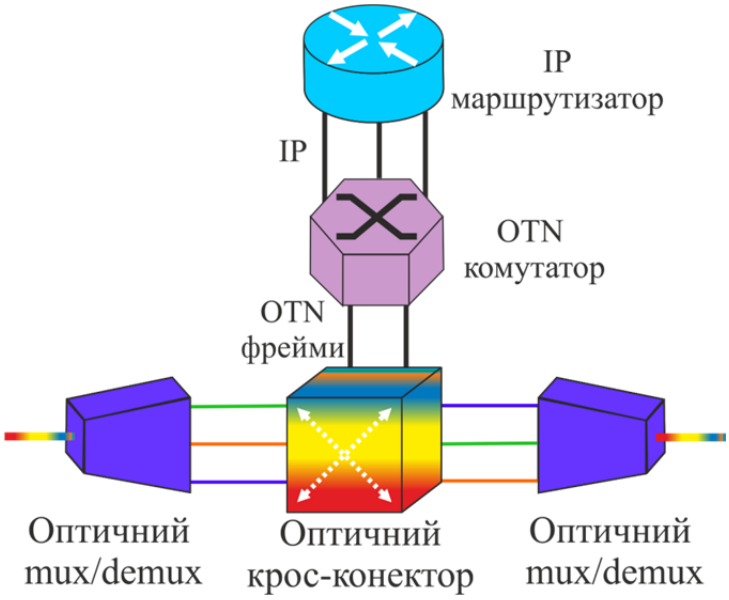


Рис. 4.5 Принцип комутації

Група вхідних мультиплексованих довжин хвиль вводиться у вузол. Демультиплексор виділяє окремі довжини хвиль та вводить їх у оптичний крос-конектор. Оптичний крос-конектор налаштований на крос-комутацію довжин хвиль, що поступають на певні вхідні порти. Довжини хвиль, що переносять дані, які повинні бути опрацьовані в поточному мережевому вузлі, крос-комутуються. Комутатор приймає оптичні сигнали (набір певних довжин хвиль), перетворює їх в електричну форму та відновлює блоки на основі методу попереднього кодування. У процесі поступлення блоків запитів комутатор здійснює демультиплексування блоків, аналізує їх заголовки. Після цього комутаційна фабрика передає скомутовану інформацію на відповідний вихід, де

знову відбувається процес мультиплексування. У контексті транспортної мережі така комутація володіє великою кількістю переваг у порівнянні з комутацією на IP/MPLS рівні. Зокрема, комутуються цілі блоки, а не окремі пакети. Це позитивно впливає на швидкість комутації (обробки), і відповідно, підвищується продуктивність мережі. Варто зазначити, що комутатор покращує ефективність використання довжин хвиль за рахунок мультиплексування низько швидкісних потоків у високо швидкісні потоки. Таким чином, можна уникнути обробки транзитного трафіку на маршрутизаторах.

4.2. Управління оптичними ресурсами між розподіленими центрами обробки даних

Відповідно до принципів пакетної передачі, кількість шляхів в одному напрямку може бути більша ніж 1. Це дозволяє балансувати навантаження та дає змогу ефективніше використовувати ресурси фізичних каналів. Проблема полягає у відсутності методів моніторингу та управління завантаженості каналів, оптимального розподілу смуги пропускання кожної хвилі (логічного каналу) одного фізичного каналу по якому здійснюється надання одного типу сервісу (з набором компонент, що працюють на окремих VM). Це пов’язано з гранульованістю тунелів, які резервуються протоколом RSVP. Прокладання тунелів через мережу базується лише на критерії мінімального завантаження каналів. Наприклад, на крайовий маршрутизатор дата-центру поступають декілька потоків на надання одного і того ж типу сервісу. Кожен сервіс складається з набору атомарних компонент, які потребують різної пропускної здатності. При цьому у кожному каналі буде частка не використаної пропускної здатності. Відповідно до процесу передачі залучено більшу кількість каналів, проте з низькою ефективністю їх використання.

Враховуючи те, що між двома компонентами сервісу в сервісно-орієнтованої мережі існує багато як фізичних так і логічних каналів і ці компоненти можуть бути розташовані у різних дата-центрах, пропонується метод локального управління ресурсами. Суть методу полягає у відкритті/прокладанні наскрізного тунелю між дата-центрами, на яких

розташовані компоненти C_i та C_{i+1}, для об'єднання та перегрупування потоків запитів на надання цього сервісу. Це дозволить гнучкіше перенаправляти потоки запитів на надання тих чи інших компонентів сервісу та удосконалити процес балансування навантаження [10, 21].

Нехай два комутатори можуть самостійно керувати оптичними каналами, які їх з'єднують. Для цього використовується протокол LMP (Link Management Protocol).

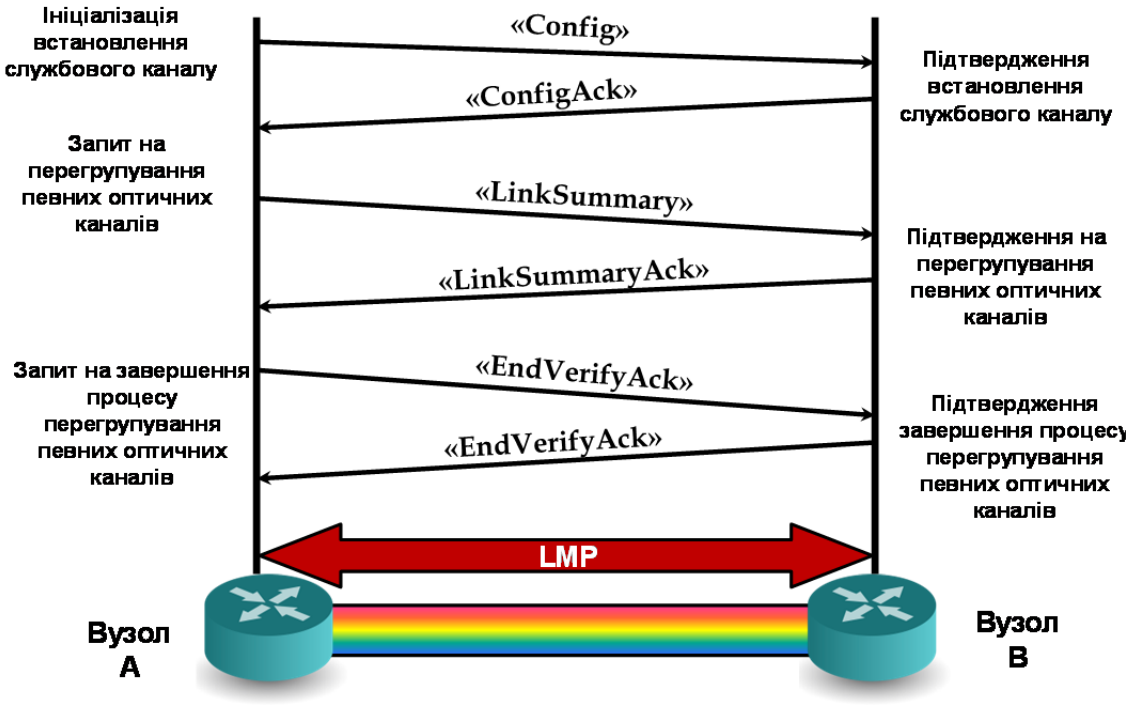


Рис. 4.6 Принцип роботи протоколу LMP

У випадку відкриття/прокладання нового Каналу кожна пара комутаторів намагається максимально ущільнити потоки у довжинах хвиль та волокнах починаючи з першого номера (рис. 4.7). Таким чином правий квадрат завжди вільний для прокладання наскрізних тунелів. Графічно це можна відобразити наступним чином.

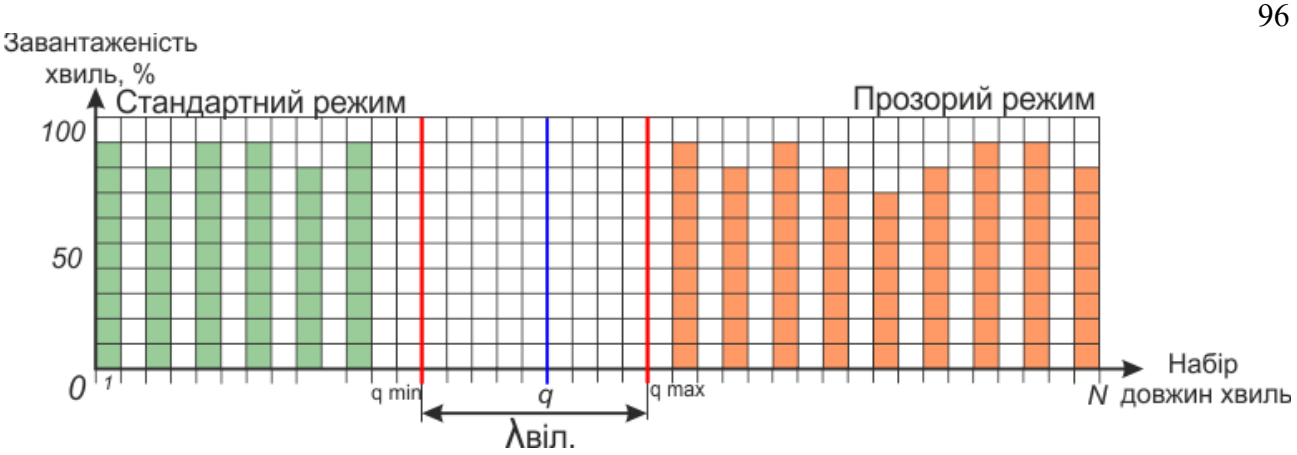


Рис. 4.7 Графічне представлення локального розподілу оптичних ресурсів

Завдяки цьому на більшості сегментів мережі хвилі з великими порядковими номерами (у другому квадраті) будуть використовуватись для прокладання наскрізних каналів. Також слід визначити мінімальний q_{min} та максимальний q_{max} поріг довжин хвиль, який унеможливить випадки, коли при стандартному чи прозорому режимі не виявиться жодної вільної несучої для передавання необхідної інформації, тобто довжини хвиль, що належать проміжку $[q_{min}; q_{max}]$ будуть вільними для використання. Параметри q_{min} і q_{max} залежать від властивостей мережі.

В технології DWDM відстань між несучими може становити від 0,4 до 3,2 нм для третього вікна прозорості. Чим ближче хвилі знаходяться одна до одної, тим більший рівень перехресних завад вони створюють [109, 115]. Звідси випливає, щоб частково зменшити взаємні впливи між оптичними каналами їх слід вибирати з доступного набору таким чином, щоб відстань між двома несучими була рівною $N \cdot \Delta\lambda$, де $\Delta\lambda$ - відстань між оптичними хвилями, N - набір сусідніх довжин хвиль, які не використовуються в даний момент часу. На рис. 4.8 представлено методику вибору хвиль при якій параметр $N=2$.

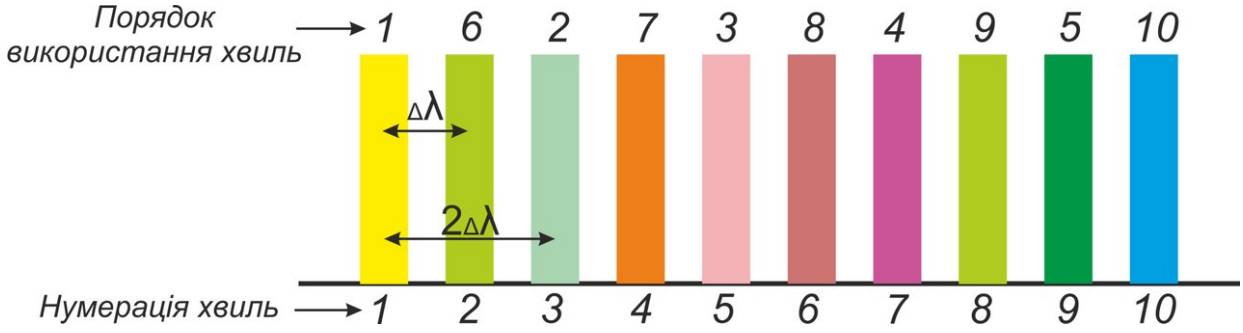


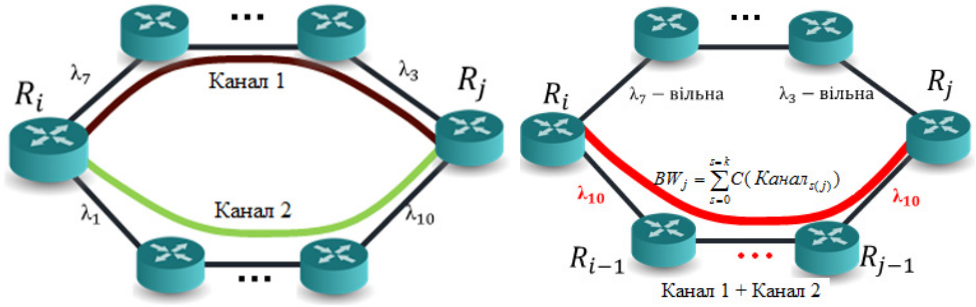
Рис. 4.8 Методика вибору хвиль для N=2

Як видно з прикладу, представленого на рис. 4.8, з десяти доступних хвиль перші п’ять вибираються через одну, яка не використовується в даний момент часу, тим самим збільшуючи відстань між оптичними каналами передавання до $2\Delta\lambda$. Якщо в певний момент часу зайнято 5 довжин хвиль і виникає необхідність використання шостої довжини хвилі(згідно нумерації це друга хвиля) то відстань між першою, другою і третьою хвилею зменшиться до $\Delta\lambda$. Проте відстань між третьою, п’ятою, сьомою і дев’ятою хвилею все ще становитиме $2\Delta\lambda$. Ця методика дозполяє мінімізувати взаємні впливи між оптичними несучими та призведе до зменшення рівня шуму в каналі зв’язку.

4.2.1. Модель управління мережними ресурсами між дата-центрами

Нехай провайдер надання послуг володіє своєю інфраструктурою з N дата-центрів (DC_N), які з’єднані між собою транспортною мережею зі спектральним ущільненням каналів [23, 99]. Взаємодію між цими центрами відслідковує програмний Оркестратор, який встановлюється на одному із DC_N відповідно до архітектури наведеної на рис. 4.1.

Для визначення можливості прокладання наскрізного тунелю використовується протокол OSPF з модифікацією, відповідно до якої, Оркестратор DC1 на якому знаходиться компонент C_i у режимі пошуку оптимального маршруту до C_{i+1} отримує всі можливі доступні довжини хвиль для наскрізного передавання. Модифікація OSPF полягає у врахуванні довжини наскрізного каналу в метриці протоколу, що дасть змогу уникнути прокладання наскрізного каналу, довжина якого негативно вплине на показник відношення сигнал/шум та коефіцієнт бітових помилок. Після цього використовується протокол RSVP, щоб зарезервувати наскрізний канал на кожному комутаторі шляху. При цьому кожному вузлу передається адреса вузла ініціатора (рис. 4.9)



а) б)

Рис. 4.9 Принцип передачі даних а) в стандартному режимі; б) при наскрізній передачі

Суть методу відкриття наскрізного каналу така:

Крок 1: При високій інтенсивності трафіку, що передається між двома вузлами R_i – R_j , приймається до уваги що кожен вузол, при стандартному режимі передавання, може здійснювати перегрупування потоків, максимально завантажувати довжини хвилі. Крім того, ці вузли повинні формувати масив даних $M_{каналів}[\_]$, який містить інформацію про пропускні здатності кожного каналу, що проходять через нього. Оновлення цього масиву буде здійснюватися через певний сталий проміжок часу *t*.

Крок 2: Оркестратор визначає всі пропускні здатності, що використовуються не ефективно на шляху між вузлами $R_i – R_j$. та формує новий масив $M_{FEC_j}[\_]$.

Крок 3: Коли сумарна пропускна здатність таких каналів $BW_j = \sum_{s=0}^{s=k} C(Канал_{s(j)})$ не менше k_1.C, де C – пропускна здатність однієї довжини хвилі, k_1.– коефіцієнт використання довжини хвилі, при якому буде здійснюватися відкриття наскрізного каналу, за умови $0{,}5<k_1<1$. Параметр k_1 вибирається адміністратором мережі, оскільки різні мережі характеризуються різними типами трафіку.

Крок 5: Для відкриття єдиного наскрізного каналу формується масив проміжних вузлів шляху $M_{R_i-R_j} = \{R_{i+1}...R_{j-1}\}$.

Крок 6: Наскрізний канал вздовж всього маршруту буде передавати дані на одній довжині хвилі. Тому, кожен вузол формує масив даних

$M_{\lambda(i+1,j-1)} = \{\lambda_{вільн\_1} ... \lambda_{вільN}\}$, в якому міститься інформація про вільні довжини хвиль на даний момент часу (6).

Крок 7: R_i після отримання інформації від усіх проміжних вузлів починає пошук спільної вільної довжини хвилі для всього маршруту. Якщо така хвиля була знайдена, то вона видаляється з масиву вільних довжин хвиль на кожному проміжному вузлі. Вузол R_j відсилає повідомлення до вузла ініціатора наскрізного каналу про підтвердження встановлення тунелю. Тепер Ri має необхідний набір даних для відкриття наскрізного каналу.

Крок 8: У випадку, коли спільна вільна хвиля не була знайдена, то відбувається вибір нових проміжних вузлів.

Закриття наскрізного каналу буде відбуватися, коли пропускна здатність довжини хвилі буде меншою за $k_2 \cdot C$, де k_2 - коефіцієнт використання довжини хвилі, при якому буде здійснюватися закриття наскрізного каналу. Повинна виконуватись наступна умова, що $k_2<0,3$, оскільки закриття наскрізного каналу має відбутися при суттєвому зниженню поступаючого трафіку, з врахуванням його сплесковості і нерівномірності.

Проте у нашому випадку можливе використання хвильової конвертації для заміни довжин хвиль оптичних сигналів. Основною перевагою хвильових конверторів є забезпечення більшої кількості оптичних шляхів та відкриття наскрізних каналів, що переносять дані на одній довжині хвилі. В свою чергу, велика вартість та складність реалізації обмежують їхнє використання в мережах. Тому, здійснення конвертації довжин хвиль оптичних сигналів пропонується здійснювати на окремих попередньо визначених вузлах. Таким чином, певні мережеві вузли будуть проводити заміну всієї довжини хвилі при потребі. Такий варіант зменшить складність та капітальні витрати апаратного забезпечення, аніж здійснення конвертації на кожному із вузлів, а також ефективніше використання ресурсів каналу у порівнянні із мережами без хвильової конвертації. Пропонується розміщувати хвильові конвертори у вузлах мережі, де є висока імовірність відсутності вільної спільної довжини

хвилі. Принцип відкриття/закриття наскрізного каналу представлений на рис. 4.10

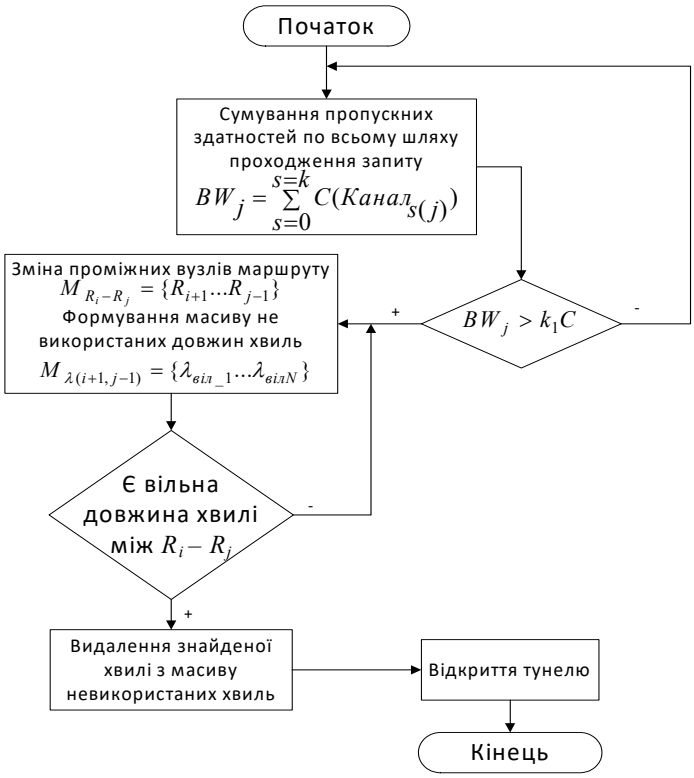


Рис. 4.10 Блок-схема алгоритму відкриття/закриття наскрізного каналу передавання в оптичній мережі ЦОД

Для дослідження ефективності запропонованої системи управління оптичними ресурсами між ЦОД, розроблено імітаційну модель транспортної оптичної мережі, що з’єднує між собою декілька географічно рознесених центри обробки даних.

Основна частина симуляторів телекомунікаційних систем та мереж функціонує по принципу дискретних подій, що не завжди повною мірою відображає особливості процесів, які відбуваються в мережі [104, 125]. Структурна схема моделі такого вузла, розроблена з використанням засобів UML, відображена на рис. 4.11.

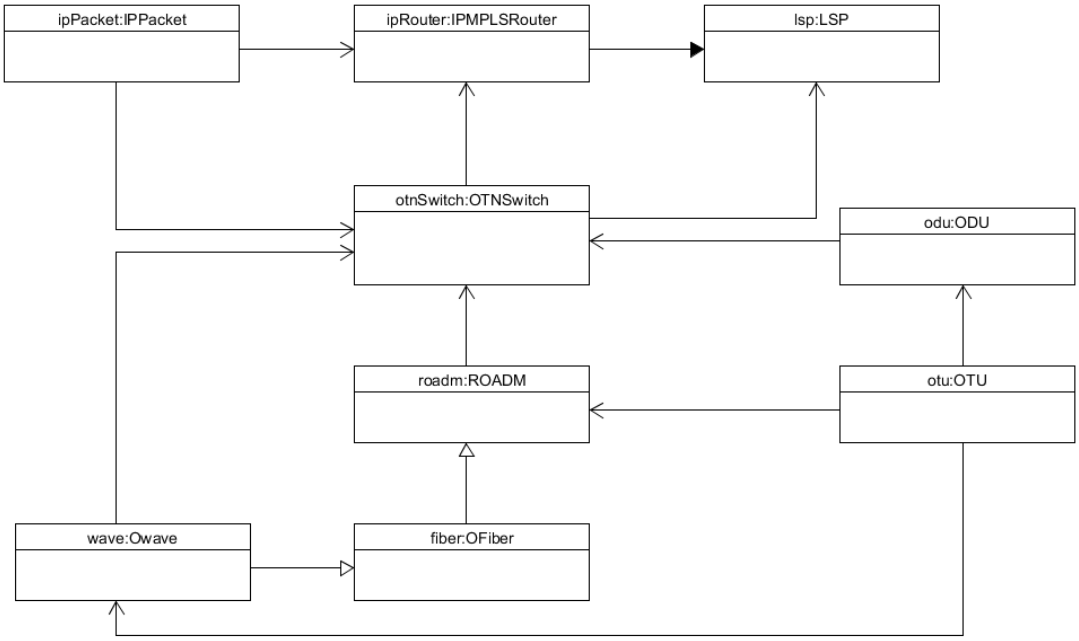


Рис. 4.11 Структурна схема імітаційної моделі вузла транспортної мережі, що з’єднує між собою декілька географічно рознесених центри обробки даних

Основними елементами моделі, які представляють логічні рівні запропонованої архітектури транспортної мережі, є:

- ROADM - реконфігурований мультиплексор вводу/виводу. Він має набір портів для підключення оптичних волокон та таблицю комутації, відповідно до якої хвиля або комутується для перетворення в електричний сигнал, або для направлення в інше волокно. Таблиця комутації формується та змінюється Оркестратором та використовує службові інтерфейси ROADM.
- Комутатор - складається з комутаційної фабрики, таблиці комутації, агрегатора блоків, мультиплексора та демультиплексора. Налаштування параметрів роботи усіх комутаторів мережі здійснює Оркестратор. Комутатор має здатність адаптуватися під будь які режими передавання даних (може перемикатися як у звичайний режим передавання потоків, так і в будь який із запропонованих у роботі).

Для передавання як службових, так і інформаційних повідомлень між рівнями використовуються буфери. Таким чином, використовуючи розроблену

модель вузла, дослідник має змогу створити модель оптичної мережі довільної конфігурації.

На основі представленої імітаційної моделі у роботі розроблений програмний інструмент з використанням мови програмування С++ та середовища програмування Qt5.4. Для створення мережі довільної конфігурації можна задати необхідні параметри в таблиці, яка відграє роль матриці суміжності. Всі вузли мережі можуть бути незалежно налаштовані, хоча мають однакову конфігурацію по замовчуванню. З'єднання між вузлами формуються на основі заповненої матриці суміжності. Для створення зв'язку між вузлами необхідно на перехресті відповідного номера рядка та стовпця задати з використанням розділового знаку "/" наступні параметри: кількість волокон, кількість довжин хвиль/пропускну здатність довжини хвилі. Основне вікно програми для налаштування конфігурації мережі та контролю над процесом моделювання відображено на рис. 4.12

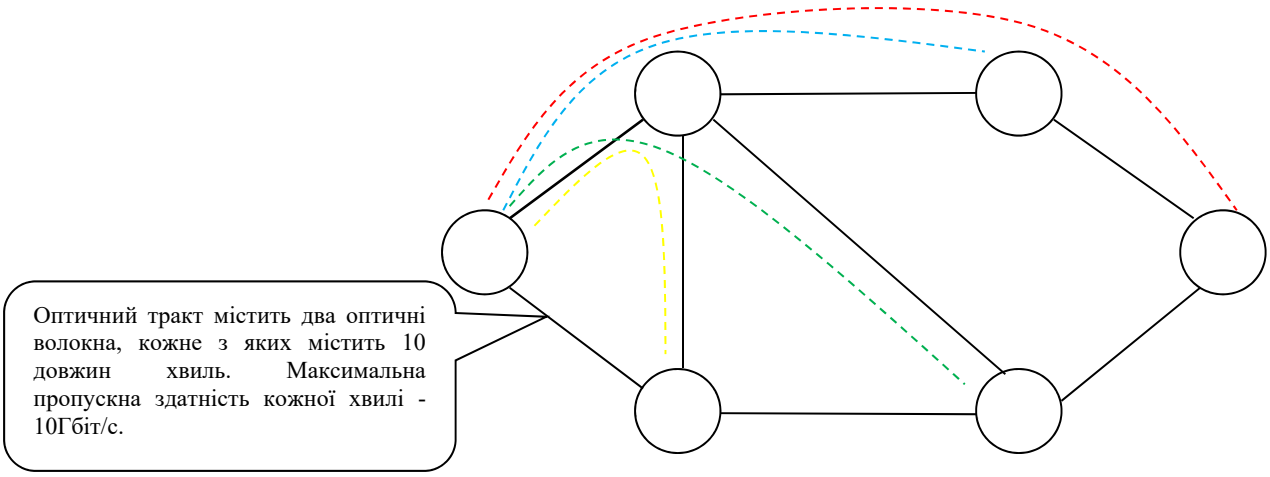


Рис. 4.12 Схема створеної моделі транспортної оптичної мережі

У мережі в ручному режимі створено набір тунелів. Параметри цих тунелів подано у таблиці 4.1.

Таблиця 4.1

Параметри тунелів

Маршрут	Колір	Вузли	Пріоритет	Режим	Середня інтенсивність навантаження, Гбіт/с	Зарезервована пропускна здатність, Гбіт/с
1	Червоний	1-2-3-4	1	Стандартний/Наскрізний	4,3	6
2	Синій	1-2-3	2	Стандартний	3,6	5
3	Зелений	1-2-5	3	Стандартний	5,4	8
4	Жовтий	1-2-6	1	Стандартний	2,3	4

Для генерації трафіку використовуються генератори на основі логнормального (міжпакетний інтервал) та експоненціального (розмір пакету) законів розподілу, які у поєднанні дають змогу отримати мультисервісний трафік з характеристиками, близькими до трафіку реальних транспортних мереж, а також мережі Інтернет.

Всі тунелі і генератори трафіку для кожного них тунелю активовано одночасно. Тривалість існування тунелів не обмежена. Моделювання відбувається в три етапи.

На першому етапі проводиться аналіз роботи мережі, при функціонуванні відповідно до існуючої архітектури, та принципів транспортної оптичної мережі. А саме: на кожному вузлі здійснюється демультиплексування потоків та комутація. На цьому етапі аналізується використання фізичних ресурсів комутаторів всіх рівнів, проводиться порівняльний аналіз їх енергоспоживання, затримки проходження пакетів з кінця в кінець, ефективність використання пропускної здатності довжин хвиль та оптичних волокон. Особлива увага у розробленому сценарії відводиться сегменту мережі, сформованому вузлами 1 та 2, та вузлу 2, оскільки саме через них проходять створені тунелі.

На другому етапі вмикається комутація на канальному рівні , яка повинна забезпечити зменшення часу передавання пакетів з кінця в кінець, знизити завантаження пограничних вузлів ЦОД.

На третьому етапі, вмикається алгоритм прокладання наскрізних тунелів та повністю запускається система локального розподілу та управління ресурсами між ЦОЖ. Завдяки узгодженій роботі алгоритму прокладання

наскрізних тунелів, системи управління та канальній комутації очікується зменшити кількість задіяних довжин хвиль, затримку передавання для пакетів першого маршруту, що в цілому має підвищити продуктивність та якість мережі, та розвантажити другий вузол.

Для проведення описаних вище експериментів необхідно встановити наступні параметри моделі:

Таблиця 4.2

Вихідні параметри моделювання

Параметр	**Значення**
Кількість вузлів	6
Кількість волокон між двома сусідніми вузлами	2
Кількість довжин хвиль в одному волокні	10
Максимальна швидкість передавання на одній довжині хвилі	10 Гбіт/с
Максимальна продуктивність мережевого вузла	200 Гбіт/с

Використовуючи наведені в таблицях 4.1 та 4.2 параметри, та на основі графового представлення мережі (рис. 4.12), формуємо матрицю суміжності. Основна панель керування програмною імітаційною моделлю відображена на рис. 4.13

Моделювання проводилося у два етапи, без застосування запропонованих рішень та із застосуванням. На рис. 4.14 відображено ефективність використання ресурсів мережевого вузла. У момент переходу з першого етапу на другий спостерігається зниження завантаженості апаратних ресурсів та енергоспоживання.

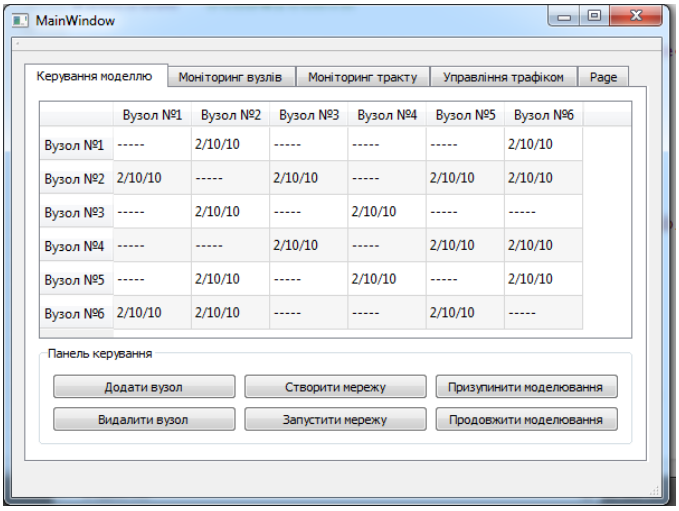


Рис. 4.13 Головна панель керування програмною імітаційною моделлю

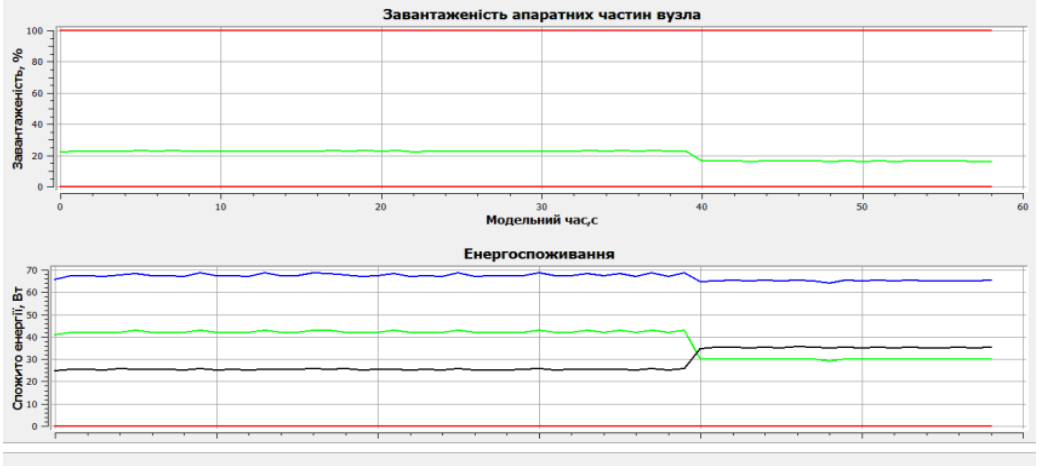
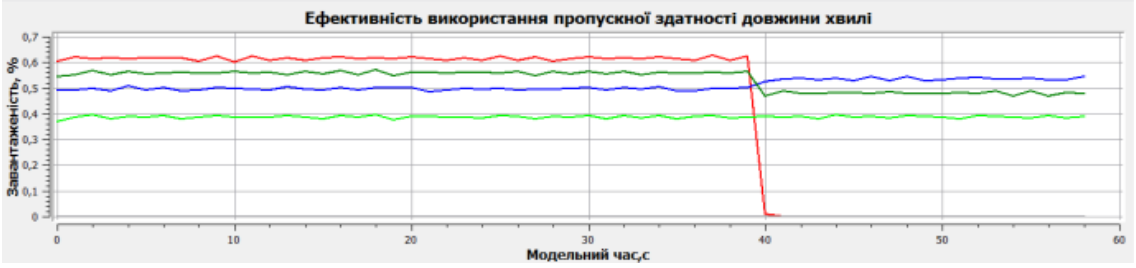


Рис. 4.14 Ефективність використання ресурсів мережевого вузла та його енергоспоживання

Ефективність використання ресурсів довжин хвиль та оптичних волокон для секції (1-2) представлені на рис. 4.15. На кожному графіку відображено стільки кривих, скільки використовується мережею у конкретний момент. У таблиці робочої панелі є можливість вибирати окрему довжину хвилі та проглядати її параметри і характеристики. Програмна модель дає змогу проводити моніторинг завантаження всіх волокон та довжин хвиль, які з'єднують два сусідні вузли.



а)

б)

Рис. 4.15. Завантаженість оптичного тракту між двома вузлами (1 та 2) мережі для (а) першого та (б) другого волокон

На рис.4.16 відображено менеджер управління трафіком, який дає змогу створювати, модифікувати, видаляти віртуальні тунелі та здійснювати моніторинг їх параметрів. На графіку зліва відображено характеристику затримки пакетів для Каналу 1 (табл. 4.2.).

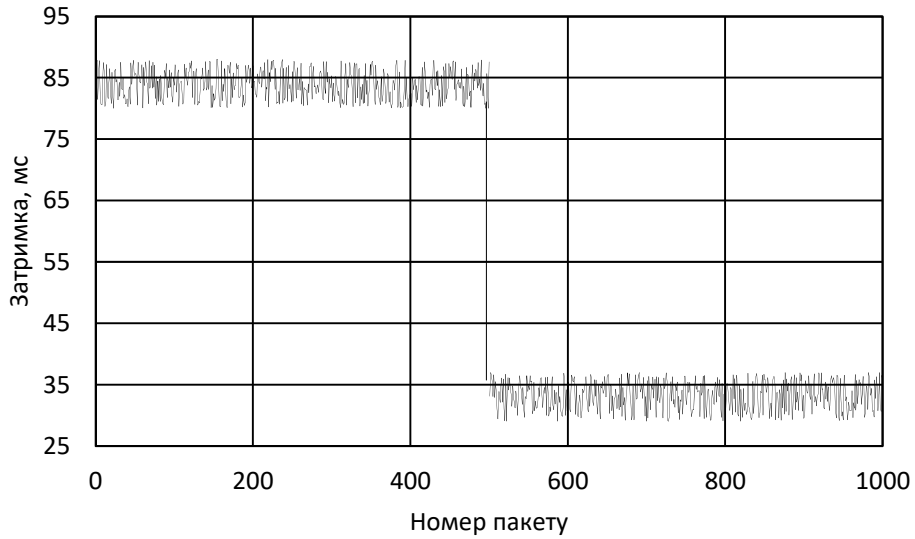


Рис. 4.16. Характеристика тривалості передавання трафіку у стандартному та прозорому режимі режимі передавання

Аналізуючи отримані результати видно, що розроблена система управління оптичними ресурсами між ЦОД забезпечила суттєве зменшення завантаженості мережевого пристрою та його енергоспоживання, приблизно у 1,5 рази. Завдяки використанню алгоритму прокладання наскрізних тунелів, система розвантажила першу довжину хвилі (позначена червоним кольором на рис. 4.15 а) , яка була завантажена більше ніж на 60% та використала одну з вільних довжин хвиль (позначена жовтим кольором на рис. 4.15 б) з високим порядковим номером для прокладання наскрізного тунелю між вузлами 1-4.

Перемикання в режим прозорої передавання дало змогу частково розвантажити найбільш завантажений вузол 2. Це особливо важливо в умовах, коли в мережі передаються великі обсяги запитів на одні і ті ж компоненти сервісу. В таких умовах мережевий вузол змушений обробляти як власний, так і транзитний, в електричному домені, та ще й дуже часто на мережевому рівні. У нашому випадку завдяки інтегрованій комутації вдалося уникнути високої затримки пакетів та енергоспоживання, спричинених комутацією.

Моніторинг якості обслуговування для Каналу 1 (рис. 4.16) показує, що після перемикання у прозорий режим передавання середня затримка пакетів з кінця в кінець зменшилася з 82 до 28 мс, тобто майже у чотири рази. Зрозуміло, що чим більше оминається вузлів, де відбувається перетворення сигналу з оптичного домену в електричний, тим більший буде виграш по затримці та енергоспоживанню. Таке суттєве зменшення затримки при передачі запитів на надання компонентів сервісів з кінця в кінець в транспортній системі ЦОД дозволяє пришвидшити процес їх обробки та зменшити час надання сервісу кінцевому користувачу. Як наслідок, це призведе до підвищення рівня якості обслуговування користувачів.

4.3. Розробка програмно-апаратного комплексу надання композитних сервісів із гарантованим рівнем QoS

На сьогоднішній день існує велика кількість програмних засобів, що дозволяють проводити моделювання як окремих мережевих пристроїв так і мережі в цілому. Великі труднощі викликають саме моделювання таких сервісно-орієнтованих систем. Основними засобів моделювання, які використовуються науковцями у всьому світі для тестування їхніх гіпотез та розробок, є: NS (Network Simulator), OPNET, CloudSim, OmNET++. Всі ці засоби дозволяють досліджувати параметри функціонування мережевих вузлів, систем, протоколів та дають змогу впроваджувати власні зміни у конфігурацію моделі того чи іншого пристрою, що дає змогу провести дослідження власних розроблених науковцями алгоритмів чи протоколів. Їх перевагою є те, що необов’язково будувати чи орендувати цілі дата-центри для дослідження, а

можна використовувати програмні аналоги серверів, маршрутизаторів чи комутаторів, розгортаючи на них необхідні додатки чи сервіси досліджуючи мережеві характеристики та параметри. Проте перераховані засоби базуються на принципі моделювання дискретних подій. Істотним недоліком цих засобів є те, що вони використовують статистичні методи для розрахунку стану системи в певний конкретний момент часу. Таким чином, година роботи реальної мережі може моделюватися протягом декількох секунд, що не дозволяє прослідкувати та адекватно оцінити зміни тих чи інших мережевих параметрів. Особливо це прослідковується при моделюванні процесів, які працюють в реальному масштабі часу. Наприклад моделювання топологічної зміни структури мережі, яка вимагає аналізу стійкості та надійності роботи мережевих компонентів, які входять до її складу, формування та обслуговування черг пакетів у маршрутизаторі та, відповідно, оцінка затримки при передачі з кінця в кінець для такої мережі.

З цієї причини у роботі було запропоновано створення цілого програмно-апаратного комплексу, що дозволить на практиці підтвердити ефективність запропонованих методів та алгоритмів, залучаючи при цьому не лише програмну складову, а й комплекс реального мережевого обладнання. Розробка здійснювалася у навчальній лабораторії Навчально – технічного центру мережевих технологій при Національному університеті «Львівська політехніка» з використанням їхнього серверного обладнання та середовища програмування Qt5.2, яке використовує мову програмування C++ (стандарт C++11, 2011р.). Основною перевагою використання даного програмного середовища є те, що написаний у ньому код може бути скомпільований на різні платформи (наприклад Windows, Linux, Mac OS).

На базі лабораторії було створено дві підмережі із фізичних машин (серверів), які локально об’єднані в два різних центри обробки даних . В основі роботи кожного окремо взятого сервера покладено модель розгортання віртуальних машин на серверах та надання сервісу. Для тестування ефективності запропонованих методів та алгоритмів кожен апаратний сервер володіє такими параметрами: центральний процесор - Intel Core i5-2410M 2.30

GHz, оперативна пам’ять - DDR3 6Gb, мережева карта - Realtek PCIe FE Family Controller 1 Gbit/s. На сервері встановлена операційна система Windows 7 Ultimate Service Pack 1 (2009). Загальний вигляд архітектури побудованої мережі представлений на рис. 4.17

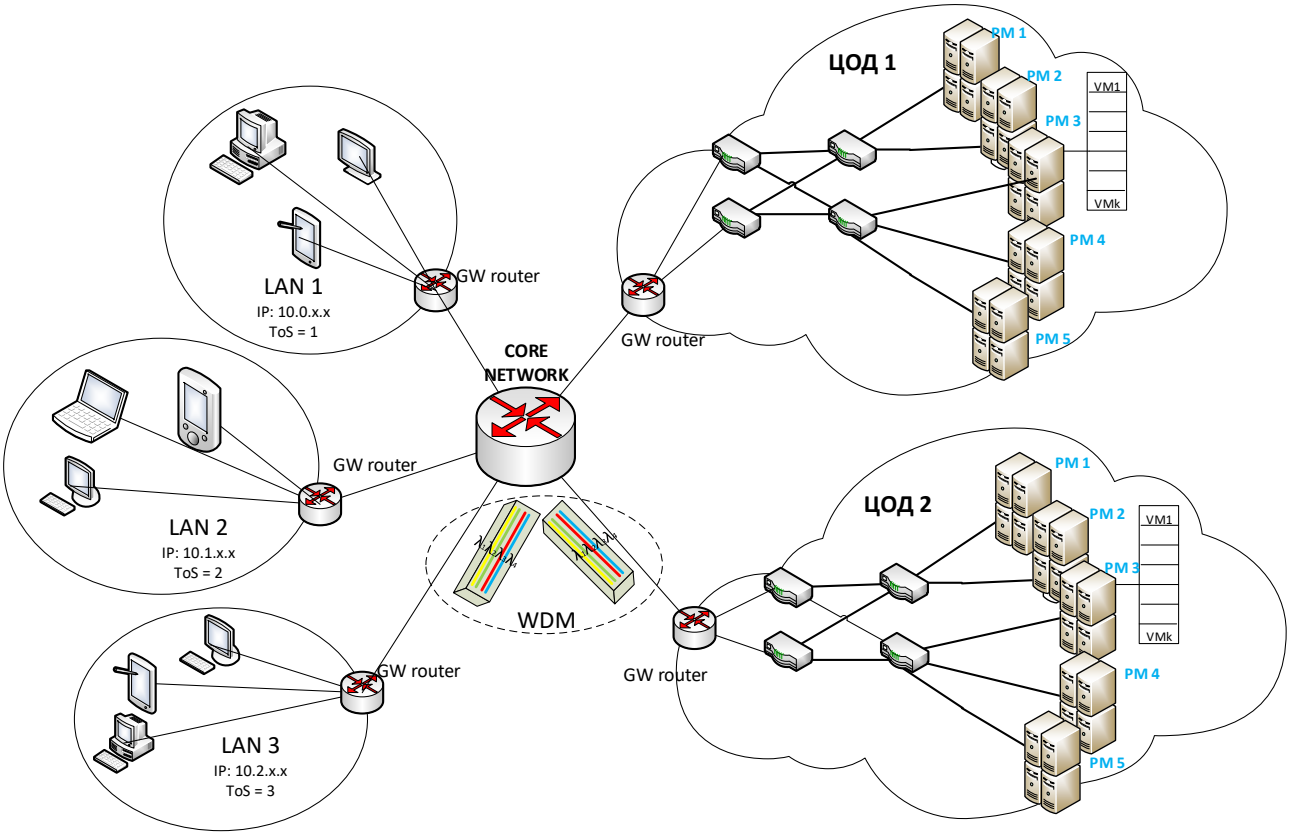


Рис. 4.17 Тестове середовище для надання композитних сервісів з використанням розподілених дата-центрів

Мережами, які підключені до даного ЦОД, є мережі студентських гуртожитків, жителям яких і будуть надаватися хмарні послуги. У зв’язку із неможливістю дослідження транспортної мережі НТЦМТ розроблено програмний аналог, який по функціональності відповідає реальній WDM системі. У мережі згенеровано набір сервісів, які розгортаються на створеній інфраструктурі центрів обробки даних, та функціонують паралельно. Параметри цих сервісів подано у таблиці 4.3, а їх розгортання на розробленій інфраструктурі на рис. 4.18

Таблиця 4.3

Параметри розгорнутих на розробленій інфраструктурі сервісів

Номер сервісу	Назва композитного сервісу	Кількість компонентів сервісу	Максимальний час обробки запиту на компоненту сервісу
Service 1	Docs online	2	100 мс
Service 2	Video streaming	2	400 мс
Service 3	Data Base	2	1 мс
Service 4	Mail	2	400 мс
Service 5	Computing	3	2 мс

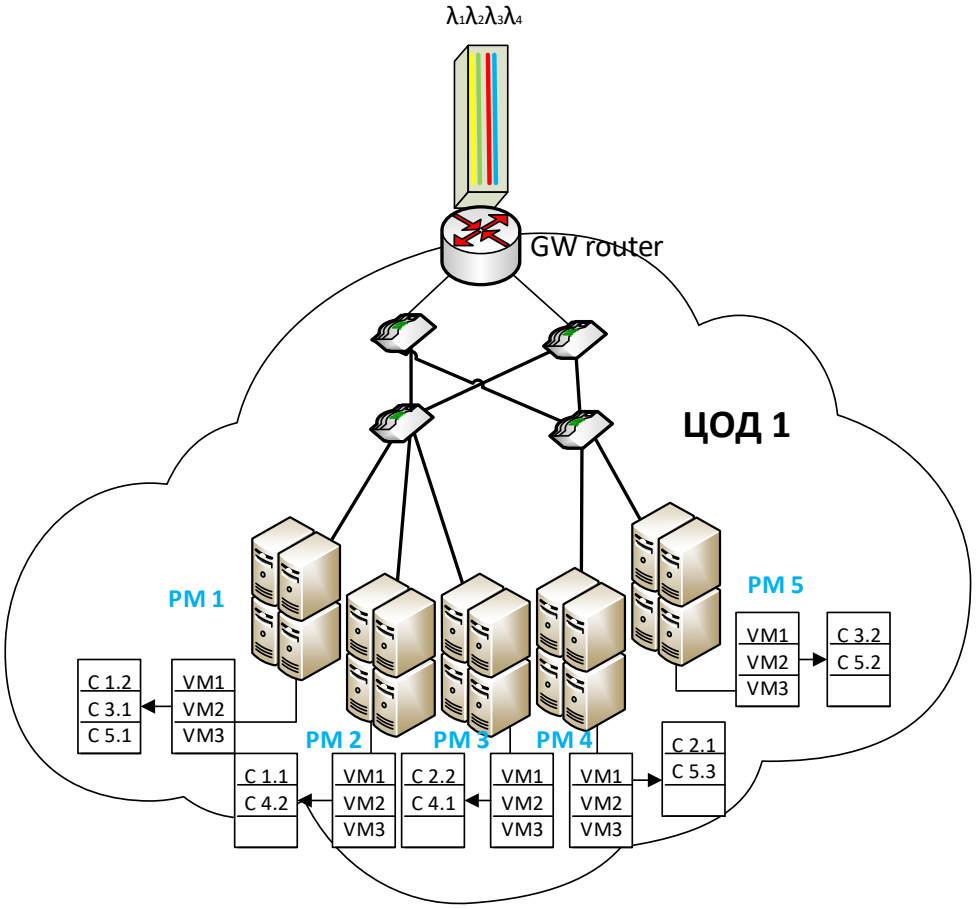


Рис. 4.18 Принцип розгортання компонентів сервісу на ЦОД 1

Кожен компонент сервісу встановлювався на окрему віртуальну машину. Тривалість існування віртуальних машин не обмежена. Для перевірки та тестування роботи системи всі компоненті сервісів на кожному фізичному сервері запущені одночасно, а запити, які надходять на кожну з них і маршрутизуються Гіпервізорами серверів. Загальний контроль та управління такої системи здійснює Оркестратор відповідно до архітектури запропонованої у п. 2.4. Продуктивність конкретної віртуальної машини розраховується як відношення одиниці часу до добутку тривалості обслуговування запиту цією

машиною на загальну кількість всіх віртуальних машин встановлених на сервері. Структурна схема розробленого комплексу із використанням засобів UML, відображена на рис. 4.19

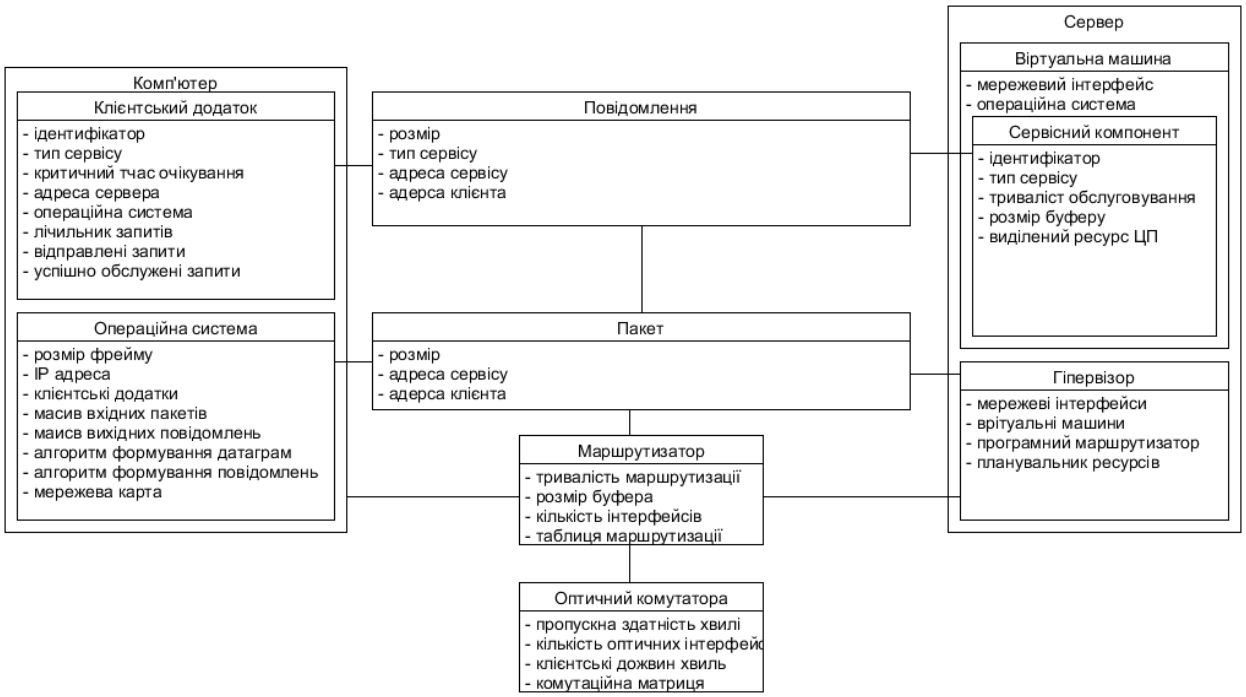


Рис. 4.19 Структурна схема розробленого комплексу

4.3.1 Дослідження якості надання композитних сервісів з використанням програмно-апаратного комплексу

Для дослідження ефективності та адекватності роботи розробленого програмно-апаратного комплексу відповідно до процесів реальної сервісно-орієнтованої мережі необхідно провести його тестування. Тестування системи проводилося у три етапи.

На першому етапі проводиться аналіз роботи мережі, при її функціонуванні згідно розробленої архітектури сервісно-орієнтованої мережі. На цьому етапі аналізується використання фізичних ресурсів серверів, проводився порівняльний аналіз їх завантаженості, затримки проходження пакетів з кінця в кінець, кількість опрацьованих та неопрацьованих запитів. Особлива увага приділялась сервісам з найгіршою якістю надання, тобто з найбільшою кількістю не опрацьованих запитів.

На другому етапі проводився моніторинг інфраструктури системи та тривалості обслуговування запитів. Використання інтегрованої архітектури управління дало змогу контролювати доступні апаратні та програмні ресурси.

На третьому етапі, запускалися усі запропоновані вище методи для сервісів, в яких тривалість обслуговування запитів є найбільшою. Досліджувалася якість надання композитних сервісів для кінцевих користувачів. Під якістю надання сервісу розуміється збільшення затримки та часу обробки запитів на компоненти сервісу.

Тестування проводилося без застосування запропонованих рішень та з їх застосуванням. Інтенсивність надходження запитів на сервер лише з одним розгорнутим сервісом та однією віртуальною машиною наведено на рис. 4.20

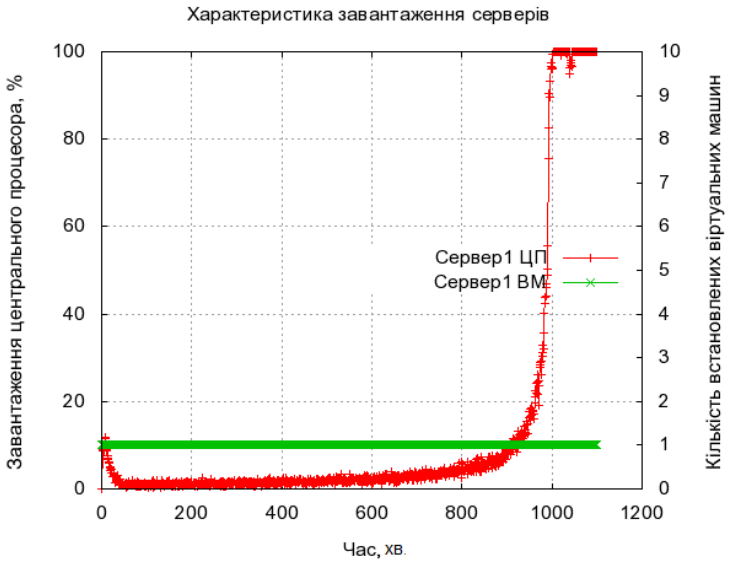


Рис. 4.20 Характеристика завантаження першого серверу

На рис. 4.21 відображено інтенсивність надходження запитів на надання сервісів на ЦОД 1 для користувачів лише однієї LAN протягом 20 годин роботи системи.

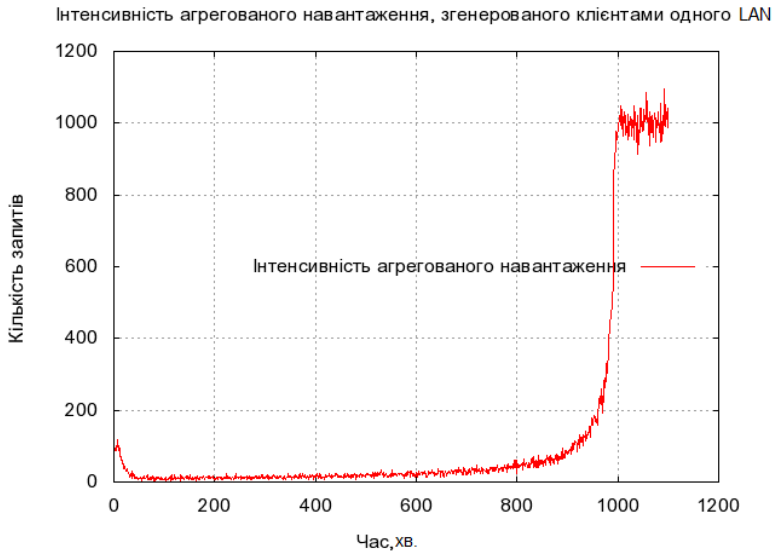
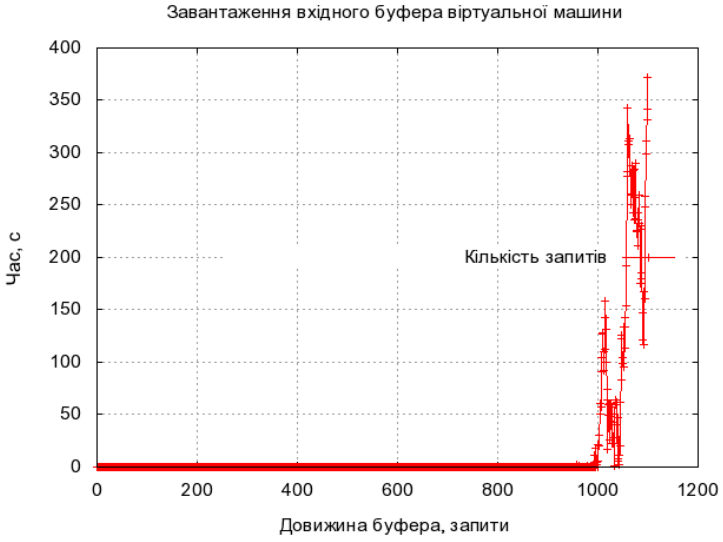
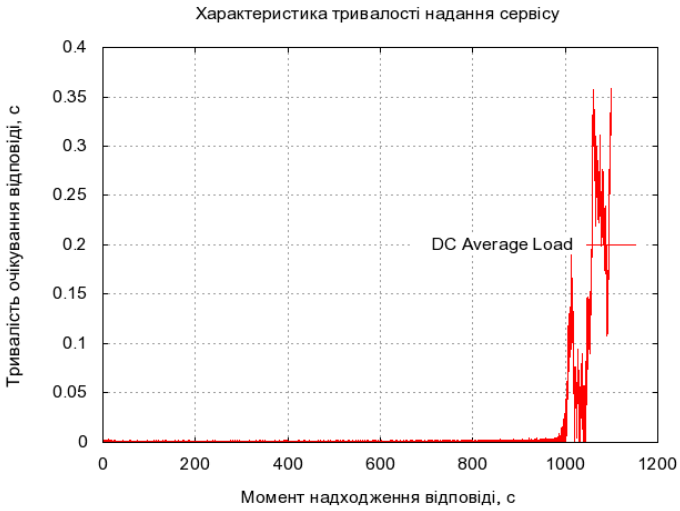


Рис. 4.21 Інтенсивність надходження запитів на надання сервісів на ЦОД 1

Як видно з рис. 4.20 - 4.21 завантаженість сервера зростає по степеневому закону розподілу, особливо у час, коли користувачі знаходяться в гуртожитках і намагаються доступитися до хмарних сервісів. Очевидно, що високий рівень завантаженості негативно впливає на кінцеву якість сприйняття послуги, оскільки зростає час обробки запитів на надання цього сервісу (в даному випадку прослідковується різке збільшення часу відгуку від 0,05 до 0,35 с (рис. 4.22)), за рахунок переповнення буферів доступних віртуальних машин.



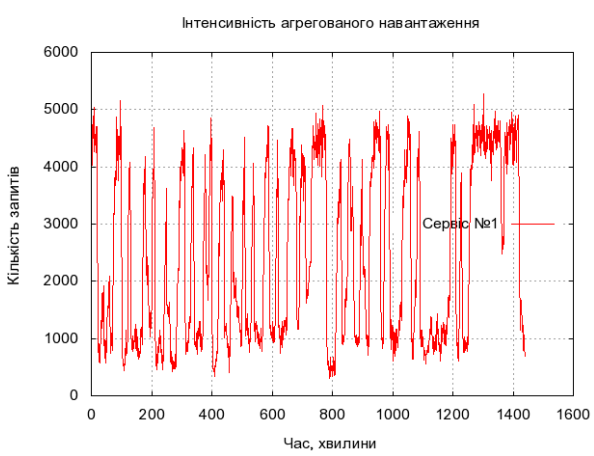
а)



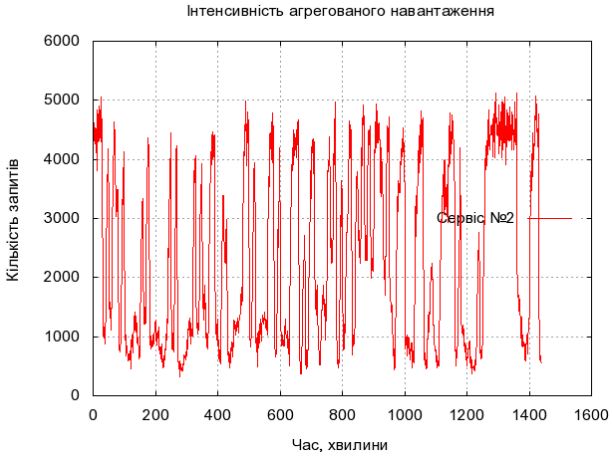
б)

Рис. 4.22 Характеристика тривалості надання сервісу а) та завантаженістю вхідного буферу VM; б) та моменту надходження відповіді

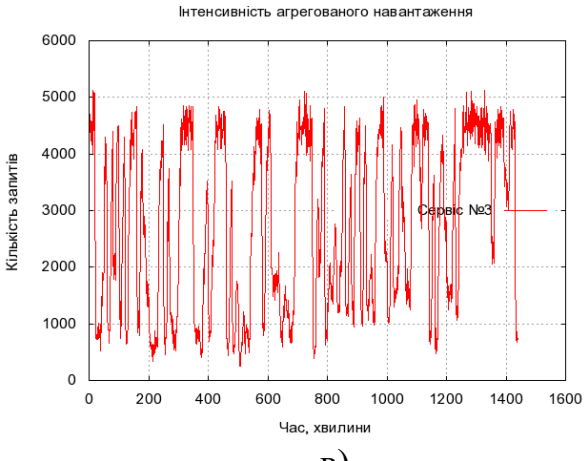
Після розгортання усіх компонентів сервісів на серверах ЦОД 1 проводився аналіз завантаженості усіх серверів взалежності від інтенсивності запитів на конкретні компоненти сервісів (рис. 4.23).



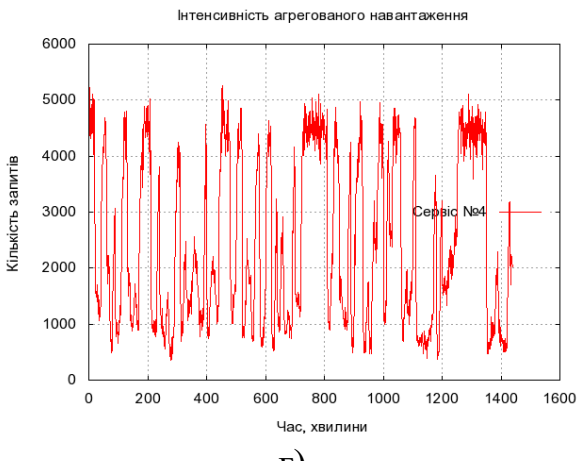
а)



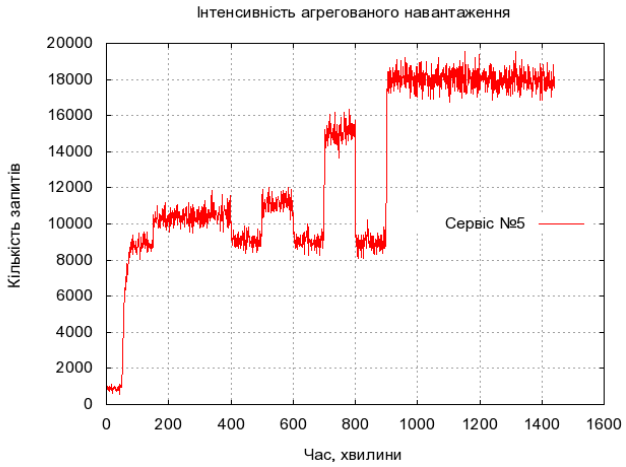
б)



в)



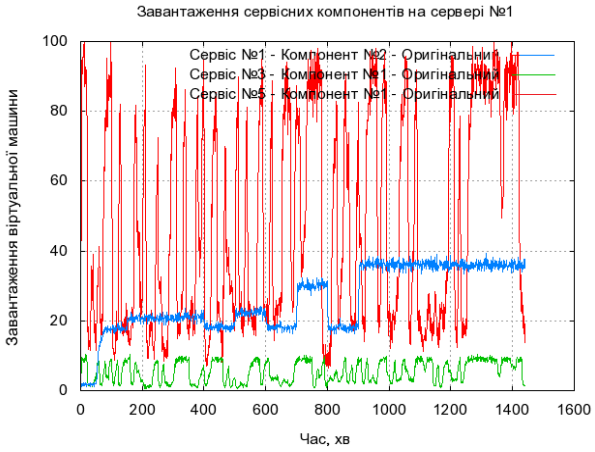
г)



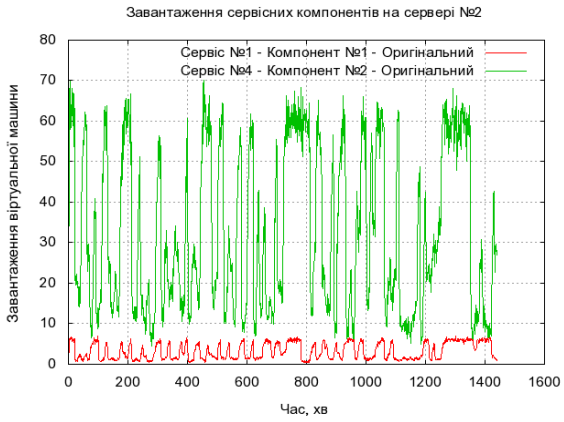
д)

Рис. 4.23 Інтенсивність агрегованого навантаження а) на перший сервіс; б) на другий сервіс; в) на третій сервіс; г) на четвертий сервіс; д) на п’ятий сервіс

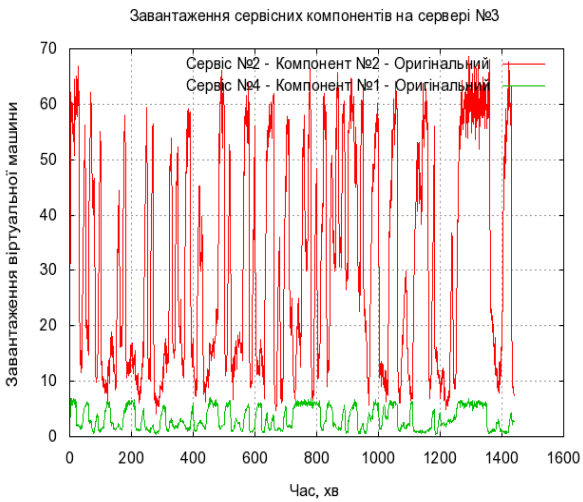
Дослідження показали, що віртуальні машини, на яких розміщені компоненти сервісу Computing, перевантажені, і у випадку надходження додаткової кількості запитів необхідно здійснювати їх міграцію (рис. 4.24 г).



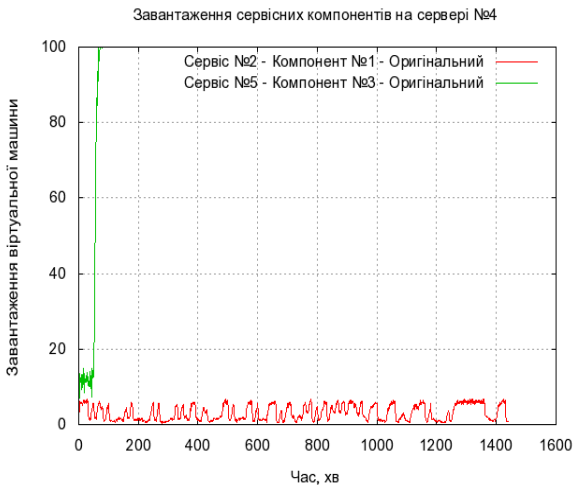
а)



б)



в)

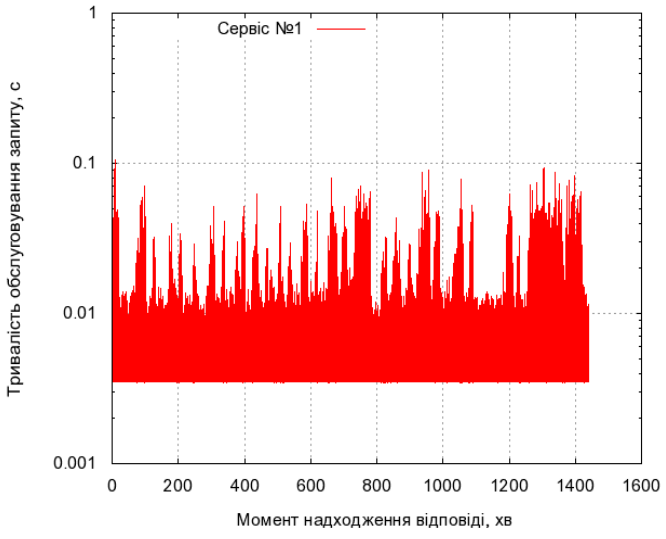


г)

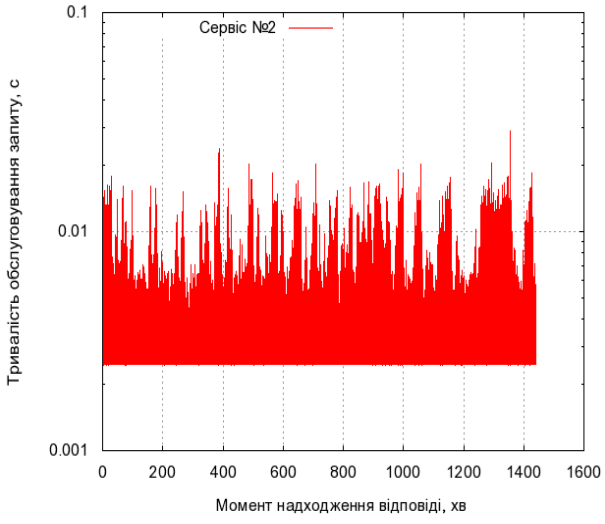
д)

Рис. 4.23 Завантаженість сервісних компонент а) на першому сервері; б) на другому сервері; в) на третьому сервері; г) на четвертому сервері; д) на п’ятому сервері

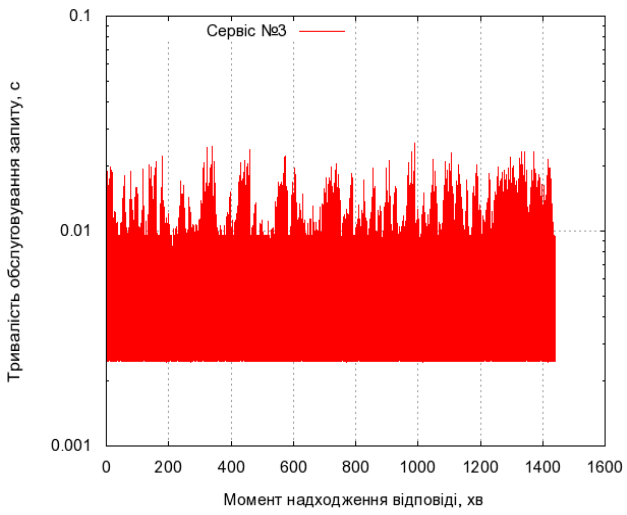
Як наслідок, збільшується тривалість обслуговування запиту та його надання кінцевому користувачеві. З рис.4.25 видно, що, якщо для першого сервісу затримка на надання сервісу в середньому становить близько 0,1 с. (рис.4.25 а), для другого – 0,07с. (рис. 4.25 б), для третього – 0,09с. (рис. 4.25 в), для четвертого – 0,06 с. (рис. 4.25 г), за той самий час спостереження, то для п’ятого сервісу (рис. 4.25 д) затримка на його надання, вже після 100 хв. доступності сервісу, перевищує максимально допустимий рівень QoS (згідно рекомендацій ITU-T) та продовжує зростати по логарифмічному закону.



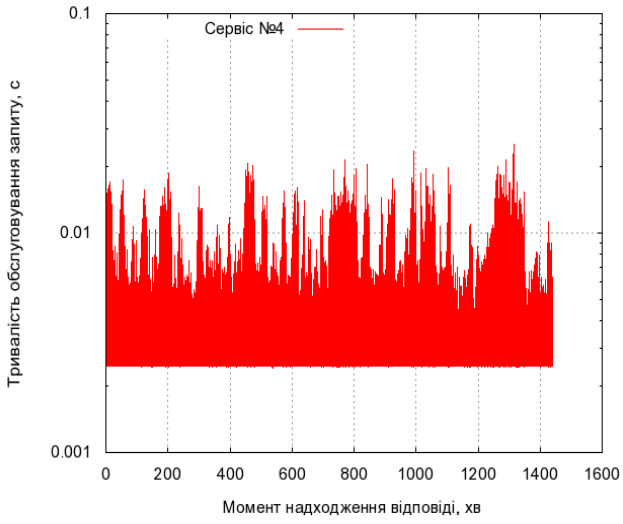
а)



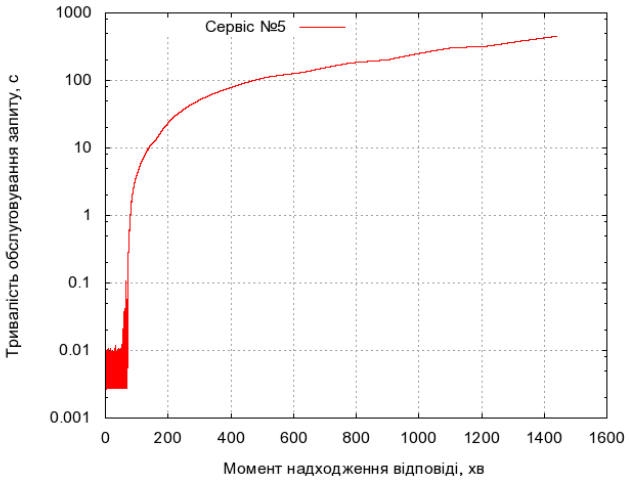
б)



в)



г)



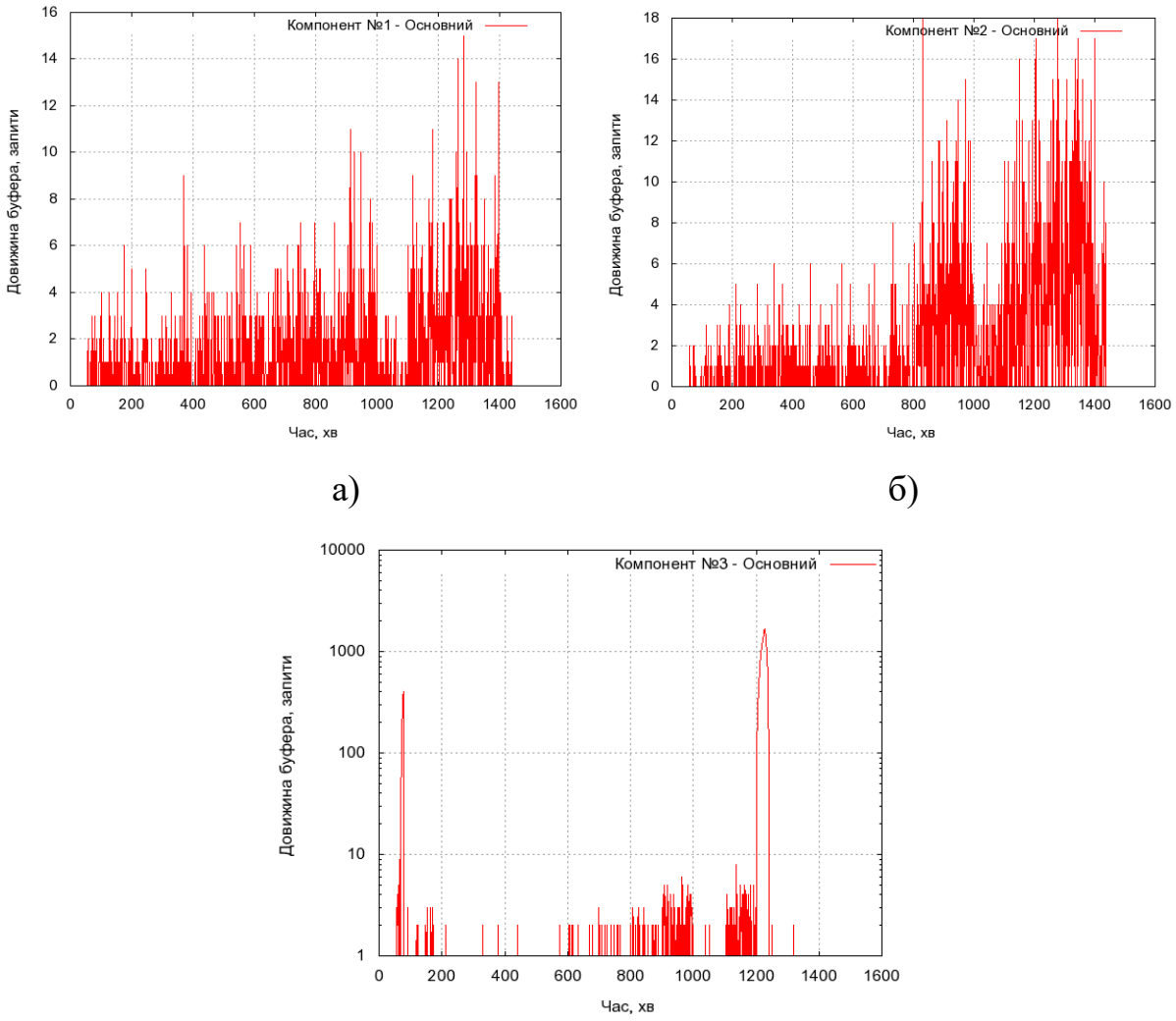
д)

Рис. 4.25 Зміна тривалості обслуговування запитів а) на перший сервіс; б) на другий сервіс; в) на третій сервіс; г) на четвертий сервіс; д) на п’ятий сервіс

Враховуючи таке суттєве погіршення якості надання сервісу, згідно методів описаних в п.2.1 та 2.4 відбувається реплікація компонентів п’ятого сервісу на сервери з незадіяними віртуальними ресурсами. В результаті роботи розробленого програмно-апаратного комплексу підтверджується необхідність ефективного балансування навантаження з врахуванням доступності компонентів та стійкості структури мереді ЦОД та зменшення тривалості обробки запитів на надання сервісу кінцевим користувачам і, як, наслідок, підвищення якості надання послуг.

4.3.2 Дослідження ефективності використання запропонованих рішень та їх вплив на якість надання композитних сервісів

Для підтвердження ефективності впровадження запропонованих у дисертаційній роботі рішень було досліджено їх вплив на якість надання композитних сервісів з використанням розробленого програмно-апартного комплексу. На 77 хв. тестування роботи системи Оркестратором на основі аналізу завантаженості буфера (рис. 4.26) було прийнято рішення про реплікацію третьої компоненти п’ятого сервісу на основі методу запропонованого у п.2.4 та розгортання на другому сервері ще однієї віртуальної машини з відповідним компонентом.



а) б)

в)

Рис. 4.26 Завантаженість буферів на надання а) першої компоненти п’ятого сервісу; б) другої компоненти п’ятого сервісу; в) третьої компоненти п’ятого сервісу

Архітеткура системи після такої реплікації зображена на рис. 4.27. Слід зазначити, що доступ користувачів до цієї компоненти та її репліки здійснюється на основі методу запропонованого у п.2.1

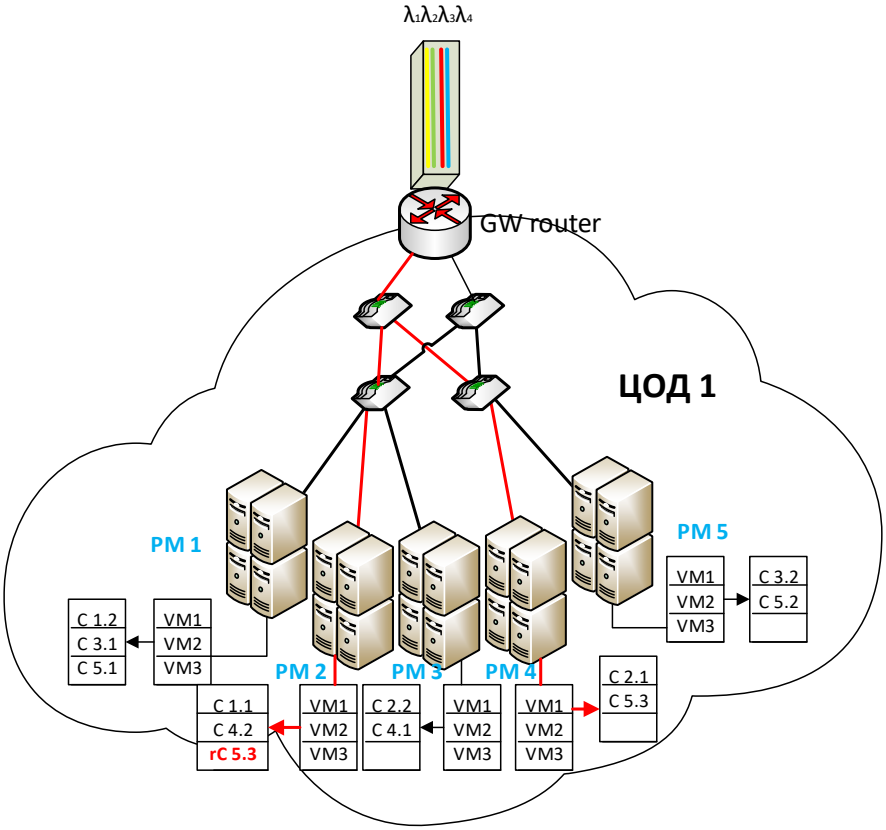
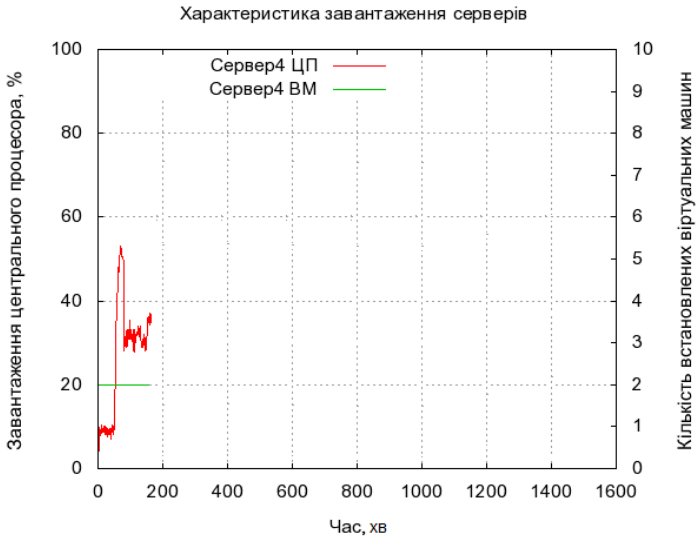


Рис. 4.27 Архітеткура системи після реплікації

Відповідно зміна завантаженості центрального процесора серверів 2 та 4 зображено на рис. 4.28 а і б

а)



б)

Рис. 4.28 Характеристика завантаження а) другого серверу; б) четвертого серверу

З рисунків видно, що завантаженість серверу 4 після балансування навантаження зменшилася у два рази, в той час коли середня завантаженість другого серверу становила 60%. На основі аналізу інтенсивності надходження запитів на компоненти п’ятого сервісу (рис. 4.29) на 178 хв,, 811 хв. та 812 хвилині тестування роботи системи Оркестратор приймає рішення про аналогічні реплікації компонентів цього сервісу на сервери в межах ЦОД 1. Зміну завантаженості серверів 2, 3, 4, 5 внаслідок балансування можна спостерігати на рис. 4.30 а-г

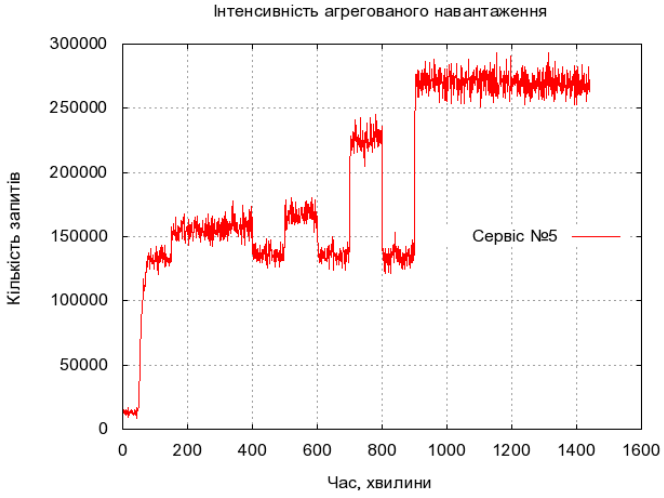


Рис. 4.29 Характеристика надходження запитів на компоненти п’ятого сервісу

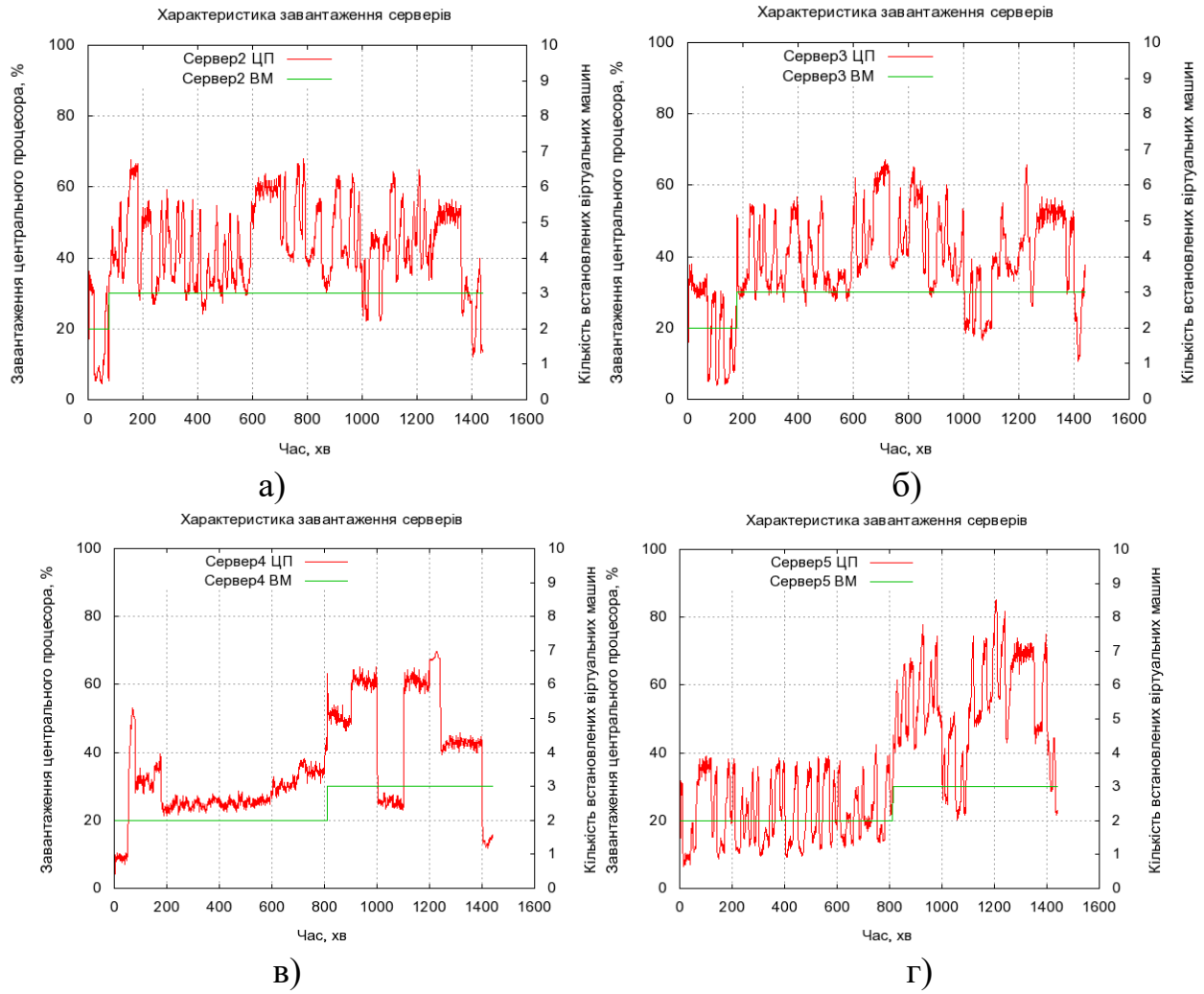
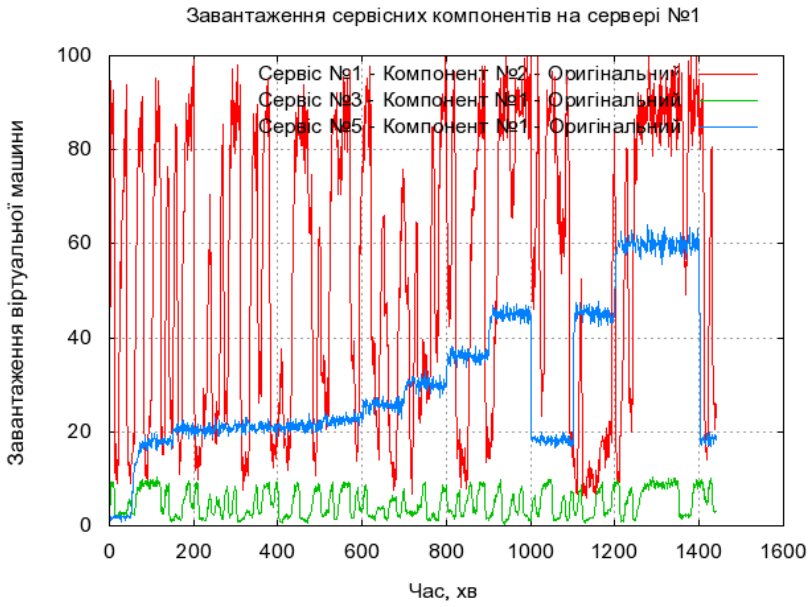
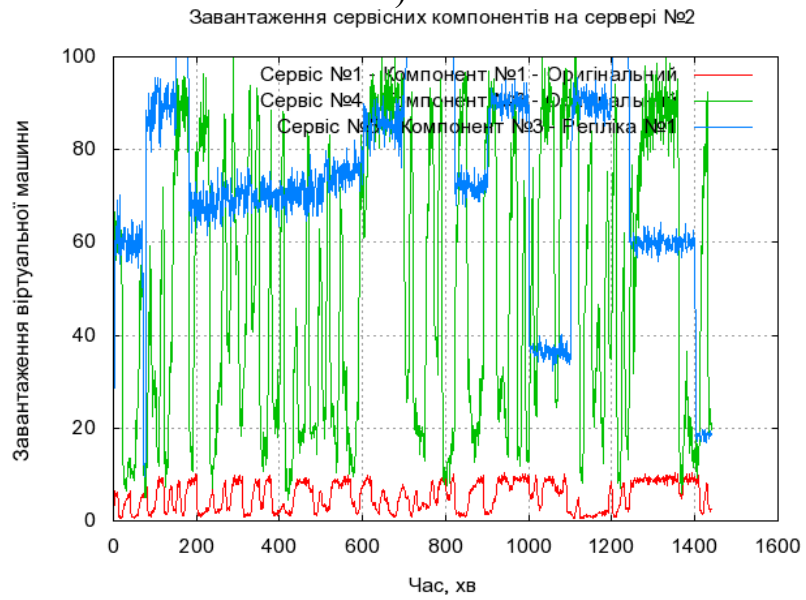


Рис. 4.30 Характеристика завантаженості а) другого серверу; б) третього серверу; в) четвертого серверу; г) п’ятого серверу внаслідок балансування

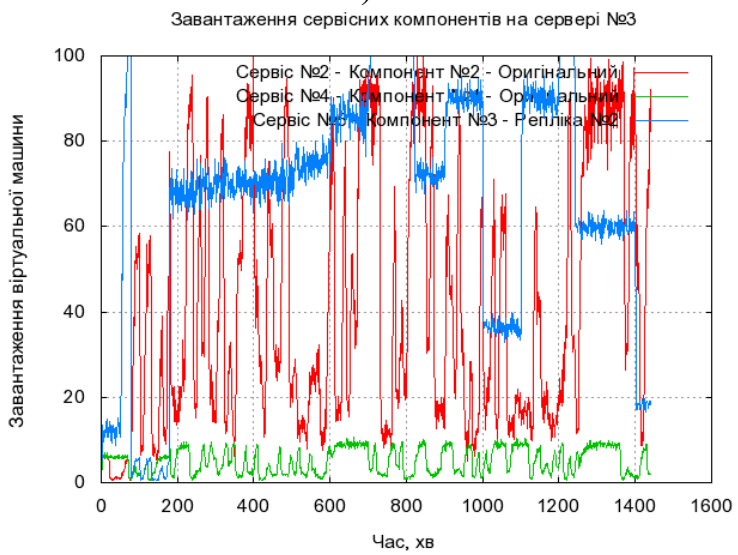
В результаті дослідження завантаженості віртуальних машин серверів (рис. 4.31) та аналізі інтенсивності поступлення запитів на компоненти п’ятого сервісу Оркестратор приймає рішення про реплікацію компонентів цього сервісу на ЦОД 2 на 1227 та 1229 хвилинах роботи системи та здійснює управління ресурсами як в межах кожного з них (за допомогою методу та архітектури описаних у п. 2.4) так і між ними (на основі системи управління оптичними ресурсами описаної в п. 4.2).


Завантаження сервісних компонентів на сервері №1
Сервіс №1 - Компонент №2 - Оригінальний
Сервіс №3 - Компонент №1 - Оригінальний
Сервіс №5 - Компонент №1 - Оригінальний
Завантаження віртуальної машини
Час, хв

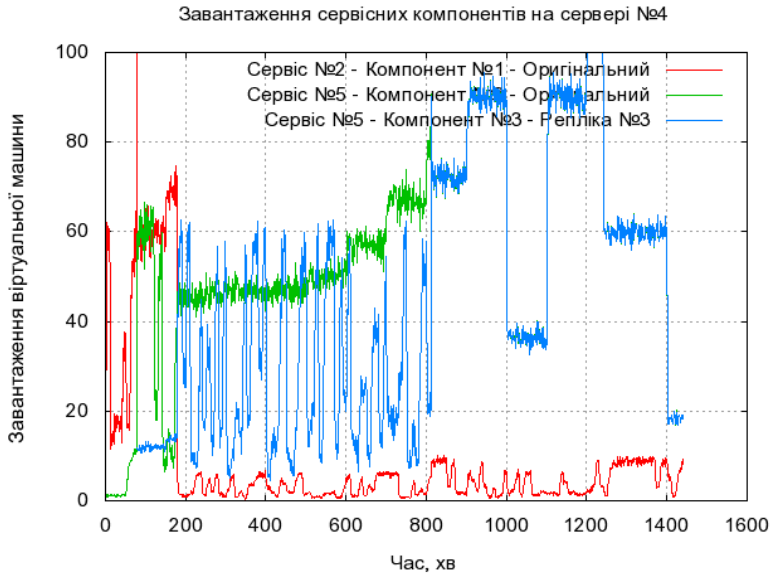
а)


Завантаження сервісних компонентів на сервері №2
Сервіс №1 - Компонент №1 - Оригінальний
Сервіс №4 - Компонент №3 - Оригінальний
Сервіс №3 - Компонент №3 - Репліка №1
Завантаження віртуальної машини
Час, хв

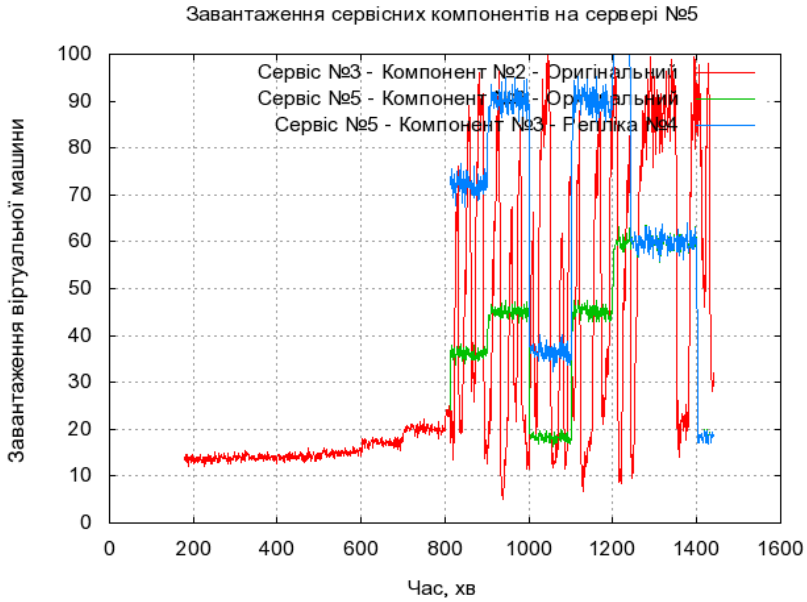
б)


Завантаження сервісних компонентів на сервері №3
Сервіс №2 - Компонент №2 - Оригінальний
Сервіс №4 - Компонент №3 - Оригінальний
Сервіс №3 - Компонент №3 - Репліка №2
Завантаження віртуальної машини
Час, хв

в)



г)



д)

Рис. 4.31 Характеристика завантаження сервісних компонент а) на першому сервері; б) на другому сервері; в) на третьому сервері; г, д) на четвертому та п’ятому серверах

В наслідок цього архітектура системи набуде вигляду представленого на рис. 4.32

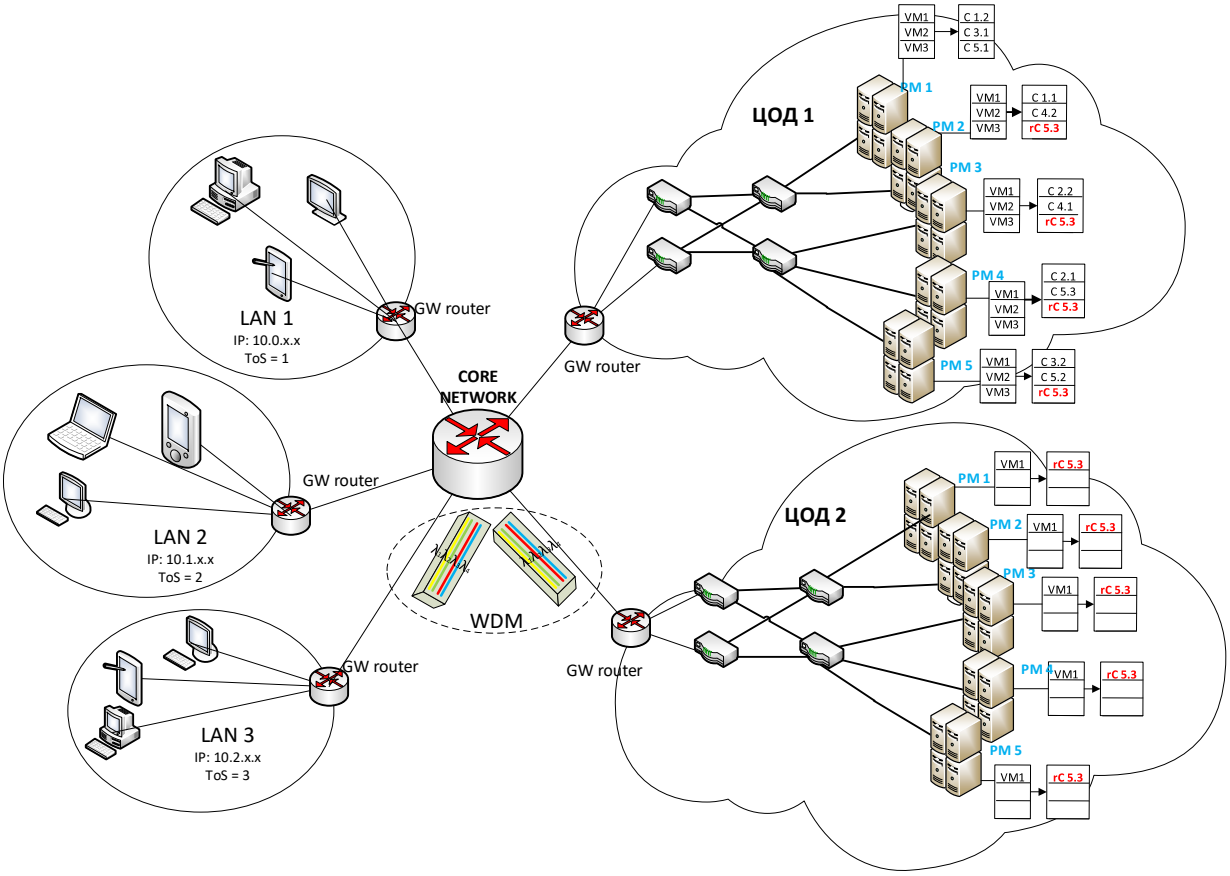
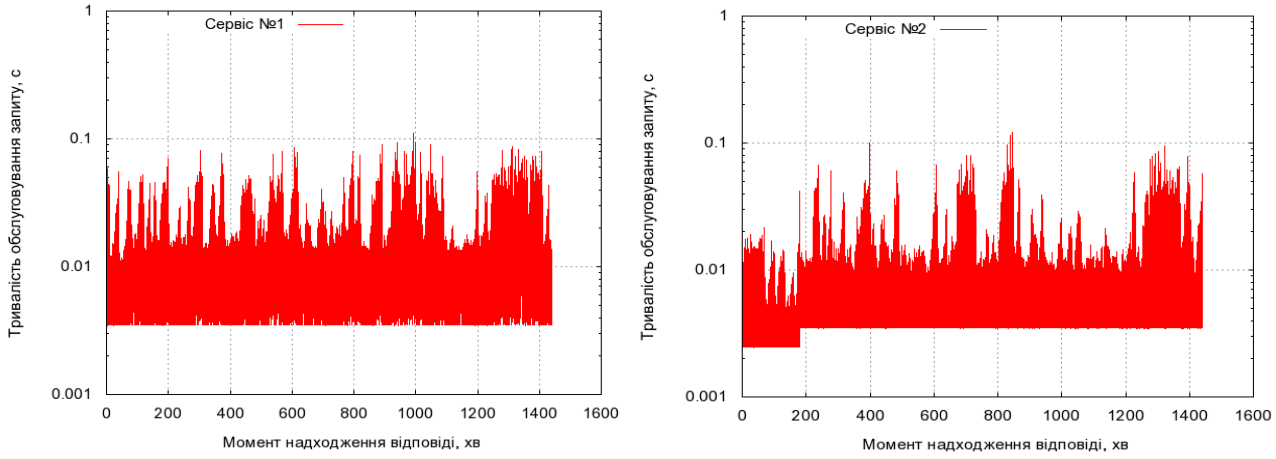


Рис. 4.32 Розташування компонентів сервісів на інфраструктурі досліджуваної мережі

Завантаженість кожного сервера в результаті роботи розробленої системи на основі запропонованих у дисертаційній роботі рішень дещо збільшилася, однак тривалість обслуговування запитів, тобто затримка на надання сервісів зменшувалась, а відтак і рівень QoS підвищився. Це видно з графіків наведених на рис. 4.33



а) б)

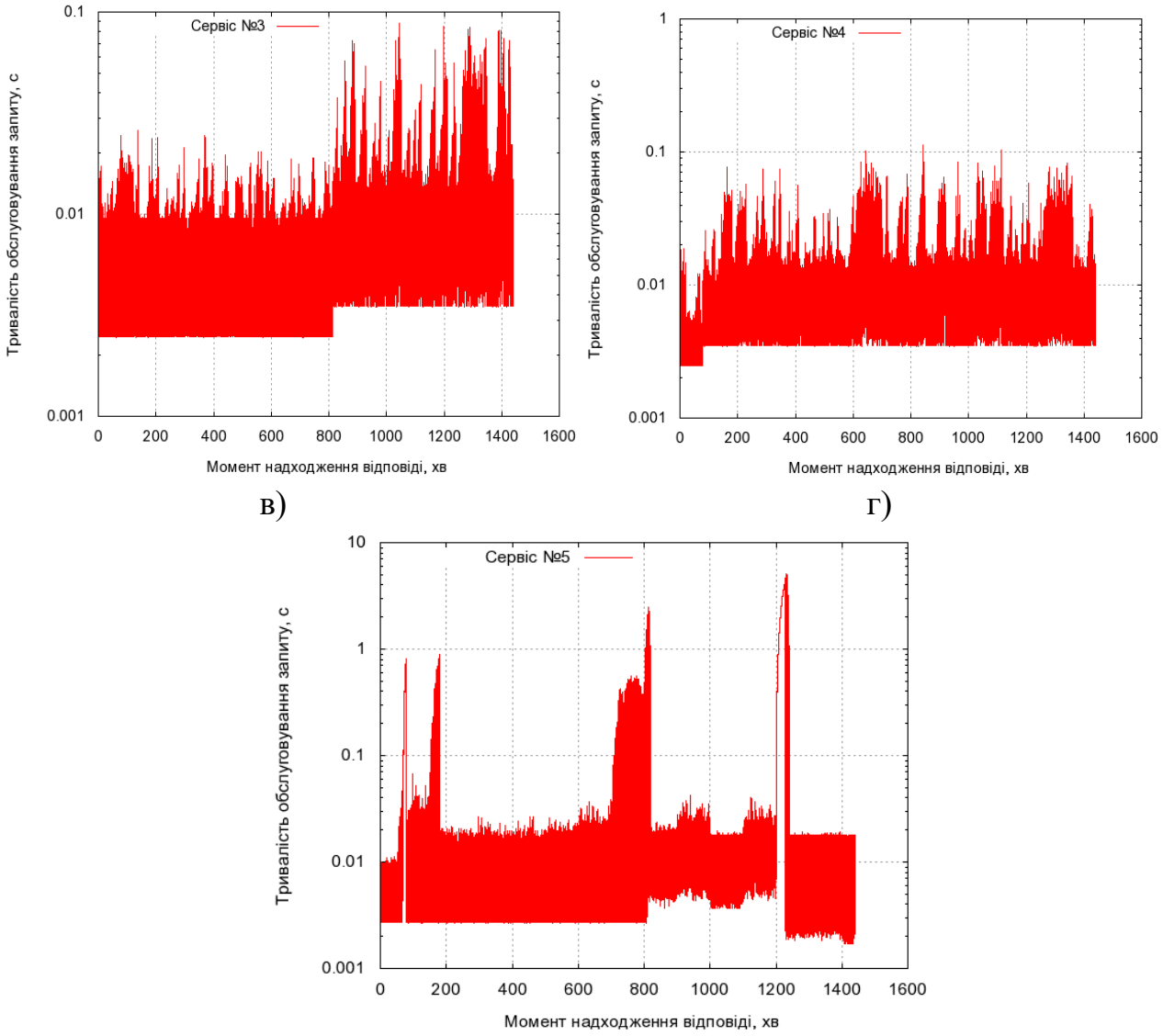


Рис. 4.33 Зміна тривалості надання а) першого сервісу; б) другого сервісу; в) третього сервісу; г) четвертого сервісу; д) п’ятого сервісу після застосування запропонованих рішень

Із залежності представленої на рис. 4.33 д чітко видно моменти ввімкнення розроблених методів та алгоритмів. Завдяки вдалій інтеграції системи управління як на локальному так і на глобальному рівні вдалося підтримувати необхідний рівень якості надання сервісів в середньому на рівні 0,02 с. Якщо розглянути детальніше інтенсивність поступлення запитів від користувачів на п’ятий сервіс та тривалість їх обслуговування (рис. 4.34), то в кожен момент балансування навантаження тривалість затримки на надання сервісу зменшувалася.

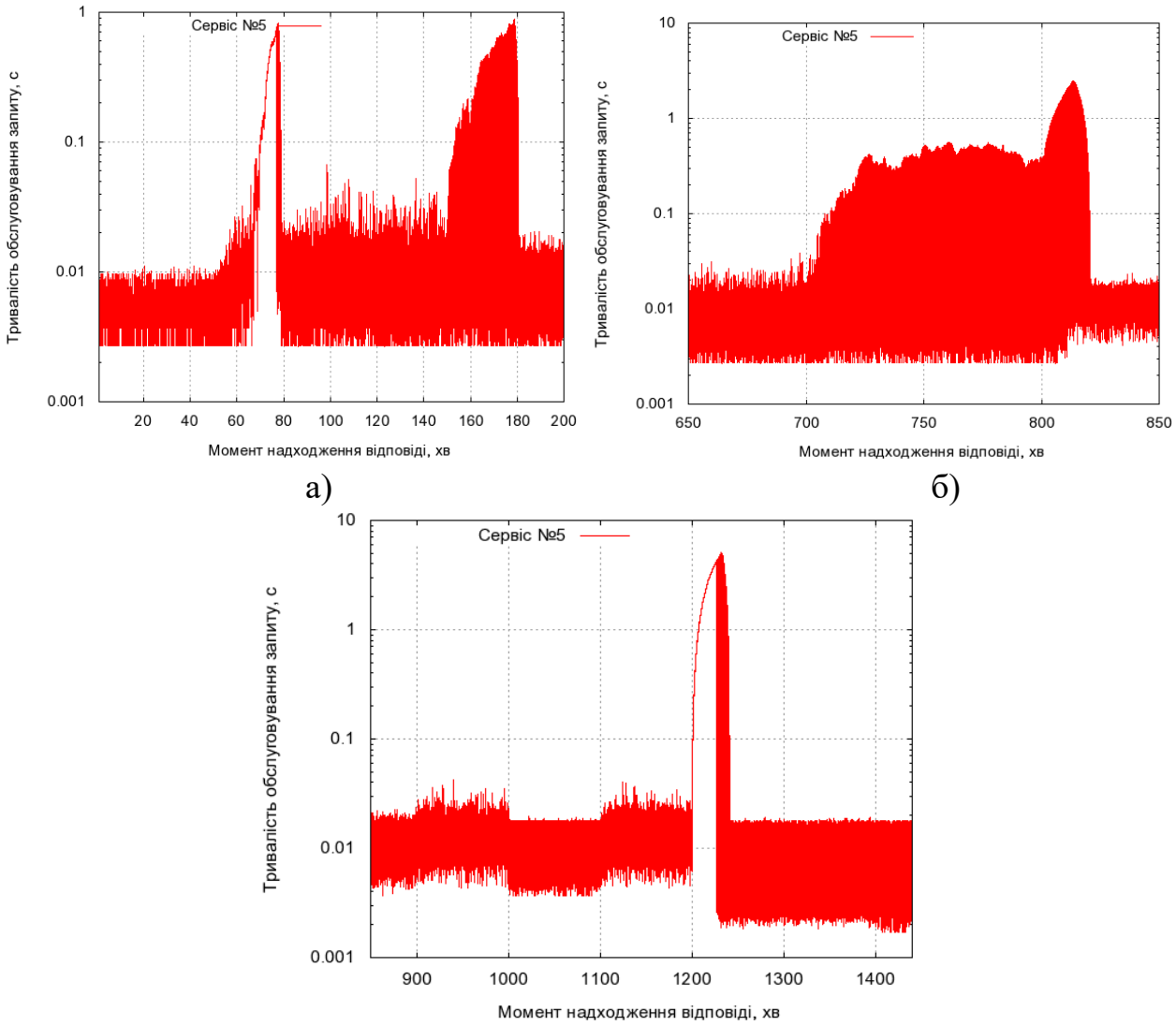


Рис. 4.34 Зміна тривалості надання п’ятого сервісу а) на 77 хв.; б) на 812 хв.; в) на 1221 хв

У перший момент (на 77 хв. рис.4.34 а) затримка на надання сервісу зменшується з 1с. до 0.02 с, тобто майже на 80%. У другий момент (на 178 хв) затримка на надання сервісу зменшується з 1с. до 0,04 с. На 811 хв., 812 хв. (рис. 4.33 б) після значної та тривалої інтенсивності поступлення запитів затримка на надання сервісу зменшилася більш ніж на 85%. Це пов’язано із ефективним застосуванням методу балансування навантаження на основі доступності фізичних ресурсів в межах одного ЦОД та методу локального розподілу оптичних ресурсів при передачі запитів на компоненти сервісу між центрамиобробки даних.

4.4. Висновки до 4-го розділу

1. Розроблено модель управління оптичними ресурсами між ЦОД, функціонування якої базується на методі локального розподілу та управління сегментом WDM мережі, який дає змогу оптимізувати використання оптичного ресурсу фізичного тракту, зменшити імовірність блокування при прокладанні нових логічних каналів, а також призведе до спрощення системи керування. Разом з цим, використання запропонованого методу дозволяє зменшити завантаженість мережевого пристрою та його енергоспоживання у 1,5 рази. Завдяки використанню алгоритму прокладання наскрізних тунелів, метод дав змогу розвантажити довжину хвилі, яка була завантажена більше ніж на 60%, та використав одну з вільних довжин хвиль з високим порядковим номером для прокладання наскрізного тунелю між вузлами. Це особливо важливо в умовах, коли в мережі передаються великі обсяги трафіку. Моніторинг якості обслуговування показує: після перемикання у прозорий режим передавання середня затримка пакетів з кінця в кінець зменшилася з 82 до 28 мс, тобто майже у чотири рази.

2. Розроблено програмно-апаратний комплекс надання композитних сервісів із гарантованим рівнем QoS, що дозволить на практиці підтвердити ефективність запропонованих методів та алгоритмів, залучаючи при цьому не лише програмну складову, а й комплекс реального мережевого обладнання. В результаті роботи розробленого програмно-апаратного комплексу підтверджується необхідність ефективного балансування навантаження з врахуванням доступності компонентів та стійкості структури мереді ЦОД та зменшення тривалості обробки запитів на надання сервісу кінцевим користувачам і, як, наслідок, підвищення якості надання послуг.

3. Завдяки вдалій інтеграційї системи управління як на локальному так і на глобальному рівні вдалося підтримувати необхідний рівень якості надання сервісів в середньому на рівні 0,02 с. Ефективне застосуванням методу балансування навантаження на основі доступності фізичних ресурсів в межах одного ЦОД та методу локального розподілу оптичних ресурсів при передачі

запитів на компоненти сервісу між центрами обробки даних дозволило у перший момент (на 77 хв.) зменшити затримку на надання сервісу з 1с. до 0,02 с, тобто на 80%; у другий момент (на 178 хв) зменшити затримку на надання сервісу з 1с. до 0,04 с. На 811 хв., 812 хв. після значної та тривалої інтенсивності поступлення запитів затримку на надання сервісу вдалося зменшити більш ніж на 85%.

ОСНОВНІ РЕЗУЛЬТАТИ ТА ВИСНОВКИ

У дисертаційній роботі розв’язано наукове завдання покращення часових параметрів надання композитних сервісів з одночасним підвищенням стійкості віртуальних топологій ЦОД, які утворюються дистанційно-векторними методами в умовах різкого зростання різноманітності потоків у сучасних гетерогенних мережах для задоволення потреб користувачів у інформаційно-комунікаційних застосуваннях реального часу.

Основні результати роботи полягають у наступному:

1. Проаналізовано основні методи надання сервісів у мережах із сервісно-орієнтованою архітектурою. Встановлено, що взаємна робота IntServ та DiffServ моделей є оптимальним варіантом для забезпечення необхідної якості QoS при передаванні запитів із кінця в кінець. За основний критерій якості обслуговування обрано затримку передавання, що ґрунтується на рекомендаціях ITU-T Y.1540. Встановлено, що при формуванні метрики на передавання компонентів сервісу не враховуються дані про віртуальну структуру мережі, що призводить до погіршення параметрів QoS у моменти міграції атомарних компонентів композитного сервісу.

2. З метою підвищення якості обслуговування та скорочення часу пошуку маршруту, що, беззаперечно, залежить від надійності окремих вузлів, запропоновано метод пошуку маршруту з урахуванням стійкості структури віртуалізованого центру обробки даних. Особливістю даного методу є компроміс між вибором оптимального маршруту та вимогами до параметрів QoS, оскільки імовірність одночасного існування потоків запитів із максимальними вимогами щодо якості надання сервісу на маршрутах, які проходять через одні і ті ж вузли, є низькою. Це дозволило враховувати не тільки особливості забезпечення різних показників QoS для різних класів трафіку, але і політику оператора з розподілу пріоритетів, відповідно до замовленого абонентом рівня якості.

3. Удосконалений метод пошуку маршруту з урахуванням стійкості структури віртуалізованого центру обробки даних дав можливість на 12% зменшити затримку у процесі пошуку маршруту передавання у динамічно

змінній мережній структурі ЦОД за рахунок доповнення метрики маршрутизації інформацією про її топологічну структуру. Моделювання показало, що застосування методу приводить до зменшення середньої затримки на 35% та пришвидшення процесу надання сервісу кінцевому користувачу.

4. Розроблено метод балансування навантаження на основі віртуалізації мережних функцій та оцінювання доступності компонентів, який максимізує продуктивність розподіленої системи, в аспекті розподілу навантаження між вузлами, що взаємодіють за допомогою реалізації інтегрованої архітектури управління. Балансування навантаження здійснюється на основі аналізу інтегрального показника доступних ресурсів фізичних машин, за принципом «міграція додатків на менш завантажені сервери» і дає змогу, за допомогою реалізації інтегрованої архітектури управління з використанням технології NVF, зменшити тривалість обслуговування запитів у 3 рази. Балансування навантаження на основі інтегральної оцінки доступних телекомунікаційних та програмно-апаратних ресурсів дало змогу мінімізувати час затримки надання сервісу користувачам. Реалізація методу дає змогу, використовуючи міграцію компонентів сервісу, зменшити середню тривалість оброблення запитів та затримку пакетів з кінця в кінець у 2,75 рази.

5. Запропоновано модель та метод управління мережними ресурсами оптичної мережі між центрами обробки даних, який дав змогу покращити часові параметри якості обслуговування, за рахунок об’єднання та перегрупування потоків запитів, забезпечивши при цьому зменшення завантаженості оптичного мережного тракту у 1,5 рази. Алгоритм «прокладання наскрізних тунелів» дав змогу, методом максимізації завантаженості оптичної несучої, зменшити затримку передавання пакетів з кінця в кінець у 2,92 рази.

6. Розроблено програмно-апаратний комплекс надання композитних сервісів із гарантованим рівнем QoS, що дало змогу на практиці підтвердити ефективність запропонованих методів та алгоритмів, залучаючи при цьому не лише програмну складову, а й комплекс реального мережного обладнання. Завдяки вдалій інтеграції системи управління як на локальному, так і на

глобальному рівні вдалося підтримувати необхідний рівень якості надання сервісів (в середньому на рівні затримки їх надання 0,2 с). Ефективне застосування методів балансування навантаження на основі оцінювання доступності фізичних ресурсів в межах одного ЦОД та локального розподілу оптичних ресурсів при передачі запитів на компоненти сервісу між центрами обробки даних дозволило зменшити затримку надання сервісу більше, ніж на 70%.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Shpur O. Improving the Quality of Composite Services Through Improvement of Cloud Infrastructure Management // O. Shpur, M. Klymash, M. Seliuchenko, B. Strykhaliuk, O.Lavriv // International Journal of Computer Science and Information Security (IJCSIS). - 2015. - Vol. 13. - No. 9 – P.36-44.

[2] Beshley M. SOA quality management subsystem on the basis of load balancing method using fuzzy sets // M. Beshley, M. Klymash, B. Strykhalyuk, O. Shpur, B. Bugil, I. Kagalo // International Journal of Computer Science and Software Engineering (IJCSSE). – 2015. - Volume 4. - Issue 1. – P.10-21.

[3] Demydov I.V. The structural-functional synthesis of cloud service delivery platform after service availability and performance criteria // Demydov I.V., Strykhalyuk B.M., Shpur O.M., Mohamed Mehdi El Hatri, Klymash Yu.V. // Системи обробки інформації : зб. наук. пр. / Х: Харк. ун-т Повітр. Сил ім. Івана Кожедуба. – 2015 - №1(126) - С. 144-149.

[4] Климаш М.М. Метод підвищення ефективності використання мережевих ресурсів інформаційно-телекомунікаційних систем // М. М. Климаш, О. М. Шпур, М. О. Селюченко, Б. В. Киричук, Т. В. Мельник // Вісник Національного університету «Львівська політехніка» №818. Радіоелектроніка та телекомунікації. – Львів. - 2015.- С. 137-151.

[5] Стрихалюк Б.М. Алгоритми пошуку шляху за критерієм мінімальної затримки для центру обробки даних //Стрихалюк Б.М., Шпур О.М., Селюченко М.О., Андрухів Т.В.// Вісник Національного університету «Львівська політехніка» №796. Радіоелектроніка та телекомунікації. – Львів. - 2014. - С.176-181.

[6] Климаш М.М., Метод диференційованого мультипотокового керування трафіком у транспортних програмно-керованих мережах // М.М. Климаш, О.М. Шпур, О.В. Багрій, А. Л. Швець // Вісник Національного університету «Львівська політехніка» №796. Радіоелектроніка та телекомунікації. – Львів. - 2014. - С.60-68.

[7] Strykhalyuk B., Service provisioning by using a structure stability algorithm in a virtualized data center based on cloud technology // B. Strykhalyuk, O. Shpur, A. Masiuk // Computational Problems of Electrical Engineering. - 2014. - Vol. 4. - №1. - P.81-87.

[8] Klymash M. Features of the cloud services implementation in the national network segment of Ukraine / M.Klymash, I.Demydov, M.Beshley, O.Shpur // Information and telecommunication science. - K.: NTUU “KPI”. - 2016. - No.1. - P.31-38.

[9] Klymash M. The model for assessment the reliability of structures in virtualized data centers // M.Klymash, O.Shpur, I. Tchaikovskyi // Information and telecommunication science. - K.: NTUU “KPI”. - 2015. - No.1. - P.33-37

[10] Стрихалюк Б.М. Визначення доступності програмних комплексів у системах із сервісно-орієнтованою архітектурою // Б.М. Стрихалюк, О.М. Шпур, М.О. Селюченко // Наукові праці ДонНТУ. Серія: обчислювальна техніка та автоматизація №2. - Донецьк – 2014. - (27)’2014. - С.109-120.

[11] Климаш М. М. Модель надання сервісів на основі методу адаптації логічної структури cloud-системи // Климаш М. М., Стрихалюк Б. М., Шпур О. М., Бешлей М. І. // Наукові записки Українського науково-дослідного інституту зв’язку. – 2014. – №5(33) - С. 27-36

[12] Shpur O. The optimal distribution of optical resources between data centers for providing the required level of QoS / O.Shpur, B.Strykhalyuk, O.Marushko, I.Bolyubash // Modern problems of radio engineering, telecommunications, and computer science. Proceedings of the International Conference TCSET’2016 (Lviv-Slavske, Ukraine February 23 – 26, 2016) – Lviv: Publishing House of Lviv Polytechnic – 2016 - P. 649-651.

[13] Strykhalyuk B. Synthesis of distributed service-oriented structures cloud networks is based on algorithm for determining hyperbolic virtual coordinates \\ B. Strykhalyuk, O. Shpur, I. Demydov, Yu. Klymash \\ Proceedings of XIIIth international conference ”The experience of designing and application of CAD Systems in microelectronics” CADSM’2015. (24-27 February, Lviv-Poljana, Ukraine) – 2015. - P. 231-235.

[14] Shpur O. Reliability of delivery services in cloud data centers \\ O. Shpur, B. Strykhalyuk, M. Klymash \\ Proceedings of XIIIth international conference"The experience of designing and application of CAD Systems in microelectronics" CADSM'2015. (24-27 February, Lviv-Poljana, Ukraine), 2015. - P.203-206.

[15] Strykhaliuk B. Improving the method for load balancing of service flows based on local management of network resources in cloud systems // B. Strykhalyuk, O. Shpur, B. Kyrychuk // Second IEEE International Scientific-Practical Conference "Problems of Infocommunications. Science and Technology"(PICS&T'2015). Conference proceedings. (13-15 October, Kharkiv, Ukraine), 2015. - Kh:KHNURE, - P. 161-163.

[16] Demydov I. Dynamic correction of routing metrics by pervasive structural routing in the scalable distributed service networks // I. Demydov, O. Shpur, Mohamed Mehdi El Hatri // Second IEEE International Scientific-Practical Conference "Problems of Infocommunications. Science and Technology"(PICS&T'2015). Conference proceedings. (13-15 October, Kharkiv, Ukraine), 2015. - Kh:KHNURE, - P. 164-166.

[17] Klymash M. Comparative analysis of methods for describing topological structures of cloud networks// O. Shpur, B. Strykhalyuk, M. Klymash// Modern problems of radio engineering, telecommunications, and computer science Proceedings of the International Conference TCSET'2014 Dedicated to the 170th anniversary of Lviv Polytechnic National University (Lviv-Slavske, Ukraine February 25 – March 1, 2014) – Lviv: Publishing House of Lviv Polytechnic – 2014 - P.50-53..

[18] Демидов І.В. Впровадження хмарних сервісних систем в національному мережному сегменті України / І.В.Демидов, О.М.Шпур // X Міжнародна науково-технічна конференція «Проблеми телекомунікацій» ПТ-2016: Збірник матеріалів конференції (19-22 квітня 2016р. м. Київ), 2016. - К.: НТТУ «КПІ» – С. 342-344.

[19] Klymash M.M. The features of cloud service delivery platform structural-functional synthesis / M.M. Klymash, I.V. Demydov, O.M. Shpur, Z.V.Kharkhalis // Міжнародна науково-технічна конференція «Сучасні інформаціно-

телекомунікаційні технології»: матеріали науково-технічної конференції (17-20 листопада 2015 р. м.Київ), 2015.– К:ДУТ - Т.3 – С. 19-21.

[20] Buhil B. Improving the effectiveness of data transfer in IP/MPLS network // B. Buhil, O.Shpur, T. Melnyk// 1st International Conference "Advanced Information and Communication Technologies" (AICT'2015). Conference proceedings. (29 October – 01 November, Lviv, Ukraine), 2015.- L. – P. 83-86.

[21] Лаврів О. Дистанційний моніторинг параметрів місця вчинення злочину з використанням мобільних технологій // О. Лаврів, Б. Стрихалюк, О. Шпур, Т. Максимюк // 1st International Conference "Advanced Information and Communication Technologies" (AICT'2015). Conference proceedings. (29 October – 01 November, Lviv, Ukraine), 2015. - L. – P. 67-69.

[22] Стрихалюк Б.М. Метод балансування навантаження на основі інтегрованої архітектури управління з використанням функції NVF / Б.М. Стрихалюк, О.М. Шпур, М. О. Селюченко // IX Міжнародна науково-технічна конференція «Проблеми телекомунікацій» ПТ-2015: Збірник матеріалів конференції.-К.: НТТУ «КПІ». -2015. – С. 322-325.

[23] Демидов І.В., Доступність композитних застосувань у сервісно-орієнтованих системах // О.А. Лаврів, О.М. Шпур, М. О. Селюченко // Вимірювальна та обчислювальна техніка в технологічних процесах: матеріали XIII Міжнародній науково-технічній конференції (6-12 червня 2014 р. м. Одеса), 2014. - С.119-122.

[24] Стрихалюк Б.М., Метод оптимізації часу надання сервісу з врахуванням структури ЦОД для мереж з cloud технологією// Б.М. Стрихалюк, О.М. Шпур // Фізико-технологічні проблеми радіотехнічних пристроїв, засобів телекомунікацій, нано- та мікроелектроніки: матеріали IV Міжнародній науково-практичній конференції (23-25 жовтня 2014 р. м. Чернівці), 2014р. - С.112-113.

[25] Стрихалюк Б.М., Віртуалізація мобільних систем зв'язку на основі технології NFV та моделей cloud-сервісів // Б.М. Стрихалюк, О.М. Шпур, А.Р. Масюк // Сучасні проблеми телекомунікацій та підготовка фахівців в

галузі телекомунікацій: матеріали конференції (30 жовтня – 02 листопада 2014 р. м. Львів), 2014р. - С.21-24

[26] Яремко О.М., Дослідження методів розподіленого передавання даних в безпровідних мережах доступу// О.М. Яремко, П.О. Гуськов, О.М. Шпур // Сучасні проблеми телекомунікацій та підготовка фахівців в галузі телекомунікацій: матеріали конференції (30 жовтня – 202 листопада 2014 р. м. Львів), 2014р. - С.179-183.

[27] Климаш М.М. Забезпечення якості обслуговування в мультисервісних мережах на основі гібридних моделей // М.М. Климаш, М.І. Бешлей, О.М. Шпур, Б.А. Бугиль // 69-та науково-технічна конференція професорсько-викладацького складу, науковців, аспірантів та студентів: матеріали конференції (3-5 грудня 2014 р. м.Одеса), 2014р.- ч.2 – С.96-99.

[28] Tanenbaum. Andrew S. Distributed systems: principles and paradigms I Andrew S. Tanenbaum, Maarten Van Steen.

[29] Sangyoon Oh “Web service architecture for mobile computing”, Indiana University, August 2006.

[30] Ptitsyin G. A. “The models of probabilistic collapse of dynamic networks”, Electrical and information complexes and systems, 2006. - vol. 2, - no. 4, - P.54-58.

[31] ITU-T E.800 Definitions of terms related to quality of service

[32] ITU-T E.860 Framework of a service level agreement

[33] RFC 2475 An Architecture for Differentiated Services

[34] Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для ВУЗов. 4-е изд. – СПб.: Питер, 2010. – 944 с.

[35] Зелингер Н. Б., Чугреев О. С., Яновский Г. Г. Проектирование сетей и систем передачи дискретных сообщений. – М.: Радио и связь, 1992. – 175 с.

[36] ЯновскийГ. .Г. Качество обслуживания в сетях IP. // Вестник связи, – 2008. – №1. – с. 1-16.

[37] Khazaei H., Mišić J., Mišić V. B. “Performance of Cloud Centers with High Degree of Virtualization under Batch Task Arrivals”, IEEE Transactions on Parallel and Distributed Systems, 2012. - vol. 10 - no. 5 - P. 1-10

[38] Mohammad Al-Fares , Loukissas A., Vahdat A. “A scalable, commodity data center network architecture” Proceedings of the ACM SIGCOMM 2008 conference on Data communication, August 17-22, 2008, Seattle, WA, USA;

[39] Costin Raiciu, Sebastien Barre, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, Mark Handley “Improving datacenter performance and robustness with multipath TCP”, SIGCOMM '11 Proceedings of the ACM SIGCOMM 2011 conference, NY, USA, pp. 266-277;

[40] Климаш М.М., Селюченко М.О. “Аналіз принципів побудови сучасних дата-центрів на основі Cloud-архітектури” / Всеукраїнська науково-практична конференція СПТЕЛ – 2013 / Львів, 2013, с.110-114;

[41] Swiatek, P. (2011) “Complex Services Availability in Service Oriented Systems, Proc. 21st International Conference on Date 16-18 Aug. 2011, pp. 227 – 232.

[42] Load balancing of virtual machines: http://lbvm.sourceforge.net/.

[43] Yi Zhao, Wenlong Huang Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud 2009 Fifth International Joint Conference on INC, IMS and IDC P.170-175

[44] Prabavathy B., Priya K., Chitra B. A Load Balancing Algorithm For Private Cloud Storage 4th ICCCNT 2013 July 4-6, 2013, Tiruchengode, India

[45] Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mondal, Santanu Dam, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing", First International Conference on Computational Intelligence: Modelling Techniques and Applications, Elsevier, Vol. 10, 2013, pp. 340-347 .

[46] Dhinesh Babu L. D. and P. Venkata Krishna, "Honey bee behaviour inspired load balancing of tasks in cloud computing environments", Applied

[47] Soares J. Cloud4NFV: a platform for virtual network functions / J. Soares, M. Dias, J. Carapinha, B. Parreira, S. Sargento // Proceedings of the 3rd International conference on cloud networking.- 2014, - P.288-293

[48] Fel Ma Distributed load balancing allocation of virtual machine in cloud data center / Fel Ma, Feng Liu та Zhen Liu // Proceedings of the 3rd International conference on software engineering and service science (ICSESS), 2012. - P.20-23.

[49] Dan Marinescu, Reinhold Kröger., “State of the art in autonomic computing and virtualization”, September 2007

[50] Daniel A. Menasce and Mohamed N. Bennani , “Autonomic Virtualized Environments”, International Council of the Aeronautical Sciences, pages 28-28, July 2006

[51] P. Ruth, J. Rhee, D. Xu, R. Kennell, S. Goasguen, “Autonomic Live Adaptation of Virtual Computational Environments in a MultiDomain Infrastructure”, 3rd IEEE International Conference on Autonomic Computing (ICAC 2006), pages 5-14 , June 2006

[52] Red Hat Linux, Piranha white paper: http://www.redhat.com/ support/wpapers/piranha/index.html

[53] Virtual Networking: http://www.gnome.org/~markmc/ virtualnetworking.html

[54] Gagan Aggarwal, Rajeev Motwani, and An Zhu., “The load rebalancing problem”, SPAA03: Proceedings of the 15th annual ACM symposium on Parallel Algorithms and architectures, Jun 2003.

[55] Travis F Vachon and James D Teresco., “Automated dynamic redistribution of Virtual operation systems under the xen virtual machine monitor”, Proceedings of the 25th conference on Proceedings of International Multi-Conference: parallel and distributed computing and networks, pages 190-195, May 2007.

[56] Jing Xu, Ming Zhao, José Fortes, Robert Carpenter, Mazin Yousif, “Autonomic resource management in virtualized data centers using fuzzy logic-based approaches”, Cluster Computing, pages.213-227, September 2008

[57] M. McNett, D. Gupta, A. Vahdat, and G. M. Voelker., “Usher: An Extensible Framework for Managing Clusters of Virtual Machines”, In Proceedings of the 21st Large Installation System Administration Conference (LISA), November 2007

[58] J. Myint, T. Naing. ”A data placement algorithm with binary weighted tree on PC cluster-based cloud storage system”. IEEE International Conference on Cloud and Service Computing (CSC), 2011, pp. 315– 320.

[59] W. Zeng, Y. Li, J. Wu, Q. Zhong and Q. Zhang. ”Load Rebalancing in Large-Scale Distributed File System.”IEEE 1st International Conference on Information Science and Engineering (ICISE), 2009, pp. 265-269.

[60] W. Zeng, Y. Zhao, K. Ou and W. Song. Research on cloud storage architecture and key technologies. In Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp. 1044-1048. ACM, 2009.

[61] Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms", Second IEEE Symposium on Network Cloud Computing and Applications, 2012, pp. 137-142.

[62] Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros, “Inter Cloud: Utility-oriented federation of cloud computing environments for scaling of application services”, 10th International Conference on Algorithms and Architectures for Parallel Processing, Springer LNCS, 2010, pp. 13-31.

[63] Ian Foster, Yon Zhao, Ioan Raicu, Shiyonglu, “Cloud Computing and Grid Computing 360-degree compared”, IEEE Workshop on Grid Computing Environments, 2008, pp. 1-10.

[64] Rajiv Ranjan, Liang Zhao, Xiaomin Wu, Anna Liu, Andres Quiroz and Manish Parashar, "Peer-to-Peer Cloud Provisioning:Service Discovery and Load Balancing", Cloud computing: Principles, Systems and Applications,computer communications and networks, Springer, 2010, pp. 195-217 .

[65] Borja Sotomayor, Ruben S.Montero, Ignacio M. Llorente, and Ian Foster, "Virtual infrastructure management in private and hybrid clouds", IEEE Internet Computing, 2009, pp. 14-22 .

[66] Ali M. Alakeel "A guide to dynamic load balancing in distributed computer systems", International Journal of Computer Science and Network Security, VOL.10 No.6 , 2010 ,153-160.

[67] Thilina Gunarathne, Tak-Lon Wu, Judy Qiu and Geoffrey Fox, "MapReduce in the Clouds for Science" 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010, pp.565-572 .

[68] Junjie Ni, Yuanqianq Huang, Zhongzhi Luan, Juncheng Zhang and Depei Qian, "Virtual machine mapping policy based on load balancing in private cloud environment", IEEE International Conference on Cloud and Service Computing, 2011, pp. 292-295

[69] Sung-Soo Kim, Ji-Hwan Byeon, Hongbo Liu, Ajith Abraham and Seán McLoone, "Optimal job scheduling in grid computing using efficient binary artificial bee colony optimization", Soft Computing, Springer, Vol. 17, No. 5, 2013, pp. 867-882.

[70] Marco Dorigo, Gianni Di Caro Luca and M. Gambardella, "Ant Algorithms for Discrete Optimization", Artificial Life, Massachusetts Institute of Technology , 1999, pp. 137-172 .

[71] Nidhi Jain Kansal and Inderveer Chana, "Cloud Load Balancing Techniques :A Step Towards Green Computing", International Journal of Computer Science Issues, Vol.9, No.1, 2012 pp.238-246

[72] Qinghai Bai, "Analysis of Particle Swarm Optimization Algorithm", Computer and Information Science, Vol. 3, No. 1, 2010, pp. 180-184.

[73] Particle Swarm Optimization, http://en.wikipedia.org/wiki/ Particle_swarm_optimization

[74] C.-W. Chiang, Y.-C. Lee, C.-N. Lee and T.-Y. Chou, "Ant colony optimization for task matching and scheduling", IEE Proceedings on Computers and Digital Techniques, Vol. 153, No. 6, 2006, pp. 373380.

[75] Salim Bitam, "Bees Life Algorithm for Job Scheduling in Cloud Computing", Proceedings of The Third International Conference on Communications and Information Techno

[76] C. Clark, K. Fraser, S. Hand, J. Hansen, E. Jul, C. Limpach,I. Pratt, and A. Warfield, “Live migration of virtual machines,”inProceedings of the 2nd conference on Symposium on Net-worked Systems Design & Implementation-Volume 2, 2005,p. 286.

[77] . Nelson, B. Lim, and G. Hutchins, “Fast transparentmigration for virtual machines,” inProceedings of the annualconference on USENIX Annual Technical Conference, 2005,p.

[78] H. Liu, H. Jin, X. Liao, L. Hu, and C. Yu, “Live migrationof virtual machine based on full system trace and replay,”inProceedings of the 18th ACM international symposium onHigh performance distributed computing, 2009, pp. 101–110.

[79] H. Jin, L. Deng, S. Wu, X. Shi, and X. Pan, “Live virtualmachine migration with adaptive memory compression,” inProc. CLUSTER, 2009.

[80] M. Hines and K. Gopalan, “Post-copy based live virtualmachine migration using adaptive pre-paging and dynamicself-ballooning,” inProceedings of the 2009 ACM SIG-PLAN/SIGOPS international conference on Virtual executionenvironments, 2009, pp. 51–60.

[81] Y. Luo, B. Zhang, X. Wang, Z. Wang, Y. Sun, and H. Chen,“Live and incremental whole-system migration of virtualmachines using block-bitmap,” in2008 IEEE InternationalConference on Cluster Computing, 2008, pp. 99–106.

[82] R. Nathuji and K. Schwan, “VirtualPower: coordinated powermanagement in virtualized enterprise systems,” inProceed-ings of twenty-first ACM SIGOPS symposium on Operatingsystems principles. ACM, 2007, p. 278.

[83] L.A.BarrosandU.Hőlzle, “The Data centeras a Computer: An Introduction to the design of Warehouse-scale Machines,” synthesis lectures on ComputerArchitecture4(1), 2009, pp.1-118

[84] T. Benson, A. Anand, A. Akella and M. Zhang, “Understanding DataCenter Traffic Characteristics”, ACM SIGCOMM 2010

[85] R. Sinha, C. Papadopoulos, J. Heidemann, “Internet Packet Size Distributions: Some Observations”, republished ISI-TR-2007-643, 2007

[86] W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)", IEEE/ACMJournal Transactions on Networking (TON), Vol. 2 Issue 1, Feb. 1994

[87] R. J. Alder, R.E. Feldman, M.S. Taqqu, “A Practical Guide to HeavyTails”, Birkhäuser, 1998

[88] L. G. Roberts, “ALOHA packet system with and without slots andcapture”, ACM SIGCOMM Computer Communication ReviewNewsletter, Vol. 5 Issue 2, New York, USA, April 1975

[89] V. W. Freeh, D. K. Lowenthal, F. Pan, N. Kappiah,R. Springer, B. L. Rountree, and M. E. Femal, “Analyzingthe energy-time trade-off in high-performance computingapplications,” IEEE Transactions on Parallel and DistributedSystems, vol. 18, no. 6, pp. 835–848, 2007.

[90] M. Curtis-Maury, F. Blagojevic, C. D. Antonopoulos, D. S. Nikolopoulos, “Prediction-based power-performanceadaptation of multithreaded scientific codes,”IEEE Transac-tions on Parallel and Distributed Systems, vol. 19, no. 10, pp.1396–1410, 200

[91] B. Lud¨ascher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao “Scientific work-flow management and the kepler system,”Concurrency andComputation: Practice and Experience, vol. 18, no. 10, pp.1039–1065, 2006.

[92] I. Taylor, I. Wang, M. Shields, and S. Majithia “Distributed computing with triana on the grid, ”Concurrency and Computation: Practice and Experience, vol. 17, no. 9, pp. 1197–1214, 2005.

[93] Y. Gil, V. Ratnakar, J. Kim, P. Gonzalez-Calero, P. Groth, J. Moody, and E. Deelman, “Wings: Intelligent workflow-based design of computational experiments” IEEE IntelligentSystems, pp. 62–72, 2010.

[94] M. Wieczorek, A. Hoheisel, and R. Prodan “Towards ageneral model of the multi-criteria workflow scheduling onthe grid” Future Generation Computer Systems, vol. 25,no. 3, pp. 237–256, 2009.

[95] J. Yu, R. Buyya, and K. Ramamohanarao “Workflow schedul-ing algorithms for grid computing” on Metaheuristics forscheduling in distributed computing environments. Springer, 2008, pp. 173–214.

[96] S. K. Garg, R. Buyya, and H. J. Siegel “Time and cost trade-off management for scheduling parallel applications on utilitygrids” Future Generation Computer Systems, vol. 26, no. 8,pp. 1344–1355, 2010.

[97] D. Yu, C. Li, and Y. Yin “Optimizing web service composi-tion for data-intensive applications” International Journal ofDatabase Theory and Application, vol. 7, no. 2, 2014

[98] A. Rostami, K. Wang, Z. Ghenretensae, P. Öhlen, and B. Skubic "First Experimental Demonstration of Orchestration of Optical Transport, RAN and Cloud based on SDN," Optical Fiber Communications Conference and Exhibition (OFC), pp. 11-13, March 2015

[99] J. Charzinski Internet client traffic measurement and characterisation results. Proceedings of the 13th International symposium on services and local access, 2000.

[100] J Julian Martin Kunkel, Thomas Ludwig, Hans Werner Meuer “Impact of future trends on exascale grid and cloud computing” 29th International Conference ISC pp. 215-231, June 2014

[101] Qiang Wang, Jie Zhang, Yongli Zhao, Hui Yang, Yiming Yu, Wei Wang “Experimental demonstration of remote centralized control platform with cloud service in software defined optical network” 9th International Conference on Communications and Networking in China (CHINACOM), pp. 298 – 302, Aug. 2014

[102] P.Gutierrez and J. Carapinha, “Cloud Networking: Implications of Agile Virtualisation on Provider Relationships”, Electronic Communications of the EASST, 2011

[103] J.Carapinha, et.al. “D2.1:Reference Scenarios and Technical System Requirements Definition”,Mobile Cloud Networking FP7 project,April 2013.

[104] A. Csaszar, et al., “Unifying Cloud and Carrier Network - EU FP7 Project UNIFY”, Workshop on Distributed Cloud Computing (DCC), December 2013.

[105] G. Xilouris1, et al., “T-NOVA: A Marketplace for Virtualized Network Functions”, European Conference on Networks and Communications (EUCNC), June 2014.

[106] Monteleone G., Paglierani P., "Session Border Controller Virtualization Towards "Service-Defined" Networks Based on NFV and SDN," Future Networks and Services, IEEE SDN, Nov. 2013

[107] Batalle J. "On the Implementation of NFV over an OpenFlow Infrastructure: Routing Function Virtualization," Future Networks and Services (SDN4FNS), IEEE SDN, vol., no., pp.1,6, 11-13 Nov. 2013

[108] J. Martins, M. Ahmed, et.al,“ClickOS and the art of network function virtualization”,inthe 11th USENIX Conference on Networked Systems Design and Implementation (NSDI'14), 2014

[109] F. Belqasmi, F. Chunyan, M. Alrubaye, and R. Glitho. Design and implementation of advanced multimedia conferencing applications in the 3gpp ip multimedia subsystem. IEEE Communications Magazine, 47(11):156-163, November 2009.

[110] Cloud ready data center network guide: [Електронний ресурс] / Juniper Networks. – Режим доступу: http://www.juniper.net/us/en/local/pdf/design-guides/8020014-en.pdf.

[111] M. Aramudhan and V. Rhymend Uthaiaraj: “A Study on Enhancing QoS Through Dynamic Service Prioritization in Web Services”. Asian Journal of Information Technology 4 (10): 954-956, 2005.

[112] M. Sato. Creating next generation cloud computing based network services and the contributions of social cloud operation support system (oss) to society. Proceedings of the IEEE International Workshops on Infrastructures for Collaborative Enterprises, pages 52-56, 2009.

[113] M. Tian, T. Voigt, T.Naumowicz, H. Ritter, J. Schiller: “Perfomance considerations for mobile web services”, Institut für Informatik, Freie Universität Berlin, Germany.

[114] S. Zhang, X. Chen, and S.Wu. Analysis and research of cloud computing system instance. Proceedings of the 2nd International Conference on Future Networks, pages 88-92 , 22-24, January 2010.

[115] Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros: Modeling and Simulation of Scalable Cloud Computing Environments and the Cloud Sim Toolkit:

Challenges and Opportunities. Grid Computing and Distributed Systems(GRIDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne,Australia, 2009.

[116] Susmita Horrow, Sanchika Gupta, Anjali Sardana, Ajith Abraham: Secure Private Cloud Architecture for MobileInfrastructure as a Service. Department of mathematics IIT Roorkee, India, 2012.

[117] Волик Б.Г., Рябинин Й.А. Эффективность, надежность и живучесть управляющих систем // Автоматика и телемеханика. -1984.- № 12.

[118] Крапивин В.Ф. О теории живучести сложных систем. — М.: Наука, 1978. — 248 с.

[119] Broder A., Kumar R., Maghoul F. et al. Graph structure in the Web / A. Broder, R. Kumar, F. Maghoul et al. // Proc. 9th Int. World Wide Web Conf. on Computer Networks: The International Journal of Computer and Telecommunications Networking. - Amsterdam, 2000. - P. 309

[120] Amazon Web Services LLC, “Amazon Elastic Compute Cloud (Amazon EC2), http://aws.amazon.com/ec2/, 2010.

[121] J. Dean and S. Ghemawat, “Mapreduce: simplified data processingon large clusters,” Commun. ACM, vol. 51, no. 1, pp.107–113, 2008.

[122] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, “Networktomography: recent developments,” Statistical Science,vol. 19, pp. 499–517, 2004.

[123] Abdull Am. S. and Ringwood G.: "Garbage Collecting the Internet: A Survey of Distributed Garbage Collection." ACM Comput. Surv., (30)3:330-373, Sept. 1998. Cited on page 186.

[124] Aberer K. and Hauswirth M.: "Peer-to-Peer Systems." In Singh, M. (ed.), The Practical Handbook of Internet Computing, chapter 35. Boca Raton, FL: eRC Press, 2005. Cited on page 15.

[125] Alonso G., Casati F., Kuno H., and MACmRAJU, V.: Web Services: Concepts, Architectures and Applications. Berlin: Springer-Verlag, 2004. Cited on pages 20, 551, 554,632.

[126] Bal H. The Shared Data-Object Model as a Paradigm for Programming Distributed Systems. Ph.D .. Thesis, Vrije Universiteit, Amsterdam, 1989. Cited on page 449.

ДОДАТОК. АКТИ ВПРОВАДЖЕННЯ ДИСЕРТАЦІЙНИХ ДОСЛІДЖЕНЬ

"ЗАТВЕРДЖУЮ"
Проректор з науково-педагогічної роботи
НУ "Львівська політехніка"
_____ доц. Давидчак О.Р.
"14" 04 2016 р.

АКТ

про використання результатів кандидатської дисертаційної роботи
Шпур Ольги Миколаївни
"Підвищення якості надання композитних сервісів у мережах із сервісно-орієнтованою архітектурою"
у навчальному процесі кафедри телекомунікацій

Даний акт складений комісією у складі:
- д.т.н., проф. Убізський С.Б., голова методичної ради Інституту телекомунікацій, радіоелектроніки та електронної техніки;
- к.т.н., доц. Озірковський Л.Д., декан базової вищої освіти Інституту телекомунікацій, радіоелектроніки та електронної техніки;
- д.т.н., проф. Климаш М.М., завідувач кафедри телекомунікацій

про те, що в навчальному процесі кафедри телекомунікацій використано результати кандидатської дисертаційної роботи "Підвищення якості надання композитних сервісів у мережах із сервісно-орієнтованою архітектурою", а саме:

- модернізовано курси лекцій для студентів напряму 6.050903 «Телекомунікації» з дисциплін: «Телекомунікаційні та інформаційні мережі, ч.1» - у частині теоретичних основ проектування розподілених сервісно-орієнтованих телекомунікаційних систем; «Телекомунікаційні системи передачі» - у частині, що стосується методики побудови та модернізації транспортних телекомунікаційних мереж на основі Cloud-технологій;
- модернізовано курси лекцій з дисциплін «Розподілені сервісні системи та Cloud-технології» та «Системне програмування інфокомунікацій» для студентів спеціальності 8.05090301 «Інформаційні мережі зв'язку», у якому використано запропоновані у роботі методи та моделі надання сервісів у мережах із сервісно-орієнтованою архітектурою.

Члени комісії: _____ Убізський С.Б.
_____ Озірковський Л.Д.
_____ Климаш М.М.

"ЗАТВЕРДЖУЮ"

Директор
ПП "Цифрові технології"

_____ Танчак З.В.

"26" 04 2016 р.

АКТ

про використання результатів кандидатської дисертаційної роботи
Шпур Ольги Миколаївни на тему:
"Підвищення якості надання композитних сервісів у мережах із сервісно-орієнтованою архітектурою"

Даний акт складений про те, що у ПП "Цифрові технології" для підвищення якості обслуговування абонентів у процесі надання розподілених хмаринкових сервісів та підвищення доступності композитних інформаційних застосувань використані результати дисертаційної роботи Шпур О.М. "Підвищення якості надання композитних сервісів у мережах із сервісно-орієнтованою архітектурою", представленої на здобуття наукового ступеня кандидата технічних наук, а саме:

- для оцінки доступності компонентів хмаринкового сервісу використано запропонований метод балансування навантаження з урахуванням доступності фізичних ресурсів, який реалізований на основі модифікованої сервісно–орієнтованої архітектури управління мережею, який враховує кількісні показники обслуговування та логічної зв'язності між елементарними сервісами, що дає можливість встановити характеристики функціонування композитних додатків;
- модель управління мережними ресурсами, яка функціонує на основі розробленого алгоритму прокладання наскрізних тунелів для передавання даних, дала змогу зменшити завантаженість граничних маршрутизаторів мереж, що забезпечило суттєве підвищення продуктивності Cloud-системи за рахунок ефективного розподілу мережних ресурсів на основі об'єднання та перегрупування потоків запитів.

Внаслідок перевірки використаних моделей на мережному обладнанні у ПП "Цифрові технології" встановлено, що результати знаходяться в межах трьохвідсоткового середньоквадратичного відхилення від поданих у дисертаційній роботі.

Провідний інженер Дрофяк А.М.

"ЗАТВЕРДЖУЮ"
Директор Львівської філії
ПАТ "Укртелеком"

_____ Андрухів Т.В., к.т.н.
"21" 04 2016 р.

АКТ
про використання результатів кандидатської дисертаційної роботи
Шпур Ольги Миколаївни
"Підвищення якості надання композитних сервісів у мережах із сервісно-орієнтованою архітектурою"

Даний акт складений про те, що у Львівській філії ПАТ "Укртелеком" використані результати кандидатської дисертаційної роботи Шпур О.М. "Підвищення якості надання композитних сервісів у мережах із сервісно-орієнтованою архітектурою ". А саме:

- завдяки використанню віртуалізації мережних функцій у розподіленій інформаційно-телекомунікаційній інфраструктурі підвищено ефективність використання та термін експлуатації апаратних ресурсів центру обробки даних та ймовірність успішного виконання запиту в моменти пікових навантажень;
- завдяки урахуванню структури центру обробки даних та процесу міграції компонентів сервісу у розподіленій інформаційно-телекомунікаційній інфраструктурі знижено обсяг службового трафіку, що виникає внаслідок втрати запитів;
- використано модель розподілу мережних ресурсів між дата–центрами на основі об'єднання та перегрупування потоків запитів, що дало змогу визначити вплив факторів перевантаження на показники відмовостійкості такої системи.

Результати експериментальних досліджень, виконаних на виробничих потужностях Львівської філії ПАТ "Укртелеком" відповідають результатам досліджень, що представлені у дисертаційній роботі, похибка не перевищує 5%.

Начальник відділу
планування мереж з КК Качан В.М.