

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2025 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи проектування
та побудови топології мереж у центрах обробки даних”

Виконав здобувач вищої освіти
II курсу, групи КІ-24М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Подолян Б.В.
« ____ » _____ 2025 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Марченко К.М.
« ____ » _____ 2025 р.
Рецензент _____

АНОТАЦІЯ

Подолян Б.В. Дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи проектування та побудови топології мереж у центрах обробки даних.

Метою розробки є дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних.

Об'єктом дослідження є процес проектування та побудови топології мереж у центрах обробки даних.

Предметом дослідження є методи проектування та побудови топології мереж у центрах обробки даних.

Методи дослідження базуються на методах побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, побудова топології мереж, центри обробки даних

ABSTRACT

Podolian B.V. Research and software implementation of the system for designing and building network topology in data centers. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the system for designing and building network topology in data centers.

The purpose of the development is the research and software implementation of the system for designing and building network topology in data centers.

The object of the research is the process of designing and building network topology in data centers.

The subject of the research is the methods of designing and building network topology in data centers.

The research methods are based on methods of building computer networks, methods of mathematical statistics, methods of software development.

The result of the work is the software implementation of the system for designing and building network topology in data centers.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with the software are provided.

The program can be used on a PC with Windows 10/11.

The program was developed in the Python environment.

Keywords: computer engineering, network topology construction, data centers

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	15
2.3 Розгорнута постановка завдання	20
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	22
3.1 Опис функціонування системи	22
3.2 Розробка структурної схеми.....	25
3.3 Розробка функціональної схеми	30
3.4 Розробка діаграми процесів.....	39
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	41
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	41
4.2 Захист розробленого програмного забезпечення.....	56
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	58
6 НАУКОВА НОВИЗНА	63

					ВКРМ-123.25.0055.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних	Літ.	Аркуш	Аркушів
Розроб.	Подольн Б.В.					М	1	87
Перев.	Марченко К.М.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-24М		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	64
7.1	Визначення цільової аудиторії кінцевого готового продукту	64
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	65
7.3	Вибір методу оцінки вартості ПЗ	65
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	66
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	68
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	68
7.7	Визначення ключових факторів успіху конкретного проєкту.....	69
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	70
8.1	Вступ.....	70
8.2	Шкідливі і небезпечні фактори при роботі з комп'ютером.....	71
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці користувача ПК	72
8.4	Розробка заходів з умов поліпшення охорони праці	75
8.5	Протипожежний захист	76
8.6	Розрахункова частина	77
9	ОСНОВНІ ВИСНОВКИ.....	79
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- IT – інформаційні технології
- ПЗ – програмне забезпечення
- ЦОД – центр обробки даних
- QoS – механізми контролю й керування якістю

КБПЗ_2025

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Надійна топологія мережі центру обробки даних є важливою для забезпечення ефективної, масштабованої та безпечної роботи мережі в сучасному цифровому середовищі.

Ви інвестували у високоякісні мережеві комутатори та маршрутизатори. Ви оптимізували свої програми для обробки запитів з мінімальною затримкою. І все ж продуктивність мережі вашого центру обробки даних все ще низька.

Виною може бути неоптимальна топологія мережі центру обробки даних. Оскільки топології мережі відіграють ключову роль у визначенні ефективності потоку мережевого трафіку в центрі обробки даних, погані топології можуть знижувати продуктивність, навіть якщо ваші програми та інфраструктура в іншому розраховані на швидкість.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем проектування та побудови топології мереж у центрах обробки даних.
- Дослідження системи проектування та побудови топології мереж у центрах обробки даних.
- Програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних.

Об'єктом дослідження є процес проектування та побудови топології мереж у центрах обробки даних.

Предметом дослідження є методи проектування та побудови топології мереж у центрах обробки даних.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Методи дослідження базуються на методах побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод проектування та побудови топології мереж у центрах обробки даних.

– Розроблено вітчизняний продукт проектування та побудови топології мереж у центрах обробки даних, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі проектування та побудови топології мереж у центрах обробки даних.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2025 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

У центрі обробки даних топологія мережі стосується структури та розташування мережевого обладнання та його зв'язку із серверами.

Щоб повністю зрозуміти, що це означає, давайте зробимо крок назад і поговоримо про те, як мережевий трафік протікає в центрі обробки даних. Як правило, трафік зрештою має переміщуватися до серверів і від них.

Однак, зазвичай не дозволяється серверам безпосередньо взаємодіяти із зовнішніми пристроями, оскільки це ускладнить ефективну маршрутизацію трафіку та може підвищити ризики безпеки.

Отже, замість безпосереднього підключення серверів до Інтернету, більшість центрів обробки даних використовують мережеву топологію, в якій мережеві маршрутизатори та комутатори служать посередниками між серверами та зовнішнім світом. Маршрутизатори та комутатори можуть виконувати операції, що допомагають захистити та підвищити ефективність потоків даних.

1.2 Область застосування

Топологія мережі – це точний спосіб розташування маршрутизаторів та комутаторів відносно серверів.

Топологія мережі центру обробки даних важлива з кількох причин:

– Продуктивність мережі: Продуктивність мережі залежить від здатності переміщувати пакети якомога швидше та з мінімальною затримкою між серверами та зовнішніми кінцевими точками. Погана топологія мережі може створювати вузькі місця, що знижують продуктивність мережі.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

– Масштабованість: Обсяг мережевого трафіку, який проходить через центр обробки даних, може змінюватися з часом. Щоб врахувати ці зміни, мережеві топології повинні бути достатньо гнучкими для масштабування.

– Економічна ефективність: мережеве обладнання може бути дорогим, а комутатори або маршрутизатори, які використовуються недостатньо, є нераціональною тратою грошей. В ідеалі, топологія мережі повинна забезпечувати ефективне використання комутаторів і маршрутизаторів, але без наближення до точки перевантаження та зниження продуктивності мережі.

– Безпека: Хоча безпека не є першочерговим фактором під час проектування топології мережі, оскільки можна застосовувати політики безпеки, використовуючи будь-яку поширену мережеву структуру, топологія все ж відіграє певну роль у визначенні того, наскільки легко сегментувати сервери з Інтернету та фільтрувати шкідливий трафік.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Центри обробки даних знову переживають трансформацію. Значні коригування будуть необхідними у 2026 році. Дослідимо це у даному розділі.

Навігація інновацій

Розвиток індустрії центрів обробки даних зумовлений попитом на більшу обчислювальну потужність, покращену стійкість та підтримку інноваційних технологій, таких як штучний інтелект та периферійні обчислення. Ми хотіли б поділитися деякими висновками з нашого щорічного огляду технологічних та ринкових розробок.

Швидка цифровізація процесів і послуг підприємств і організацій продовжуватиме генерувати безпрецедентні обсяги даних у 2026 році та надалі. Ці дані чинять тиск на традиційні інфраструктури на ринку даних, особливо в центрах обробки даних. Центри обробки даних повинні одночасно транспортувати, обробляти, аналізувати та зберігати зростаючі обсяги інформації.

Це має наслідки для архітектури, інфраструктури, конфігураторів та підключення. Попит стимулює розробку (менших) високопродуктивних обчислювальних блоків з розширеними можливостями. Водночас центри обробки даних повинні вирішувати термінові завдання, пов'язані з щільністю упаковки, оглядом системи, затримкою, управлінням температурою, моніторингом систем та продуктивності в режимі реального часу, а також ефективним налаштуванням.

Зростаючий попит і складність підвищують ризики підключеності. Рішення, які раніше вважалися перспективними, з 2026 року повинні будуть зіткнутися з новими та швидкозмінними викликами. Водночас, у всіх програмах

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

мають бути забезпечені безперебійна обробка даних, безпека та операційна продуктивність.

Кожен, хто сьогодні думає про мережеві та комунікаційні рішення, повинен також забезпечити ще більшу ємність та ефективність, одночасно підвищуючи надійність та спрощуючи обслуговування. Цінний простір у комп'ютерних залах та стійках має використовуватися ще ефективніше, оскільки щільність інфраструктури та пристроїв зростає.

Ось чому центри обробки даних повинні шукати інтелектуальні шляхи міграції. Такі критерії, як беззрощені з'єднання, роз'єми дуже малого форм-фактора, стрічкові волокна товщиною лише 82 мікрони, платформи високої щільності, а також моніторинг та управління активами, мають критичне значення. Попередньо налаштовані шафи, які вже оснащені блоком живлення, охолодженням, функціями безпеки та підключення, можуть бути рішенням та спростити цей процес. Вони дозволяють реалізувати модульні концепції.

Готові до штучного інтелекту та передових технологій

Там, де застосунки штучного інтелекту переміщуються в центри обробки даних – тенденція, яка продовжуватиме різко зростати у 2026 році – вимоги до інфраструктури швидко змінюються. Процеси обробки даних, такі як машинне навчання (ML), моделі великих мов (LLM), автономне водіння, інтелектуальний аналіз даних, моделювання та візуалізація, вимагають високопродуктивних комп'ютерів, розташованих поблизу місць дії. Вища обчислювальна потужність означає вищу щільність живлення, що призводить до вищих вимог до охолодження.

Інноваційні програми, що потребують інтенсивної обробки даних, часто вимагають вийти за рамки традиційних хмарних інфраструктур. Йдеться про врахування затримки, ефективності, витрат та безпеки. Сучасні концепції центрів обробки даних ставлять під сумнів. Сьогоднішнє планування має враховувати налаштування завтрашнього дня.

Опитування McKinsey, проведене на початку 2024 року, підкреслює,

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

наскільки терміновою є ця справа. Згідно зі звітом, 72% підприємств вже інтегрували штучний інтелект принаймні в одну бізнес-функцію, що є значним зростанням порівняно з попередніми роками*.

Центри обробки даних на периферії представляють собою альтернативу в контексті зростання штучного інтелекту. MarketsandMarkets прогнозує, що ринок периферійних обчислень зросте з 60,0 млрд доларів США у 2024 році до 110,6 млрд доларів США у 2029 році.

Вони пропонують локальну обчислювальну потужність, яка дозволяє приймати рішення в режимі реального часу на основі штучного інтелекту. Пристрої Інтернету речей можуть бути тісніше інтегровані в периферійні центри обробки даних. Контроль над даними може бути посилений. Завдяки своїй віддаленій архітектурі периферійні центри обробки даних пропонують низьку затримку та швидку обробку даних, що робить їх ідеальними для робочих навантажень зі штучним інтелектом.

Дотримання вимог

Центри обробки даних на периферії висувають свої унікальні вимоги до планувальників та операторів. Планування рішення на периферії зі штучним інтелектом включає в себе конкретні міркування, такі як вибір місця розташування, термін служби інфраструктури, безпека, стабільність мереж та електропостачання, дистанційне керування та автономна робота. Їх слід розглядати як архітектуру для кожного окремого випадку використання.

Рішення на периферії повинні бути розроблені модульно, гнучко та масштабовано, щоб вони могли відповідати різним вимогам. Вони повинні підтримувати дистанційний моніторинг та прогнозне обслуговування. За будь-яких обставин має бути можливо максимізувати їхню потужність та запобігти перебоям.

Традиційного повітряного охолодження може бути недостатньо. Це означає, що необхідно враховувати передові рішення, такі як рідинне та гібридне охолодження. Це буде у випадку, коли електрична потужність на стійку

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

перевищує значний поріг у 30 кВт.

Обираючи гібридні шляхи

Варто відзначити деякі інші трендові теми, які відіграватимуть важливу роль у 2026 році та надалі: 5G, IoT (Інтернет речей) та гібридизація.

Інноваційні послуги 5G належать до додатків з інтенсивним використанням даних, які стимулюють периферійні обчислення та ставлять постачальників центрів обробки даних перед новими викликами. На периферії необхідної низької затримки та високої швидкості передачі даних можна досягти надійніше, ніж у хмарі.

Інтернет речей змушує діяти. Він вимагає від центрів обробки даних збільшення ємності сховищ та розробки ефективніших функцій обробки даних.

Самі центри обробки даних розроблятимуть певні підходи або використовуватимуть їх інтенсивніше для управління даними та отримання з них аналітичних даних. З цією метою вони дедалі більше покладатимуться на штучний інтелект, машинне навчання, автоматизацію та оркестрацію. У майбутньому з'являтиметься дедалі більше складних застосувань штучного інтелекту, які вимагатимуть гібридної інфраструктури центрів обробки даних з можливостями периферійних, локальних та хмарних ресурсів.

Ефективніше використання енергії

Оскільки споживання енергії зростає через потребу в зберіганні даних та обчислювальній продуктивності, сталий розвиток є важливішим, ніж будь-коли. Нові центри обробки даних проектуються з урахуванням енергоефективності. Вони інтегрують системи рідинного охолодження. Вони покладаються на вдосконалені структури сітки. А також використовують компактніші оптичні комутатори для покращення підключення.

Для центрів обробки даних державні стандарти сталого розвитку вже не є необов'язковими, а радше важливою операційною метою. Державні цілі щодо енергоефективності стимулюють інвестиції в інфраструктуру та модернізацію в усій галузі.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Goldman Sachs прогнозує, що до кінця десятиліття попит на електроенергію в центрах обробки даних може зрости на 160% і до 2030 року становитиме від 3% до 4% світового попиту на електроенергію.

Початкові операційні моделі на основі штучного інтелекту та інструменти їх управління можуть допомогти скоротити споживання енергії на 20-40 % для окремих технологій, таких як охолодження. Однак, згідно з поточними дослідженнями Uptime, інструменти штучного інтелекту для операційної ефективності втрачають довіру керівників підприємств. Але, як і ажіотаж навколо штучного інтелекту загалом, галузь також повинна залишатися професійною, коли йдеться про енергоспоживання та операційну ефективність. Тоді вона зможе зробити наступні великі кроки для покращення роботи центрів обробки даних.

Оптичні комутатори мають вирішальне значення для майбутнього центрів обробки даних. Завдяки їм центри обробки даних можуть задовольнити зростаючий попит на високошвидкісну передачу даних, масштабованість та енергоефективність. Вони підтримують новітні технології, такі як 800 Gigabit Ethernet.

Оптичні комутатори підтримують набагато вищу пропускну здатність та швидкість передачі даних порівняно зі звичайними електричними комутаторами. Ця здатність є надзвичайно важливою. Це єдиний спосіб впоратися зі зростанням обсягу даних, що генеруються хмарними обчисленнями, додатками штучного інтелекту та пристроями Інтернету речей, масштабованим та сталим способом з мінімальною затримкою та втратою сигналу.

Цікаво, що суто електрична передача даних без електрооптичного перетворення також може подолати вимоги до високої затримки. Це можливо завдяки інфраструктурі категорії 8 на коротких відстанях у стійках.

Ресурси

Для центрів обробки даних головне – це стиснення енергії загалом. Однак, більша потужність у комп'ютерних залах призводить до значно більшого виділення тепла. Саме тому системи рідинного охолодження та інші передові

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

технології охолодження стають дедалі важливішими для забезпечення надійності серверів та мережевої інфраструктури.

Такі рішення, як модульні стійки та інтегровані програмні платформи, дозволяють центрам обробки даних розподіляти ресурси більш інтелектуально. Це дозволяє їм ефективніше керувати своїми потребами в охолодженні, живленні та підключенні. Інноваційні інтелектуальні мережеві структури, які мінімізують втрати енергії, користуватимуться зростаючим попитом.

Крім того, власникам центрів обробки даних доводиться враховувати споживання простору та ємність. Найновіші моделі показують, що на стійку можливо виділити понад 160 кВт потужності. Раніше для цієї потужності використовувалося 20 або більше стійок у традиційному корпусі.

Гарантування безпеки

Зі зростанням складності мереж передачі даних зросли також вимоги до кібербезпеки. Gartner очікує, що витрати на інформаційну безпеку зростуть на 15% до майже 212 мільярдів доларів США у 2026 році.

Центри обробки даних повинні бути оснащені найсучаснішими засобами захисту, щоб забезпечити цілісність даних та запобігти порушенням безпеки. Інтелектуальні рішення для моніторингу на основі штучного інтелекту пропонують проактивне виявлення загроз. Вони дозволяють адміністраторам швидко реагувати та захищати критично важливу інфраструктуру.

Точки атаки можуть навіть зрости, якщо впроваджувати віддалене керування, що вимагає локальної хмарної інфраструктури. Якщо дані спільно використовуються на кількох сайтах, усі вони повинні бути захищені на однаково високому рівні безпеки.

Підготовка до завтрашнього дня

Інвестиції в інфраструктуру та модернізацію периферійних, колорадських та гіпермасштабованих центрів обробки даних адаптуються до змінних вимог. Центри обробки даних повинні мати можливість адаптуватися до кожного використання та кожного клієнтського випадку.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Вимоги до сталого розвитку, регулювання та ефективності швидко змінюються. Подальшого розвитку інфраструктури центри обробки даних можуть легше адаптуватися до цих вимог. Таким чином, вони можуть гарантувати безпеку від збоїв, зменшити вплив на навколишнє середовище та відповідати вимогам і нормативним актам клієнтів.

Модернізація виходить за рамки оновлення обладнання та сьогодні обирає цілісний підхід. Цей підхід інтегрує апаратне забезпечення, програмне забезпечення, енергоефективність та безпеку в універсальну екосистему.

Планування потужностей та масштабування інфраструктури з одночасним збереженням високої доступності та мінімальних перебоїв у наданні послуг залишаються проблемою. Зокрема, це пов'язано з тим, що сьогодні пристрої постійно додаються, переміщуються або замінюються, оскільки центри обробки даних масштабуються та змінюються.

Центри обробки даних дедалі більше покладаються на автоматизоване управління та рішення DCIM. Експертний рівень може отримувати аналітичні дані, пов'язані з KPI, з усіх систем у центрі обробки даних та представляти їх у зрозумілій та практичній формі. Інтеграція штучного інтелекту та доповненої реальності (AR) в управління активами може призвести до покращення використання ресурсів та прийняття рішень.

Цифрові двійники на основі штучного інтелекту можуть бути використані для оптимізації планування та будівництва центрів обробки даних. Багато факторів необхідно враховувати під час будівництва нових центрів обробки даних, але ще більше – під час перепроєктування існуючих приміщень з усіма їхніми застарілими операційними технологіями.

Комплексні рішення з R&M

R&M пропонує масштабовані, адаптивні та перспективні рішення для центрів обробки даних, які можна налаштувати відповідно до нових трансформацій, вимог та технологій. Міцна мережа партнерських відносин з інтеграторами, додатковими постачальниками та іншими однодумцями збагачує

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

асортимент продуктів. Це дозволяє R&M надавати комплексні, інтегровані інфраструктурні рішення.

Ми можемо обслуговувати всі аспекти, включаючи рішення для периферійних та мікроцентрів обробки даних. Це забезпечує гнучкий та ефективний спосіб управління даними в джерелі, зберігаючи при цьому продуктивність хмари, яка покриває всі відповідні вимоги ОТ. Інноваційні рішення для центрів обробки даних інтегрують програмне та апаратне забезпечення для безперебійної координації.

Чого ще очікувати у 2026 році? R&M розширює високопродуктивну мережеву платформу Netscale. Це дозволяє центрам обробки даних ще більше підвищити ефективність та потужність у необчислювальній сфері. Нові розробки, такі як оптична розподільна рамка PRIME (ODF), налаштовані для задоволення зростаючих вимог за допомогою масштабованих та компактних систем управління оптичними оптоволоконками. Завдяки загальносистемному підходу до інфраструктури ми підвищуємо рівень обслуговування та забезпечуємо чудову підтримку, продуктивність та адаптивність, щоб центри обробки даних могли впоратися з майбутніми трансформаціями.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – динамічна інтерпретована об'єктно-орієнтована скриптова мова програмування із строгою динамічною типізацією. Офіційний сайт мови програмування Python <https://www.python.org/>. Python – багатоцільова мова програмування, яка дозволяє писати код, що добре читається. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Python – багатоплатформова мова програмування. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Ще однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм тощо. Програми, написані на мові програмування Python, можуть бути як невеликими скриптами, так і складними системами. Python абсолютно безкоштовний.

Швидкість виконання коду Python

Один з можливих недоліків Python – швидкість виконання коду. Python не є компільованою мовою. Код на Python спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. У більшості випадків при використанні Python виходять програми повільніші в порівнянні з такими мовами, як C.

Втім, сучасні комп'ютери мають таку обчислювальну потужність, що для більшості застосунків швидкість розробки важливіша швидкості виконання, а програми на Python зазвичай пишуться набагато швидше.

Окрім того, Python легко розширюється модулями, написаними на C або C++. Такі модулі можуть використовуватися для виконання частин програми, що створюють інтенсивне навантаження на процесор.

Використання Python

Python використовується для різних цілей: для створення ігор і веб-застосунків, розробки внутрішніх інструментів для різноманітних проектів. Мова також широко застосовується в науковій області для досліджень і розв'язування прикладних завдань.

Застосування мови програмування Python:

1. BitTorrent – протокол для обміну даними.
2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

3. Blender – програма для створення тривимірної комп’ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.

4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.

5. World of Tanks.

6. Вільна енциклопедія Вікіпедія.

7. Пошукова система Google.

8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.

9. YouTube – популярне відеосховище.

Версії Python

Мови програмування з часом змінюються – розробники додають в них нові можливості, а також виправляють помилки. Так з’являються різні версії мови. Наприклад, код написаний на Python 2 у більшості випадків не буде працювати у версії Python 3 без внесення додаткових змін.

Процесор є найважливішим компонентом в комп’ютері. Одна з основних функцій процесора – це обробка даних згідно комп’ютерної програми, яка є списком інструкцій, шляхом виконання арифметичних і логічних операцій над фрагментами даних.

Кожна інструкція в програмі – це команда, яка «повідомляє» процесору, яку операцію він повинен виконати. Процесор комп’ютера може розуміти лише ті інструкції, які написані на машинній мові. Машинна мова – це штучна мова, створена для передачі команд комп’ютеру. За допомогою машинної мови створюються ефективні програми, оскільки розробник отримує доступ до всіх можливостей процесора. Машинна мова – мова низького рівня.

Інструкція машинної мови існує для кожної операції, яку процесор здатний виконати – є інструкція для додавання чисел, є інструкція для віднімання

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

чисел і т.д. Увесь набір інструкцій, який центральний процесор може виконати, відомий як набір інструкцій процесора.

Наприклад, у вас є певна програма, яка зберігається на диску вашого комп'ютера. Для виконання програми, ви здійснюєте подвійний клік на значку програми. Це змушує програму копіюватися з диска в оперативну пам'ять, після чого процесор комп'ютера виконує копію програми, яка знаходиться в оперативній пам'яті.

Коли процесор виконує інструкції програми, він бере участь у процесі, який є відомим як цикл `fetch – decode – execute` (отримати – декодувати – виконати). Цей цикл виконується для кожної інструкції у програмі і складається з трьох кроків:

Отримати

Програма – це послідовність інструкцій на машинній мові. Першим кроком циклу є завантаження (отримання) наступної інструкції з пам'яті в процесор.

Декодувати

Інструкція машинної мови – це двійкове число, яке представляє команду, що повідомляє процесору виконати певну операцію. На цьому кроці процесор декодує інструкцію, яку було «витягнуто» з пам'яті, для визначення того, яка операція повинна виконуватись.

Виконати

Останній крок циклу – виконати операцію.

Хоча процесор комп'ютера розуміє тільки машинну мову, людині непрактично писати програми на машинній мові. Така програма може мати тисячі або навіть мільйони бінарних інструкцій, і написання такої програми буде дуже обтяжливим процесом.

З цієї причини була створена мова асемблера як альтернатива машинній мові. Замість використання двійкових чисел для написання інструкцій, мова асемблера використовує короткі слова, відомі як мнемокоди.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Незважаючи на те, що мова асемблера не вимагає двійкових інструкцій, як у випадку машинної мови, проте вона вимагає високих знань про процесор. Використовуючи мову асемблера, навіть для найпростішої програми, необхідно написати велику кількість інструкцій.

Мова програмування високого рівня дозволяє створювати складні програми, не знаючи, як працює процесор, і не записуючи великої кількості інструкцій низького рівня. Крім того, більшість мов програмування високого рівня використовують слова, які легко зрозуміти.

Python – одна із популярних сучасних мов програмування високого рівня. Python – інтерпретована мова програмування. Python – це високорівнева інтерпретована мова програмування, на відміну від C++, яка є прикладом компільованої мови програмування. Назва Python відноситься як до мови програмування, так і до інтерпретатора – комп'ютерної програми, яка зчитує початковий код (написаний на Python) і виконує інструкції (команди).

Для перекладу мови високого рівня на машинну мову доступні два типи програм:

1. Компілятор.
2. Інтерпретатор.

Завантаження Python

Версії інтерпретатора Python для різних операційних систем доступні для безкоштовного завантаження за адресою <https://www.python.org/downloads>.

Середовище програмування для Python

Для написання програм використовують текстові редактори або інтегровані середовища розробки, які включають в себе різні інструменти для роботи з кодом: засіб для написання коду (текстовий редактор), інтерактивний інтерпретатор, відлагоджувач тощо.

Текстові редактори та інтегровані середовища програмування для Python:

– IDLE – стандартний редактор Python. Встановлюється разом з Python для користувачів Windows, окремим пакунком для користувачів Linux.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

– Notepad++ – безкоштовний текстовий редактор початкового коду, який підтримує велику кількість мов, в тому числі і Python. Лише для користувачів Windows.

– Visual Studio Code – це легкий, але потужний редактор початкового коду, який розповсюджується безкоштовно і доступний у версіях для платформ Linux, Windows і macOS.

– PyScripter – інтегроване середовище розробки для мови програмування Python. Для користувачів Windows. Поширюється безкоштовно.

– Wing IDE 101 – вільне інтегроване середовище для Python, розроблене для навчання програмістів-початківців. Для користувачів Linux, Windows і macOS. Поширюється безкоштовно.

– Geany – вільний текстовий редактор з базовими елементами інтегрованого середовища розробки, доступний для операційних систем Linux, Windows і macOS.

– PyCharm – інтегроване середовище розробки для мови програмування Python. PyCharm є власницьким програмним забезпеченням. Наявна безкоштовна версія Community з усіченим набором можливостей. Для користувачів Linux, Windows і macOS.

– Thonny – IDE для вивчення програмування мовою Python. Для користувачів Linux, Windows і macOS.

– Mu – редактор коду Python для програмістів-початківців. Для користувачів Linux, Windows і macOS.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи проектування та побудови топології мереж у центрах обробки даних.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Існує кілька способів проектування топології мережі центру обробки даних. Ось найпоширеніші:

1. Трирівнева топологія

Традиційно більшість центрів обробки даних використовували так звану трирівневу мережеву топологію. Згідно з цією схемою, мережеві ресурси організовані на три рівні:

- Рівень доступу: забезпечує пряме підключення до пристроїв кінцевих користувачів, що дозволяє здійснювати зв'язок з мережею.

- Рівень розподілу: діє як посередник, агрегуючи трафік від комутаторів доступу та забезпечуючи дотримання політик для керування маршрутизацією та безпекою.

- Базовий рівень: забезпечує високошвидкісне та надійне з'єднання між рівнями розподілу та зовнішніми мережами, обробляючи трафік магістральної мережі.

Трирівнева топологія відносно проста в реалізації. Її головним недоліком є те, що вона не дуже гнучка та нелегка в масштабуванні, оскільки відносно невелика кількість комутаторів та маршрутизаторів повинна обробляти великий обсяг мережевого трафіку. Отже, ці пристрої зазвичай є дорогими моделями високого класу, а це означає, що якщо ви хочете збільшити їхню ємність, вам доведеться додати дорожче обладнання.

Додавання такого обладнання не є ні простим, ні швидким, і може бути неефективним з точки зору витрат, якщо обсяги трафіку коливаються, а розширені можливості комутації та маршрутизації потрібні лише зрідка. У такому разі ви отримаєте дороге обладнання, яке буде використовуватися

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

недостатньо. З цих причин трирівнева топологія зазвичай не ідеальна для центрів обробки даних, обсяги мережевого трафіку яких коливаються, хоча вона може добре працювати, коли рівні трафіку є стабільними та передбачуваними.

2. Верхня частина полиці

У топології центру обробки даних «top-of-rack» ви встановлюєте мережеві комутатори в кожній серверній стійці. Таким чином, топологія «top-of-rack» спрощує швидке та економічно ефективно додавання потужності.

Оскільки обсяг трафіку, який має обробляти кожен комутатор, менший, ніж у трирівневій топології (де кілька стійок можуть бути підключені до одного комутатора), підхід «верхня частина стійки» може використовувати менш дорогі комутатори.

Таким чином, топологія «top-of-rack» спрощує дешеве та швидке додавання потужності. Конструкції «top-of-rack» також масштабуються природним чином залежно від потужності центру обробки даних, оскільки чим більше серверних стійок ви додаєте, тим більше комутаторів ви додаєте.

Недоліком є те, що вам доведеться встановлювати та обслуговувати більше комутаторів, що збільшує експлуатаційне навантаження на команди, відповідальні за розгортання та управління мережевим обладнанням. Конструкції, що встановлюються зверху стійки, також можуть знижувати ефективність у ситуаціях, коли серверна стійка не надсилає або не отримує достатньо трафіку для використання своїх комутаторів на повну потужність.

3. Перемикання мереж

Комутована структура мереж у деяких аспектах схожа на трирівневу топологію, але з більшою кількістю комутаторів її легше масштабувати. Вона також нагадує топологію «top-of-rack», з тією ключовою відмінністю, що сервери не призначені постійно до конкретного комутатора, що дозволяє ефективніше використовувати його потужність.

Комутована структура (fabric) у деяких аспектах схожа на трирівневу топологію, але в ній є більша кількість комутаторів, що полегшує

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

підключаються, зазвичай не займають забагато місця порівняно із серверами, фізичні обмеження простору можуть бути проблемою, особливо в невеликих центрах обробки даних або серверних кімнатах. Якщо так, то трирівнева топологія, як правило, працює краще, оскільки вона не вимагає стільки окремих мережевих пристроїв або кабелів.

3.2 Розробка структурної схеми

Використання хмарних обчислювальних сервісів зростає щороку [1] і має багатообіцяюче майбутнє [2] [3]. З 3,0 зеттабайтами IP-трафіку на рік та очікуваним трафіком 8,6 зеттабайтів у 2029 році [1], хмарний трафік зростає майже втричі. Однак цей трафік поділяється на три типи: від центрів обробки даних до користувачів, від центрів обробки даних до центрів обробки даних та всередині центрів обробки даних. Найбільша частина цього трафіку залишається всередині центрів обробки даних і відповідає типу трафіку всередині центрів обробки даних.

Щоб керувати таким величезним трафіком, приблизно 3342 екзабайти (ЕВ) у 2025 році та 7566 ЕВ, що очікуються у 2029 році [1], хмарні мережі перебувають під пильною увагою з метою пошуку нових рішень, що забезпечують достатню продуктивність та масштабованість. Серед цих нових рішень ми зосереджуємося як на TRILL (прозоре взаємозв'язок багатьох каналів) [4], так і на OpenFlow [5].

Перший має децентралізовану площину керування, тоді як другий має централізовану площину керування та центральний контролер. У розділах III-B та III-A ми детально розглядаємо відповідно TRILL та OpenFlow.

Другим за важливістю типом трафіку за кількістю є трафік від центру обробки даних до користувача. Очікується, що цей трафік зростає на 25%

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

протягом наступних чотирьох років. Це показує як збільшення кількості клієнтів хмар, так і збільшення завдань, делегованих інфраструктурі хмари.

Останній тип трафіку, між центрами обробки даних, є найменш важливим за кількістю. Однак, він матиме найсильніше зростання, зі збільшенням на 31% протягом наступних чотирьох років. У цій статті ми зосереджуємося на цьому типі трафіку, який становить приблизно 7% від загального хмарного трафіку. Крім того, цей трафік між центрами обробки даних має інші обмеження, ніж два інших. Той факт, що він є міжцентровим, має бути прихований від користувачів хмар. У більшості випадків, коли відбувається зв'язок між центрами обробки даних, центри обробки даних належать одному хмарному постачальнику. Тому здебільшого вони використовують одну й ту саму технологію. У цих випадках для рішення TRILL ми пропонуємо в [6] рішення, яке дозволяє взаємодіяти з кількома кампусами TRILL без їх об'єднання.

Однак у деяких випадках обидва центри обробки даних використовують різні технології, оскільки вони належать різним постачальникам. У [7] наведено декілька прикладів.

Перший – це хмарна федерація. Вона об'єднує хмарних постачальників, які спільно використовують свої ресурси за допомогою спільного регулювання для задоволення потреб своїх клієнтів.

Другий приклад – міжхмарне середовище. У цьому середовищі мережі кількох постачальників хмарних послуг взаємопов'язані, а їхні центри обробки даних використовують різні технології. У цих випадках трафік між центрами обробки даних має бути модифікований, щоб його приймав кожен центр обробки даних. Ця модифікація має бути прозорою для користувачів.

Третім прикладом може бути випадок, коли користувачі бажають скористатися перевагами продуктів різних постачальників хмарних послуг залежно від їхніх пропозицій та обмежень. Користувач може обрати одного постачальника хмарних послуг для зберігання величезної кількості даних, використовуючи іншого постачальника для його обчислювальної продуктивності,

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

і, нарешті, обрати третього постачальника з високою пропускнуою здатністю для розміщення своїх веб-сервісів.

Четвертим прикладом може бути компанія, яка має кілька філій. Кожна філія підключена до найближчого центру обробки даних, щоб зменшити час доступу. Ми бачимо, що якщо немає міжмарного доступу, то користувачі не зможуть досягти такого використання хмарних сервісів.

Спосіб взаємоз'єднання центрів обробки даних можна розділити на два підходи, кожен з яких має два рішення [7]. Якщо саме постачальник послуг хоче встановити взаємозв'язок центрів обробки даних, то це підхід, орієнтований на постачальника. В іншому випадку, коли клієнт хоче використовувати хмару кількох постачальників, це підхід, орієнтований на клієнта.

Серед підходів, орієнтованих на постачальника, є два рішення: федеративна хмара та гібридна хмара. У федеративному хмарному рішенні кілька постачальників хмарних послуг домовляються про правила та політики, що дозволяють взаємодію між своїми ресурсами.

Таким чином, вони можуть обмінюватися своїми невикористаними ресурсами один з одним. Клієнт хмари може навіть не знати про цей обмін ресурсами та використовувати інфраструктуру іншого постачальника.

Друге рішення – це гібридна хмара. У цьому рішенні організація з приватною хмарою, яка потребує більшої кількості ресурсів, використовуватиме хмару третьої сторони замість того, щоб розширювати власну інфраструктуру. Це рішення має економічну привабливість для клієнтів, оскільки таким чином їм не потрібно модифікувати свою приватну хмарну інфраструктуру.

Клієнтоорієнтований підхід містить два інших рішення: Мультихмарне рішення та Агрегований сервіс від брокера.

Мультихмарне рішення перекладає все управління кількома хмарними постачальниками та кількома технологіями на клієнта. У цьому випадку клієнту, можливо, доведеться розробити адаптер для API кожного постачальника. Таким

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

чином, постачальники не знають про своє хмарне з'єднання, оскільки саме клієнт керує ним.

Останнє рішення, Агрегований сервіс від брокера, схоже на попереднє, за винятком того, що клієнт не повинен керувати абстракцією API. Натомість клієнти використовують послуги третьої сторони, яка називається брокером. Він надає єдиний API для кількох хмарних постачальників, абстрагуючи кожен з API постачальників.

У даній роботі ми зосереджуємося на підході, орієнтованому на постачальника, а точніше на федеративному хмарному рішенні. Ми розробили рішення для з'єднання кількох центрів обробки даних за допомогою двох різних технологій. Це рішення є шлюзом, який з'єднує центри обробки даних за допомогою TRILL або OpenFlow.

Передумови OpenFlow

OpenFlow має централізовану площину керування, один центральний контролер та стандартні комутатори OpenFlow. Контролер OpenFlow має глобальне уявлення про топологію мережі та віртуальні мережі клієнтів. Контролер приймає кожне рішення щодо маршрутизації та кожне рішення щодо комутації всередині цієї мережі. Протокол OpenFlow здатний взаємодіяти з обладнанням, сумісним з OpenFlow, для заповнення їхньої таблиці комутації.

Ці таблиці комутації дозволяють застосовувати певний набір правил до кожного потоку. У разі відсутності таких правил для потоку, цей потік надсилається комутатором до контролера. Контролер є єдиним пристроєм, здатним вирішувати, які правила застосовувати до цього потоку.

Сумісні комутатори OpenFlow можуть бути як апаратними, так і програмними. Наприклад, Open vSwitch – це віртуальний комутатор, призначений для програмних гіпервізорів. Він пропонує мережеве підключення для віртуальних машин. Open vSwitch працює через таблицю потоків, яка визначає правила та дії для кожного потоку. Для ізоляції потоків кожного орендаря, комутатори OpenFlow можуть використовувати протоколи тунелювання, такі як

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Передумови прозорого з'єднання багатьох каналів (TRILL)

На відміну від централізованої площини керування OpenFlow, TRILL [4] має децентралізовану площину керування. Це призводить до більшої стійкості TRILL, ніж OpenFlow. TRILL не має єдиної точки відмови, на відміну від OpenFlow та його контролера. TRILL – це протокол другого рівня з повною неблокуючою структурою, яка дозволяє будь-яке з'єднання в мережі. Новий компонент (RBridge) визначається специфікаціями TRILL для заміни традиційних комутаторів. Навіть якщо не всі комутатори потрібно замінити для роботи TRILL, вибір комутаторів для заміни є важливим. Фактично, RBridge діє як проксі, додаючи свою адресу до кадру.

3.3 Розробка функціональної схеми

Перейдемо до розгляду функціональної схеми. Сформулюємо й вирішимо завдання аналізу функціональної структури системи проектування та побудови топології мереж у центрах обробки даних. Для цього розроблена концепція проведення аналізу, вирішені завдання формального опису структури й обчислення її характеристик: навантаження на канали зв'язку, вузли й структуроутворююче устаткування системи проектування та побудови топології мереж у центрах обробки даних.

На рисунку 3.2 представлена функціональна схема системи, які відповідає запропонованому підходу.

Основною метою аналізу структури є визначення параметрів потоків даних, що проходять по каналах зв'язку системи проектування та побудови топології мереж у центрах обробки даних й вступні на вузли системи проектування та побудови топології мереж у центрах обробки даних. Однак, завдання структури системи проектування та побудови топології мереж у центрах обробки даних тільки як сукупності вузлів і зв'язків між ними, не дозволяє досліджувати потоки даних, оскільки потоки даних формуються розв'язуваними на системи проектування та побудови топології мереж у центрах обробки даних

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

завданнями, точніше додатками, які запускаються на вузлах системи проектування та побудови топології мереж у центрах обробки даних й обмінюються між собою даними. Отже, для аналізу системи проектування та побудови топології мереж у центрах обробки даних необхідно відомості про функціональну структуру доповнити відомостями про додатки і їхнє розміщення в системі проектування та побудови топології мереж у центрах обробки даних.

Результатами аналізу повинні стати математичні залежності, що дозволяють обчислювати чисельні значення характеристик системи проектування та побудови топології мереж у центрах обробки даних з урахуванням особливостей конкретної структури системи проектування та побудови топології мереж у центрах обробки даних.

Запропоновано концептуальний підхід до аналізу функціональної структури системи проектування та побудови топології мереж у центрах обробки даних, заснований на дослідженні взаємодії додатків (завдань) як незалежних джерел і приймачів даних. У цьому випадку спочатку визначаються параметри потоків даних між додатками при виконанні всього комплексу завдань (будується інформаційна модель), а потім, залежно від розміщення додатків по вузлах системи проектування та побудови топології мереж у центрах обробки даних й використовуваного устаткування, визначаються параметри потоків даних між вузлами системи проектування та побудови топології мереж у центрах обробки даних (будується технічна модель).

При цьому не тільки повністю враховуються всі взаємодії між додатками, але й з'являється можливість проведення аналізу складних ієрархічних мережних структур, шляхом декомпозиції на підмережі системи проектування та побудови топології мереж у центрах обробки даних, що часто застосовується в технологіях VLAN і VPN. Даний підхід розвивається й застосовується в даній роботі.

Відзначимо, що отримані результати аналізу потоків даних, можна надалі використовувати для проведення більше глибоких досліджень із застосуванням відомих методів і моделей, наприклад, СеМО.

Для аналізу функціональної структури й розрахунку характеристик системи проектування та побудови топології мереж у центрах обробки даних

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

визначені правила її формального опису, що однозначно задають властивості додатків і структуру системи проектування та побудови топології мереж у центрах обробки даних, що дозволяють будувати моделі для проведення розрахунків. Відповідно до концептуального підходу до проведення аналізу функціональної структури системи проектування та побудови топології мереж у центрах обробки даних виділені дві складові її структури інформаційна й технічна:

– Інформаційна структура визначає інформаційні потоки між інформаційними вузлами, на яких встановлене програмне забезпечення й представляє сукупність вузлів і інформаційних ресурсів, розміщених на цих вузлах. Маючи у своєму розпорядженні дані про інформаційну структуру системи проектування та побудови топології мереж у центрах обробки даних можна приймати рішення про організацію каналів зв'язку між вузлами системи проектування та побудови топології мереж у центрах обробки даних, визначати необхідні параметри телекомунікаційної системи, формувати структуру системи проектування та побудови топології мереж у центрах обробки даних.

– Технічна структура – сукупність структуроутворюючого устаткування, технічних вузлів системи проектування та побудови топології мереж у центрах обробки даних й каналів зв'язку. Технічному вузлу можуть відповідати декілька інформаційних.

Таким чином, для повноцінного аналізу функціональної структури реальної системи проектування та побудови топології мереж у центрах обробки даних необхідно провести аналіз складових її інформаційної й технічної структур і зв'язати результати аналізу.

Зв'язування результатів аналізу інформаційної й технічної структур має на увазі відображення характеристик інформаційної структури в характеристики технічної структури й визначення параметрів технічної структури на основі параметрів і характеристик інформаційної структури.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

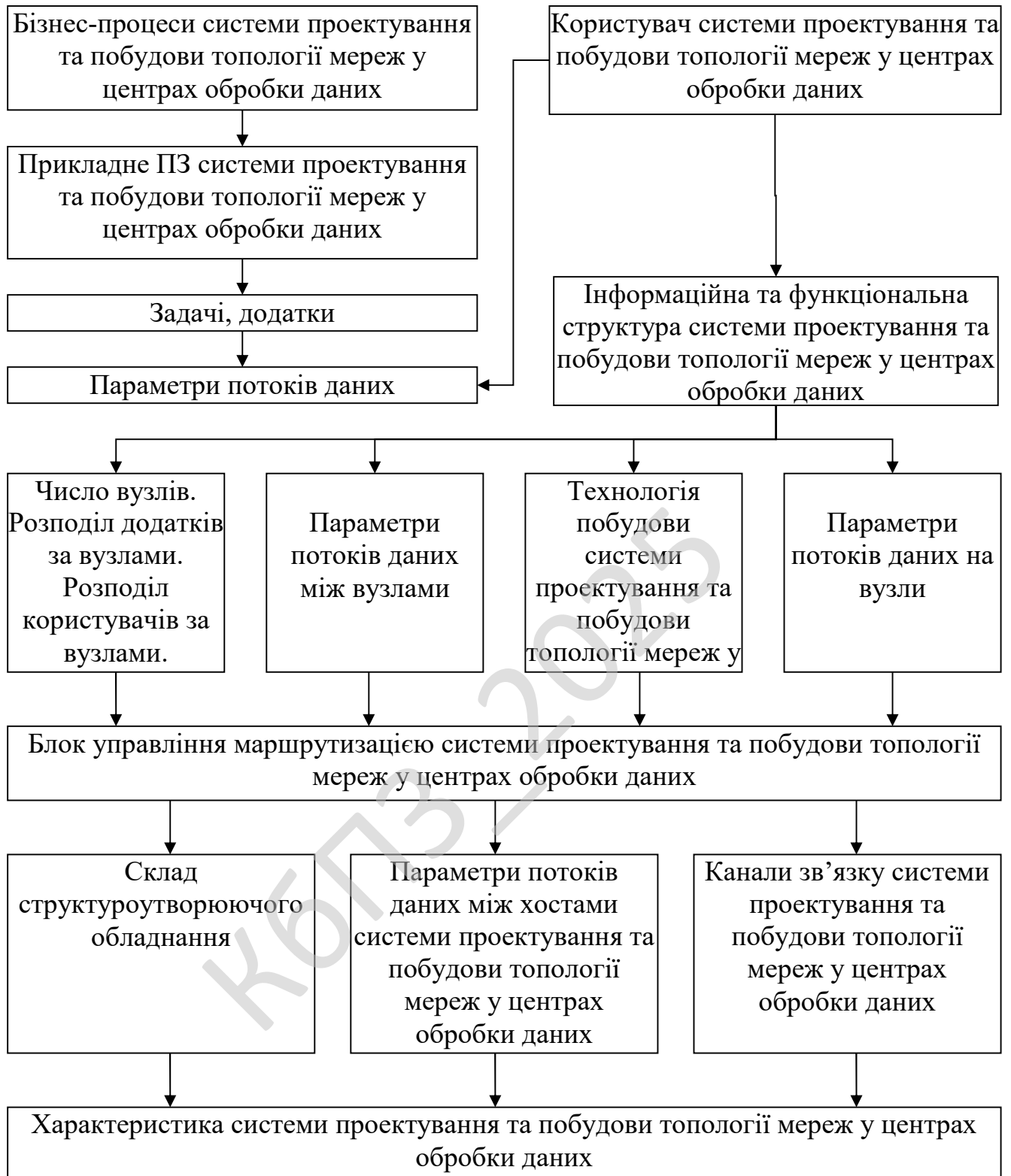


Рисунок 3.2 – Функціональна схема системи

Інформаційна структура системи проектування та побудови топології мереж у центрах обробки даних задається набором параметрів:

$$SI = \{N, M, D, L, R, S_k (k=1, 2, \dots, L), A_{km} (k=1, 2, \dots, L; m=1, 2, \dots, D), G, H, S\},$$

де:

- N – число працюючих користувачів;
- M – число задіяних вузлів;
- D – число використовуваних додатків;
- L – число розв'язуваних завдань;
- R – число баз даних;
- $S_k = \{p_k, d_k, u_k, W_k\}$, ($k=1, 2, \dots, L$) – набір даних, що описують

розв'язувані завдання, де:

- $p_k = (p_{k1}, p_{k2}, \dots, p_{kD})$, – вектор-рядок, що визначає завданням k додатка, що запускаються;

- $d_k = (d_{k1}, d_{k2}, \dots, d_{kR})$ – вектор-рядок, що визначає використовувані завданням k бази даних (сховища даних);

- $u_k = (u_{k1}, u_{k2}, \dots, u_{kN})$, – вектор-рядок, що визначає завдання k користувачів, що запускають $W_k = \|w_{kij}\|$ ($i=1, 2, \dots, D; j=1, 2, \dots, D$) – матриця, що встановлює послідовність запуску додатків завданням k ;

- $A_{km} = \{v_{km}, b_{km}\}$, ($k=1, 2, \dots, L; m=1, 2, \dots, D$) – набір даних, що описують додатки, використовувані завданнями, де:

- $v_{km} = (v_{km1}, v_{km2}, \dots, v_{kmR})$, – вектор-рядок, що задає обсяги даних, якими обмінюється додаток m з базами даних, при рішенні завдання k ;

- $b_{km} = (b_{km1}, b_{km2}, \dots, b_{kmD})$ – вектор-рядок, що задає обсяги даних, якими обмінюється додаток m з іншими додатками, при рішенні завдання k ;

- G матриця розміщення додатків по вузлах;

- H – матриця підключення користувачів до вузлів;

- S – матриця розміщення баз даних по вузлах.

Набір однозначно визначає інформаційну структуру системи проектування та побудови топології мереж у центрах обробки даних.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Для заданого набору **SI** отримані результати, що дозволяють визначити параметри потоків даних між вузлами системи проектування та побудови топології мереж у центрах обробки даних.

При цьому використовувалася матриця інтенсивностей потоків запитів користувачів на запуск завдань $\Lambda = \|\lambda_{ij}\|$, ($i=1,2,\dots,N$; $j=1,2,\dots,L$).

Показано, що матриця інтенсивностей потоків даних між вузлами системи проектування та побудови топології мереж у центрах обробки даних при рішенні завдання k обчислюється за формулою: $A_k = \lambda_k Z_k$, ($k=1,2,\dots,L$). , де Z_k – матриця обсягів даних, переданих між вузлами, при рішенні завдання. Обчислені також матриця інтенсивностей потоку запитів на запуск додатка номер j , на вузлі номер i – B^* і матриця сумарної інтенсивності потоку запитів до бази даних номер j , на вузлі номер i – Φ .

Таким чином, визначена множина параметрів потоків даних інформаційної структури системи проектування та побудови топології мереж у центрах обробки даних: $PSI(SI) = \{\Lambda, Z_k, A_k (k=1,2,\dots,L), A, B^*, \Phi\}$.

Розроблено моделі для аналізу ієрархічної трьохрівневої інформаційної структури системи проектування та побудови топології мереж у центрах обробки даних, що дозволяють визначити завантаження каналів зв'язку й мережного устаткування в системі проектування та побудови топології мереж у центрах обробки даних, що складається із сукупності підмереж і корпоративних серверів.

Отримано наступні результати:

– матриці інтенсивностей потоків даних між групами й усередині груп кожного рівня $A_1(C_1) = C_1 A(C_1)^T$, $A_2(C_2) = C_2 [A_1(C_1) - dg(A_1(C_1))](C_2)^T$;

– матриці інтенсивностей потоків даних утворених кожним завданням $A_{1k}(C_1)$ і $A_{2k}(C_1)$, ($A = \sum_{k=1}^L A_k$).

Досліджено властивості матриць, що відбивають взаємозв'язки потоків даних окремих завдань, сформульовані у вигляді доведених тверджень.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

Як міра ефективності ієрархічної структури запропоновані коефіцієнти поглинання інтенсивностей потоків даних на кожному рівні. Коефіцієнт поглинання на кожному рівні відбиває частку сумарної інтенсивності потоків, що локалізується усередині рівня.

Результати аналізу інформаційної структури дозволяють визначати параметри потоків даних між логічними об'єднаннями вузлів системи проектування та побудови топології мереж у центрах обробки даних. Інформаційна структура реалізується конкретними технічними засобами й втілюється у вигляді технічної структури. При цьому можливо невідповідність інформаційної й технічної структур, пов'язане з технічними характеристиками й можливостями апаратури. Наприклад, на одному технічному вузлі можуть бути встановлені кілька інформаційних вузлів. У зв'язку із цим розроблені методи й засоби аналізу технічної структури системи проектування та побудови топології мереж у центрах обробки даних, створеної на базі інформаційної структури.

Для аналізу роботи реальної системи проектування та побудови топології мереж у центрах обробки даних розроблений метод формального опису її технічної структури. Структура реальної системи проектування та побудови топології мереж у центрах обробки даних формується із застосуванням структуроутворюючого устаткування (комутатори, маршрутизатори), до якого підключаються вузли системи проектування та побудови топології мереж у центрах обробки даних, при цьому створюється багаторівнева (часто трьохрівнева) мережа. Такий підхід застосовується, як правило, при створенні системи проектування та побудови топології мереж у центрах обробки даних на базі технології VLAN. При цьому групи першого рівня інформаційної структури становлять віртуальні локальні системи проектування та побудови топології мереж у центрах обробки даних. Другий і третій рівні інформаційної структури призначені для з'єднання цих мереж між собою.

Число комутаторів, використовуваних для з'єднання вузлів технічної структури системи проектування та побудови топології мереж у центрах обробки

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

даних при створенні груп першого рівня (комутатори першого рівня), визначається особливостями реальної системи проектування та побудови топології мереж у центрах обробки даних, технічними можливостями комутаторів. Позначимо це число K_1^* , ($K_1^* \geq K_1 \geq 1$). Число комутаторів, для з'єднання комутаторів першого рівня й створення груп другого рівня (комутатори другого рівня) – K_2^* , ($K_2^* \geq K_2 \geq 1$). Число комутаторів третього рівня для з'єднання комутаторів другого рівня – $K_3^* \geq 0$.

Технічна структура системи проектування та побудови топології мереж у центрах обробки даних задається множиною **ST**, елементами якого є:

– $Y_1^* = \|y_{1ij}^*\|$, ($i = 1, 2, \dots, M; j = 1, 2, \dots, K_1^*$) матриця з'єднань технічних вузлів системи проектування та побудови топології мереж у центрах обробки даних (робітники станції, сервери) з комутаторами першого рівня;

– $Y_2^* = \|y_{2ij}^*\|$, ($i = 1, 2, \dots, K_1^*; j = 1, 2, \dots, K_2^*$) матриця з'єднань комутаторів першого рівня з комутаторами другого рівня;

– $Y_3^* = \|y_{3ij}^*\|$, ($i = 1, 2, \dots, K_2^*; j = 1, 2, \dots, K_3^*$) матриця з'єднань комутаторів третього рівня з комутаторами другого рівня;

– $X_1^* = \|x_{1ij}^*\|$, ($i, j = 1, 2, \dots, K_1^*$) матриця з'єднань комутаторів першого рівня;

– $X_2^* = \|x_{2ij}^*\|$, ($i, j = 1, 2, \dots, K_2^*$) матриця з'єднань комутаторів другого рівня;

– $X_3^* = \|x_{3ij}^*\|$, ($i, j = 1, 2, \dots, K_3^*$) матриця з'єднань комутаторів третього рівня.

Для кожного рівня задані матриці, що задають пропускні здатності каналів зв'язку, використовуваних при побудові мереж:

$$C_1^*(Y_1^*), C_2^*(Y_2^*), C_3^*(Y_3^*), C_1^*(X_1^*), C_2^*(X_2^*), C_3^*(X_3^*).$$

Для заданої множини **ST** отримані наступні результати:

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

– матриця інтенсивностей інформаційних потоків, між комутаторами першого рівня й усередині груп технічних вузлів, підключених до комутаторів першого рівня: $A_1^*(Y_1^*) = (Y_1^*)^T A Y_1^*$;

– матриця сумарних інтенсивностей інформаційних потоків для комутаторів другого рівня: $A_2^*(Y_2^*) = (Y_2^*)^T [A_1^*(Y_1^*) - dg(A_1^*(Y_1^*))] Y_2^*$;

– матриця інтенсивностей інформаційних потоків, між і усередині комутаторів третього рівня: $A_3^*(Y_3^*) = (Y_3^*)^T [A_2^*(Y_2^*) - dg(A_2^*(Y_2^*))] Y_3^*$;

– вектори інтенсивностей інформаційних потоків, що надходять на комутатори всіх рівнів: $\lambda_1^* = (\lambda_{11}^*, \lambda_{12}^*, \dots, \lambda_{1K_1}^*)$, $\lambda_2^* = (\lambda_{21}^*, \lambda_{22}^*, \dots, \lambda_{2K_2}^*)$, $\lambda_3^* = (\lambda_{31}^*, \lambda_{32}^*, \dots, \lambda_{3K_3}^*)$;

– вектор сумарних інтенсивностей потоків даних, переданих по каналах зв'язку $\gamma_i^* = (\gamma_{i1}^*, \gamma_{i2}^*, \dots, \gamma_{iM}^*)$, ($i = 1, 2, \dots, M$).

Отримані також формули для обчислення параметрів потоків даних кожного завдання.

Оскільки в корпоративних мережах великої розмірності використовується маршрутизація, то для обчислення параметрів потоків необхідно застосовувати відповідні алгоритми маршрутизації на графах. Для цього побудована матриця графа зв'язків між комутаторами Ω . Показано, що матриця Ω може бути представлена у вигляді:

$$\Omega = \begin{pmatrix} X_1^* & Y_2^* & \mathbf{0} \\ (Y_2^*)^T & X_2^* & Y_3^* \\ \mathbf{0} & (Y_3^*)^T & X_3^* \end{pmatrix}.$$

Таким чином, параметри потоків даних у системи проектування та побудови топології мереж у центрах обробки даних для заданої технічної структури визначаються множиною:

$$PST(ST) = \{A_1^*(Y_1^*), A_2^*(Y_2^*), A_3^*(Y_3^*), \lambda_1^*, \lambda_2^*, \lambda_3^*, \gamma_1^*, \Omega\}.$$

Множина **PST(ST)** визначає параметри, як сумарних, так і часток потоків даних для окремих завдань (додатків), переданих по каналах зв'язку.

Якість рішення завдань залежить від того, яка частина пропускної здатності каналу (смуга) виділена для кожного завдання, що звичайно досягається шляхом застосування режиму гарантованої якості обслуговування (QoS). Тому при рішенні завдань розрахунку потоків із застосуванням систем з гарантованою якістю обслуговування проводиться розширення множини **PST(ST)** шляхом додавання множини коефіцієнтів поділу каналів – **PKST**. Це дозволяє визначити залежність характеристик системи проектування та побудови топології мереж у центрах обробки даних від смуги пропускання в каналі (комутаторі), що відводиться для завдання.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерської роботи, наведена на рисунку 3.3.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування). Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

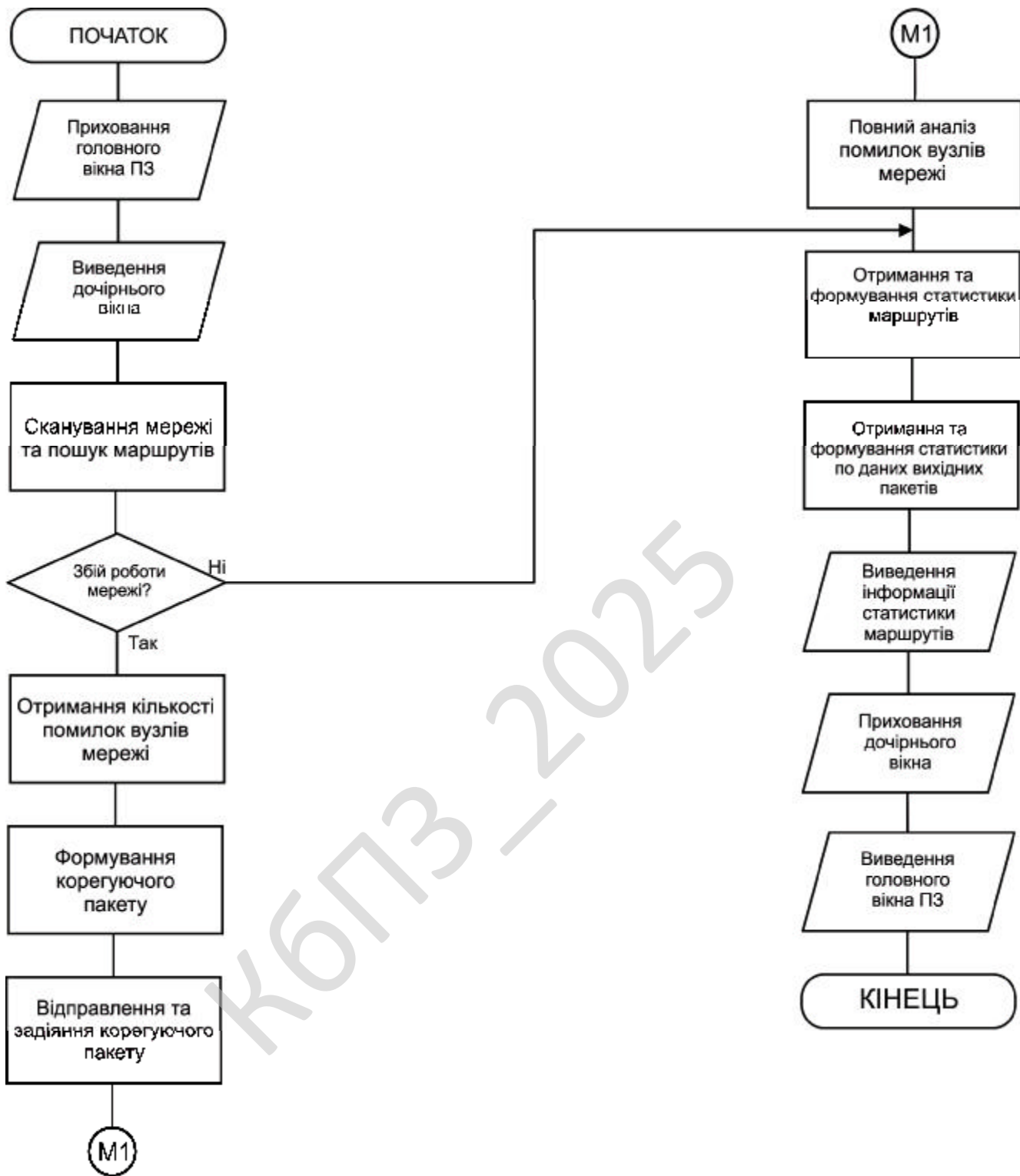


Рисунок 4.2 – Блок-схема роботи підпрограми

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку

дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Також при розробці магістерської роботи було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів; Діаграма компонент.

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації (association relationship);
- включення (include relationship);
- розширення (extend relationship);
- узагальнення (generalization relationship).

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Діаграма класів це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

В UML існують наступні типи зв'язків які використовуються у діаграмі класів: Асоціації; Агрегація; Композиція.

Асоціації це якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарної. Можна, хоча це рідко буває необхідним, створювати асоціації, що зв'язують відразу кілька класів. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами.

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає необхідність посилатися на них і відрізнити один від одного. Ім'я буде особливо корисним, якщо між одними і тими ж класами існує кілька різних асоціацій.

Клас, що бере участь в асоціації, грає в ній деяку роль. По суті, це "обличчя", яким клас, що знаходиться на одній стороні асоціації, звернений до класу з іншого її боку. Можна явно позначити роль, яку клас грає в асоціації.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

Діаграма компонент в UML це діаграма, на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись.

Модуль програмного забезпечення може бути представлено в якості компоненти. Деякі компоненти існують під час компіляції, деякі – під час компонування, а деякі під час роботи програми.

Діаграма компонент відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортання.

Компоненти об'єднуються разом використовуючи структурні зв'язки (assembly connector) щоб об'єднати інтерфейси двох компонент. Це ілюструє зв'язок типу «клієнт-сервер».

Структурна взаємодія – «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту».

При використанні діаграми компонент щоб показати внутрішню структуру компонента, клієнтські та серверні інтерфейси можуть утворювати пряме з'єднання з внутрішніми. Таке з'єднання називається з'єднанням делегації.

Призначення та загальна характеристика системи

Програмний комплекс проектування та побудови топології мереж у центрах обробки даних виконує автоматизоване формування логічної та фізичної структури мережевої інфраструктури датацентру.

Система приймає на вхід вимоги замовника, параметри майданчика, обмеження щодо кількості стійок, кількості серверів, рівнів резервування та допустимого коефіцієнта надпідписки.

На основі цих вхідних даних система формує структуровану модель топології, виконує валідацію проектних рішень та генерує вихідні артефакти у вигляді конфігураційних даних.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Система працює на мові програмування Python. Вона використовує об'єктно орієнтований підхід.

Основу становлять класи, які описують типи пристроїв, вузли мережі, з'єднання, стійки і весь центр обробки даних. Поверх моделей працює модуль алгоритмів проектування, який реалізує побудову спайн ліф топології, а також модуль валідації, який контролює дотримання заданих обмежень.

Користувач задає основні параметри через текстовий інтерфейс сценарію. Система відповідає читабельним звітом з коротким описом побудованої топології та збереженими у файли результатами у форматі JSON.

Це дозволяє легко інтегрувати результати з іншими засобами автоматизації, такими як системи керування конфігураціями.

Архітектура програмного комплексу має кілька логічних рівнів.

На рівні моделей працює модуль, який описує базові сутності. Клас DeviceType описує тип мережевого пристрою, його роль, кількість портів та пропускну здатність. Клас Node описує конкретний екземпляр пристрою у певній стійці. Клас Link описує фізичне або логічне з'єднання між двома вузлами. Клас Rack описує стійку та містить перелік пристроїв, що в ній розміщуються. Клас DataCenter описує центр обробки даних як сукупність стійок, вузлів та з'єднань.

```
from dataclasses import dataclass, field
from typing import List, Dict, Optional, Tuple
import json
import math
import random

# Клас описує тип мережевого пристрою у центрі обробки даних
@dataclass
class DeviceType:
    name: str
    role: str
    port_count: int
    port_speed_gbps: int

# Клас описує вузол мережі у топології
@dataclass
```

```

class Node:
    node_id: str
    name: str
    device_type: DeviceType
    rack: Optional[str] = None
    metadata: Dict[str, str] = field(default_factory=dict)

# Клас описує з'єднання між двома вузлами мережі
@dataclass
class Link:
    node_a: str
    port_a: int
    node_b: str
    port_b: int
    capacity_gbps: int

# Клас описує стійку у центрі обробки даних
@dataclass
class Rack:
    rack_id: str
    position_index: int
    nodes: List[str] = field(default_factory=list)

# Клас описує центр обробки даних
@dataclass
class DataCenter:
    name: str
    racks: Dict[str, Rack] = field(default_factory=dict)
    nodes: Dict[str, Node] = field(default_factory=dict)
    links: List[Link] = field(default_factory=list)

```

На рівні проектування працює модуль генерації топології. Центральний елемент цього рівня клас `SpineLeafDesigner`. Він приймає екземпляр `DataCenter` та параметри майбутньої топології. Модуль створює спайн рівень, лист рівень та комутатор рівня стійки. Для кожного рівня модуль додає вузли та відповідні зв'язки. У результаті формується повна модель топології, яка зберігається у структурі `DataCenter`.

```

# Клас реалізує побудову спайн ліф топології
class SpineLeafDesigner:
    def __init__(

```

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

```

        self,
        dc: DataCenter,
        num_spines: int,
        racks_per_pod: int,
        tor_per_rack: int,
        leafs_per_pod: int,
        server_ports_per_tor: int,
        target_oversubscription: float
    ):
        self.dc = dc
        self.num_spines = num_spines
        self.racks_per_pod = racks_per_pod
        self.tor_per_rack = tor_per_rack
        self.leafs_per_pod = leafs_per_pod
        self.server_ports_per_tor = server_ports_per_tor
        self.target_oversubscription = target_oversubscription
        self.device_library = self._build_default_device_library()

# Метод формує стандартну бібліотеку типів пристроїв
def _build_default_device_library(self) -> Dict[str, DeviceType]:
    library = {}
    library["SPINE"] = DeviceType("SPINE_100G", "SPINE", 32, 100)
    library["LEAF"] = DeviceType("LEAF_40G", "LEAF", 48, 40)
    library["TOR"] = DeviceType("TOR_10G", "TOR", 48, 10)
    library["SERVER"] = DeviceType("SERVER_10G", "SERVER", 2, 10)
    return library

```

Модуль валідації топології працює поверх сформованої моделі. Клас `TopologyValidator` обчислює коефіцієнт надпідписки, перевіряє резервування та аналізує зв'язність. Він обходить структуру центр обробки даних, агрегує пропускну здатність ліній, формує таблиці сумарних навантажень та порівнює їх з заданими проектними значеннями.

```

# Клас відповідає за перевірку коректності топології
class TopologyValidator:
    def __init__(self, dc: DataCenter):
        self.dc = dc

# Метод розраховує коефіцієнт надпідписки для стійок
def calculate_oversubscription_per_rack(self) -> Dict[str, float]:
    result: Dict[str, float] = {}
    for rack_id, rack in self.dc.racks.items():
        down = 0

```

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

```

        up = 0
        for node_id in rack.nodes:
            node = self.dc.nodes[node_id]
            if node.device_type.role == "SERVER":
                down += node.device_type.port_count * node.device_type.port_speed_gbps
            if node.device_type.role == "TOR":
                for link in self.dc.links:
                    if link.node_a == node_id or link.node_b == node_id:
                        if link.capacity_gbps > 10:
                            up += link.capacity_gbps

        if up > 0:
            ratio = down / up
        else:
            ratio = math.inf
        result[rack_id] = ratio
    return result

```

На рівні представлення результатів працює модуль експорту. Клас `TopologyExporter` перетворює модель `DataCenter` у структуру словників та списків, зручну для серіалізації у формат `JSON`. Окремий метод створює коротке текстове резюме топології для вставки у пояснювальну записку.

```

# Клас виконує експорт топології у зовнішні формати
class TopologyExporter:
    def __init__(self, dc: DataCenter):
        self.dc = dc

    # Метод повертає представлення топології у вигляді словника
    def to_dict(self) -> Dict:
        racks_data = {}
        for rack_id, rack in self.dc.racks.items():
            racks_data[rack_id] = {
                "position_index": rack.position_index,
                "nodes": rack.nodes
            }
        nodes_data = {}
        for node_id, node in self.dc.nodes.items():
            nodes_data[node_id] = {
                "name": node.name,
                "device_type": node.device_type.name,
                "role": node.device_type.role,
                "rack": node.rack,
            }

```

```

        "metadata": node.metadata
    }
links_data = []
for link in self.dc.links:
    links_data.append(
        {
            "node_a": link.node_a,
            "port_a": link.port_a,
            "node_b": link.node_b,
            "port_b": link.port_b,
            "capacity_gbps": link.capacity_gbps
        }
    )
return {
    "name": self.dc.name,
    "racks": racks_data,
    "nodes": nodes_data,
    "links": links_data
}
# Метод зберігає топологію у файл формату JSON
def save_to_json(self, filename: str) -> None:
    data = self.to_dict()
    with open(filename, "w", encoding="utf8") as f:
        json.dump(data, f, indent=4, ensure_ascii=False)
# Метод формує коротке текстове резюме топології
def summary(self) -> str:
    total_nodes = len(self.dc.nodes)
    total_links = len(self.dc.links)
    racks_count = len(self.dc.racks)
    roles_count: Dict[str, int] = {}
    for node in self.dc.nodes.values():
        role = node.device_type.role
        roles_count[role] = roles_count.get(role, 0) + 1
    lines = []
    lines.append(f"Назва датацентру {self.dc.name}")
    lines.append(f"Кількість стійок {racks_count}")
    lines.append(f"Кількість вузлів {total_nodes}")
    lines.append(f"Кількість з'єднань {total_links}")
    for role, count in roles_count.items():
        lines.append(f"Кількість вузлів ролі {role} дорівнює {count}")
    return "\n".join(lines)

```

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

Алгоритм побудови спайн ліф топології

Алгоритм побудови спайн ліф топології виконує такі дії. Система створює спайн комутатори та додає їх у окрему групу. Далі створюються стійки та комутатори рівня стійки. Для кожної групи стійок формується лист рівень, який представлений лагом лист комутаторів. Кожен лист комутатор отримує зв'язки з усіма спайн комутаторами з однаковою пропускну здатністю. Стійкові комутатори отримують зв'язки з лист комутаторами таким чином, щоб забезпечити принаймні два незалежні шляхи у верхній рівень.

Алгоритм використовує параметри кількості стійок, кількості спайн комутаторів, кількості лист комутаторів на групу стійок та кількості серверів на стійку. На основі цих параметрів розраховується очікуваний коефіцієнт надійності та виконується орієнтовна перевірка відповідності проектним вимогам.

```
# Метод будує повну спайн ліф топологію у датацентрі
def build_topology(self) -> None:
    spines = self._create_spine_layer()
    pods = self._create_racks_and_tors()
    leafs = self._create_leaf_layer(pods)
    self._connect_leafs_to_spines(leafs, spines)
    self._connect_tors_to_leafs(pods, leafs)

# Метод створює спайн вузли
def _create_spine_layer(self) -> List[str]:
    spines = []
    spine_type = self.device_library["SPINE"]
    for index in range(1, self.num_spines + 1):
        node_id = f"SPINE_{index}"
        node = Node(
            node_id=node_id,
            name=node_id,
            device_type=spine_type,
            rack=None
        )
        self.dc.nodes[node_id] = node
        spines.append(node_id)
    return spines
```

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Перевірка та розрахунки правильності обраних рішень

Після побудови топології система виконує розрахунки для оцінювання правильності обраних проектних рішень. Валідація починається з аналізу коефіцієнта надпідписки.

Для кожної стійки система обчислює сумарну пропускну здатність з боку серверів та сумарну пропускну здатність з боку аплінків з ToR комутаторів. Відношення цих величин дає коефіцієнт надпідписки. У разі, якщо значення перевищує заданий поріг, система попереджає користувача.

Далі виконується перевірка резервування. Система проходить усі ToR вузли і переконується, що кожен з них має з'єднання щонайменше з двома різними лист комутаторами. Це забезпечує стійкість до відмови одного з них.

```
# Метод перевіряє чи має кожен стійковий
# комутатор підключення до двох лист вузлів
def check_tor_redundancy(self) -> Dict[str, bool]:
    result: Dict[str, bool] = {}
    for node_id, node in self.dc.nodes.items():
        if node.device_type.role != "TOR":
            continue
        upstream_leafs = set()
        for link in self.dc.links:
            if link.node_a == node_id:
                neighbor = self.dc.nodes[link.node_b]
                if neighbor.device_type.role == "LEAF":
                    upstream_leafs.add(link.node_b)
            if link.node_b == node_id:
                neighbor = self.dc.nodes[link.node_a]
                if neighbor.device_type.role == "LEAF":
                    upstream_leafs.add(link.node_a)
        result[node_id] = len(upstream_leafs) >= 2
    return result
```

Окремо система виконує простий аналіз зв'язності. Для цього формується граф, у якому вершинами є вузли, а ребрами є з'єднання. За допомогою обходу у ширину система визначає, чи можна дістатися від довільного ToR до будь якого спайн вузла. Якщо для деяких вузлів це неможливо, система повідомляє про помилки у проекті.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

2. Якщо молодший значущий біт $x_0 = 1$, то $x_0 = x_0 \oplus C$. Якщо молодший значущий байт $x_3 = 0$, то $x_3 = x_3 \oplus C$.

3. $x_i = x_{i-1} \oplus x_i \oplus x_{i+1}$ для $i = 0..3$.

Всі операції з індексами виконуються по модулю 4. Операція множення на кроці 1 виконується по модулі $2^{32}-1$. Спеціальний випадок для даного алгоритму: якщо другий операнд дорівнює $2^{32}-1$, результат теж дорівнює $2^{32}-1$. В алгоритмі використовуються наступні константи:

$$C = 2\text{aaaaaaa}, c_0 = 025\text{f1cdb}, c_1 = 2 * c_0, c_2 = 2^3 * c_0, c_3 = 2^7 * c_0.$$

Константа C – «найпростіша» константа без кругової симетрії, високою трійковою вагою й нульовим молодшим значущим бітом. У константи c_0 є інші особливі характеристики. Константи c_1 , c_2 і c_3 – зрушені версії c_0 , і служать для запобігання атак, заснованих на симетрії.

Розшифрування виконується у зворотному порядку, Етапи 2 і 3 інверсні їм самим. На етапі 1 замість c_i використовується c_i^{-1} . Значення $c_0^{-1} = 0\text{dad4694}$.

КБПЗ-2025

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Для перегляду короткої довідки про програму слід натиснути на основному вікні кнопку авторського права, після чого на екрані з'явиться вікно показане на рисунку 5.2.

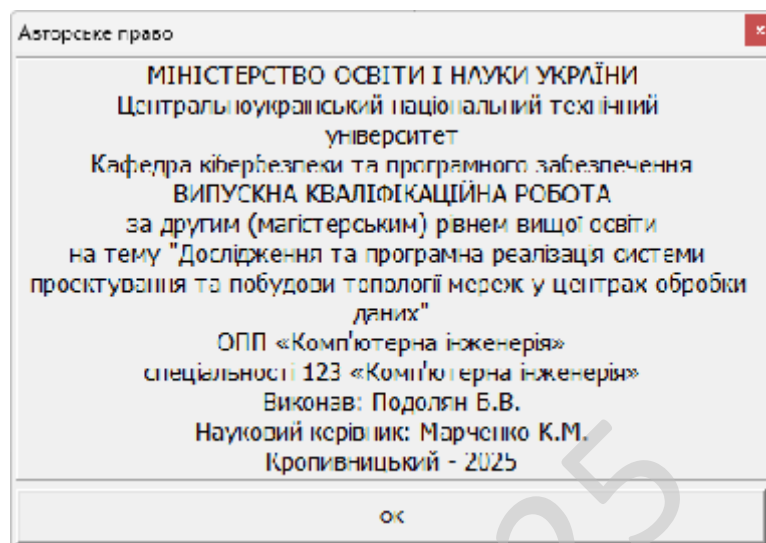


Рисунок 5.2 – Вікно розробника ПЗ

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом білої скриньки та чорної скриньки.

Тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

- При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

- Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

- Некоректних чи відсутніх функцій;
- Помилки інтерфейсу;
- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних;
- Помилки характеристик (необхідна ємність пам'яті і т.д.);
- Помилки ініціалізації та завершення.

Обрано умови розповсюдження – Freeware.

Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи проектування та побудови топології мереж у центрах обробки даних.

Метою розробки є дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних.

Об'єктом дослідження є процес проектування та побудови топології мереж у центрах обробки даних.

Предметом дослідження є методи проектування та побудови топології мереж у центрах обробки даних.

Методи дослідження базуються на методах побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод проектування та побудови топології мереж у центрах обробки даних.

– Розроблено вітчизняний продукт проектування та побудови топології мереж у центрах обробки даних, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження і програмна реалізація системи автоматизованого проектування топології мереж можуть бути цікавими перш за все великим організаціям, що мають власні центри обробки даних, або планують їх модернізацію. Це можуть бути банки, телекомунікаційні компанії, державні структури, університети, хмарні провайдери чи ІТ-компанії, які надають послуги розміщення серверів або обчислювальних ресурсів. Їхній основний інтерес полягає в тому, щоб оптимізувати мережеву інфраструктуру, скоротити витрати на її побудову і підвищити надійність.

Також така система буде корисною для проєктних і інтеграційних компаній, які займаються розробкою інфраструктурних рішень. Вона допоможе їм автоматизувати процес планування, уникати помилок у трасуванні та швидше формувати технічну документацію. Це зменшує трудомісткість і час, необхідний для реалізації проєкту, що дає можливість обслуговувати більше клієнтів при менших затратах.

Не менш важливо, що результати дослідження мають наукову цінність для академічного середовища. Студенти та дослідники можуть використовувати розроблену систему як навчальний інструмент для моделювання складних мережевих структур у віртуальному середовищі. Це дозволяє краще розуміти архітектуру дата-центрів і принципи побудови високонавантажених мережевих систем.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості системи проектування топології мереж можна залучити групу експертів із суміжних галузей – спеціалістів із мережевої інженерії, адміністраторів ЦОД, проєктантів і економістів. Кожен експерт оцінює систему за кількома ключовими параметрами: ступінь інноваційності, очікувана економія часу та ресурсів, складність впровадження, потенційний ринок збуту, рівень ризиків і масштабованість рішення.

Наприклад, після проведення опитування серед 10 експертів середня оцінка інноваційності системи склала 9,2 бала з 10, очікуваної економічної вигоди – 8,7 бала, а складності впровадження – 4,5 бала, що вважається прийнятним рівнем. Після нормалізації показників інтегральний індекс привабливості може досягати 8,4 бала, що свідчить про високий інтерес потенційних споживачів до продукту.

Такий підхід дозволяє кількісно підтвердити, що система має не лише технічну, а й комерційну перспективу. Крім того, експертні оцінки допомагають визначити напрями подальшого вдосконалення – наприклад, якщо кілька експертів відзначили потребу у кращій інтеграції з наявними CAD- або DCIM-рішеннями, це може стати пріоритетом на наступному етапі розробки.

7.3 Вибір методу оцінки вартості ПЗ

Найбільш доцільним методом оцінки вартості в цьому випадку є підхід, заснований на аналізі ТСО (Total Cost of Ownership) – повної вартості володіння системою. Цей метод дозволяє врахувати не тільки вартість створення програмного забезпечення, але й витрати на навчання персоналу, ліцензування, технічну підтримку, оновлення і майбутню масштабованість системи.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

ТСО допомагає побачити реальну фінансову картину в довгостроковій перспективі. Наприклад, початкові витрати на розробку можуть виглядати високими, однак, якщо система дає можливість скоротити кількість помилок у проектуванні, мінімізувати простої та знизити експлуатаційні витрати, загальна економія протягом кількох років може бути суттєво більшою за початкові інвестиції.

Крім того, метод ТСО можна поєднати з розрахунком ROI (Return on Investment), що дозволить оцінити, за який період інвестиції окупляться. Для системи проектування топологій мереж цей показник часто становить від 1 до 2 років, що робить її привабливою для комерційного впровадження у великих ІТ-компаніях.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Економічний ефект від впровадження системи проектування та побудови топології мереж у ЦОД (інструменти класу DCIM/Network Digital Twin/Intent-Based Design) для середнього дата-центру може бути визначена наступним чином. Вхідні дані зафіксовано в таблиці 7.1.

Вартість системи: ліцензія та інтеграція – 1 400 000 грн (одноразово), підтримка – 200 000 грн/рік.

Розрахунок економічного ефекту демонструє наступне: зменшення перепроєктування/перерозгортання (економія на «залізі») – 1 200 000грн, скорочення часу проектування – 960 000грн, менше простоїв – 300 000грн, економія на кабельній інфраструктурі – 200 000грн, енергозаощадження від консолідації – 525 600грн, автоматизація документації – 210 000грн, чистий річний ефект (до окупності сарех) – 3 195 600 грн, термін окупності – 0,44 року (~5,3 міс.), ROI (річний) \approx 228%.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Таблиця 7.1 – Вихідні дані для розрахунку

Показник	До впровадження	Після впровадження
Надлишкове розгортання портів/комутаторів	~30% від потреби	≤10%
Простої мережі на рік	10 год × 50 000 грн/год	4 год × 50 000 грн/год
Тривалість проектування мережі (1 проєкт)	20 днів	12 днів
К-сть мережевих проєктів на рік	12	12
Річний бюджет на мережеве «залізо»	6 000 000 грн	-20% завдяки точному плануванню
Річні витрати на кабель/аксесуари	800 000 грн	-25% завдяки оптимальному трасуванню
Споживана потужність мережевого шару	N кВт	-15 кВт (консолідація)
ІТ-персонал на документацію/облік	0,5 FTE економії	0,5 FTE вивільнено

Додаткові вигоди (які важко одразу монетизувати): нижчий ризик помилок у схемах та адресації завдяки цифровому «твіну» мережі, швидше введення потужностей (time-to-service): запуск нових POD/стійок з днів → у години, стандартизація і комплаєнс (аудит, трасування змін, актуальні схеми L2/L3), вища мережева надійність: автоматична перевірка політик, виявлення SPOF до релізу.

Система проектування та побудови топології мереж у ЦОД створює поєднаний фінансовий ефект: менше «заліза» та кабелю, нижчі енерговитрати,

швидше проєктування та введення сервісів, менше простоїв і рутини для інженерів. У наведеному прикладі інвестиція окупається менш ніж за пів року, а далі генерує понад 3,1 млн грн чистого ефекту щороку. Для організацій, що масштабують ЦОД або модернізують мережевий шар, це – одна з найбільш швидкоокупних ініціатив цифрової трансформації інфраструктури.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Просування такої системи доцільно починати з визначення цільових клієнтів – компаній, що вже мають розвинену мережеву інфраструктуру або планують її розбудову. Першим кроком може стати створення демонстраційної версії або тестового стенду, який покаже ефективність системи у реальних сценаріях – наприклад, при моделюванні складної багаторівневої топології мережі ЦОД.

Далі важливо забезпечити інформаційну підтримку проєкту: публікації у професійних журналах, виступи на IT-конференціях, участь у виставках, а також ведення тематичних сторінок у LinkedIn або GitHub, де розробники можуть демонструвати нові можливості продукту. Це сприятиме формуванню довіри серед IT-спільноти.

На завершальному етапі доцільно створити партнерські програми з інтеграторами обладнання та виробниками мережевих компонентів. Це дозволить просувати продукт у комплексних рішеннях, де він стане невід’ємною частиною інфраструктурного пакета. Такий підхід значно підвищує видимість проєкту на ринку і розширює його потенційну аудиторію.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації збуту слід поєднати кілька стратегій – прямі продажі великим клієнтам, дистрибуцію через партнерів та SaaS-модель. Пропозиція

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

системи як хмарного сервісу дозволить клієнтам тестувати її без складного розгортання, що значно знижує бар'єр входу. Такий підхід відкриває шлях до середнього бізнесу, який не має власних дата-центрів, але використовує віртуальні інфраструктури.

Додатково можна впровадити гнучкі умови ліцензування: наприклад, оплату за кількість вузлів у мережі або за час використання. Це дозволить клієнтам починати з невеликих інсталяцій і масштабувати систему поступово, відповідно до власних потреб.

Сильним каналом просування також є демонстраційні проєкти спільно з навчальними закладами чи технопарками. Вони створюють довіру до продукту, допомагають формувати спільноту користувачів і відкривають перспективи для участі в державних або корпоративних грантових програмах цифровізації.

7.7 Визначення ключових факторів успіху конкретного проєкту

Головним фактором успіху є реальна користь для ІТ-команд – система має не просто візуалізувати топологію, а зменшувати час проєктування, знижувати кількість помилок і підвищувати прозорість управління інфраструктурою. Якщо користувач відчує, що рішення економить його час і ресурси, воно швидко знайде своє місце на ринку. Другою умовою є простота впровадження. Навіть найпотужніший інструмент не отримає популярності, якщо для його налаштування потрібні місяці. Тому інтуїтивний інтерфейс, підтримка інтеграцій із відомими DCIM- і CAD-системами, а також наявність технічної документації стають важливими елементами успіху. Третім фактором є довіра до надійності рішення. Компанії, що працюють із критичними інфраструктурами, очікують стабільності, регулярних оновлень і якісної підтримки. Якщо ці умови дотримані, система має всі шанси стати не просто продуктом, а базовим елементом у побудові сучасних дата-центрів нового покоління.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

У сучасному житті електронно-обчислювальна машина (ЕОМ) займає важливе місце. Нині ми можемо уявити свою професійну діяльність без такого обов'язкового елемента нашого побуту. Зараз за допомогою ЕОМ керуються життєво важливі процеси. Це і управління різними електростанціями, аеропортами, постачання та логістика, документообіг, комерція та розробки, управління виробничими процесами тощо.

Але ЕОМ має і зворотний бік, а саме – небезпеку. Адже вона є складним електронним пристроєм, у якому є безліч процесів, які можуть стати джерелом небезпеки. З ЕОМ потрібно поводитися обережно та знати, як убезпечити себе від негативних наслідків його використання. Оскільки захворювання можуть бути спричинені надмірним фізичним або розумовим навантаженням, через велику нервово-емоційну напругу, або через виробниче середовище. Також програмісти у процесі роботи отримують негативний вплив на органи зору та руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. Ці шкідливі фактори можуть привести до професійних захворювань. В даному розділі магістерської роботи проведемо аналіз основних чинників при роботі програміста.

Законом України “Про охорону праці” [7] регламентуються загальні положення державної політики в галузі охорони праці, а також конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [8], яким затверджено нормативно-правовий

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

Законодавством України чітко врегульовано норми та вимоги до використання комп'ютерної техніки на підприємстві, безпосередньо й охорона праці на підприємстві при роботі за комп'ютером., зокрема «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», затверджені наказом Мінсоцполітики від 14.02.2018 № 207 [1], «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98. [2].

Загальні вимоги пожежної безпеки під час експлуатації комп'ютерної техніки визначають «Правила пожежної безпеки в Україні» (затверджені наказом МВС від 30.12.2014 № 1417) [3], комп'ютерних класів – пункт 3 розділу VIII «Правил пожежної безпеки для навчальних закладів та установ системи освіти України» (затверджені наказом МОН від 15.08.2016 № 974). [4] та інші державні стандарти, що регламентують експлуатування комп'ютерної техніки як радіоелектронної апаратури.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Можна виділити наступні основні фактори, що впливають на стан здоров'я людей, які працюють за комп'ютером:

- сидяче положення на протязі тривалого періоду;
- вплив електромагнітного випромінювання монітора;
- втома очей, навантаження на зір;
- перевантаження суглобів кистей;
- стрес при втраті інформації або виникненні критичних помилок.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

У кожному з цих випадків ступінь ризику прямо пропорційний часу, що проводиться за комп'ютером і поблизу від нього. В сучасних умовах взаємодія людини з технікою значно ускладнилась, що вимагає комплексного підходу, який передбачає розгляд людини, технічних засобів праці та виробничого середовища, як взаємозв'язаних елементів єдиної системи. Все вищесказане в повній мірі відноситься й до системи «людина–комп'ютер–середовище».

Вагомий вплив на працездатність та здоров'я користувачів комп'ютерів здійснює виробниче середовище. Це середовище у виробничих приміщеннях (офісах), в основному, визначається мікрокліматом, освітленням, наявністю шкідливих речовин у повітрі, рівнем шуму та випромінювання.

Для того, щоб об'єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам та запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини необхідно скласти санітарно-гігієнічну характеристику умов працівника, який працює з програмним продуктом.

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці користувача ПК

Розглянемо приміщення в якому працює користувач ПК з даним програмним продуктом.

Приміщення має одностороннє природне освітлення і загальне штучне освітлення. Стіни і стеля обклеєні світлими шпалерами, підлога вкрита темним ламінатом. У приміщенні відсутні сильні вібрації та шкідливі речовини. Склад повітря відповідає нормі. У кімнаті знаходиться ПК з 4-ядерним процесором і 23-дюймовим IPS-монітором, а також меблі.

Приміщення має довжину 4 м, ширину 3,5 м, висоту стелі 2,7 м. Кількість робочих місць – одне. Площа – 14 м², об'єм – 37,8 м³. Виходячи з цього, отримано дані, наведені в таблиці 8.1.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

По отриманим замірам параметрів мікроклімату можна зробити висновок, що всі показники задовольняють вимогам, зазначеним для робіт категорії легка 1а і є задовільними для здоров'я людини.

Щодо освітлення, то згідно з ДБН В.2.5-28:2006 «Природне і штучне освітлення» [6] ця робота відноситься до Va розряду зорових робіт. Передбачається використання природного, штучного і змішаного освітлення.

Природне освітлення здійснюється за допомогою вікна, площа якого складає $S' = 1,8 \times 1,5 = 2,7 \text{ м}^2$ і є бічним освітленням. У світильниках місцевого і загального освітлення використовуються світлодіодні лампи потужністю 20 Вт із світловим потоком однієї лампи 900 лм. Згідно замірів рівень освітлення в даному приміщенні і на робочому місці складає в межах 350 -500 лк, що відповідає нормованому значенню.

Джерелом шуму в приміщенні є комп'ютер. Вентилятори (кулери) системного блоку, процесора, відеокарти і блоку живлення є сучасними і мають низький рівень шуму. Згідно з технічною документацією шум, зумовлений кулером в блоці живлення складає 25 дБ, кулером процесора – 30 дБ, загальний – 34 дБ. Враховуючи незначний рівень шуму від персонального комп'ютера і незначний рівень фонового шуму від іншого устаткування, можна стверджувати, що сумарний рівень шумового забруднення приміщення не перевищує максимально допустимий рівень коригованої звукової потужності і складає не більше 50 дБА, що відповідає рівню шуму для приміщень з комп'ютерною технікою згідно Державних санітарних правил і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98. У приміщенні відсутні джерела інфрачервоного, ультрафіолетового і електромагнітного випромінювання, бо монітор ПК вироблений на основі рідкокристалічної матриці, підсвітка якої здійснюється неоновією лампою, які не мають сильного електромагнітного випромінювання і сертифіковані в Україні.

Блок живлення є екранованим і не випускає вищезазначених видів випромінювання.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

8.4 Розробка заходів з умов поліпшення охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці користувача ПК.

З точки зору забезпечення електробезпеки до цих заходів можна віднести: устаткування розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв; періодична перевірка всіх приладів і пристроїв; щорічна здача іспитів з охорони праці.

З точки зору забезпечення оптимальних умов мікроклімату, рівня звуку і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою дефлектора, для забезпечення необхідного повітрообміну в приміщенні вузла; організацію системи центрального опалювання, для підтримки оптимальної температури в холодний період року; організацію штучного загального освітлення, для забезпечення необхідних умов зорової роботи, що відповідають, оформлення паспорта на приміщення вузла, з занесенням в нього вимірювань освітленості і рівня звуку, проведених відділом охорони праці.

Крім рекомендацій щодо конкретного приміщення, де було проведено дослідження умов праці, існують загальні вимоги, які зарекомендовані відповідними нормативними документами.

Правильна організація робочих місць запобігає передчасній втомлюваності користувача і сприяє збереженню здоров'я. Організація робочого місця передбачає:

- правильне розміщення робочого місця у виробничому приміщенні;
- вибір ергономічного обґрунтованого робочого положення, виробничих меблів з урахуванням характеристик людини;
- раціональне компонування обладнання на робочих місцях;
- урахування характеру й особливостей трудової діяльності.

Стосовно робочих місць користувача ВДТ, то організація робочого місця має забезпечуватися відповідно до ДСанПіН 3.3.2-007-98. Для запобігання

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

перевтомленню необхідно виконувати вправи для очей та дотримуватись розпорядку роботи та відпочинку. На робочому місці реалізовувався режим відпочинку: кожні дві години – перерва для виконання фізичних вправ для м'язів очей.

8.5 Протипожежний захист

Пожежі в приміщеннях з оргтехнікою становлять особливу небезпеку, бо поєднані з великими матеріальними збитками. Пожежа може виникнути при взаємодії горючих речовин і джерел запалювання. Горючими речовинами є будівельні та опоряджувальні матеріали, пластмасові корпуси техніки, шнури тощо. Джерелами запалювання можуть бути електронні схеми комп'ютерів, принтерів, пристроїв електроживлення, де внаслідок різних порушень виникає перегрівання елементів, утворюються електричні іскри та дуги, здатні спричинити займання горючих матеріалів.

При обслуговуванні, ремонтних та профілактичних роботах використовуються різні легкозаймисті рідини, прокладаються тимчасові електропровідники, здійснюється паяння. Виникає додаткова пожежна небезпека, яка потребує відповідних заходів пожежного захисту.

До засобів гасіння пожежі, призначених для локалізації невеликих займань, належать вогнегасники, сухий пісок, азбестові ковдри. Приміщення, в якому встановлено комп'ютери і де немає необхідності влаштування систем автоматичного пожежогасіння, необхідно оснащувати переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 м² в приміщеннях.

Звуковбирне облицювання стін, стель приміщень треба виконувати з негорючих та важко горючих матеріалів. З метою виявлення початкової стадії займання необхідно використовувати пристрої систем автоматичного пожежогасіння там, де цього вимагають Правила пожежної безпеки.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення у випадку пожежі, прикріплену на вхідні двері.

8.6 Розрахункова частина

В приміщенні (де відсутні джерела виділення шкідливих речовин) працює двоє людей. Робота пов'язана з використанням ПЕОМ. Розміри приміщення: $A = 4$ м, $B = 7$ м, $H = 2.8$ м, устаткування займає 15% об'єму. Визначимо найменшу необхідну кількість повітря для вентиляції.

Для приміщень, в яких відсутні виділення шкідливих речовин у повітрі, розрахунок вентиляції здійснюється залежно від кількості працюючих.

Необхідна кількість повітря ($\text{м}^3/\text{год.}$), яка забезпечує відповідність параметрів повітря робочої зони нормованим значенням, визначається за наступною формулою:

$$L = L' \cdot N,$$

де

L' – нормативна кількість повітря на одного працюючого, яка залежить від питомого об'єму приміщення, $\text{м}^3/(\text{год.}-\text{люд.})$;

N – кількість працюючих.

Питомий об'єм приміщення V_n , ($\text{м}^3/\text{люд.}$), визначається за формулою:

$$V_n = V/N,$$

де V – об'єм приміщення, м^3 .

Визначаємо вільний об'єм приміщення

$$V = A \cdot B \cdot H \cdot 0,85 = 4 \cdot 7 \cdot 2,8 \cdot 0,85 = 66,4 \text{ м}^3.$$

Питомий вільний об'єм складає

$$V' = V / N = 66,4 / 2 = 33,2 \text{ м}^3/\text{люд.} > 20 \text{ м}^3/\text{люд.}$$

Нормована кількість повітря на одну людину при $V' > 20 \text{ м}^3 / \text{люд.}$ становить $30 \text{ м}^3 / (\text{год.} \cdot \text{люд.})$.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Висновки до розділу

У даному розділі магістерської роботи проведено аналіз умов працівника робота якого пов'язана з комп'ютерною технікою. Проведено аналіз основних санітарно-гігієнічних показників в заданому приміщенні, де працівник зайнятий постійною роботою за комп'ютером. Створені умови повинні забезпечувати комфортну роботу. На підставі вивченої літератури з даної проблеми, були зазначені оптимальні параметри мікроклімату, освітлення, допустимі рівні шуму та іонізуючого випромінювання при роботі з ПЕОМ, а також розраховано найменшу необхідну кількість повітря для вентиляції.

Дотримання умов, що визначають оптимальну організацію робочих місць працівників, дозволить зберегти гарну працездатність протягом усього робочого дня, підвищить як в кількісному, так і в якісному відношеннях продуктивність їх праці.

КБПЗ - 2025

					VKPM-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи проектування та побудови топології мереж у центрах обробки даних.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів проектування та побудови топології мереж у центрах обробки даних.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем проектування та побудови топології мереж у центрах обробки даних.
- Досліджена система проектування та побудови топології мереж у центрах обробки даних.
- На основі отриманих результатів досліджень створена програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання проектування та побудови топології мереж у центрах обробки даних.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ММВ.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Подолян Б.В. Дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних // Збірник праць молодих науковців ЦНТУ. – Вип. 15. – Кропивницький: ЦНТУ, 2025.
2. Оліфер В.Г. Комп'ютерні мережі. Принципи, технології, протоколи. Підручник / В.Г. Оліфер, Н.А.Оліфер. – [5-е вид.]. – 2016. – 944 с.
3. Е. Таненбаум, Д. Уезеролл «Комп'ютерні мережі». – [5-е вид.]. – 2016. – 960 с.
4. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.
5. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.
6. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
7. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
8. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
9. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
10. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка» (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.)*. Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

11. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.

12. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

13. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

14. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

15. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

16. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

17. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

18. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

19. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». CEUR Workshop Proceedings Volume 3156, 2022, Pages 390-399.

20. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

21. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

22. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

23. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

24. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

25. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

26. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

27. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

28. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

29. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

30. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

31. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

32. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties».

International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

33. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

34. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.517-522.

35. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

36. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

37. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv*, Ukraine, 2-6 July, 2019, P. 395-399.

38. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

39. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising

Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

40. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.*

41. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering.* – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

42. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи.* 2021. Т. 5, № 4. С. 79-95

43. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки.* №4. С. 103-110. 2020.

44. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка.* № 3(7). С. 43-62. 2020.

45. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

46. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія.* – Х.: ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

					ВКРМ-123.25.0055.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

47. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки.* № 2(33). с. 161-172, 2019.

48. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

49. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

50. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х.: Вид. Рожко С.Г. 2019. С. 123-139

51. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. *Центральноукраїнський науковий вісник. Технічні науки.* № 1(32). с. 173-183, 2019.