

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
“ ____ ” _____ 2024 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 та дата-
центрах”

Виконав здобувач вищої освіти
II курсу, групи КІ-23М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Бондаренко Д.С.
« ____ » _____ 2024 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Петренюк В.І.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
« 6 » вересня 2024 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Бондаренку Дмитру Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних та дата-центрах

2. Керівник роботи Петренюк Володимир Ілліч, канд. фіз.-мат. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 19-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту 2.12.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних та дата-центрах

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

- | | |
|--|--|
| <u>1. Призначення та область використання.</u> | <u>6. Наукова новизна.</u> |
| <u>2. Перегляд аналогічних існуючих систем.</u> | <u>7. Маркетингове та економічне обґрунтування ІТ-проєкту.</u> |
| <u>3. Опис і обґрунтування проектних рішень.</u> | <u>8. Заходи з охорони праці та техніки безпеки.</u> |
| <u>4. Етапи програмування системи.</u> | <u>9. Висновки.</u> |
| <u>5. Впровадження системи в промислову експлуатацію</u> | |

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

- | | |
|--|-----------------|
| <u>Наукова новизна</u> | <u>1 аркуш</u> |
| <u>Структурна схема системи</u> | <u>1 аркуш</u> |
| <u>Функціональна схема системи</u> | <u>1 аркуш</u> |
| <u>Діаграма процесів</u> | <u>1 аркуш</u> |
| <u>Блок-схема алгоритму роботи додатку</u> | <u>2 аркуша</u> |
| <u>Показники економічної ефективності</u> | <u>1 аркуш</u> |

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2024 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Бондаренко Д.С. Дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних та дата-центрах. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

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 та дата-центрах.

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

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

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

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

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

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, топологія мереж, центр обробки даних, дата-центр

ABSTRACT

Bondarenko D.S. Research and software implementation of the network topology design and construction system in data processing centers and data centers. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this graduation thesis for the second (master's) level of higher education, software was developed, which is intended for the system of designing and building network topology in data processing centers and data centers.

The goal of the development is the research and software implementation of the network topology design and construction system in data processing centers and data centers.

The object of the study is the process of designing and constructing network topology in data processing centers and data centers.

The subject of research is methods of designing and constructing network topology in data processing centers and data centers.

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

The result of the work is the software implementation of the network topology design and construction system in data processing centers and 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 software tools are provided.

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

The program was developed in the Python environment.

Keywords: computer engineering, network topology, data processing center, data center

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	14
2.3 Розгорнута постановка завдання	14
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	16
3.1 Опис функціонування системи	16
3.2 Розробка структурної схеми.....	34
3.3 Розробка функціональної схеми	40
3.4 Розробка діаграми процесів.....	49
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	51
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	51
4.2 Захист розробленого програмного забезпечення.....	63
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	66
6 НАУКОВА НОВИЗНА	71

					ВКРМ-123.24.0004.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних та дата-центрах</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Бондаренко Д.С.</i>					М	1	99
<i>Перев.</i>	<i>Петренюк В.І.</i>							
Н.контр.	<i>Коваленко А.С.</i>					ЦНТУ КІ-23М		
Затв.	<i>Смірнов О.А.</i>							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	72
7.1	Визначення цільової аудиторії кінцевого готового продукту	72
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	73
7.3	Вибір методу оцінки вартості ПЗ	74
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	75
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	77
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	77
7.7	Визначення ключових факторів успіху конкретного проєкту.....	80
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	81
8.1	Вступ.....	81
8.2	Характеристика умов праці програміста	82
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці користувача ПК.....	84
8.4	Розробка заходів з охорони праці	86
8.5	Розрахункова частина	87
9	ОСНОВНІ ВИСНОВКИ.....	91
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	93

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

КБПЗ_2024

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Еволюція центрів обробки даних стала одним із найважливіших досягнень сучасної комп'ютерної техніки. Оскільки організації продовжують оцифровувати величезні обсяги даних і покладатися на них, потреба в ефективній, масштабованій і надійній мережевій архітектурі центру обробки даних стає першорядною. Ця робота заглиблюється в тонкощі архітектури мережі центрів обробки даних, вивчаючи її компоненти, принципи проектування та нові тенденції.

Центр обробки даних є технологічним центром роботи сучасного підприємства. Центр обробки даних забезпечує критично важливу ІТ-інфраструктуру, необхідну для надання ресурсів і послуг співробітникам, партнерам і клієнтам у всьому світі.

Малий або середній бізнес часто може створити корисний «центр обробки даних» у межах шафи чи іншої зручної кімнати з невеликими змінами, якщо такі є. Однак величезний масштаб корпоративних обчислень вимагає великого виділеного простору, ретельно спроектованого для забезпечення потреб ІТ-інфраструктури в просторі, живленні, охолодженні, управлінні, надійності та безпеці.

У результаті центр обробки даних є найбільшим і найдорожчим активом, яким матиме бізнес – як з точки зору капіталовкладень, так і поточних операційних витрат. Бізнес- та ІТ-керівники повинні приділяти пильну увагу питанням, пов'язаним з проектуванням і будівництвом центру обробки даних, щоб забезпечити відповідність кінцевого об'єкта бізнес-потреbam протягом усього життєвого циклу об'єкта та мінливих умов бізнесу.

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

– Огляд існуючих систем проектування та побудови топології мереж у центрах обробки даних та дата-центрах.

– Дослідження системи проектування та побудови топології мереж у центрах обробки даних та дата-центрах.

– Програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних та дата-центрах.

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

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

Методи дослідження базуються на методах побудови топології 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 та дата-центрах.

– Розроблено вітчизняний продукт проектування та побудови топології мереж у центрах обробки даних та дата-центрах, 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 та дата-центрах.

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.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 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 та дата-центрах, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2024

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Архітектура мережі центру обробки даних складається з кількох критичних компонентів, кожен з яких виконує унікальну функцію для забезпечення безперебійної та ефективної роботи. Ці компоненти можна розділити на фізичну інфраструктуру, мережеві пристрої та мережеві топології.

Фізична інфраструктура

1. Кабелі: основою будь-якої мережі центру обробки даних є його кабельна інфраструктура. Це стосується як мідних, так і оптоволоконних кабелів, які з'єднують різні мережеві пристрої. Оптоволокну є кращим для міжміських і високошвидкісних з'єднань через їхню більшу пропускну здатність і меншу затримку.

2. Стійки та корпуси: вони використовуються для розміщення серверів, комутаторів та іншого мережевого обладнання. Правильне управління стійкою забезпечує оптимальний потік повітря та охолодження, що є вирішальним для підтримки продуктивності та довговічності мережевих пристроїв.

3. Системи охолодження: Ефективні системи охолодження є життєво важливими для запобігання перегріву, що може призвести до збоїв обладнання та втрати даних. Сучасні центри обробки даних використовують поєднання кондиціонування повітря, рідинного охолодження та інших інноваційних технологій охолодження.

4. Джерело живлення: Джерела безперебійного живлення (UPS) і резервні генератори необхідні для забезпечення роботи центру обробки даних під час відключень електроенергії. Блоки розподілу електроенергії (PDU) розподіляють електроенергію на всі мережеві пристрої в центрі обробки даних.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Мережеві пристрої

1. Комутатори: комутатори є основними будівельними блоками мережі центру обробки даних. Вони з'єднують різні пристрої в центрі обробки даних, полегшуючи передачу даних і зв'язок. Комутатори можна класифікувати на основні, розподільні та комутатори доступу, кожен з яких обслуговує різні рівні мережі.

2. Маршрутизатори: маршрутизатори з'єднують мережу центру обробки даних із зовнішніми мережами, включаючи Інтернет. Вони керують трафіком між різними мережами та забезпечують доставку пакетів даних у правильні місця призначення.

3. Брандмауери: брандмауери мають вирішальне значення для безпеки мережі. Вони відстежують і контролюють вхідний і вихідний мережевий трафік на основі заздалегідь визначених правил безпеки, захищаючи центр обробки даних від несанкціонованого доступу та кіберзагроз.

4. Балансувальники навантаження: балансувальники навантаження розподіляють мережевий трафік між декількома серверами, гарантуючи, що жоден сервер не буде перевантажений занадто великим трафіком. Це покращує загальну продуктивність і надійність центру обробки даних.

5. Пристрої зберігання: Мережі зберігання даних (SAN) і мережеві пристрої зберігання даних (NAS) використовуються для зберігання та керування великими обсягами даних. Ці пристрої забезпечують доступність даних для обробки та пошуку.

Топології мережі

Топологія мережі стосується розташування різних мережевих пристроїв і способу їх з'єднання між собою. У мережах центрів обробки даних зазвичай використовується кілька топологій, кожна з яких має свої переваги та недоліки.

1. Трирівнева архітектура: ця традиційна архітектура складається з трьох рівнів: ядра, розподілу та доступу. Базовий рівень забезпечує високошвидкісне підключення між різними частинами центру обробки даних,

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Резервування та відмовостійкість

Резервування передбачає наявність кількох шляхів для переміщення даних, гарантуючи, що збій одного компонента не призведе до простою мережі. Відмовостійкість – це здатність мережі продовжувати коректну роботу навіть у разі збоїв апаратного чи програмного забезпечення. Цього можна досягти за допомогою резервних джерел живлення, мережевих з'єднань і апаратних компонентів.

Продуктивність і низька затримка

Мережі центрів обробки даних повинні забезпечувати високу продуктивність і низьку затримку, щоб забезпечити ефективну обробку даних і обмін даними. Високошвидкісні комутатори та маршрутизатори, оптимізовані топології мережі та ефективне балансування навантаження сприяють досягненню цих цілей.

Безпека

Безпека має першорядне значення в мережах центрів обробки даних для захисту конфіденційних даних і запобігання несанкціонованому доступу. Це передбачає впровадження надійних брандмауерів, систем виявлення вторгнень, шифрування та механізмів контролю доступу. Регулярні перевірки безпеки та оновлення також необхідні для вирішення нових загроз.

Керованість

Мережею центру обробки даних має бути легко керувати та контролювати. Інструменти та програмне забезпечення для керування мережею забезпечують видимість продуктивності мережі, визначають вузькі місця та полегшують усунення несправностей. Інструменти автоматизації також можуть допомогти керувати рутинними завданнями, такими як зміни та оновлення конфігурації.

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

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

Нові тенденції в мережевій архітектурі ЦОД

Швидкий розвиток технологій продовжує формувати мережеву архітектуру центрів обробки даних. Кілька нових тенденцій стимулюють інновації та покращують ефективність сучасних центрів обробки даних.

Програмно-визначена мережа (SDN)

SDN відокремлює рівень керування від рівня даних, дозволяючи мережевим адміністраторам керувати та налаштовувати мережу за допомогою програмного забезпечення, а не апаратного забезпечення. Цей підхід забезпечує більшу гнучкість, масштабованість і автоматизацію, полегшуючи керування великими та складними мережами центрів обробки даних.

Віртуалізація мережевих функцій (NFV)

NFV передбачає віртуалізацію мережевих функцій, таких як брандмауери, балансувальники навантаження та маршрутизатори, і їх запуск на звичайному обладнанні. Це зменшує потребу у спеціальному обладнанні, знижує витрати та підвищує гнучкість мережі.

Граничні обчислення

Граничні обчислення наближають обчислення та зберігання даних до місця, де це необхідно, зменшуючи затримку та використання пропускну здатності. Ця тенденція особливо актуальна для додатків, які вимагають обробки в реальному часі, таких як пристрої Інтернету речей і автономні транспортні засоби.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Гіперконвергентна інфраструктура (НСІ)

НСІ об'єднує обчислення, сховище та мережу в єдину систему, спрощуючи керування центром обробки даних і підвищуючи ефективність. Цей підхід зменшує складність розгортання інфраструктури та керування нею, полегшуючи її масштабування та адаптацію до мінливих вимог.

Штучний інтелект і машинне навчання

Штучний інтелект і машинне навчання все частіше використовуються для оптимізації роботи центру обробки даних, починаючи від прогнозування апаратних збоїв і закінчуючи оптимізацією споживання енергії. Ці технології можуть аналізувати величезні обсяги даних, забезпечуючи розуміння та автоматизацію, що підвищує продуктивність і ефективність мережі.

Інтеграція 5G

Розгортання мереж 5G обіцяє істотне покращення швидкості, затримки та підключення. Центри обробки даних повинні будуть адаптуватися для підтримки зростаючого попиту на пропускну здатність і додатки з низькою затримкою, такі як доповнена реальність і віддалена хірургія.

Проблеми в архітектурі мережі ЦОД

Хоча досягнення в архітектурі мережі центрів обробки даних пропонують численні переваги, вони також створюють кілька проблем, які необхідно вирішити.

Складність

У міру того, як центри обробки даних зростають у розмірах і ускладнюються, управління мережею та налаштування мережі стає все більш складним. Впровадження нових технологій, таких як SDN і NFV, вимагає спеціальних навичок і знань для впровадження та обслуговування.

Загрози безпеці

Зростаюча складність кіберзагроз створює серйозну проблему для безпеки центрів обробки даних. Забезпечення надійних заходів безпеки та їх оновлення для вирішення нових загроз – це постійна боротьба.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Енергоефективність

Центри обробки даних споживають значну кількість енергії, що сприяє високим експлуатаційним витратам і негативному впливу на навколишнє середовище. Розробка енергоефективних мереж і впровадження стійких практик є важливими для вирішення цих проблем.

Вартість

Побудова та підтримка мережі центрів обробки даних передбачає значні витрати, починаючи від придбання обладнання та програмного забезпечення до найму кваліфікованого персоналу. Збалансувати потребу у високій продуктивності та надійності з бюджетними обмеженнями є постійною проблемою.

Управління даними

Експоненціальне зростання даних вимагає ефективних методів управління даними. Важливо забезпечити ефективне зберігання, обробку та доступ до даних, зберігаючи цілісність даних і відповідність нормам.

Висновок

Мережна архітектура центру обробки даних є критично важливим компонентом сучасної обчислювальної інфраструктури, що дозволяє організаціям зберігати, обробляти та ефективно керувати величезними обсягами даних. Розуміючи компоненти, принципи проектування та нові тенденції, підприємства можуть створювати надійні, масштабовані та безпечні мережі, які відповідають поточним і майбутнім вимогам.

Швидкий розвиток технологій, таких як SDN, NFV, периферійні обчислення та штучний інтелект, стимулює інновації в архітектурі мережі центрів обробки даних, пропонуючи нові можливості для підвищення продуктивності, ефективності та керованості. Однак ці досягнення також викликають проблеми, які вимагають ретельного планування та впровадження.

Підсумовуючи, добре спроектована мережева архітектура центру обробки даних має важливе значення для підтримки цифрової трансформації організацій,

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

дозволяючи їм використовувати весь потенціал своїх даних і програм. Залишаючись у курсі останніх тенденцій і найкращих практик, компанії можуть створювати стійкі та ефективні мережі, які стимулюють зростання та інновації.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це потужна мова програмування, яка проста у вивченні. Він має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованим характером роблять його ідеальною мовою для створення сценаріїв і швидкої розробки додатків у багатьох сферах на більшості платформ.

Інтерпретатор Python і обширна стандартна бібліотека доступні у вихідному або двійковому вигляді для всіх основних платформ на веб-сайті Python <https://www.python.org/> і можуть вільно поширюватися. Цей же сайт також містить дистрибутиви та вказівники на багато безкоштовних сторонніх модулів Python, програм і інструментів, а також додаткову документацію.

Інтерпретатор Python легко розширюється за допомогою нових функцій і типів даних, реалізованих у C або C++ (або інших мовах, які можна викликати з C). Python також підходить як мова розширення для настроюваних програм.

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 та дата-центрах.

В процесі розробки випускної кваліфікаційної роботи за другим

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

(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-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Розглянемо призначення та структуру ЦОД. У будь-якому центрі обробки даних є два основні аспекти: об'єкт та ІТ-інфраструктура, яка знаходиться в об'єкті. Ці аспекти співіснують і працюють разом, але їх можна обговорювати окремо.

Об'єкт

Об'єкт – це фізична будівля, яка використовується для центру обробки даних. Простіше кажучи, центр обробки даних – це просто великий відкритий простір, де буде розгорнута інфраструктура. Хоча майже будь-який простір має потенціал для роботи певної ІТ-інфраструктури, належним чином спроектований об'єкт враховує такі фактори:

– **Простір.** Повинна бути достатня площа – проста міра квадратних футів або квадратних метрів – для розміщення всієї ІТ-інфраструктури, яку бізнес має намір розгорнути зараз і в майбутньому. Приміщення має бути розташоване на добре продуманій ділянці з доступними податками та доступом. Простір часто поділяють для різних цілей або типів використання.

– **Потужність.** Повинна бути достатня потужність – у ватах, часто до 100 мегават – для роботи всієї ІТ-інфраструктури. Електропостачання має бути доступним, *чистим*, тобто без коливань і збоїв, і надійним. Необхідно включити відновлювану та додаткову/допоміжну енергію.

– **Охолодження.** Величезна кількість електроенергії, що надходить до центру обробки даних, перетворюється на обчислення, тобто роботу, і багато тепла, яке потрібно відвести від ІТ-інфраструктури за допомогою звичайних систем HVAC, а також інших нетрадиційних технологій охолодження.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

– **Безпека.** Враховуючи цінність центру обробки даних і його критичну важливість для бізнесу, центр обробки даних повинен передбачати контрольований доступ із застосуванням різноманітних тактик, починаючи від доступу співробітників до відеоспостереження.

– **Управління.** Сучасні центри обробки даних, як правило, включають систему керування будівлею (BMS), розроблену для того, щоб допомогти ІТ-керівникам і бізнесменам контролювати середовище центру обробки даних у режимі реального часу, включаючи нагляд за температурою, вологістю, рівнями живлення та охолодження, а також доступом і реєстрацією безпеки.

Великим підприємствам можуть знадобитися дуже великі центри обробки даних, як-от цей центр обробки даних Google у окрузі Дуглас, штат Джорджія.

Інфраструктура

Інфраструктура являє собою широкий спектр ІТ-обладнання, розгорнутого на об'єкті. Це обладнання, яке запускає програми та надає послуги бізнесу та його користувачам. Типова ІТ-інфраструктура включає такі компоненти:

– **Сервери.** На цих комп'ютерах розміщуються корпоративні програми та виконуються обчислювальні завдання.

– **Зберігання.** Підсистеми, такі як дискові масиви, використовуються для зберігання та захисту прикладних і бізнес-даних.

– **Мережа.** Обладнання, необхідне для створення бізнес-мережі, включає комутатори, маршрутизатори, брандмауери та інші елементи кібербезпеки.

– **Кабелі та стійки.** Милі дротів з'єднують між собою ІТ-обладнання, а фізичні серверні стійки використовуються для організації серверів та іншого обладнання в межах приміщення.

– **Резервне живлення.** Джерело безперебійного живлення (UPS), маховик та інші системи аварійного живлення мають вирішальне значення для забезпечення впорядкованої роботи інфраструктури в разі збою основного живлення.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

– **Платформи управління.** Одна або кілька платформ керування інфраструктурою центру обробки даних (DCIM) необхідні для нагляду та керування ІТ-інфраструктурою, яка звітує про працездатність, доступність, потужність і конфігурацію системи.

Коли компанія вирішує спроектувати та побудувати центр обробки даних, основна увага приділяється проекту та будівництву об'єкта. Але ІТ-лідери також повинні розглянути інфраструктуру, яка входить в об'єкт для перевірки проекту.

Як спроектувати центр обробки даних

Немає обов'язкових або обов'язкових стандартів для проектування або будівництва центрів обробки даних. Центр обробки даних призначений для задоволення унікальних потреб бізнесу в цілому, а не навпаки. Однак основною метою будь-якого стандарту є створення спільної платформи передового досвіду. Існує низка поточних стандартів центру обробки даних, і підприємство може включити один або кілька стандартів або частини стандартів у проект центру обробки даних. Стандарти допомагають забезпечити належну увагу таким факторам, серед іншого:

- Концептуальний проект.
- Планування та планування приміщення.
- Вимоги до конструкції будівлі.
- Питання фізичної безпеки.
- Внутрішні пристрої (механічні, електричні, сантехнічні та пожежні системи).
- Операції та робочі процеси.
- Технічне обслуговування.

Нижче наведено лише деякі з основних стандартів проектування центрів обробки даних та інфраструктури:

- **Стандарт рівня Uptime Institute.** Стандарт Uptime Institute Tier зосереджується на проектуванні, будівництві та введенні в експлуатацію центру

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

обробки даних і використовується для визначення стійкості об'єкта відповідно до чотирьох рівнів резервування/надійності.

– **ANSI/TIA 942-B.** Цей стандарт включає планування, проектування, будівництво та введення в експлуатацію будівель, а також протипожежний захист, ІТ та технічне обслуговування. Він також використовує чотири рівні рейтингів надійності, які впроваджують спеціалісти, сертифіковані BICSI.

– **Серія EN 50600.** Ця серія стандартів зосереджена на розробці ІТ-кабелю та мереж, а також містить різні концепції резервування інфраструктури та надійності, які приблизно базуються на Tier Standard Tier Institute Uptime Institute.

– **ANSI/BICSI 002-2019.** Документ під назвою «Найкращі практики проектування та впровадження центрів обробки даних» містить обговорення систем відведення тепла та охолодження, використання технологій літій-іонних акумуляторів, планування спільного розташування та підтримку ініціатив Open Compute Project.

– **ASHRAE .** Рекомендації ASHRAE , які не стосуються ІТ-центрів або центрів обробки даних, стосуються проектування та реалізації систем опалення, вентиляції, кондиціонування повітря, охолодження та інших суміжних сфер.

Стандарти рівня центрів обробки даних окреслюють те, що потрібно для забезпечення надійності та продуктивності.

Крім того, існує багато різноманітних нормативних та операційних стандартів, які можна застосовувати до центрів обробки даних. Нормативні стандарти включають HIPAA, GDPR, Закон Сарбейнса-Окслі, SAS 70 Тип I або II і Закон Грамма-Ліча-Блайлі. Операційні стандарти можуть включати ISO 9000 для якості, ISO 14000 для управління навколишнім середовищем, ISO 27001 для інформаційної безпеки, стандарт безпеки даних індустрії платіжних карток для безпеки платіжних карток і EN 50600-2-6 щодо управління та операційної інформації.

Хоча нормативні стандарти та практики безпосередньо не пов'язані з дизайном центру обробки даних, дотримання цих стандартів може гарантувати,

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Крім фізичного простору, проекти центрів обробки даних повинні включати ретельний розгляд розташування обладнання та макетів, тобто місця розміщення ІТ-інфраструктури в об'єкті. Найпоширенішою особливістю будь-якого макета центру обробки даних є серверна стійка, яку також називають *стійкою*. Стійка – це порожня металева рама зі стандартними відстанями та варіантами кріплення, призначена для розміщення стандартизованого ІТ-обладнання, яке встановлюється в стійку, наприклад серверів, підсистем зберігання даних, мережевого обладнання, кабелів, допоміжних систем живлення, таких як пристрої ДБЖ, і опцій введення/виведення, таких як клавіатури та монітори для адміністративного доступу.

На цьому зображенні ІТ-спеціаліст встановлює та обслуговує стійкі системи великої місткості в центрі обробки даних.

Стійки також відіграють важливу роль у схемах охолодження центрів обробки даних. Стелажі механізмів зазвичай організуються для створення гарячих і холодних проходів, які можуть підвищити ефективність охолодження, забезпечуючи введення охолодженого повітря в *холодний коридор*, яке нагрівається механізмом і подається в *гарячий коридор*, де нагріте повітря може бути ефективно видалено з кімнати. Організація проходів також може сприяти введенню додаткових дверей і заходів безпеки в кінцях кожного проходу, щоб обмежити доступ людей.

Безпека центру обробки даних

Безпека центру обробки даних зазвичай включає три різні аспекти безпеки доступу, безпеки об'єктів і кібербезпеки.

Безпека доступу

Будь-яке обговорення засобів центру обробки даних повинно включати розгляд фізичної безпеки. Фізична безпека – це управління людським персоналом і захист фізичного об'єкта та його ІТ-інфраструктури. При належному застосуванні безпека гарантує, що лише уповноважений персонал має доступ до

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

об'єкта та спорядження, і що всі людські дії документуються. Безпека може включати наступний комплекс заходів:

- Доступ до закладу та навколо нього (включаючи зони обладнання).
- Ключовий доступ до певних стійок і серверів.
- Журнали для доступу співробітників і відвідувачів/постачальників.
- Супровід непрацюючих.
- Відеоспостереження.
- Персонал охорони на місці.

Охорона об'єкта

Фізична безпека також поширюється на цілісність середовища центру обробки даних, включаючи температуру, вологість і умови диму/пожежі/повені. Цим аспектом захисту центру обробки даних часто керує BMS, яка відстежує умови навколишнього середовища чи надзвичайні ситуації та повідомляє керівникам будівель.

Кібербезпека

Кібербезпека зосереджується на контролі доступу до корпоративних даних і програм, розміщених в IT-інфраструктурі центру обробки даних. Кібербезпека призначена для того, щоб лише належним чином автентифіковані користувачі могли отримати доступ до даних або використовувати програми, а також щоб про будь-які порушення повідомлялося та негайно усувалося. Наприклад, фізична безпека не дозволяє людині торкатися диска в центрі обробки даних, тоді як кібербезпека запобігає цій самій людині отримати доступ до даних на диску на відстані сотень миль через мережу. Кібербезпека використовує поєднання засобів захисту від зловмисного програмного забезпечення, керування конфігурацією, виявлення/запобігання вторгненням, реєстрації активності та інших інструментів, щоб контролювати мережеву активність і виявляти потенційні загрози.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Вимоги до потужності та продуктивності центру обробки даних

Потужність – це постійний виклик для будь-якого центру обробки даних корпоративного класу. Велике об'єкт може споживати близько 100 мегават – цього достатньо для забезпечення електроенергією близько 80 000 будинків. Power представляє найбільший єдиний Орех для центру обробки даних корпоративного класу; тому оператори центрів обробки даних висувають наступні вимоги до електроенергії:

- **Ємність.** Для роботи центру обробки даних має бути достатньо електроенергії .
- **Вартість.** Енергія повинна бути якомога дешевшою.
- **якість.** Потужність має бути електрично чистою (тобто без небажаних електричних перешкод, стрибків напруги та стрибків).
- **Надійність.** Електропостачання має бути без перебоїв, відключень або інших збоїв.

Ці проблеми дедалі частіше вирішуються за допомогою місцевих і все більш відновлюваних джерел енергії, включаючи вітрову, сонячну та локальну генерацію.

Але для того, щоб бізнес розумів проблеми з електроживленням для будь-якого центру обробки даних, важливо, щоб дизайнери центрів обробки даних і ІТ-лідери розраховували потреби в електроенергії об'єкта та його ІТ-інфраструктури. Саме цей контрольний показник дає змогу бізнесу зрозуміти приблизну вартість електроенергії та обговорити потужності з регіональними комунальними службами.

Немає єдиного способу оцінки потреби в потужності. Для об'єкта потужність є простою оцінкою потреб у освітленні та опаленні, вентиляції та кондиціонуванні. Вимоги до потужності ІТ-інфраструктури можуть бути складнішими, оскільки вимоги до потужності сервера залежать від *робочого навантаження* , тобто того, скільки роботи виконують програми, і конфігурації

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

кожного сервера, включаючи вибір ЦП, встановленої пам'яті та інших пристроїв розширення, таких як графічні процесори.

Традиційні оцінки потужності включають підходи на основі стійки та таблиць.

Підхід на основі стійки зазвичай призначає стандартизовану оцінку потужності на стійку. Наприклад, ІТ-керівник може призначити приблизно від 7 кВт до 10 кВт на стійку. Якщо центр обробки даних планує розгорнути 50 стійок, оцінка потужності є простим кратним. Подібним підходом є загальна оцінка центру обробки даних у ватах на квадратний фут. Однак, оскільки цей підхід приділяє мало уваги обладнанню, встановленому в кожній стійці, він часто є найбільш неточним засобом оцінки потужності.

Підхід, заснований на паспортній таблиці, дозволяє ІТ-лідерам скласти вимоги до потужності, зазначені на паспортній таблиці кожного сервера чи іншого ІТ-пристрою. Це більш детальний підхід, який зазвичай дає кращі оцінки. Тим не менш, потреба в потужності, зазначена на таблиці кожного пристрою, може бути завідомо неточною та не враховувати роботу, яку пристрій виконує.

Більш сучасний підхід полягає у використанні фактичних вимірювань потужності на сервер, отриманих за допомогою ІТ-пристроїв керування живленням, таких як інтелектуальні блоки розподілу електроенергії, розташовані в кожній стійці. Фактичні вимірювання можуть дати найточніші оцінки та дати операторам центрів обробки даних краще зрозуміти, як потреби в електроенергії та витрати можуть коливатися залежно від вимог до робочого навантаження.

Нарешті, енергопостачання неминуче зазнає випадкових збоїв у виробництві та розподілі, тому центри обробки даних повинні включати один або кілька варіантів резервного або резервного живлення. Може бути встановлено кілька рівнів вторинної влади, залежно від того, від яких проблем бізнес має намір захищатися.

На рівні об'єкта центр обробки даних може включати резервні генератори, що працюють на дизельному паливі або природному газі, здатні забезпечувати

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

перед кожним рядом стелажів, тоді як нагріте повітря збирається та виводиться із загального гарячого проходу. Для запобігання змішуванню нагрітого повітря з охолодженим додаються додаткові фізичні бар'єри. Такі схеми стримування пропонують дуже ефективно використання потужності ОВК.

У центрах обробки даних, які не використовують конструкцію гарячого/холодного проходу, блоки охолодження не завжди здатні ефективно охолоджувати обладнання.

У центрах обробки даних, сконструйованих навколо гарячих/холодних проходів, блоки охолодження можуть більш ефективно охолоджувати обладнання.

Інші підходи до охолодження включають системи кондиціонування повітря *в кінці ряду* та *у верхній частині стелажа*, які вводять охоложене повітря в частини ряду стелажів і випускають нагріте повітря в гарячі проходи.

Деякі центри обробки даних навіть застосовують нові технології рідинного охолодження, які занурюють ІТ-обладнання в ванну з охолодженими електрично нейтральними рідинами, такими як мінеральні масла. Охолоджувач рідини невеликий і енергоефективний, і рідини можуть запропонувати в рази більшу ефективність теплопередачі, ніж повітряне охолодження. Однак рідинне охолодження стикається з іншими проблемами, включаючи витіки/затоплення, корозію частин або сприйнятливість до проникнення рідини, фільтрацію рідини та чистоту та безпеку людини.

На цій діаграмі показані основні концепції кімнатних, рядних і стійкових архітектур охолодження. Сині стрілки вказують на відношення шляхів первинного охолодження до приміщення.

Ефективність і стійкість ЦОД

Сучасні занепокоєння щодо впливу на навколишнє середовище викидів вуглекислого газу від виробництва електроенергії спонукали багато організацій зробити новий акцент на ефективності та стійкості центру обробки даних.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

енергії, що забезпечує нульовий чистий викид. Щоб досягти чистого нуля, підприємства повинні прийняти енергозбереження, енергоефективність – наприклад, ініціативи PUE – і поновлювані екологічно чисті джерела енергії.

Зменште споживання енергії для зберігання даних, щоб працювати в екологічно чистішому центрі обробки даних.

Інші проблеми сталого розвитку включають практику утилізації та переробки. ІТ-обладнання створює значні відходи у вигляді пакувальних матеріалів. Нове спорядження все частіше виготовляється з використанням біорозкладаних пакувальних матеріалів із мінімальним вмістом токсинів, які легко руйнуються у потоці відходів.

Практика утилізації застарілого та знятого з експлуатації обладнання може додати до потоку відходів метали та неорганічні – потенційно токсичні – компоненти. Багато організацій перепрофілюють старе обладнання, щоб продовжити термін служби переміщеного обладнання в інших несуттєвих ролях, або навіть дарують використане обладнання іншим підприємствам.

Найкращі практики проектування ЦОД

Не існує єдиного способу проектування центру обробки даних, і існує незліченна кількість конструкцій, які задовольняють унікальні потреби кожного бізнесу. Але такі стратегії можуть створити центр обробки даних із надзвичайною ефективністю та стійкістю:

– **Виміряйте енергоефективність.** Оператори ЦОД не можуть керувати тим, що вони не вимірюють. Використовуйте такі показники, як PUE, щоб контролювати ефективність центру обробки даних. PUE має бути безперервним вимірюванням, яке проводиться через часті проміжки часу протягом усього року, оскільки сезони та погода можуть впливати на споживання електроенергії.

– **Розгляньте креативний дизайн простору.** Замість традиційного великого відкритого простору для ІТ-обладнання подумайте про більш відокремлений або розділений простір центру обробки даних. Це може включати зони, які можна залишити порожніми, якщо вони непотрібні, використовувати

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

для розміщення різних типів обладнання з подібними вимогами до живлення/охолодження або інших підходів для зменшення вимог до живлення та охолодження.

– **Перегляньте потік повітря.** Охолодження має важливе значення для безпечної роботи IT-інфраструктури, але необхідно керувати повітряним потоком і оптимізувати його. Це може включати обмеження змішування гарячого/холодного повітря, використання схем утримання гарячого/холодного коридору та навіть використання заглушок для покриття невикористаних отворів стійки, що запобігає надходженню охолодженого повітря до місць, де не охолоджується жодне обладнання.

– **Підвищити температуру.** Чим холодніше в серверній кімнаті, тим вона енергоємніша та дорожча. Замість того, щоб підтримувати серверну кімнату холодніше, оцініть ефект фактичного підвищення температури. Наприклад, замість того, щоб запускати холодний коридор при температурі від 68 до 72 градусів за Фаренгейтом, розгляньте можливість запуску холодного проходу при температурі від 78 до 80 градусів за Фаренгейтом. Більшість IT-обладнання може витримувати високу температуру таким чином, якщо обладнання працює при постійній температурі.

– **Спробуйте альтернативне охолодження.** Система HVAC може бути стандартною для центрів обробки даних, але розгляньте способи зменшити або усунути залежність від традиційного HVAC. Наприклад, центри обробки даних у прохолодніших кліматичних умовах можуть зменшити використання систем опалення, вентиляції, вентиляції та кондиціонування повітря та запровадити в приміщення холодніше зовнішнє повітря, що називається *природним охолодженням*. Подібним чином HVAC можна доповнити або замінити чиллерами з водяним охолодженням, тобто *економайзерами*, або іншими технологіями теплообміну, які споживають набагато менше енергії.

– **Покращити розподіл енергії.** Енергоефективність центрів обробки даних часто втрачається через неефективність пристроїв керування та розподілу

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

проблемою розуміння найпоширеніших загроз і розробки належної стійкості до цих загроз.

– **Зміна.** Нові обчислювальні технології та нові вимоги постійно розробляються та впроваджуються. Розробники центрів обробки даних повинні розглянути, як адаптувати та впроваджувати часто непередбачувані зміни без необхідності фундаментального перепроєктування ІТ-інфраструктури для кожної зміни.

Програмне забезпечення та інструменти для керування інфраструктурою центру обробки даних

Центри обробки даних – це складні організми, які вимагають постійного моніторингу та управління як на рівні об’єкта, так і на рівні ІТ-інфраструктури. Оператори центрів обробки даних зазвичай використовують інструменти DCIM, щоб забезпечити перспективу роботи як об’єкта, так і інфраструктури. Набір загальних завдань керування, необхідних для роботи центру обробки даних, включає наступне:

– **Спостереження та нагляд.** Завдання спостереження включають моніторинг потужності, температури та вологості в об’єкті. Спостереження в інфраструктурі може включати доступну ємність, тобто які системи використовуються, а які вільні; працездатність програми для моніторингу належної роботи ключових робочих навантажень підприємства; і загальний час безвідмовної роботи або доступність. Завдання спостереження зазвичай пов’язані з системами сповіщень і квитків для встановлення пріоритетів і усунення проблем у міру їх виявлення.

– **Безпека.** Значні інструменти та інфраструктури розгортаються для безпеки центру обробки даних. Фізична безпека передбачає електронний доступ і все більш «розумне» відеоспостереження, таке як розпізнавання обличчя за допомогою ШІ. ШІ може допомогти визначити, чи повинен хтось бути в кімнаті з обладнанням і хто ця особа. Багато додаткових інструментів допомагають ІТ-адміністраторам і фахівцям із безпеки захищатися від мережеских атак, вторгнень

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

у систему, зловмисного програмного забезпечення та неправомірних дій співробітників.

– **Підготовка та санація.** Управління центром обробки даних також включає завдання підготовки, такі як процеси аварійного відновлення та резервного копіювання, а також можливості міграції робочого навантаження для забезпечення своєчасного виконання завдань системного обслуговування. Завдання з відновлення включають регулярне обслуговування, а також періодичне оновлення системи, усунення несправностей системи та ремонт.

– **Місткість і спроможність.** Інструменти керування даними можуть контролювати поточну ємність, тобто використані та вільні ресурси, і допомагають операторам центрів обробки даних відстежувати використання, щоб планувати збільшення ємності. Вони також допомагають підтримувати регулярні вдосконалення можливостей центру обробки даних, такі як оновлення системи, оновлення технологій і впровадження нових технологій центру обробки даних.

Управління є ключовим елементом гарантії бізнес-послуг і угод про рівень обслуговування (SLA). Багато центрів обробки даних пов'язані певною формою SLA – або з внутрішніми департаментами чи відділами, або із зовнішніми діловими партнерами та клієнтами. Моніторинг і керування за допомогою DCIM та інших інструментів є важливими для гарантування дотримання SLA або виявлення порушень SLA, які можна негайно виділити та виправити. Крім того, комплексний моніторинг і управління допомагають забезпечити безперервність роботи та аварійне відновлення, що може бути життєво важливим для сучасних нормативних зобов'язань.

Центри обробки даних і хмарні обчислення

Технології хмарних обчислень змінили обличчя сучасних корпоративних обчислень. Часи, коли бізнесу необхідно було побудувати та запуснути власний повноцінний центр обробки даних як важливу вартість ведення бізнесу, минули, і їх замінили сторонні послуги, які надають глобальні постачальники IaaS, PaaS і SaaS. Сьогодні робочі навантаження та служби можна запускати в

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

загальнодоступній хмарі так само легко і часто за таку саму або меншу вартість, ніж виконання цих робочих навантажень у традиційному центрі обробки даних.

Компанія може вибрати приватну хмару в своєму центрі обробки даних. Це може допомогти привнести в ІТ-операції такі переваги, як хмара, наприклад гнучкість і самообслуговування. Вибір впровадження приватної хмари не змінює значним чином дизайн або робочі аспекти центру обробки даних. На практиці приватна хмара використовуватиме те саме апаратне забезпечення та інфраструктуру, що й інші робочі навантаження. Приватна хмара вмикається шляхом додавання рівня спеціалізованого програмного забезпечення або структури, наприклад OpenStack.

Вибір використання хмари може сильно вплинути на загальний розмір, обсяг і витрати, пов'язані з центром обробки даних. Деякі з найпрогресивніших «хмарних» компаній, що працюють сьогодні, можуть відмовитися від виділеного центру обробки даних на користь проектування та розгортання всіх робочих навантажень у загальнодоступній хмарі.

Однак більшість підприємств вирішують підтримувати певну кількість робочих навантажень у локальному центрі обробки даних, де бізнес безпосередньо контролює робоче навантаження та свою інфраструктуру. У цих випадках використання хмари може дозволити компанії виконувати менш критичні, експериментальні або тимчасові робочі навантаження в хмарі. Це потенційно може зменшити обсяг простору, обладнання, масштабованість і витрати, необхідні для традиційного локального центру обробки даних.

3.2 Розробка структурної схеми

Розглянемо механізми роботи TRILL і SPB. У міру того, як мережі Ethernet стають все більшими, алгоритм spanning-tree стає дедалі більшою проблемою, насамперед через те, що корисні зв'язки відключаються, а резервування втрачається. У високопродуктивній мережі, наприклад у центрі

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

обробки даних, відключені зв'язки є витраченим ресурсом. Другорядною проблемою є те, що у випадку збою зв'язку створення нового дерева та відновлення підключення за допомогою сполучного дерева може зайняти багато секунд.

Щоб вирішити ці проблеми, зараз існують протоколи, які *дозволяють* Ethernet мати активні петлі в топології, забезпечуючи першокласне використання *всіх* каналів. Ідея полягає в тому, щоб створити таблиці пересилання в комутаторах Ethernet – або принаймні в деяких з них – які направляють кожен пакет за найкоротшим шляхом – або принаймні наближеним до найкоротшого шляху – на основі всіх доступних посилань. Це вже давно є основним продуктом у світі IP (9 алгоритмів маршрутизації та оновлення), але, безперечно, є розривом із традицією на рівні локальної мережі.

Тут є два конкуруючих протоколи: TRILL (Transparent Interconnection of Lots of Links) і SPB (Shortest-Path Bridging). TRILL задокументовано в [RP04] і RFC 6325 [<https://tools.ietf.org/html/rfc6325.html>] і супутніх документах, тоді як SPB стандартизовано IEEE 802.1aq [[en.Wikipedia.org/wiki/IEEE_802.1aq](https://en.wikipedia.org/wiki/IEEE_802.1aq)]. Тут ми зосередимося на TRILL.

І TRILL, і SPB передбачають, що спочатку лише кілька комутаторів будуть достатньо розумними, щоб виконувати маршрутизацію за найкоротшим шляхом, так само, як колись лише кілька комутаторів реалізували алгоритм spanning-tree. Але з часом цілком імовірно, що більшість, якщо не всі комутатори Ethernet, будуть знати про найкоротший шлях. У високопродуктивних центрах обробки даних особливо ймовірно, що переадресація буде заснована на TRILL або SPB.

У TRILL комутатори Ethernet, які підтримують TRILL, відомі як Router-Bridges або RBridges (терміни RSwitches і TRILL Switches також можуть бути доречними). Між RBridges знаходяться *Legacy Ethernet* (звані «лінками» в [RP04] і RFC 6325 [<https://tools.ietf.org/html/rfc6325.html>], хоча цей термін вводить в оману); Застарілі мережі Ethernet складаються з максимальної кількості підмереж

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

хостів Ethernet і комутаторів, які не підтримують TRILL. Намір RBridges полягає в тому, щоб розділити весь Ethernet на відносно невеликі Legacy Ethernet. У кінцевому випадку, коли *всі* комутатори є RBridges, Legacy Ethernet є просто окремими хостами. На наведеній нижче діаграмі чотири RBridges ізолюють Legacy Ethernet 1, 2, 3 і 4, хоча Legacy Ethernet 5 представляє певний ступінь неефективності розділення.

Кожна успадкована мережа Ethernet обирає *один* підключений RBridge, який представлятиме її. Вище є унікальний вибір для LE1–LE4, але LE5 має прийняти рішення. Цей обраний RBridge відомий як *призначений RBridge* або DRB. Кожна успадкована мережа Ethernet буде власне охоплююче дерево, можливо (хоча і не обов'язково), яке базується на визначеному мосту RBridge.

Трафік від Legacy Ethernet назовні, як правило, пересилатиметься через призначений RBridge; підключення до інших RBridges не використовуватимуться. Ідея полягає в тому, щоб пакети від однієї застарілої мережі Ethernet до іншої доставлялися спочатку до DRB вузла джерела, а потім до DRB вузла призначення за допомогою справжньої переадресації за найкоротшим шляхом між RBridges, а звідти до вузла призначення. Звичайно, в кінцевому випадку, коли кожен комутатор є RBridge, трафік пройде найкоротший шлях від початку до кінця.

Єдиним винятком із цього правила щодо пересилання через призначений RBridge є те, що DRB може делегувати це завдання пересилання іншим RBridges для різних VLAN у межах Legacy Ethernet. Якщо це зробити, кожна VLAN завжди використовуватиме той самий RBridge для всього зовнішнього трафіку.

Друга частина процесу полягає в тому, щоб кожен з RBridges з'ясував загальну топологію; тобто кожен створює повну карту всіх RBridges та їхніх взаємозв'язків. Це робиться за допомогою *протоколу оновлення маршрутизації стану зв'язку*, описаного в розділі 9.5 Алгоритм оновлення маршрутизації стану зв'язку. З двох основних протоколів стану зв'язку, IS-IS та OSPF, TRILL вибрав перший, оскільки його легше адаптувати до налаштувань, у яких, як тут, вузли не

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

обов'язково мають IP-адреси. Кожен RBridges надсилає відповідні «пакети стану зв'язку», використовуючи багатоадресну розсилку та використовуючи бази даних для кожного RBridge, щоб гарантувати, що ці пакети не перенаправляються нескінченно. Ці пакети стану зв'язку можна порівняти з повідомленнями Hello spanning-tree. Щойно кожен RBridge має повну карту всіх RBridges, кожен RBridge може розрахувати оптимальний маршрут до будь-якого іншого RBridge.

Оскільки призначені RBridges бачать пакети зі своїх застарілих мереж Ethernet, вони вивчають MAC-адреси активних хостів усередині за допомогою звичайного протоколу навчання Ethernet. Потім вони діляться цими адресами з іншими RBridges, використовуючи протокол стану зв'язку IS-IS, тому інші RBridges зрештою дізнаються, як досягти більшості, якщо не всіх адрес Ethernet, присутніх у загальній мережі.

Однак для доставки до раніше невідомих пунктів призначення все ще потрібно використовувати резервне заповнення. З цією метою RBridges узгоджують між собою охоплююче дерево, що охоплює всі RBridges. Будь-який пакет із невідомим адресатом передається вздовж цього охоплюючого дерева, а потім, коли пакет досягає призначеного мосту RBridge для застарілої мережі Ethernet, розповсюджується вздовж охоплюючого дерева цього Ethernet. Цей процес також використовується для доставки широкомовних і багатоадресних пакетів.

Коли RBridges спілкуються один з одним, вони узгоджують компактні двобайтові адреси – відомі як «ніки» – один для одного, проти стандартних шестибайтових адрес Ethernet. Це економить місце в комунікаціях RBridge-to-RBridge.

Під час проходження пакетів між RBridges додається спеціальний заголовок TRILL. Цей заголовок містить поле підрахунку переходів, інакше воно не присутнє в Ethernet, що означає, що будь-які пакети, спіймані в тимчасових циклах маршрутизації, зрештою будуть відхилені. IS-IS іноді може генерувати такі петлі маршрутизації, хоча це рідко.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Заголовок TRILL також містить псевдоніми джерела та призначення RBridges. Це означає, що фактичне пересилання пакетів між RBridges не включає MAC-адресу хоста призначення; який використовується лише після того, як пакет досягне призначеного RBridge для призначення Legacy Ethernet, після чого заголовок TRILL видаляється.

Якщо зв'язок між двома RBridges дає збій, то кінцеві точки зв'язку надсилають повідомлення про оновлення IS-IS, щоб сповістити всі інші RBridges про збій. Потім інші RBridges можуть перерахувати свої таблиці пересилання, щоб не використовувати пошкоджене посилання. Час відновлення зазвичай становить менше 0,1 секунди, що приблизно в сто разів більше, ніж час відновлення spanning-tree.

TRILL підтримує використання кількох однакових шляхів для покращення пропускної здатності між двома RBridges; див. 9.7 ECMP . У високопродуктивному центрі обробки даних ця функція дуже важлива.

Як і TRILL, SPB використовує IS-IS між мостами, що підтримують SPB, щоб знайти найкоротші шляхи, і інкапсулює пакети зі спеціальним заголовком, коли вони переміщуються між RBridges. SPB не включає кількість переходів у заголовок інкапсуляції; замість цього він ретельніше контролює пересилання. SPB також використовує оригінальну MAC-адресу призначення для переадресації між RBridge.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

– В TRILL свій формат кадру, а SPB використовує стандартний кадр MAC-in-MAC (стандарт IEEE 802.1ah).

– У контрольній площині обидва протоколи використовують IS-IS, однак TRILL поки не підтримує чітку прив'язку VLAN до сервісу, що утрудняє «розтягування» сервісу між пристроями.

– Обидва протоколи мають досить швидку збіжність – близько 100 мс.

– В SPB шляхи симетричні, в TRILL таке буває не завжди.

– Для багатоадресної передачі в TRILL використовуються уніфіковані дерева (не більше п'яти-шести на домен), а в SPB дерево розраховується для кожного вхідного вузла (ingress node), що ефективніше в плані розподілу трафіку, але більш затратне з погляду обчислень.

– На відміну від SPB, поточний варіант OAM в TRILL не підтримується.

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

досліджувати потоки даних, оскільки потоки даних формуються розв'язуваними на системи проектування та побудови топології мереж у центрах обробки даних 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 використовуюваного 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.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

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

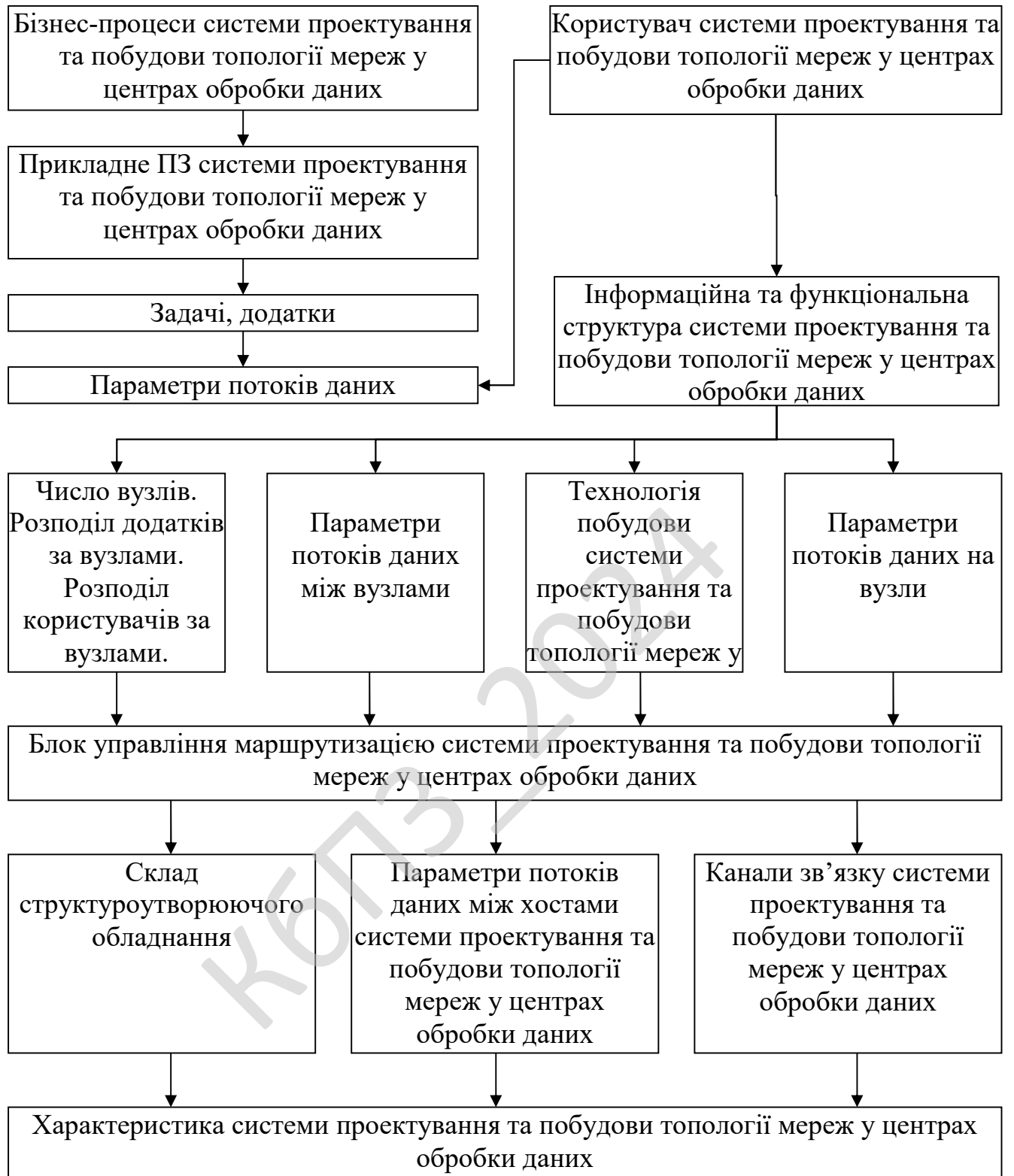


Рисунок 3.2 – Функціональна схема системи

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.

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

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

де:

- N – число працюючих користувачів;
- M – число задіяних вузлів;
- D – число використовуваних додатків;
- L – число розв'язуваних завдань;
- R – число баз даних;
- $\mathbf{S}_k = \{\mathbf{p}_k, \mathbf{d}_k, \mathbf{u}_k, \mathbf{W}_k\}$, ($k=1, 2, \dots, L$) – набір даних, що описують

розв'язувані завдання, де:

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

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

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

- $\mathbf{A}_{km} = \{\mathbf{v}_{km}, \mathbf{b}_{km}\}$, ($k=1, 2, \dots, L; m=1, 2, \dots, D$) – набір даних, що описують додатки, використовувані завданнями, де:

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

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

- \mathbf{G} матриця розміщення додатків по вузлах;

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

- **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.

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: $\mathbf{A}_k = \lambda_k \mathbf{Z}_k, (k=1,2,\dots,L)$, де \mathbf{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 – \mathbf{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: $\mathbf{PSI}(\mathbf{SI}) = \{\Lambda, \mathbf{Z}_k, \mathbf{A}_k (k=1,2,\dots,L), \mathbf{A}, \mathbf{B}^*, \Phi\}$.

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 $\mathbf{A}_1(\mathbf{C}_1) = \mathbf{C}_1 \mathbf{A}(\mathbf{C}_1)^T, \mathbf{A}_2(\mathbf{C}_2) = \mathbf{C}_2 [\mathbf{A}_1(\mathbf{C}_1) - dg(\mathbf{A}_1(\mathbf{C}_1))](\mathbf{C}_2)^T$;
- матриці інтенсивностей потоків даних утворених кожним завданням

$$A_{1k}(C_1) \text{ і } A_{2k}(C_1), (A = \sum_{k=1}^L A_k).$$

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

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

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 дипломної 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.

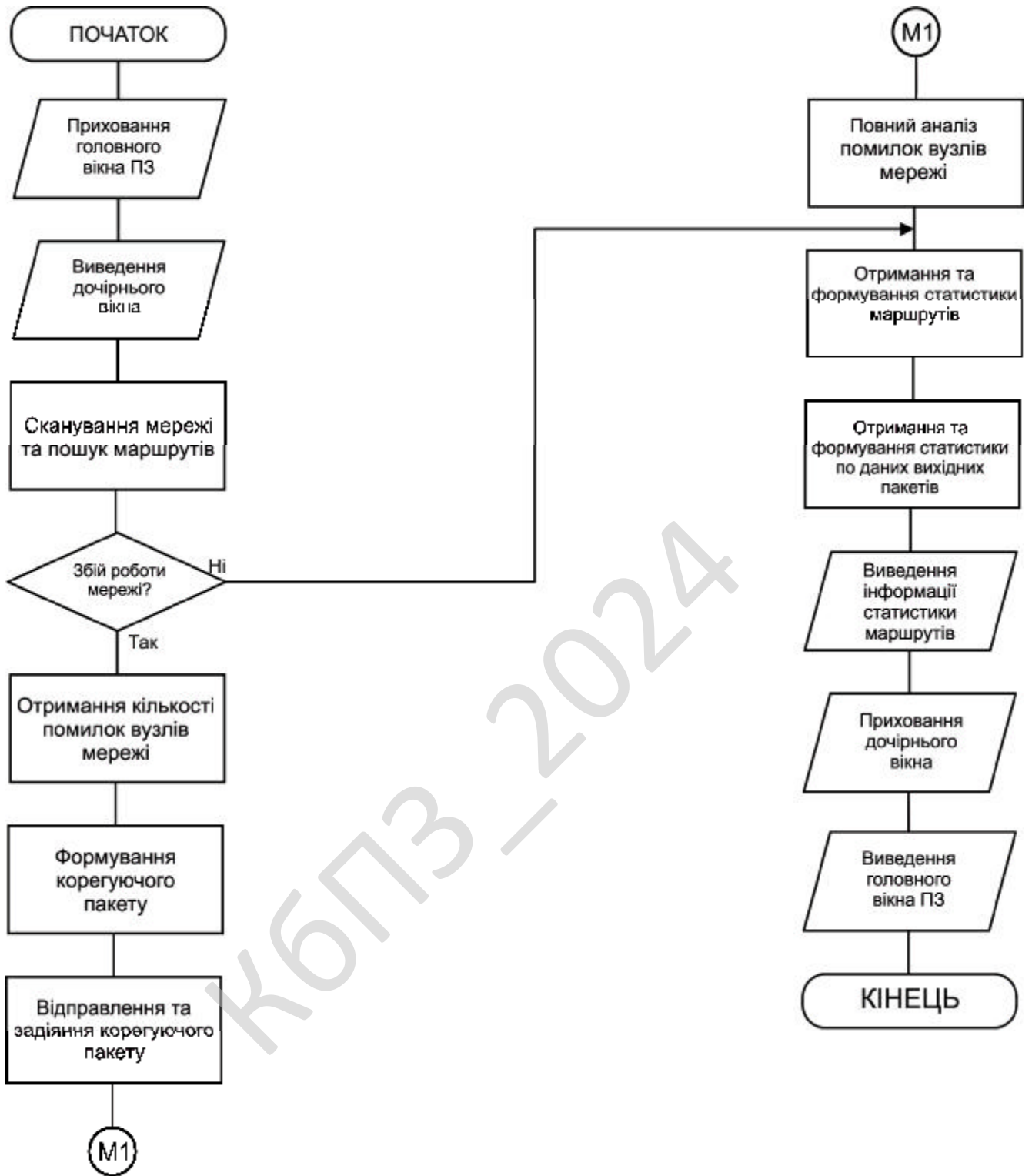


Рисунок 4.2 – Блок-схема роботи підпрограми

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

основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи проектування та побудови топології мереж у центрах обробки даних.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

При розробці використовувались концепції діаграм діяльності. Тобто в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Це фундаментальна одиниця 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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

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

Також при розробці магістерської дипломної 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.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

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.


```

def allocate_ip(self):
    """Виділення нової IP-адреси з підмережі"""
    if not self.available_ips:
        raise ValueError("No more available IP addresses in the
subnet!")

    return self.available_ips.pop(0)

def available_ips_count(self):
    """Показує кількість доступних IP-адрес"""
    return len(self.available_ips)

# Клас для створення топології мережі
class NetworkTopology:
    def __init__(self, name):
        self.name = name
        self.devices = []
        self.connections = []
        self.graph = nx.Graph()

    def add_device(self, device: NetworkDevice):
        """Додати пристрій до топології"""
        self.devices.append(device)
        self.graph.add_node(device.name)

    def connect_devices(self, device1: NetworkDevice, device2:
NetworkDevice):
        """З'єднати два пристрої у топології"""
        device1.add_connection(device2)
        self.connections.append((device1, device2))
        self.graph.add_edge(device1.name, device2.name)

    def visualize_topology(self):
        """Візуалізація мережевої топології"""
        plt.figure(figsize=(10, 8))
        pos = nx.spring_layout(self.graph)
        nx.draw(self.graph, pos, with_labels=True, node_color="skyblue",
node_size=2000, edge_color="gray", font_size=10, font_weight="bold")
        plt.title(f"Network Topology: {self.name}")
        plt.show()

    def list_devices(self):
        """Вивести список пристроїв у мережі"""

```

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

```

        for device in self.devices:
            print(device)

    def list_connections(self):
        """Вивести список з'єднань між пристроями"""
        for conn in self.connections:
            print(f"{conn[0].name} <--> {conn[1].name}")

# Клас для управління конфігурацією пристроїв
class DeviceConfiguration:
    def __init__(self, device: NetworkDevice):
        self.device = device
        self.settings = {}

    def set_setting(self, key, value):
        """Задати налаштування для пристрою"""
        self.settings[key] = value

    def get_setting(self, key):
        """Отримати значення налаштування"""
        return self.settings.get(key, None)

    def apply_configuration(self):
        """Застосування конфігурації до пристрою"""
        print(f"Applying configuration to {self.device.name}:")
        for key, value in self.settings.items():
            print(f"    {key}: {value}")

# Демонстрація роботи системи
# Створення підмережі з IP адресами
subnet_planner = SubnetPlanner("192.168.1.0", 24)

# Створення пристроїв
router1 = NetworkDevice("Router", "Router1")
router2 = NetworkDevice("Router", "Router2")
switch1 = NetworkDevice("Switch", "Switch1")
switch2 = NetworkDevice("Switch", "Switch2")
firewall = NetworkDevice("Firewall", "Firewall1")

# Призначення IP-адрес пристроям

```

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

```

router1.assign_ip(subnet_planner.allocate_ip())
router2.assign_ip(subnet_planner.allocate_ip())
switch1.assign_ip(subnet_planner.allocate_ip())
switch2.assign_ip(subnet_planner.allocate_ip())
firewall.assign_ip(subnet_planner.allocate_ip())

# Створення мережевої топології
topology = NetworkTopology("Data Center Topology")

# Додавання пристроїв до топології
topology.add_device(router1)
topology.add_device(router2)
topology.add_device(switch1)
topology.add_device(switch2)
topology.add_device(firewall)

# З'єднання пристроїв у топології
topology.connect_devices(router1, switch1)
topology.connect_devices(router1, switch2)
topology.connect_devices(router2, switch1)
topology.connect_devices(switch1, firewall)

# Візуалізація топології
topology.visualize_topology()

# Виведення списку пристроїв та з'єднань
topology.list_devices()
topology.list_connections()

# Налаштування конфігурацій для пристроїв
config_router1 = DeviceConfiguration(router1)
config_router1.set_setting("OSPF", True)
config_router1.set_setting("Static Routes", ["192.168.2.0/24"])
config_router1.apply_configuration()

config_firewall = DeviceConfiguration(firewall)
config_firewall.set_setting("Firewall Rules", ["Allow HTTP", "Allow HTTPS",
"Block FTP"])
config_firewall.apply_configuration()

# Перевірка залишку доступних IP-адрес
print(f"Available          IPs          in          the          subnet:
{subnet_planner.available_ips_count()}")

```

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Алгоритм призначений в основному для додатків, у яких ключ міняється нечасто, до того ж існує фаза початкового рукостискання, під час якої відбувається автентифікація сторін і узгодження загальних параметрів і секретів. При реалізації на 32-бітних мікропроцесорах з більшим кешем даних Blowfish значно швидше DES.

Алгоритм складається із двох частин: розширення ключа й шифрування даних. Розширення ключа перетворює ключ довжиною, принаймні, 448 біт у кілька масивів підключів загальною довжиною 4168 байт.

В основі алгоритму лежить мережа Фейштеля з 16 ітераціями. Кожна ітерація складається з перестановки, що залежить від ключа, і підстановки, що залежить від ключа й даних. Операціями є XOR і додавання 32-бітних слів.

Blowfish використовує велику кількість підключів. Ці ключі повинні бути обчислені заздалегідь, до початку будь-якого шифрування або дешифрування даних. Елементи алгоритму:

1. P – масив, що складається з вісімнадцяти 32-бітних підключів:

$$P_1, P_2, \dots, P_{18}.$$

2. Чотири 32-бітних S -boxes с 256 входами кожний. Перший індекс означає номер S -box, другий індекс – номер входу.

$$S_{1,0}, S_{1,1}, \dots S_{1,255};$$

$$S_{2,0}, S_{2,1}, \dots S_{2,255};$$

$$S_{3,0}, S_{3,1}, \dots S_{3,255};$$

$$S_{4,0}, S_{4,1}, \dots S_{4,255};$$

Шифрування

Входом є 64-бітний елемент даних X , що ділиться на дві 32-бітні половини, X_l і X_r .

$$X_l = X_l \text{ XOR } P_i$$

$$X_r = F(X_l) \text{ XOR } X_r$$

$$\text{Swap } X_l \text{ and } X_r$$

Функція F

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

Розділити X_1 на чотири 8-бітних елементи A, B, C, D .

$$F(X_1) = ((S_{1,A} + S_{2,B} \bmod 2^{32}) \text{ XOR } S_{3,C}) + S_{4,D} \bmod 2^{32}$$

Дешифрування відрізняється від шифрування тим, що P_i використовуються у зворотному порядку.

Генерація підключів

Підключи обчислюються з використанням самого алгоритму Blowfish.

1. Ініціалізувати перший P -масив і чотири S -boxes фіксовані рядки.
2. Виконати операцію XOR P_1 з першими 32 бітами ключа, операцію XOR P_2 із другими 32 бітами ключа й т.д. Повторювати цикл доти, поки весь P -масив не буде побітово складний з усіма бітами ключа. Для коротких ключів виконується конкатенація ключа із самим собою.
3. Зашифрувати нульовий рядок алгоритмом Blowfish, використовуючи підключи, описані в пунктах (1) і (2).
4. Замінити P_1 і P_2 виходом, отриманим на кроці (3).
5. Зашифрувати вихід кроку (3), використовуючи алгоритм Blowfish з модифікованими підключами.
6. Замінити P_3 і P_4 виходом, отриманим на кроці (5).
7. Продовжити процес, замінюючи всі елементи P -масиву, а потім всі чотири S -boxes, виходами відповідним чином модифікованого алгоритму Blowfish.

Для створення всіх підключів потрібна 521 ітерація.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання магістерської дипломної роботи. Розрахунок проводиться через консольний додаток з передачею результатів у інтерфейс. Розроблене програмне забезпечення системи проектування та побудови топології мереж у центрах обробки даних складається з наступних функціональних блоків:

- Навігаційне меню: Формат; Налаштування; Довідка.
- Вікно виведення результату роботи системи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ: Пошук об'єктів мережі; Статистика; База даних статичної інформації мережі; Перегляд сформованої мережі; Налаштування фільтру; Вікно прогнозування завантаженості мережі; Параметри потоків даних; Побудова карти мережі.

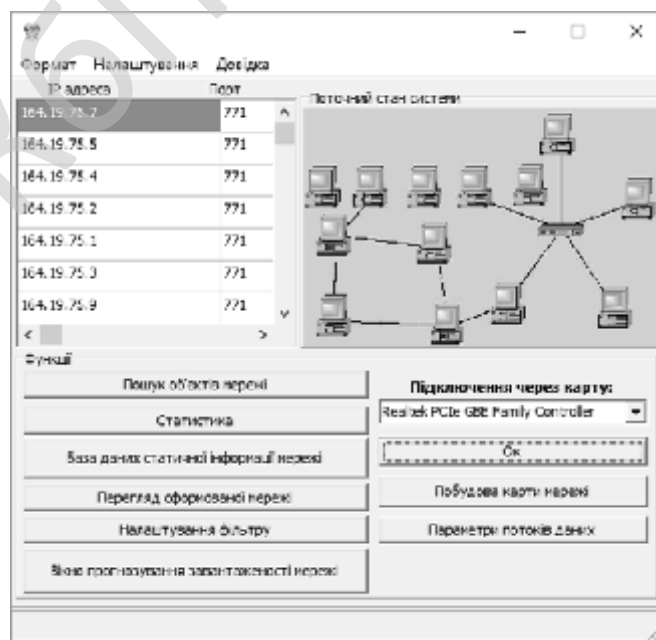


Рисунок 5.1 – Головне вікно розробленого ПЗ

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.

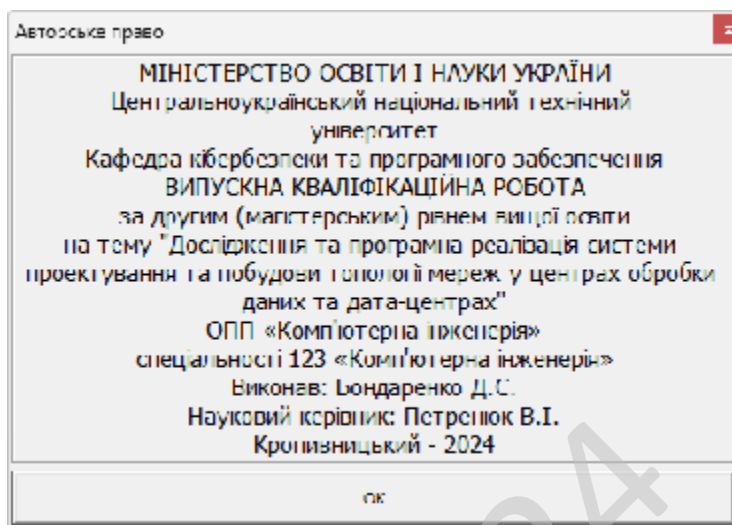


Рисунок 5.2 – Вікно розробника ПЗ

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-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

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 та дата-центрах.

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

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

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

Методи дослідження базуються на методах побудови топології 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 та дата-центрах.

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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи проектування і побудови топології мереж у центрах обробки даних можуть зацікавити аудиторію, схематично подану на рисунку 7.1.

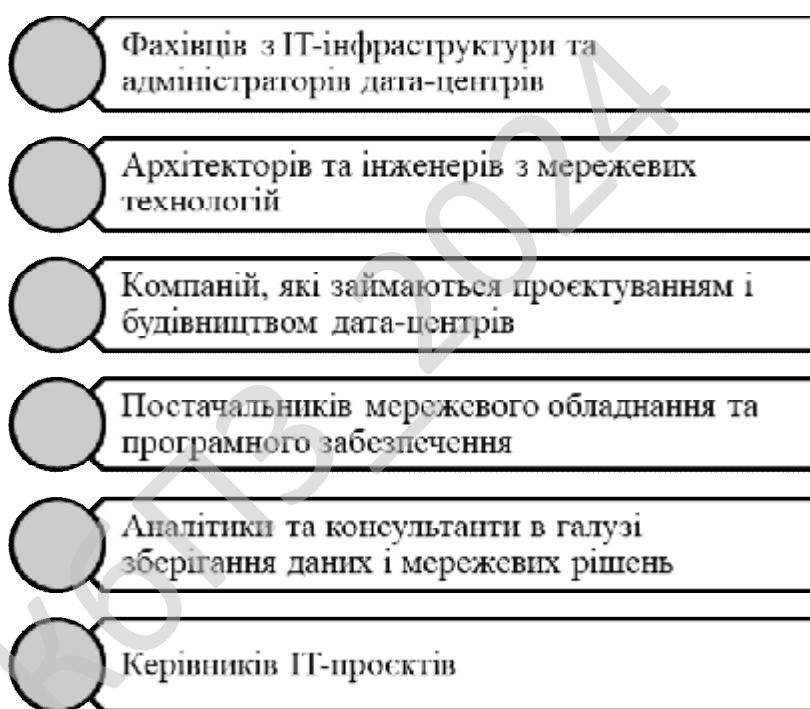


Рисунок 7.1 – Ключові сегменти цільової аудиторії

Ці зацікавлені сторони можуть отримати користь від дослідження, щоб покращити свої рішення для дата-центрів, забезпечити відповідність сучасним стандартам та збільшити ефективність мережевої інфраструктури.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості програмної реалізації системи проектування і побудови топології мереж у дата-центрах можна застосувати метод експертних оцінок із застосуванням бальної шкали.

За критерії обираємо:

- надійність і безперервність роботи мережі;
- стійкість до збоїв;
- гнучкість топології (масштабованість);
- простота інтеграції з іншими системами;
- економічна ефективність;
- зручність в управлінні та адмініструванні;
- підтримка сучасних стандартів безпеки.

Експертами обираємо фахівців, які мають досвід роботи з мережами та IT-інфраструктурою: архітекторів мереж, системних адміністраторів, аналітиків і технічних директорів.

Встановлюємо шкалу оцінювання, від 1 до 5, де:

- 1 – дуже низька привабливість,
- 5 – дуже висока привабливість.

Кожен експерт оцінює систему за всіма критеріями, присвоюючи бали відповідно до своїх знань і досвіду. Після того, як усі експерти виставили свої оцінки, обчислюємо середнє значення для кожного критерію, щоб визначити загальний рівень привабливості.

Оцінки кожного критерію аналізуємо і за їхніми результатами складаємо загальний рейтинг привабливості розробленої системи. Якщо певні критерії мають низькі оцінки, це можна вказувати на аспекти, що потребують покращення.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Таблиця 7.1 – Зведені результати опитування

Критерії	Експерт 1	Експерт 2	Експерт 3	Середній бал
Надійність і безперервність	5	4	5	4.67
Стійкість до збоїв	4	5	5	4.67
Гнучкість топології	5	4	4	4.33
Простота інтеграції	4	3	4	3.67
Економічна ефективність	3	4	3	3.33
Зручність в управлінні	5	4	4	4.33
Підтримка стандартів безпеки	5	5	5	5.00
Загальний середній бал: 4.33				

На основі отриманих даних робимо висновок про високий рівень привабливості системи для дата-центрів. Водночас варто звернути увагу на економічну ефективність, яка потребує вдосконалення. Такий підхід дозволяє якісно оцінити потенційну привабливість системи на етапі планування та доопрацювання.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи проектування та побудови топології мереж у дата-центрах варто вибрати метод, який враховує специфічність проекту, витрати на ресурси та необхідні технічні вимоги. Варто

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

обирати між методом оцінки на основі функціональних точок, методом оцінки вартості на основі витрат та методом аналогів. Для програмної реалізації системи проектування мережевої топології в дата-центрах, де важлива точна оцінка функціоналу і витрат на ресурси, метод оцінки на основі функціональних точок (FPA) або метод витрат (Cost-Based Estimation) можуть бути оптимальними. Але все ж метод аналогів рекомендуємо як над швидкий. Адже є аналогічні проекти із подібними функціональними та технічними вимогами. Це швидкий метод, який саме і варто застосувати.

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

Економічна ефективність від впровадження системи проектування та побудови топології мереж у центрах обробки даних може проявлятися через скорочення витрат, оптимізацію роботи мережі та покращення загальної продуктивності.

Зменшення витрат на проектування та розгортання мережі першочергово проявляється через можливість швидко створювати і тестувати різні варіанти топології без додаткових витрат на інженерів та проектувальників та зниження потреби в повторних проектних і монтажних роботах завдяки точності та оптимізації початкового проекту. Розрахунково витрати на проектування і тестування мережевої топології можна зменшити на 20–30%.

Оптимізація використання обладнання шляхом аналізу та визначення найбільш ефективних маршрутів і конфігурації, що допомагає зменшити кількість обладнання або замінити дороге обладнання на більш ефективні альтернативи; зменшення капітальних витрат на обладнання через зниження кількості непотрібних мережевих пристроїв. Розрахунково на 10%.

Скорочення операційних витрат завдяки зниженню кількості простоїв. Оптимальна топологія зменшує ризики простоїв та проблем із продуктивністю,

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

до нової системи. Покращення якості обслуговування клієнтів. Стабільність та гнучкість мережевої інфраструктури підвищують якість обслуговування кінцевих користувачів. Задоволеність клієнтів допомагає утримувати поточних клієнтів і залучати нових, що позитивно впливає на дохідність компанії. Покращення якості обслуговування на 10% може зменшити відтік клієнтів на 5%.

Економічна ефективність досягається щорічно завдяки зменшенню операційних витрат, оптимізації витрат на проектування, зниженню енергоспоживання та підвищенню продуктивності мережевої інфраструктури.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Покроковий алгоритм для просування проєкту програмної реалізації системи проектування та побудови топології мереж у центрах обробки даних (ЦОД) та дата-центрах представлено графічно на рисунку 7.2. Цей алгоритм орієнтований на залучення цільової аудиторії та побудову довіри до продукту.

Завдяки комплексному підходу до просування проєкту можна ефективно розширити ринок, залучити цільову аудиторію та створити довіру до продукту.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту та шляхів реалізації проєкту програмної системи для проектування та побудови топології мереж у центрах обробки даних, можна застосувати наступні стратегії:

- напряму пропонувати продукт великим дата-центрам і компаніям з високими вимогами до інфраструктури (телекомунікаційні оператори, великі підприємства тощо);
- встановити партнерські стосунки з компаніями, які надають консалтингові або інтеграційні послуги для дата-центрів;

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

1. Аналіз ринку та конкурентів	Оцінка ринкової ніші: дослідити, як інші рішення задовольняють потреби дата-центрів у проєктуванні мереж.
	Визначення унікальних переваг (USP): сформулювати ключові особливості, які відрізняють продукт від конкурентів (наприклад, зменшення витрат, простота інтеграції, гнучкість).
	Аналіз цільової аудиторії: визначити основних споживачів – адміністратори дата-центрів, мережеві архітектори, інженери.
2. Позиціонування та брендинг	Розробка повідомлення бренду: створити чітке та зрозуміле повідомлення, що підкреслює ефективність системи в автоматизації та оптимізації мережевої топології.
	Розробка логотипу та візуальних матеріалів: створити візуальні матеріали, що будуть асоціюватися з продуктом.
	Розробка сайту і лендінгу продукту: підготувати сайт із описом продукту, його перевагами, кейсами і формою для заявок.
3. Маркетингові дослідження та аналіз попиту	Опитування та інтерв'ю з потенційними клієнтами: щоб краще зрозуміти очікування та основні потреби клієнтів.
	Створення демо-версії продукту: розробити демонстраційну версію або віртуальний тур для потенційних клієнтів, щоб вони могли ознайомитися з можливостями системи.
4. Контент-маркетинг	Написання статей та white papers: публікувати матеріали про переваги оптимізації мережевої інфраструктури в ЦОДах, використовуючи приклади економічної ефективності та надійності.
	Блог на веб-сайті: ділитися результатами впровадження, кейсами та інсайтами у сфері автоматизації проєктування мереж.
	Вебінари та навчальні матеріали: організувати вебінари для показу можливостей продукту, створення записів із прикладами використання.
5. Соціальні мережі та професійні спільноти	LinkedIn та Twitter: публікувати анонси, новини та статті про продукт, а також брати участь в обговореннях на професійних платформах.
	GitHub та форуми мережевих технологій: якщо є відкриті компоненти продукту або бібліотеки, варто публікувати їх на GitHub і залучати спільноту до обговорень та тестування.
6. PR та участь у виставках	Прес-релізи: анонсувати вихід продукту на спеціалізованих ресурсах та у виданнях, орієнтованих на IT-інфраструктуру та технології дата-центрів.
	Участь у виставках та конференціях: представляти продукт на конференціях з тематики дата-центрів, таких як Data Center World, де є потенційні клієнти та партнери.
	Публікації у технічних журналах та на порталах: готувати матеріали для ресурсів, що спеціалізуються на технологіях дата-центрів та мережевих рішеннях.
7. Сформування партнерств	Співпраця з постачальниками мережевого обладнання та програмного забезпечення: запропонувати інтеграцію та оптимізацію їхнього обладнання з вашою системою.
	Альянси з консалтинговими компаніями: співпрацювати з компаніями, які надають послуги із проєктування дата-центрів і можуть рекомендувати вашу систему своїм клієнтам.
8. Програма тестування для ключових клієнтів	Плотний запуск у партнерських організаціях: запропонувати безкоштовне тестування або зв'язки для перших клієнтів, щоб зібрати зворотний зв'язок і показати переваги рішення.
	Відгуки та кейси: зібрати відгуки і створити реальні кейси, що демонструють позитивні зміни після впровадження продукту.
9. Оцінка та оптимізація стратегії просування	Аналіз даних просування: регулярно аналізувати відгуки, відвідуваність вебсайту, конверсії та залучення аудиторії в соціальних мережах.
	Коригування маркетингової стратегії: на основі зібраної інформації та відгуків клієнтів змінювати акценти просування, наприклад, більше наголошувати на тих перевагах продукту, які найбільше цінуються ринком.
10. Післяпродажна підтримка та постійне вдосконалення	Служба підтримки клієнтів: створити команду технічної підтримки для швидкого реагування на запити користувачів.
	Регулярні оновлення та нові функції: забезпечувати стабільні оновлення, пропонувати нові функціональні можливості та вдосконалення на основі відгуків клієнтів.

Рисунок 7.2 – Алгоритм просування проєкту

- використовувати точкову рекламу на професійних платформах, таких як linkedin, де зосереджені фахівці, пов'язані з керуванням дата-центрами;
- співпраця з відомими експертами в галузі дата-центрів та публікації на спеціалізованих ресурсах можуть забезпечити довіру до продукту і підвищити його видимість;
- публікація матеріалів, статей і кейсів, які вирішують реальні проблеми проектування та оптимізації мереж у дата-центрах, що допоможе покращити позиції у пошукових системах і привернути увагу цільових клієнтів;
- надавати можливість інтеграції з уже існуючими рішеннями для управління мережевою інфраструктурою.
- запровадження гнучкої передплатної моделі (місячної, річної тощо) зробить систему доступною для ширшого кола клієнтів, які можуть випробувати її можливості без значних початкових інвестицій;
- розробити програми для швидкого освоєння системи, що підвищить рівень задоволення користувачів і сприятиме довготривалим відносинам з клієнтами;
- забезпечити високоякісну технічну підтримку та допомогу з налаштуванням системи, що зменшить ризики та підвищить рівень довіри до продукту;
- використовувати систему CRM для відстеження ефективності різних каналів та прийняття рішень щодо оптимізації роботи з партнерами та клієнтами;
- завдяки CRM проводити сегментацію за потребами, наприклад, між тими, хто потребує простих рішень для малих ЦОДів, і тими, хто розгортає складні, масштабовані мережі;
- випуск оновлень продукту на основі зворотного зв'язку допомагає втримувати клієнтів і створює позитивний імідж продукту як того, що активно розвивається.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Ці заходи допоможуть ефективно оптимізувати канали збуту, підвищити конкурентоспроможність продукту і розширити його ринкове охоплення серед центрів обробки даних та дата-центрів.

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи проектування та побудови топології мереж для центрів обробки даних та дата-центрів наведені на рисунку 7.3.

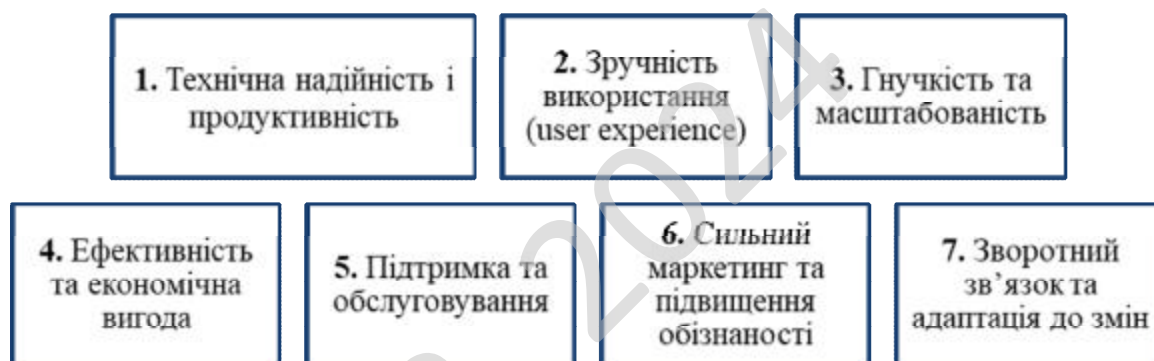


Рисунок 7.3 – Ключові фактори успіху проєкту

Виконання кожного з цих факторів допоможе забезпечити конкурентоспроможність і високу привабливість системи для ринку, а також стійке зростання клієнтської бази.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Програмісти у процесі роботи отримують негативний вплив на органи зору, а також мають значну розумову напругу і нервово-емоційне навантаження. Руки (м'язи рук та суглоби пальців) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій спеціалісти відносять високочастотні електромагнітні коливання роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

До недоліків умов праці користувачів комп'ютерної техніки можна віднести:

- недостатню площу і обсяг виробничого приміщення;
- недотримання вимог, мікроклімату на робочих місцях;
- низький рівень освітленості у приміщеннях і на робочих поверхнях апаратури;
- підвищений рівень низькочастотних магнітних полів від моніторів;
- порушення вимог організації робочих місць;
- недотримання вимог до режимам праці та відпочинку;
- надмірне виробничу навантаження працівників;
- відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [3] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

виконання; інструктаж з охорони праці; організацію контролю над станом умов праці в робочих місцях; проведення атестації робочих місць в умовах праці.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – ось одна з головних задач охорони праці [5].

8.2 Характеристика умов праці програміста

В приміщенні, в якому проводиться розробка і дослідження програмного продукту, відсутні умови, які можуть створювати підвищену або особливо підвищену небезпеку, тому воно відноситься до класу звичайних приміщень згідно Правил улаштування електроустановок (ПУЕ). Джерелом живлення є трифазна мережа напруги 380/220 В з глухо заземленою нейтраллю, з частотою 50 Гц згідно За пожежною та вибухонебезпекою приміщення відноситься до класу В. В таблиці 8.1 наведена загальна характеристика приміщення щодо пожежної та вибухонебезпеки та важкості робіт.

Таблиця 8.1. – Загальна характеристика приміщення щодо вибухопожежонебезпеки та важкості робіт

Характеристика приміщень за вибухопожежною категорією та класом зони	Загальна характеристика приміщення	Категорія за важкістю робіт згідно ГН 3.3.5-8.6.6.1 -2002
В – пожежонебезпечне клас П – II	Звичайне без ознак хімічного забруднення та нормальної вологості і за санітарними нормами	1а.....до 139 Вт/м ² 1б.....до 140-174 Вт/м ² Клас умов праці – оптимальний

Температура повітря в приміщенні визначається температурою зовнішнього повітря і тепловою енергією, що виділяється всередині приміщення. Джерелами теплоти в даному приміщенні є люди, електроустаткування, а також освітлювальні прилади в темний час доби. Зовнішнім джерелом надлишкового тепла є сонячна радіація у світлий час доби. Робота, виконувана в даному приміщенні, відноситься до категорії І-а. Людиною в цьому випадку виділяється до 120 ккал теплової енергії в годину. Вологість повітря в приміщенні визначається вологістю атмосферного і видихуваного людьми повітря, а також випарами з поверхні шкіри.

У приміщенні немає виділення шкідливих газів. Тому що в ньому не проводиться монтажних робіт, пайки чи інших робіт, при яких виділяються шкідливі гази.

Для нормалізації параметрів повітряного середовища також періодично здійснюється провітрювання приміщення і вологе прибирання. У всьому будинку діє встановлена загально обмінна витяжна вентиляція.

Раціональне освітлення приміщення сприяє кращому виконанню виробничого завдання і забезпеченню комфорту при роботі. Для забезпечення нормального освітлення застосовуються природне, однобічне, бічне і штучне освітлення, а також сполучене, нормуються згідно ДБН В.2.5-28-2006 Природне і штучне освітлення [6].

За результатами виміру освітленості величина освітленості від системи загального штучного освітлення дорівнює 310 лк, що відповідає вимогам, які пред'являються до даного приміщення.

Основними джерелами шуму на робочих місцях, обладнаних відео дисплейними терміналами, є принтер, сканер факс і обладнання для кондиціонування повітря, в самих відео дисплейних терміналів – вентилятори систем охолодження і трансформатори.

Згідно ДСанПіН 3.3.2.007-98 [6] допустимий еквівалентний рівень шуму для робочого місця програміста складає 50 дБА (акустичних децибела).

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

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 в нормі. У кімнаті знаходиться ПК з 4-ядерним процесором і 23-дюймовим IPS монітором, а також меблі.

Приміщення має довжину 4м, 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 8.2.

Таблиця 8.2 – Фактичні та нормативні значення параметрів приміщення

Параметр	Норма *	Реальні параметри
Площа, S	не менше 6 м ²	14 м ²
Об'єм, V	не менше 20 м ³	37,8 м ³

*Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

За даними, які наведено у табл. 8.2, можна зробити висновок, що отримані показники, площа та об'єм приміщення у розрахунку на одно робоче місце користувача ПК відповідає чинним нормам і вимогам.

Щодо мікроклімату, то згідно з ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» [7] роботу з ПК можна віднести до категорії легка 1а. Джерелами тепла в цьому приміщенні є люди, електроустаткування, освітлювальні прилади в темний час доби і система опалювання взимку. Оператором виділяється до 120ккал теплової енергії за

годину. Оптимальні та фактичні значення параметрів мікроклімату приведені в таблиці 8.3.

Таблиця 8.3 – Значення параметрів мікроклімату

Період року	Параметр	Оптимальний*	Фактичний
Теплий	Температура	23 – 25 ⁰ С	24 ⁰ С
	Вологість	40 – 60%	50%
	Швидкість повітря	< 0,1м/с	
Холодний	Температура	22 – 24 ⁰ С	23 ⁰ С
	Вологість	40 – 60%	55%
	Швидкість повітря	< 0,1м/с	

*ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень»

По отриманим замірам параметрів мікроклімату можна зробити висновок, що усі 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» [7] ця робота відноситься до V_a розряду зорових робіт. Передбачається використання природного, штучного і змішаного освітлення.

Природне освітлення здійснюється за допомогою вікна, площа якого складає $S' = 1,8 \cdot 1,5 = 2,7 \text{ м}^2$ і є боковим освітленням. У світильниках місцевого і загального освітлення використовуються світлодіодні лампи потужністю 20 Вт із світловим потоком 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, що не має сильного електромагнітного випромінювання і сертифіковані в Україні.

Блок живлення є екранованим і не випускає вищезазначених видів випромінювання.

8.4 Розробка заходів з охорони праці

Перерахуємо проведені заходи щодо забезпечення умов праці на робочому місці програміста.

Для зменшення шуму в приміщенні пропоную використовувати замість матричного принтера, що створює багато шуму, більш тихий – лазерний принтер.

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 і освітленості до цих заходів можна віднести: організацію природної вентиляції, за допомогою

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

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, проведених відділом охорони праці.

Як міри по зниженню шуму можна запропонувати:

- облицювання стелі і стін звукопоглинаючим матеріалом (знижують шум на 6-8 до);
- екранування робочого місця (постановкою перегородок, діафрагм);
- установка в комп'ютерних приміщеннях устаткування, що робить мінімальний шум;
- раціональне планування приміщення.

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

8.5 Розрахункова частина

При роботі з комп'ютерною та офісною технікою головними причинами ураження електричним струмом є двофазне дотикання; однофазне дотикання; дотик до корпусу обладнання, яке не проводить струм, але опинилося під напругою; перебування в зоні дії атмосферної або статичної електрики; вхід у зону дії електромагнітного поля.

Правильна організація експлуатації й обслуговування комп'ютерної та офісної техніки, контрольно-вимірювальної апаратури оговорена "Правилами технічної експлуатації електроустановок споживачів і правил техніки безпеки при експлуатації електроустановок споживачів" [8]. Розрахуємо конструкцію захисного заземлення (рис. 8.1) за методикою [9, 10].

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

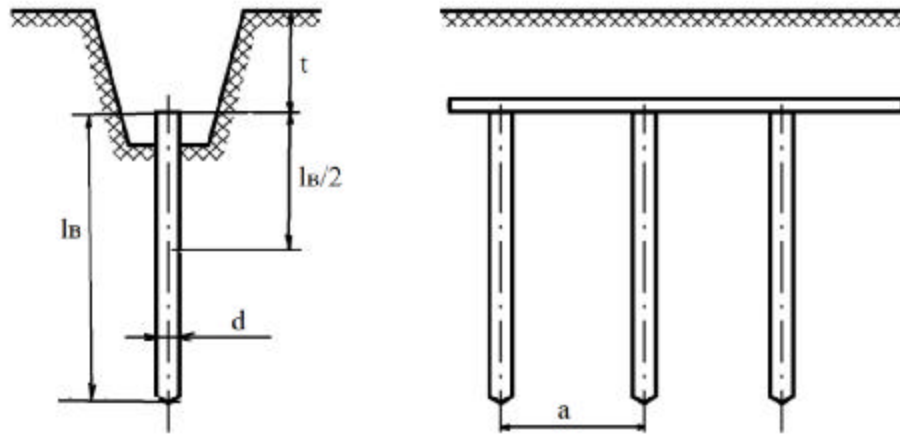


Рисунок 8.1 – Схема електричного захисного заземлення

Початкові дані для розрахунку:

- напруга живлення 380 В;
- режим нейтралі трансформатора – ізольована; – природне заземлення присутнє, $R_e = 20 \text{ Ом}$;
- ґрунт – глина; питомий опір – $\rho_r = 60 \text{ Ом}\cdot\text{м}$;
- вологість глини під час вимірювання її опору – нормальна; – кліматична зона – третя.

Відповідно рекомендацій ПУЕ умови виконання завдання визначають значення

$$R_{nye} = 4 \text{ Ом.}$$

Природні заземлювачі наявні, тому допустимий опір R_d штучних заземлювачів

$$R_d \leq \frac{20 \cdot 4}{20 - 4} \leq 5 \text{ Ом,}$$

Для виконання заземлення обираємо вертикальні електроди із кутової сталі шириною профілю $b_k = 50 \text{ мм}$, довжиною $l_v = 3 \text{ м}$; для таких вертикальних електродів коефіцієнт сезонності $\psi = 1,3$;

- відстань між електродами $a = 3 \text{ м}$, тобто $a / l_v = 1$;
- горизонтальний з'єднувальний електрод – стальна смуга шириною $b_g = 45 \text{ мм}$;

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

– глибина заглиблення смуги $t = 0.8$ м.

Визначаємо розрахунковий питомий електричний опір глини ρ_1 , де буде розміщуватися заземлювальний пристрій, для третьої кліматичної зони місцевості $\psi = 1.3$:

$$\rho_1 = 60 \cdot 1,3 = 78 \text{ Ом м.}$$

Визначаємо опір розтіканню струму з одного вертикального заземлювача, попередньо визначивши відстань від поверхні землі до середини вертикального заземлювача t_v :

$$t_v = 0,8 + 0,5 \cdot 3 = 2,3 \text{ м}$$

$$R_v = \frac{78}{2 \cdot 3 \cdot 1,3 \cdot 3} \ln \frac{2 \cdot 3}{0,95 \cdot 0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} = 21,424 \text{ Ом}$$

Розраховуємо кількість вертикальних електродів n методом послідовних наближень. У першому наближенні приймаємо $\eta_v = 1$. Виконуючи округлення результату розрахунку у більший бік отримуємо:

$$n_1 = \frac{21,424}{5} \approx 5,4$$

Для n_1 за таблицею визначаємо η_{v1} . Оскільки значення η_{v1} для n_1 у таблиці відсутнє, знаходимо середнє арифметичне:

$$\eta_{v1} = \frac{0,73 + 0,65}{2} = 0,69.$$

Для η_{v1} визначаємо нове число n_2 , округляємо його до більшого цілого:

$$n_2 = \frac{21,424}{5 \cdot 0,69} \approx 7.$$

Для $n_2 = 7$ визначимо η_{v2} :

$$\frac{n_2 - 6}{10 - 6} = \frac{\eta_{v2} - 0,65}{0,59 - 0,65}$$

Для η_{v3} визначаємо нове число n_3 та округляємо результат до більшого цілого:

$$n_3 = \frac{21,424}{5 \cdot 0,635} \approx 7.$$

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

Таким чином отримали, що кількість електродів не змінюється (різниця між останніми числами заземлювачів менше одиниці). Тобто число вертикальних заземлювачів складає 7.

Визначаємо довжину горизонтального електроду. Для заземлювачів, розташованих у ряд

$$L_g = 1,05 \cdot 1 \cdot (7 - 1) = 18,9 \text{ м.}$$

Знаходимо опір розтікання струму горизонтального електроду, визначивши спочатку величину ρ_2 , $\psi = 2,5$:

$$\rho_2 = 60 \cdot 2,5 = 150 \text{ Ом}\cdot\text{м},$$

$$R_g = \frac{150}{2\pi \cdot 18,9} \ln \frac{2 \cdot 18,9 \cdot 18,9}{0,045 \cdot 0,8} = 12,5 \text{ Ом}$$

Визначаємо загальний результуючий опір системи заземлення, спочатку визначивши коефіцієнт використання горизонтального електроду за допомогою табличними даними. Для цього, виконуючи лінійну інтерполяцію, напишемо рівняння:

$$\eta_g = -0,00667 \cdot n_3 + 0,7933 = 0,747;$$

$$R_{gr} = \frac{21,424 \cdot 12,5}{21,424 \cdot 0,747 + 7 \cdot 12,5 \cdot 0,636} = 3,742 \text{ Ом.}$$

Порівнюємо значення R_{gr} з R_d . Співвідношення $R_{gr} < R_d$ виконується, тобто вибрана конструкція заземлення задовольняє вимогам ПУЕ.

Таким чином, приймаємо наступну конструкцію заземлення: вертикальні електроди розміщені в ряд в кількості 7 штук, довжина з'єднувального горизонтального електроду складає 18,9 м.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

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 та дата-центрах.

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 та дата-центрах.

Рішення даного завдання полягало у вирішенні наступних задач:

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

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 та дата-центрах.

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-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

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 Blowfish.

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-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бондаренко Д.С. Дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних та дата-центрах // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
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. 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.
11. 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.

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

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

13. 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.

14. 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.

15. 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.

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

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.*

28. 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.*

29. 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.*

30. 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.*

31. 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.*

32. 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.*

33. 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.*

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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

40. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки.* №4. С. 103-110. 2020.

41. 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.

42. 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.

43. 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.

44. 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.

45. 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.

46. 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).

47. 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

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

48. 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.

49. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

50. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

51. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

КБПЗ – 2024

					ВКРМ-123.24.0004.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1	Найменування та область застосування.....	2
2	Підстава для розробки.....	2
3	Мета та призначення розробки.....	2
4	Джерела розробки.....	2
5	Технічні вимоги.....	2
5.1	Вміст проекту.....	2
5.2	Показники призначення.....	3
5.3	Вимоги до функціональних характеристик.....	3
5.4	Вимоги до архітектури.....	3
5.5	Вимоги до надійності.....	3
5.6	Умови експлуатації.....	4
5.7	Вимоги до складу та параметрів технічних засобів.....	4
5.8	Вимоги до інформаційної і програмної сумісності.....	4
5.8.1	Обладнання.....	4
5.8.2	Мова програмування.....	4
5.8.3	Вхідні дані.....	5
5.8.4	Вихідні дані.....	5
6	Вимоги до програмної документації.....	5
7	Економічні вимоги.....	5
8	Вимоги щодо охорони праці.....	5
9	Перелік документів, що розробляються.....	6
10	Етапи розробки.....	6
11	Порядок контролю та приймання.....	6

					ВКРМ-123.24.0004.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Бондаренко Д.С.				Літ.	Аркуш	Аркушів
Перевірів	Петренко В.І.						
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М		
Затв.	Смірнов О.А.						
Дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних та дата- центрах							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи проектування та побудови топології мереж у центрах обробки даних та дата-центрах.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 19-13 від 07.08.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи проектування та побудови топології мереж у центрах обробки даних та дата-центрах.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи проектування та побудови топології мереж у центрах обробки даних та дата-центрах;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута характеристика умов праці програміста.

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 99 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 16.12.2024 р.

					ВКРМ-123.24.0004.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти
_____ Петренюк В.І.

*Дослідження та програмна реалізація
системи проектування та побудови топології мереж у центрах обробки
даних та дата-центрах*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 15

Літера: РП

Кропивницький – 2024 року

Основна програма

```
class NetworkDevice:
    def __init__(self, name, device_type, ip_address, mac_address):
        self.name = name
        self.device_type = device_type
        self.ip_address = ip_address
        self.mac_address = mac_address

    def display_info(self):
        return f"Device Name: {self.name}, Type: {self.device_type}, IP: {self.ip_address}, MAC: {self.mac_address}"

class Rack:
    def __init__(self, rack_id):
        self.rack_id = rack_id
        self.devices = []

    def add_device(self, device):
        self.devices.append(device)

    def display_rack_info(self):
        info = f"Rack ID: {self.rack_id}\n"
        for device in self.devices:
            info += device.display_info() + "\n"
        return info

class DataCenter:
    def __init__(self, name, location):
        self.name = name
        self.location = location
        self.racks = []

    def add_rack(self, rack):
        self.racks.append(rack)

    def display_data_center_info(self):
        info = f>Data Center Name: {self.name}, Location: {self.location}\n"
        for rack in self.racks:
            info += rack.display_rack_info() + "\n"
        return info

class NetworkTopology:
    def __init__(self):
        self.data_centers = []

    def add_data_center(self, data_center):
        self.data_centers.append(data_center)

    def display_topology(self):
        topology_info = ""
        for data_center in self.data_centers:
            topology_info += data_center.display_data_center_info() + "\n"
        return topology_info

# Створення мережевих пристроїв
router1 = NetworkDevice("Router1", "Router", "192.168.1.1", "00:1A:2B:3C:4D:5E")
switch1 = NetworkDevice("Switch1", "Switch", "192.168.1.2", "00:1A:2B:3C:4D:5F")
firewall1 = NetworkDevice("Firewall1", "Firewall", "192.168.1.3", "00:1A:2B:3C:4D:6A")
server1 = NetworkDevice("Server1", "Server", "192.168.1.4", "00:1A:2B:3C:4D:6B")

# Створення стійки з пристроями
rack1 = Rack("Rack1")
rack1.add_device(router1)
rack1.add_device(switch1)
rack1.add_device(firewall1)
```

```

# Створення ще однієї стійки
rack2 = Rack("Rack2")
rack2.add_device(server1)

# Створення центру обробки даних та додавання стійок
dc1 = DataCenter("MainDC", "Kyiv")
dc1.add_rack(rack1)
dc1.add_rack(rack2)

# Створення топології мережі
topology = NetworkTopology()
topology.add_data_center(dc1)

# Виведення топології
print(topology.display_topology())

# Додавання нових пристроїв та центрів обробки даних
router2 = NetworkDevice("Router2", "Router", "10.0.0.1", "00:2B:3C:4D:5E:6F")
switch2 = NetworkDevice("Switch2", "Switch", "10.0.0.2", "00:2B:3C:4D:5E:6G")
firewall2 = NetworkDevice("Firewall2", "Firewall", "10.0.0.3",
"00:2B:3C:4D:5E:6H")

rack3 = Rack("Rack3")
rack3.add_device(router2)
rack3.add_device(switch2)
rack3.add_device(firewall2)

dc2 = DataCenter("BackupDC", "Lviv")
dc2.add_rack(rack3)

topology.add_data_center(dc2)

# Виведення оновленої топології
print(topology.display_topology())

class Link:
    def __init__(self, device1, device2, link_type):
        self.device1 = device1
        self.device2 = device2
        self.link_type = link_type

    def display_link_info(self):
        return f"Link between {self.device1.name} and {self.device2.name}, Type:
{self.link_type}"

class NetworkMap:
    def __init__(self):
        self.links = []

    def add_link(self, link):
        self.links.append(link)

    def display_network_map(self):
        map_info = ""
        for link in self.links:
            map_info += link.display_link_info() + "\n"
        return map_info

# Створення зв'язків між пристроями
link1 = Link(router1, switch1, "Ethernet")
link2 = Link(switch1, firewall1, "Fiber")
link3 = Link(firewall1, server1, "Ethernet")
link4 = Link(router2, switch2, "Ethernet")
link5 = Link(switch2, firewall2, "Fiber")

# Створення карти мережі та додавання зв'язків
network_map = NetworkMap()
network_map.add_link(link1)

```

```

network_map.add_link(link2)
network_map.add_link(link3)
network_map.add_link(link4)
network_map.add_link(link5)

# Виведення карти мережі
print(network_map.display_network_map())

# Функція для обчислення навантаження на мережу
def calculate_network_load(device):
    # Симуляція навантаження на основі типу пристрою
    if device.device_type == "Router":
        return 70 # Відсоток завантаженості
    elif device.device_type == "Switch":
        return 40
    elif device.device_type == "Firewall":
        return 60
    elif device.device_type == "Server":
        return 85
    else:
        return 10

# Функція для перевірки доступності пристрою
def check_device_status(device):
    # Симуляція перевірки стану пристрою
    return True if device.ip_address else False

# Виведення навантаження для кожного пристрою в центрі обробки даних
for dc in topology.data_centers:
    for rack in dc.racks:
        for device in rack.devices:
            print(f"{device.name} Load: {calculate_network_load(device)}%")
            print(f"{device.name} Status: {'Online' if
check_device_status(device) else 'Offline'}")

# Додавання функціоналу для динамічної зміни IP-адреси пристрою
def update_ip_address(device, new_ip):
    device.ip_address = new_ip
    print(f"Updated {device.name} IP to {new_ip}")

# Зміна IP-адреси одного з пристроїв
update_ip_address(router1, "192.168.10.1")

# Виведення оновленої топології після зміни IP
print(topology.display_topology())

# Функція для моніторингу трафіку між пристроями
def monitor_traffic(link):
    # Симуляція моніторингу трафіку
    return f"Traffic on link {link.device1.name} <--> {link.device2.name}:
100Mbps"

# Моніторинг трафіку для кожного зв'язку в карті мережі
for link in network_map.links:
    print(monitor_traffic(link))

```

Файл vlan_backup.py

```
import json
import os

class VLAN:
    def __init__(self, vlan_id, name, devices):
        self.vlan_id = vlan_id
        self.name = name
        self.devices = devices

    def display_vlan_info(self):
        devices_info = ", ".join([device.name for device in self.devices])
        return f"VLAN ID: {self.vlan_id}, Name: {self.name}, Devices: {devices_info}"

class VLANManager:
    def __init__(self):
        self.vlans = []

    def create_vlan(self, vlan_id, name, devices):
        vlan = VLAN(vlan_id, name, devices)
        self.vlans.append(vlan)

    def display_all_vlans(self):
        return "\n".join([vlan.display_vlan_info() for vlan in self.vlans])

    def backup_vlan_config(self, file_path):
        vlan_config = [
            {
                "vlan_id": vlan.vlan_id,
                "name": vlan.name,
                "devices": [device.name for device in vlan.devices]
            } for vlan in self.vlans
        ]

        with open(file_path, 'w') as backup_file:
            json.dump(vlan_config, backup_file)
            print(f"Backup saved to {file_path}")

    def restore_vlan_config(self, file_path):
        if os.path.exists(file_path):
            with open(file_path, 'r') as backup_file:
                vlan_config = json.load(backup_file)
                self.vlans = [VLAN(cfg['vlan_id'], cfg['name'], []) for cfg in
                    vlan_config]
            print(f"VLAN configuration restored from {file_path}")
        else:
            print(f"Backup file {file_path} not found")

# Приклад використання
vlan_manager = VLANManager()
vlan_manager.create_vlan(10, "Management", [router1, switch1])
vlan_manager.create_vlan(20, "Production", [server1, firewall1])

# Виведення інформації про всі VLAN
print(vlan_manager.display_all_vlans())

# Створення резервної копії конфігурації VLAN
vlan_manager.backup_vlan_config("vlan_backup.json")

# Відновлення конфігурації VLAN з файлу
vlan_manager.restore_vlan_config("vlan_backup.json")
```

Файл performance_simulation.py

```
import random
import time

class DevicePerformanceMonitor:
    def __init__(self, device):
        self.device = device

    def monitor_cpu_usage(self):
        # Симуляція випадкової завантаженості процесора
        return random.randint(20, 90)

    def monitor_memory_usage(self):
        # Симуляція випадкової завантаженості пам'яті
        return random.randint(30, 95)

    def monitor_network_traffic(self):
        # Симуляція випадкового мережевого трафіку
        return random.randint(50, 500) # Мегабіти в секунду

    def display_performance(self):
        cpu_usage = self.monitor_cpu_usage()
        memory_usage = self.monitor_memory_usage()
        network_traffic = self.monitor_network_traffic()
        return (f"{self.device.name} Performance:\n"
                f"CPU Usage: {cpu_usage}%\n"
                f"Memory Usage: {memory_usage}%\n"
                f"Network Traffic: {network_traffic} Mbps")

class FirmwareUpdateSimulator:
    def __init__(self, device):
        self.device = device

    def simulate_update(self):
        print(f"Starting firmware update for {self.device.name}...")
        time.sleep(2) # Симуляція часу на оновлення
        success = random.choice([True, False])
        if success:
            print(f"Firmware update for {self.device.name} completed successfully.")
        else:
            print(f"Firmware update for {self.device.name} failed.")

# Приклад використання
performance_monitor = DevicePerformanceMonitor(router1)
print(performance_monitor.display_performance())

# Симуляція оновлення прошивки
update_simulator = FirmwareUpdateSimulator(router1)
update_simulator.simulate_update()
```

```
import requests

class MonitoringSystemIntegration:
    def __init__(self, system_url):
        self.system_url = system_url

    def send_device_status(self, device):
        data = {
            "name": device.name,
            "ip": device.ip_address,
            "status": "Online" if check_device_status(device) else "Offline"
        }
        try:
            response = requests.post(self.system_url, json=data)
            if response.status_code == 200:
                print(f"Status for {device.name} sent successfully to monitoring
system.")
            else:
                print(f"Failed to send status for {device.name}:
{response.status_code}")
        except requests.exceptions.RequestException as e:
            print(f"Error sending status: {e}")

class RoutingProtocol:
    def __init__(self, protocol_name):
        self.protocol_name = protocol_name
        self.routes = {}

    def add_route(self, destination, next_hop):
        self.routes[destination] = next_hop

    def display_routes(self):
        routes_info = f"{self.protocol_name} Routing Table:\n"
        for destination, next_hop in self.routes.items():
            routes_info += f"Destination: {destination}, Next Hop: {next_hop}\n"
        return routes_info

# Приклад використання інтеграції з системою моніторингу
monitoring_integration = MonitoringSystemIntegration("http://monitoring-
system.local/api")
monitoring_integration.send_device_status(router1)

# Створення та налаштування таблиці маршрутів
ospf = RoutingProtocol("OSPF")
ospf.add_route("192.168.2.0/24", "192.168.1.1")
ospf.add_route("10.0.0.0/8", "192.168.1.2")
print(ospf.display_routes())
```

Файл distributed_firewall.py

```
class FirewallRule:
    def __init__(self, rule_id, source_ip, destination_ip, action):
        self.rule_id = rule_id
        self.source_ip = source_ip
        self.destination_ip = destination_ip
        self.action = action # "ALLOW" або "DENY"

    def display_rule_info(self):
        return (f"Rule ID: {self.rule_id}, Source: {self.source_ip}, "
                f"Destination: {self.destination_ip}, Action: {self.action}")

class DistributedFirewall:
    def __init__(self):
        self.rules = []

    def add_rule(self, rule_id, source_ip, destination_ip, action):
        rule = FirewallRule(rule_id, source_ip, destination_ip, action)
        self.rules.append(rule)

    def display_firewall_rules(self):
        return "\n".join([rule.display_rule_info() for rule in self.rules])

    def apply_rules(self):
        # Симуляція застосування правил брандмауера
        for rule in self.rules:
            print(f"Applying rule: {rule.display_rule_info()}")
        print("All rules applied.")

# Приклад використання розподіленого брандмауера
firewall = DistributedFirewall()
firewall.add_rule(1, "192.168.1.10", "192.168.1.20", "ALLOW")
firewall.add_rule(2, "192.168.1.15", "10.0.0.5", "DENY")
# Виведення правил брандмауера та їх застосування
print(firewall.display_firewall_rules())
firewall.apply_rules()
```

Файл traffic_analysis_qos.py

```

import random

class TrafficAnalyzer:
    def __init__(self):
        self.traffic_data = {}

    def generate_traffic_data(self, link):
        # Генерація випадкових даних про трафік для кожного зв'язку
        self.traffic_data[link] = random.randint(50, 1000) # трафік в Mbps

    def analyze_traffic(self):
        analysis_report = "Traffic Analysis Report:\n"
        for link, traffic in self.traffic_data.items():
            analysis_report += f"Link {link.device1.name} <-->
{link.device2.name}: {traffic} Mbps\n"
            if traffic > 800:
                analysis_report += "Warning: High traffic detected!\n"
        return analysis_report

class QoSManager:
    def __init__(self):
        self.qos_rules = {}

    def add_qos_rule(self, device, priority_level):
        # Призначення рівня пріоритету для пристрою (1-найвищий, 5-найнижчий)
        self.qos_rules[device] = priority_level

    def apply_qos(self):
        for device, priority in self.qos_rules.items():
            print(f"Applying QoS: Device {device.name}, Priority Level:
{priority}")

# Приклад використання
traffic_analyzer = TrafficAnalyzer()
# Генерація трафіку для кожного зв'язку
traffic_analyzer.generate_traffic_data(link1)
traffic_analyzer.generate_traffic_data(link2)

# Аналіз трафіку
print(traffic_analyzer.analyze_traffic())

# Динамічне управління смугою пропускання (QoS)
qos_manager = QoSManager()
qos_manager.add_qos_rule(router1, 1)
qos_manager.add_qos_rule(switch1, 2)
qos_manager.apply_qos()

```

Файл disaster_simulation.py

```
import random

class DisasterScenarioSimulator:
    def __init__(self, topology):
        self.topology = topology

    def simulate_device_failure(self):
        # Вибір випадкового пристрою з топології
        all_devices = []
        for data_center in self.topology.data_centers:
            for rack in data_center.racks:
                all_devices.extend(rack.devices)

        if all_devices:
            failed_device = random.choice(all_devices)
            print(f"Simulating failure of device: {failed_device.name}")
            return failed_device
        else:
            print("No devices available for simulation.")
            return None

    def simulate_link_failure(self, network_map):
        if network_map.links:
            failed_link = random.choice(network_map.links)
            print(f"Simulating failure of link: {failed_link.device1.name} <-->
{failed_link.device2.name}")
            return failed_link
        else:
            print("No links available for simulation.")
            return None

# Приклад використання
disaster_simulator = DisasterScenarioSimulator(topology)

# Моделювання відмови пристрою
failed_device = disaster_simulator.simulate_device_failure()

# Моделювання відмови зв'язку
failed_link = disaster_simulator.simulate_link_failure(network_map)
```

Файл encryption.py

```

from cryptography.fernet import Fernet

class EncryptionManager:
    def __init__(self):
        # Генерація ключа для шифрування
        self.key = Fernet.generate_key()
        self.cipher = Fernet(self.key)

    def encrypt_data(self, data):
        # Шифрування даних
        encrypted_data = self.cipher.encrypt(data.encode())
        return encrypted_data

    def decrypt_data(self, encrypted_data):
        # Розшифрування даних
        decrypted_data = self.cipher.decrypt(encrypted_data).decode()
        return decrypted_data

class SecureLink:
    def __init__(self, device1, device2, encryption_manager):
        self.device1 = device1
        self.device2 = device2
        self.encryption_manager = encryption_manager

    def send_encrypted_message(self, message):
        encrypted_message = self.encryption_manager.encrypt_data(message)
        print(f"Encrypted message sent from {self.device1.name} to {self.device2.name}: {encrypted_message}")
        return encrypted_message

    def receive_encrypted_message(self, encrypted_message):
        decrypted_message = self.encryption_manager.decrypt_data(encrypted_message)
        print(f"Decrypted message received by {self.device2.name}: {decrypted_message}")

# Приклад використання шифрування
encryption_manager = EncryptionManager()

# Створення захищеного зв'язку
secure_link = SecureLink(router1, switch1, encryption_manager)

# Відправка та отримання зашифрованого повідомлення
encrypted_msg = secure_link.send_encrypted_message("Test message")
secure_link.receive_encrypted_message(encrypted_msg)

```

Файл device_discovery.py

```
class DeviceDiscovery:
    def __init__(self):
        self.discovered_devices = []

    def discover_device(self, ip_address, mac_address, device_type):
        # Додавання нового пристрою
        new_device = NetworkDevice(f"Discovered-{len(self.discovered_devices) +
1}", device_type, ip_address, mac_address)
        self.discovered_devices.append(new_device)
        print(f"Discovered new device: {new_device.name}, IP:
{new_device.ip_address}, MAC: {new_device.mac_address}")

    def display_discovered_devices(self):
        for device in self.discovered_devices:
            print(f"Device: {device.name}, IP: {device.ip_address}, MAC:
{device.mac_address}")

# Приклад використання
device_discovery = DeviceDiscovery()
device_discovery.discover_device("192.168.2.5", "00:1C:2D:3E:4F:6A", "Switch")
device_discovery.discover_device("192.168.2.6", "00:1C:2D:3E:4F:6B", "Router")

# Виведення інформації про знайдені пристрої
device_discovery.display_discovered_devices()
```

Файл scalability.py

```
class NetworkScaler:
    def __init__(self):
        self.devices = []

    def add_device(self, device):
        self.devices.append(device)

    def scale_network(self, new_device_count):
        print(f"Scaling network by adding {new_device_count} new devices...")
        for i in range(new_device_count):
            new_device = NetworkDevice(f"Device-{len(self.devices) + 1}",
                "Router", f"10.0.0.{len(self.devices) + 1}", f"00:AA:BB:CC:{len(self.devices) +
                1:02X}")
            self.add_device(new_device)

    def display_scaled_network(self):
        for device in self.devices:
            print(f"Device: {device.name}, IP: {device.ip_address}, MAC:
                {device.mac_address}")

# Приклад масштабування мережі
network_scaler = NetworkScaler()
network_scaler.add_device(router1)
network_scaler.add_device(switch1)

# Масштабування мережі
network_scaler.scale_network(5)

# Виведення інформації про масштабовану мережу
network_scaler.display_scaled_network()
```

Файл error_analysis.py

```
class ErrorAnalyzer:
    def __init__(self):
        self.errors = []

    def log_error(self, device, error_message):
        error_entry = {
            "device": device.name,
            "error": error_message,
            "status": "unresolved"
        }
        self.errors.append(error_entry)

    def resolve_error(self, error_index):
        if 0 <= error_index < len(self.errors):
            self.errors[error_index]["status"] = "resolved"
            print(f"Resolved error for {self.errors[error_index]['device']}")
        else:
            print("Error index out of range.")

    def display_errors(self):
        for error in self.errors:
            print(f"Device: {error['device']}, Error: {error['error']}, Status: {error['status']}")

# Приклад використання
error_analyzer = ErrorAnalyzer()
error_analyzer.log_error(router1, "Packet loss detected")
error_analyzer.log_error(switch1, "High latency")

# Виведення інформації про помилки
error_analyzer.display_errors()

# Вирішення однієї з помилок
error_analyzer.resolve_error(0)

# Виведення оновленої інформації про помилки
error_analyzer.display_errors()
```

Файл cloud_integration.py

```
class CloudDataCenter:
    def __init__(self, cloud_provider, region):
        self.cloud_provider = cloud_provider
        self.region = region
        self.resources = []

    def add_resource(self, resource_name, resource_type):
        self.resources.append({"name": resource_name, "type": resource_type})
        print(f"Added {resource_type} {resource_name} to {self.cloud_provider}
in {self.region}")

    def display_cloud_resources(self):
        for resource in self.resources:
            print(f"Resource: {resource['name']}, Type: {resource['type']}")

# Приклад використання інтеграції з хмарою
cloud_dc = CloudDataCenter("AWS", "eu-central-1")
cloud_dc.add_resource("Virtual Router", "Router")
cloud_dc.add_resource("Virtual Server", "Server")

# Виведення ресурсів хмарного дата-центру
cloud_dc.display_cloud_resources()
```

КБПЗ_2024