

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 віртуалізації**  
**мережі Nuage Networks для автоматизації SDN і SD-WAN”**

Виконав здобувач вищої освіти  
II курсу, групи КІ-23М  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Бондаренко О.Д.  
« \_\_\_\_ » \_\_\_\_\_ 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 Олегу Дмитровичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN*

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 віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN*

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

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

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

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

## 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 О.Д. Дослідження та програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN. 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 віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

Метою розробки є дослідження та програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

Об'єктом дослідження є процес віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

Предметом дослідження є методи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

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

Результат роботи – програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

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, Nuage Networks, SDN, SD-WAN

## ABSTRACT

**Bondarenko O.D. Research and software implementation of the Nuage Networks network virtualization system for SDN and SD-WAN automation. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the Nuage Networks network virtualization system for SDN and SD-WAN automation.

The goal of the development is the research and software implementation of the Nuage Networks network virtualization system for SDN and SD-WAN automation.

The object of the study is the virtualization process of Nuage Networks for SDN and SD-WAN automation.

The subject of research is Nuage Networks network virtualization methods for SDN and SD-WAN automation.

Research methods are based on methods of virtualization of computer networks, methods of mathematical statistics, methods of software development.

The result of the work is a software implementation of the Nuage Networks network virtualization system for SDN and SD-WAN automation.

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, Nuage Networks, SDN, SD-WAN

## 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.....	10
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	12
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	12
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	25
2.3 Розгорнута постановка завдання .....	28
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	30
3.1 Опис функціонування системи .....	30
3.2 Розробка структурної схеми.....	40
3.3 Розробка функціональної схеми .....	48
3.4 Розробка діаграми процесів.....	56
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	58
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	58
4.2 Захист розробленого програмного забезпечення.....	72
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	76
6 НАУКОВА НОВИЗНА .....	82

					ВКРМ-123.24.0005.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN	Літ.	Аркуш	Аркушів
Розроб.	Бондаренко О.Д.					М	1	110
Перев.	Петренюк В.І.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-23М		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ .....	83
7.1	Визначення цільової аудиторії кінцевого готового продукту .....	83
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	83
7.3	Вибір методу оцінки вартості ПЗ .....	85
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	86
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ .....	87
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ .....	89
7.7	Визначення ключових факторів успіху конкретного проєкту.....	91
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	92
8.1	Вступ.....	92
8.2	Шкідливі і небезпечні фактори при роботі з комп'ютером.....	93
8.3	Аналіз умов праці на робочому місці фахівця .....	94
8.4	Розробка заходів з поліпшення охорони праці .....	98
8.5	Розрахункова частина .....	99
9	ОСНОВНІ ВИСНОВКИ.....	102
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	104

КБПЗ-2024

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>2</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- IT – інформаційні технології
- ПЗ – програмне забезпечення
- QoS – механізми контролю й керування якістю

КБПЗ\_2024

					ВКРМ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** Питання побудови й експлуатації сучасної віртуалізованої мережі є найважливішими для фахівців в області центрів обробки даних. Значна увага приділена актуальним тенденціям галузі: віртуалізації й програмно визначаємих мережам (SDN), продуктам для побудови мережної інфраструктури центрів обробки даних – LAN і SAN, а також організації комунікацій між віддаленими площадками й створенню розподілених центрів даних.

Віртуалізація серверів дозволяє декільком серверам працювати одночасно на єдиному фізичному вузлі; при цьому зазначені сервери залишаються взаємно ізольованими. По суті, кожна віртуальна машина працює як єдиний сервер на даному фізичному комп'ютері. Віртуалізація мереж забезпечує аналогічну можливість, при якій кілька віртуальних мережних інфраструктур функціонують в одній фізичній мережі (з потенційним перекриттям IP-адрес), і кожна віртуальна мережна інфраструктура працює як єдина віртуальна мережа в загальній мережній інфраструктурі.

У термінах віртуалізації мережі (HNV) клієнт визначається як "власник" групи віртуальних машин, розгорнутих у центрі обробки даних. Клієнтом може бути корпорація або підприємство в багатоклієнтському суспільному центрі реєстрації й обробки даних, або підрозділ або філія компанії в приватному центрі реєстрації й обробки даних. Кожний клієнт може мати в центрі обробки даних одну або кілька мереж віртуальних машин, а кожна мережа віртуальних машин містить у собі одну або декілька віртуальних підмереж.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

– Огляд існуючих систем віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

– Дослідження системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

– Програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

*Об'єктом дослідження є процес віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.*

*Предметом дослідження є методи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.*

*Методи дослідження базуються на методах віртуалізації 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 віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

– Розроблено вітчизняний продукт віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

**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.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	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 віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ\_2024

					ВКРМ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

### Мережа віртуальних машин

Кожна мережа віртуальних машин складається з однієї або більше віртуальних підмереж. Мережа віртуальних машин формує ізоляційні границі, у межах яких віртуальні машини мережі можуть зв'язуватися один з одним. У результаті віртуальні підмережі однієї мережі віртуальних машин не повинні використовувати префікси, що перекриваються, IP-адрес.

У кожній мережі віртуальних машин є домен маршрутизації, що ідентифікує дану мережу. ID домену маршрутизації (RDID), що ідентифікує мережу віртуальних машин, призначається адміністраторами центра обробки даних або керуючим ПЗ центра обробки даних, наприклад System Center 2012 R2 Virtual Machine Manager (VMM) системні центри. RDID має формат GUID Windows – наприклад "{ 3333-4444-000000000000}".

### Віртуальні підмережі

Віртуальна підмережа реалізує семантику підмережі IP рівня 3 для віртуальних машин однієї віртуальної підмережі. Віртуальна підмережа є доменом розсилання (аналогічно віртуальної локальної мережі (VLAN)). Віртуальні машини в одній віртуальній підмережі повинні використовувати однаковий префікс IP-адреси.

Кожна віртуальна підмережа належить до однієї мережі віртуальних машин (RDID), і їй призначається унікальний ID віртуальної підмережі (VSID). VSID повинен бути унікальний у рамках центра обробки даних. Його значення може перебувати в діапазоні від 4096 до  $2^{24-2}$ .

Ключова перевага використання мережі віртуальних машин і домену маршрутизації – те, що це дозволяє клієнтам перенести топологію своїх мереж у

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

хмару. Оскільки в даних мереж різні ID доменів маршрутизації, вони не можуть взаємодіяти один з одним. Таким чином, мережа R&D компанії А ізольована від мережі Sales тої ж компанії. Мережа R&D компанії А містить три віртуальні підмережі. Відзначимо, що як RDID, так і VSID у центрі реєстрації й обробки даних унікальні.

Пакети віртуальних машин з VSID 5001 можуть направлятися або перенаправлятися із застосуванням віртуалізації мережі на віртуальні машини з VSID 5002 або VSID 5003. Перед доставкою пакета на комутатор віртуалізації мережі обновляє VSID вхідного пакета, перетворюючи його в VSID кінцевої віртуальної машини. Винятковою умовою для цього є знаходження обох VSID у межах одного RDID. Якщо VSID, пов'язаний з пакетом, не відповідає VSID віртуальної машини призначення, пакет відкидається. Таким чином, віртуальні мережні адаптери з RDID1 не можуть відправляти пакети на віртуальні мережні адаптери з RDID2.

В описі потоку пакетів, наведеному вище, термін "віртуальна машина" фактично означає "віртуальний мережний адаптер" віртуальної машини. Як правило, віртуальна машина має лише один віртуальний мережний адаптер. У цьому випадку поняття віртуальної машини й віртуального мережного адаптера можуть уважатися взаємозамінними. Оскільки віртуальна машина може мати кілька віртуальних мережних адаптерів і ці віртуальні мережні адаптери можуть мати різні ID віртуальної підмережі (VSID) і ID домену маршрутизації (RDID), у віртуалізації мережі особлива увага приділяється відправленню й одержанню пакетів між віртуальними мережними адаптерами.

Кожна віртуальна підмережа визначає підмережа IP рівня 3 і границі домену розсилання рівня 2 (P2) аналогічно VLAN. При передачі пакета віртуальною машиною дане розсилання обмежене віртуальними машинами, підключеними до портів комутатора з однаковою VSID. Кожний VSID може бути пов'язаний з адресою багатоадресного розсилання в адреси постачальника (АП). Весь трафік розсилання для певного VSID відправляється на дану адресу багатоадресного розсилання.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Віртуалізація мережі НЕ залежить від розсилання або багатоадресного розсилання. Для широкомовних або багатоадресних пакетів у мережі віртуальних машин використовується IP-адреса багатоадресного розсилання в просторі адрес постачальника, якщо він настроєний. Проте багато операторів центрів обробки даних не використовують багатоадресні розсилання у своїх середовищах. У результаті, якщо адреса багатоадресного розсилання в просторі адрес постачальника недоступний, використовується інтелектуальна одноадресна реплікація в просторі адрес постачальника. Це значить, що пакети направляються в режимі одноадресного розсилання тільки на адреси постачальника, настроєні для конкретної віртуальної підмережі, у якій перебуває пакет. Крім того, для кожного вузла відправляється тільки один одноадресний пакет – незалежно від числа віртуальних машин на вузлі.

Крім подання домену розсилання, VSID забезпечує ізоляцію. Віртуальний мережний адаптер у віртуалізації мережі підключається до порту комутатора, що має ACL VSID. При надходженні на даний порт комутатора пакета з невідповідним VSID такий пакет відкидається. Пакети доставляються на порт комутатора тільки при відповідності VSID пакета й VSID порту комутатора. От чому в наведеному вище прикладі на рисунку 2 пакети, що надходять від VSID 5001 на 5003, повинні були містити VSID, змінений перед відправленням на кінцеву віртуальну машину.

Якщо порт комутатора не має ACL VSID, віртуальний мережний адаптер, підключений до такого порту комутатора, не включається у віртуальну підмережу віртуалізації мережі. Пакети, що відправляються з віртуального мережного адаптера, що не має ACL VSID, проходять через комутатор без зміни.

При відправленні пакета віртуальною машиною VSID порту комутатора зв'язується з даним пакетом. На приймаючій стороні механізм віртуалізації мережі доставляє на комутатор VSID у значенні OOB разом з декапсульованим пакетом. На приймаючій стороні механізм віртуалізації мережі виконує пошук політики й додає VSID у значення OOB перед передачею пакета на комутатор.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Розширення комутатора можуть працювати як у просторі адреси постачальника (АП), так і в просторі адреси клієнта (АК). Це значить, що VSID доступно розширенням комутатора. Це дозволяє розширенню комутатора користуватися підтримкою архітектури обслуговування одним екземпляром додатка декількох розгортань. Наприклад, розширення комутатора брандмауера може відрізнити IP-адреса 10.1.1.5 типу АК зі значенням ЗПД, що містить VSID 5001, від такого ж IP-адреси типу АК, що містить VSID 6001.

## 1.2 Область застосування

Областю застосування є центри обробки даних. Центр обробки даних (ЦОД) – це відказостійка комплексна централізована система, що забезпечує автоматизацію бізнес-процесів з високим рівнем продуктивності і якістю надаваних сервісів. Призначення ЦОД – забезпечення гарантованої безвідмовної роботи інформаційної системи підприємства із заданими рівнями доступності, надійності, безпеці й керованості. Використання технології створення центрів обробки даних дозволяє створювати резервні штаб-квартири підприємств зі збереженням максимально можливої функціональності інформаційної системи при надзвичайних обставинах.

Центри обробки даних включають:

- високонадійне серверне встаткування;
- системи зберігання й передачі даних;
- програмне забезпечення;
- архітектурно-технічні рішення;
- інженерну інфраструктуру, що забезпечує;
- фізичний захист приміщень;
- комплекс організаційних заходів;
- систему моніторингу й керування.

Переваги створення ЦОД:

- підвищення ефективності й надійності експлуатації обчислювальних ресурсів;

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>10</b>

- надання відказостійких інфраструктурних сервісів у режимі 24 години x 7 днів у тиждень x 365 днів у році;
- просте й прозоре централізоване адміністрування;
- зниження витрат на надання інженерних комунікацій;
- високий рівень захисту інформаційної системи;
- централізоване керування й облік ресурсів ЦОД;
- контроль доступу до ЦОД;
- просте й зручне масштабування обчислювальних ресурсів.

У рамках рішення забезпечується комплексна безпека центрів обробки даних, що включає захист від наступних погроз:

- відмова встаткування й програмного забезпечення;
- збої енергоживлення;
- пожежа й задимлення;
- несанкціонований доступ, злом, крадіжки;
- віруси;
- затоплення, різкі температурні зміни, пил;
- часткове руйнування будинку;
- електромагнітні випромінювання.

Проектування ЦОД виконується з обліком розв'язуваних бізнес завдань, рівня вимог до безпеки, побажань замовника, використання вже наявного встаткування, і знаходить втілення в архітектурно-технічних рішеннях проекту. Цей підхід дозволяє створювати захищені гетерогенні центри обробки даних, що складаються з устаткування й програмного забезпечення різних виробників, включаючи наслідувані системи замовника.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

## 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**

### **Sandboxie**

Домашня сторінка: <http://www.sandboxie.com/>

Розроблювач Ronen Tzur порівнює дію програми Sandboxie з невидимим шаром, нанесеним поверх паперу: на нього можна наносити будь-які написи; при знятті захисту, аркуш залишиться недоторканим.

Можна виділити 4 основних способи застосування пісочниць в Sandboxie:

- Захищений інтернет-серфінг.
- Поліпшення приватності.
- Безпечна email-переписка.
- Збереження ОС у первісному стані.

Останній пункт має на увазі, що в пісочниці можна встановлювати й запускати будь-які клієнтські додатки – браузері, ІМ-месенджери, гри – без впливу на систему. Sandboxie контролює доступ до файлів, дисковим пристроям, ключам реєстру, процесам, драйверам, портам і іншим потенційно незахищеним джерелам.

Насамперед, SandboxIE корисна тим, що дозволяє користувачеві гнучко набудовувати пісочниці й привілеї за допомогою оболонки Sandboxie Control.

Тут, через контекстне й головного меню, доступні основні операції:

- Запуск і зупинка програм під контролем Sandboxie.
- Перегляд файлів усередині пісочниці.
- Відновлення потрібних файлів з пісочниці.
- Видалення всіх результатів роботи або вибіркового файлів.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12



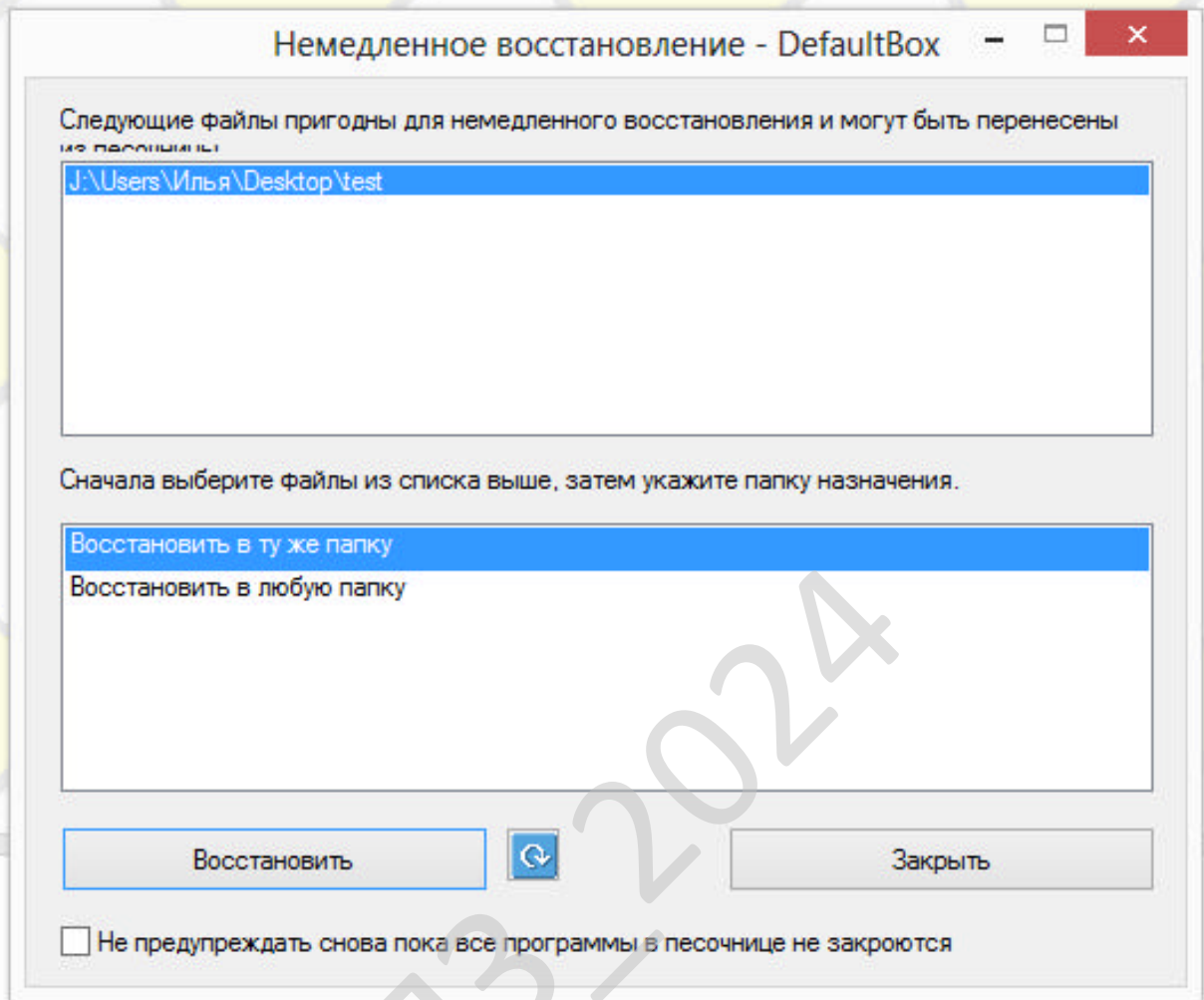


Рисунок 2.2 – Негайне відновлення файлів

Більш детальні налаштування доступу розташовані в розділах «Обмеження» і «Доступ до ресурсів». Вони можуть знадобитися в тому випадку, якщо додаток не може працювати без певних привілеїв (потрібно певна системна бібліотека, драйвер або т.п.). В «Обмеженнях», стосовно до програм або груп, настраюється доступ в Інтернет, до апаратних засобів, ІРС-Об'єктам, а також доступ низького рівня. В «Доступі до ресурсів» – відповідні налаштування для файлів, директорій, до реєстру й прочимой системним ресурсам.

Також у налаштуваннях Sandboxie перебуває важливий розділ «Додатка», де зібрані групи програм, для яких наданий доступ до зазначених ресурсів.

Споконвічно всі елементи списку деактивовані, для застосування змін для конкретного додатка потрібно відзначити його в списку й натиснути кнопку «Додати».

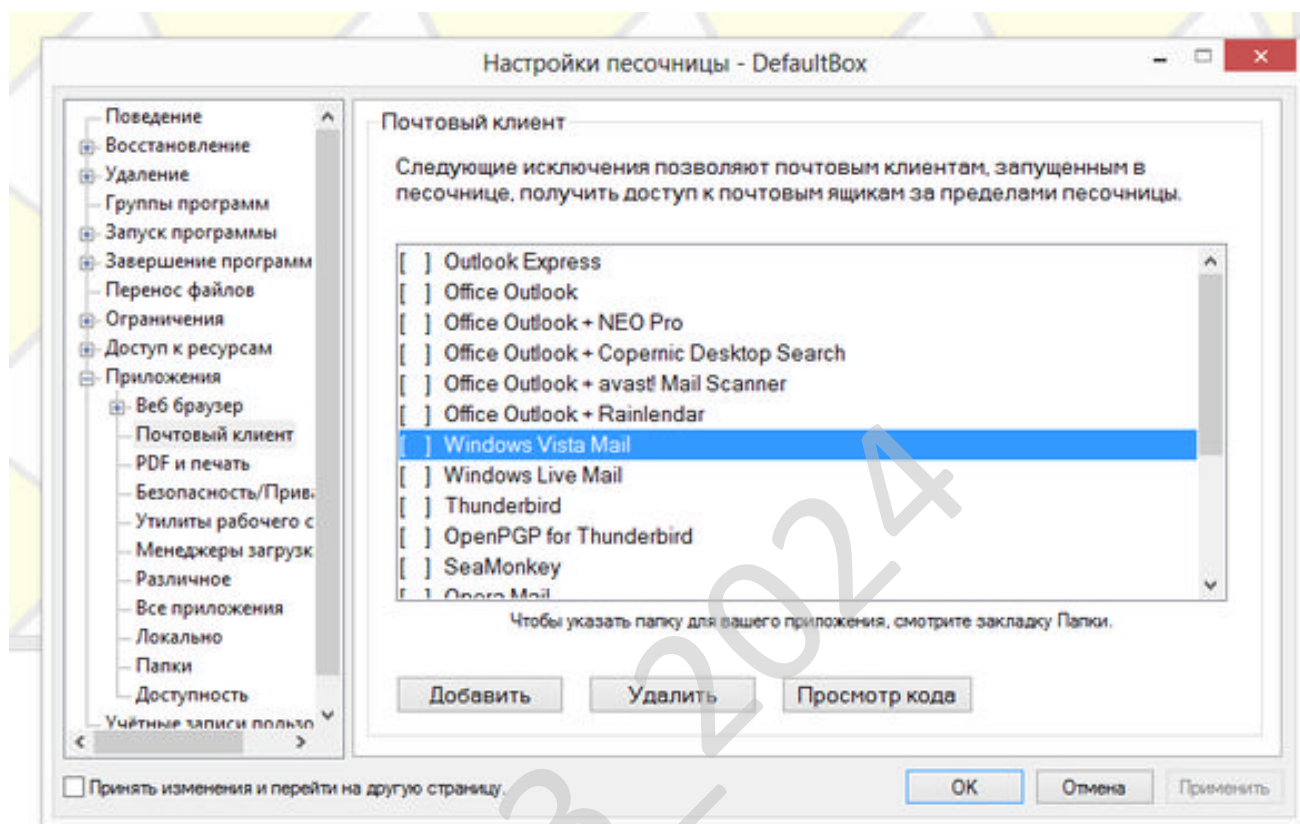


Рисунок 2.3 – Налаштування доступу додатків

Таким чином, можна створювати пісочниці з різними параметрами. Дозволяється клонувати конфігурацію вже наявної пісочниці, для цього, при створенні нової, зі списку, що випадає, потрібно вибрати те середовище, з якого потрібно перенести налаштування.

### Резюме

За допомогою додатка Sandboxie можна створювати віртуальні середовища будь-яких конфігурацій, без обмежень для користувача. Sandboxie надає велика кількість налаштувань як для окремих додатків, так і для пісочниць.

[+] Гнучке налаштування кожної пісочниці.

[+] Створення правил для групи програм.

[-] Не можна створювати дистрибутиви.

[-] Відсутність майстра налаштування.

### **Evalaze**

Домашня сторінка: <http://www.evalaze.de/en/evalaze-oxid/>

Символічно, що Evalaze бере свій початок від програми Thinstall 2007, на даний момент приналежної компанії VMware.

Evalaze не настільки відома, як Sandboxie, серед програм для роботи з пісочницями, однак має ряд цікавих особливостей, що виділяє її з ряду подібних рішень. Завдяки віртуалізації, додатка можна запускати в автономному середовищі з будь-якого комп'ютера, поза залежністю від наявності драйверів, бібліотек, більше нових версій додатка, що запускається. При цьому не потрібно ні попереднє налаштування, ні додаткові конфігураційні файли або бібліотеки або ключі реєстру.

Evalaze не вимагає установки, один нюанс: для роботи знадобиться Microsoft .NET Framework версії 2.0 або вище. У безкоштовній версії, так само як і в професійній редакції, доступний майстер налаштування віртуалізації й необмежена кількість віртуальних додатків. Скачати trial-версію із сайту розроблювачів можна тільки за запитом (email розроблювачів див. на сайті).

На сторінці Compatibility опублікований список додатків, протестованих на сумісність із різними ОС і версіями Evalaze. Наприклад, такі програми, як Adobe Reader, AutoCAD, Google Earth, Outlook, Skype не можна віртуалізувати, у той же час, як більшість додатків з великого списку повноцінно працюють із Evalaze.

У головному вікні Evalaze доступний майстер створення проекту (Start Wizard). На першому етапі вказується ім'я й розташування файлів для роботи в пісочниці. Перед установкою нового додатка Evalaze створює знімок файлової системи й реєстру. В PRO-версії можна вибрати глибину сканування, у безкоштовній даній опції немає.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16



Отримана конфігурація може бути збережена в проект. Від початку й до кінця процес налаштування віртуального додатка займає більше часу, чим, скажемо, в Sandboxie, однак він більше послідовний і зрозумілий.

Слід зазначити дві додаткові можливості Evalaze, які, імовірно, зацікавлять розроблювачів ПЗ, тестувальників: це робота з віртуальною файловою системою й віртуальним реєстром. Дані автономні середовища Evalaze можна редагувати на свій розсуд, додаючи файли, директорії, ключі, необхідні для функціонування тої або іншої віртуальної програми.

Також в Evalaze можна набувати асоціації «з коробки»: віртуальний додаток при запуску відразу створить необхідні асоціації з файлами в ОС.

### **Резюме**

Програма, за допомогою якої можна створювати автономні додатки, які зручно використовувати у всіляких ситуаціях, що в цілому полегшує міграцію, сумісність, безпеку. На жаль, безкоштовна версія практично марна, вона цікава тільки для дуже поверхневого вивчення функцій Evalaze.

[–] Малофункціональна ознайомлювальна версія.

[–] Висока ціна Pro-версії.

[+] Є присутнім майстер налаштування.

[+] Віртуальні файлова система й реєстр

### **Enigma Virtual Box**

Домашня сторінка: <http://www.enigmaprotector.com/en/downloads.html>

Програма Enigma Virtual Box призначена для запуску додатків в ізольованому віртуальному середовищі. Список підтримуваних форматів містить у собі dll, осх (бібліотеки), avi, mp3 (мультимедіа), txt, doc (документи) і ін.

Enigma Virtual Box моделює віртуальне середовище навколо додатка в такий спосіб. Перед запуском додатка спрацьовує завантажник Virtual Box, що зчитує інформацію, що необхідна для роботи програми: бібліотеки й інших компонентів – і надає їхньому додатку замість системних. У результаті програма працює автономно стосовно ОС.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

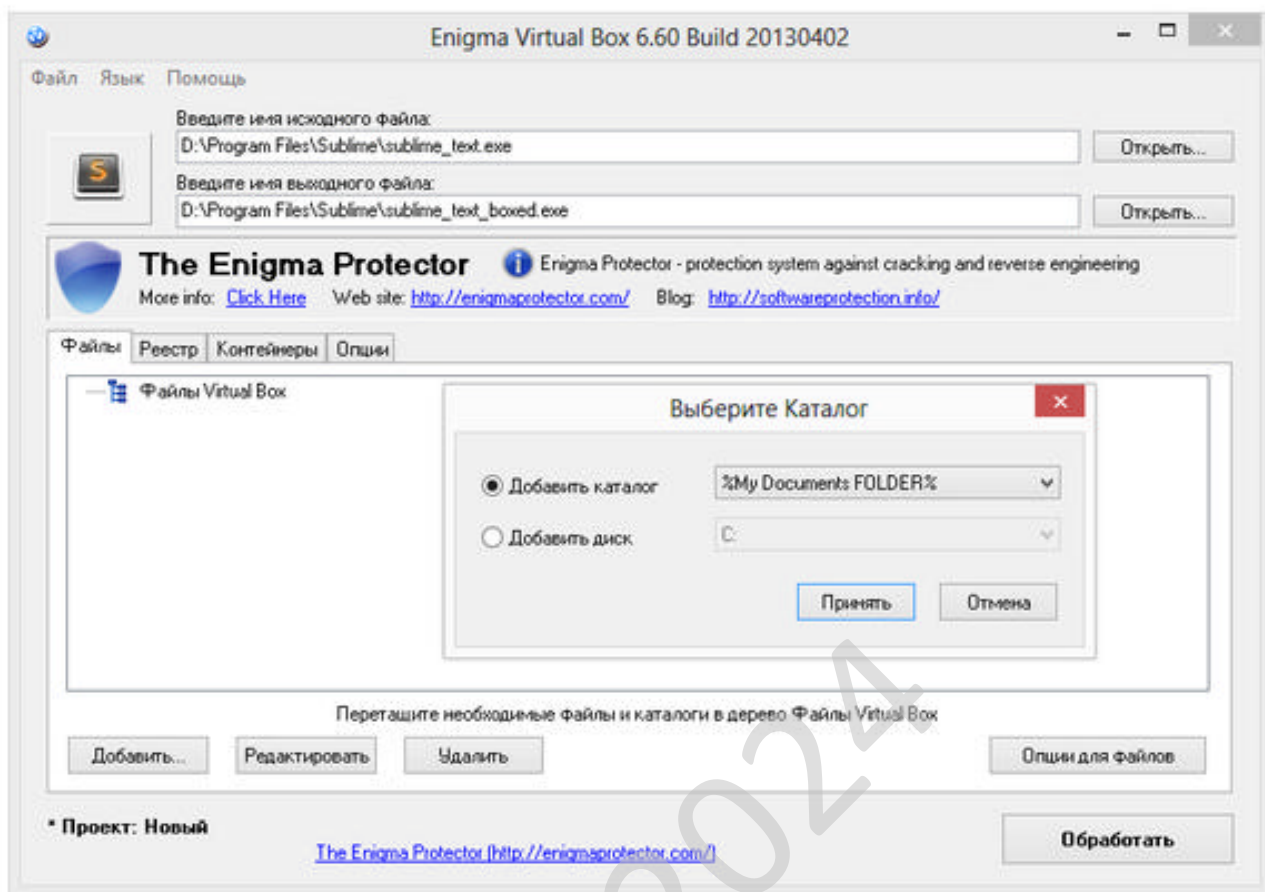


Рисунок 2.5 – Налаштування віртуального додатка в Enigma Virtual Box

На конфігурацію пісочниць Sandboxie або Evalaze, як правило, іде хвилин 5. На перший погляд, в Virtual Box також не передбачається тривале налаштування. У документації використання програми вміщається фактично в одна пропозиція.

Усього 4 вкладки – «Файли», «Реєстр», «Контейнери» і, властиво, «Опції». Потрібно вибрати здійснений файл, указати розташування кінцевого результату й запустити обробку. Але згодом виявляється, що віртуальне середовище потрібно створювати самостійно. Для цього й призначені три поруч ідуть роздягнула «Файли», «Реєстр» і «Контейнери», де вручну додаються потрібні дані. Після чого можна нажати обробку, запустити вихідний файл і перевірити працездатність програми.





Рисунок 2.6 – Бібліотека додатків Cameyo

Ще одна цікава можливість – створення віртуального додатка через веб-інтерфейс. Установник можна завантажити з комп'ютера або вказати URL файлу.

Процес конвертації, по заявах, займає від 10 до 20 хвилин, але частий час очікування менше в кілька разів. По закінченні, на email приходить повідомлення з посиланням на опублікований пакет.

При всіх хмарних зручностях потрібно відзначити два важливих моменти. Перший: кожна програма час вчасно обновляється, а в бібліотеці присутні досить застарілі екземпляри. Другий аспект: додатка, додані користувачами, може йти врозріз із ліцензією окремо взятої програми. Необхідно це розуміти й урахувати при створенні користувальницьких дистрибутивів. І третє – ніхто не

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21



[+] Мережна синхронізація.

[+] Доступ до користувальницьких додатків.

[+] Створення віртуальних додатків онлайн.

[-] Відсутність налаштування пісочниць

### **Spoon.net**

Домашня сторінка: <http://www.spoon.net/>

Spoon Tools – це комплекс інструментів для створення віртуальних додатків. Крім професійного середовища Spoon Studio, spoon.net заслуговує на увагу як хмарний сервіс, що інтегрується з Робочим столом, дозволяючи швидко створювати пісочниці.

Для інтеграції з Робочим столом необхідно зареєструватися на сервері spoon.net і встановити спеціальний віджет. Після реєстрації користувач одержує можливість завантажувати із сервера віртуальні додатки через зручну оболонку.

Чотири можливості, що привносяться віджетом:

– Створення пісочниць для файлів і додатків.

– Наведення порядку на Робочому столі за допомогою ярликів, меню швидкого запуску.

– Безпечне тестування нових додатків, запуск застарілих версій поверх нових.

– Скасування змін, зроблених пісочницею.

Швидкий доступ до віджету spoon.net можливий за допомогою сполучення клавіш Alt + Win. Оболонка містить у собі рядок пошуку, по сумісництву – консоль. У ній виробляється пошук додатків на комп'ютері й на веб-сервісі.

Дуже зручна організація десктопа: на віртуальний Робочий стіл можна перетягнути потрібні файли, які будуть синхронізуватися з spoon.net. Нові пісочниці можна створювати буквально двома кліками.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

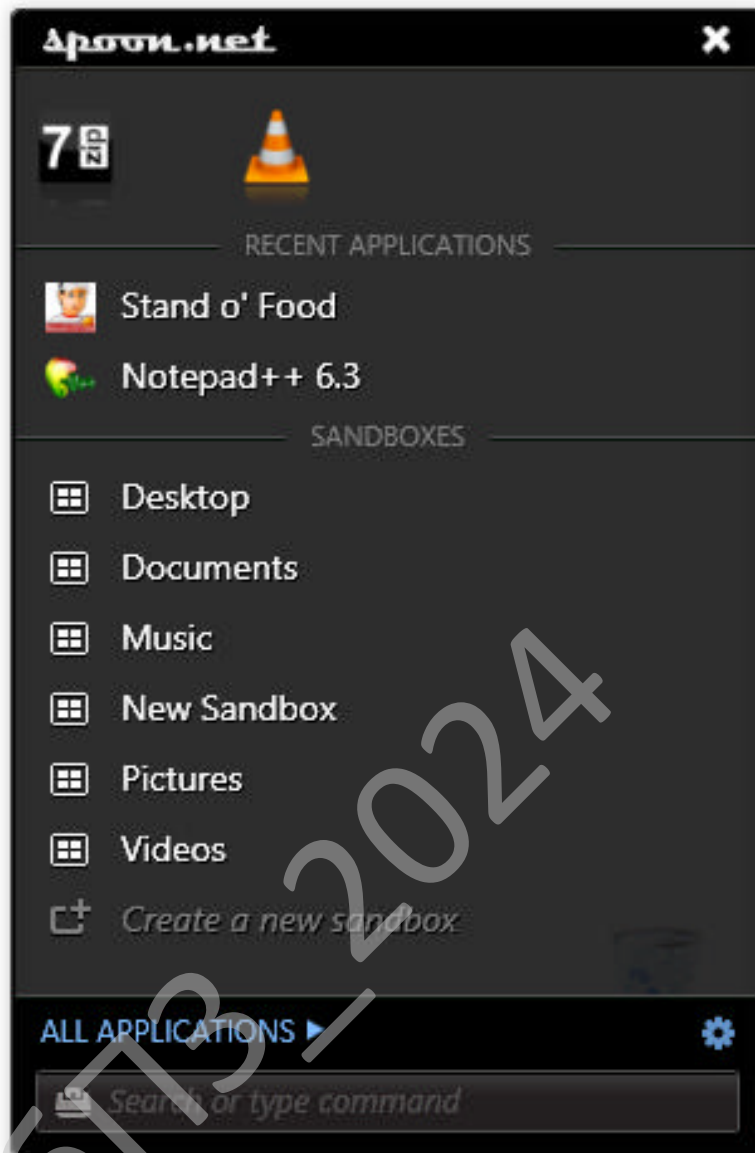


Рисунок 2.8 – Консоль і рядок пошуку spool.net

Безумовно, по частині налаштування пісочниць Spoon не може скласти конкуренцію Sandboxie або Evalaze з тієї причини, що в Spoon вони попросту відсутні. Не можна встановлювати обмеження, конвертувати «звичайне» додаток у віртуальне. Для цих цілей призначений комплекс Spoon Studio.

### Резюме

Spoon – «сама хмарна» оболонка для роботи з віртуальними додатками й, разом з тим, найменш яка піддається налаштуванню. Даний продукт прийде по

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

смаку користувачам, яким важлива не стільки безпека роботи за допомогою віртуалізації, скільки зручність роботи з необхідними програмами повсюдно.

[+] Інтеграція віджета з Робочим столом.

[+] Швидке створення пісочниць.

[-] Відсутність налаштувань по обмеженню віртуальних програм

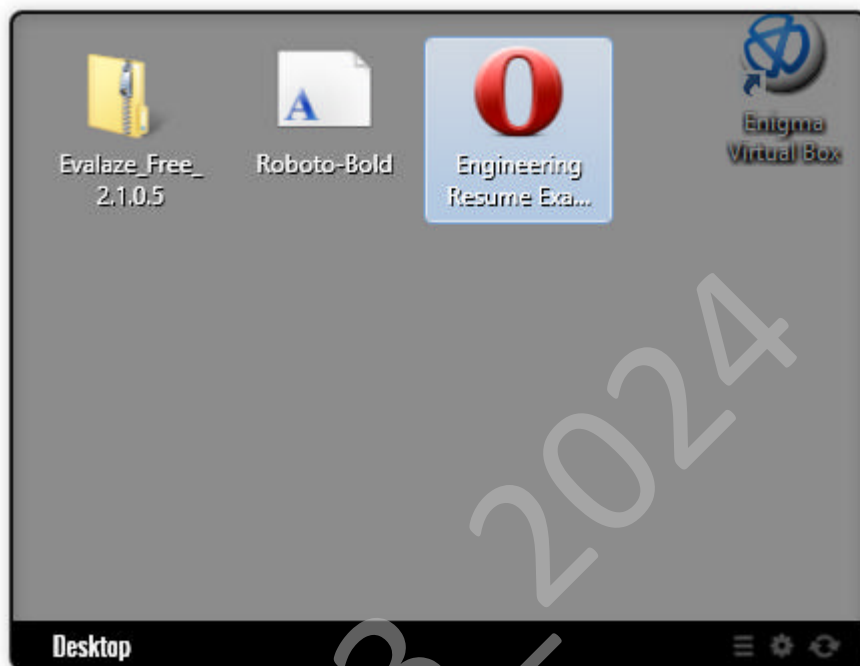


Рисунок 2.9 – Пісочниця, створена в Spoon

## 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – високорівнева мова програмування, яку називають другою за популярністю в світі. Її використовують для розробки вебзастосунків, програмного забезпечення, машинного навчання. Python застосовують для вирішення робочих завдань у компаніях Google, Instagram, Facebook, IBM, NASA, Dropbox, Netflix та інших. Розробники цінують цю мову програмування за простоту у вивченні, ефективність та мультиплатформність.

					VKPM-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

## Мова Python: що це?

Python – скриптова мова програмування з досить простим синтаксисом. Для розуміння достатньо порівняти принципи написання найпростішої програми, яка виводить на екран текстове повідомлення. Саме тому мова програмування Python більш доступна для новачків, а професіонали встигли адаптувати її для вирішення великої кількості завдань. Це мультиплатформне рішення, тому знання Python дає можливість працювати у різних сферах: від розробки мобільних застосунків до ігрової індустрії та штучного інтелекту.

У мови програмування динамічна типізація: є можливість передавати до функцій будь-який тип даних без попереднього вказання. Інтерпретованість дозволяє знаходити помилки у коді ще до повної збірки у робочий застосунок. При цьому Python дуже чітко дає зрозуміти, де та через що виникла помилка.

Це мова об'єктноорієнтованого програмування (ООП). Програмне забезпечення на Python оформлене у вигляді моделей, які можуть бути зібраними у пакети. Тип та структуру кожного об'єкта можна запитати під час виконання програми. Для кожного з об'єктів можна отримати всю інформацію щодо його внутрішньої структури. Окрім того:

- у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;
- гнучкість та масштабованість Python дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність;
- розробка на Python у більшості випадків проходить швидше, ніж на інших мовах програмування;
- Python – інтерпретована мова програмування. Це значить, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запустити;
- у Python – колосальна спільнота однодумців. Тож будь-які складнощі конкретних розробників вирішуються колективно.

					VKPM-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Проте є декілька особливостей, які можна віднести до недоліків. Це повільність (ця мова програмування хоч і універсальна, проте повільніша за інші), велика кількість ресурсів, необхідних для роботи та «прив'язаність» до системних бібліотек.

Мова програмування Python використовується у наступних сферах:

1. Розробка програмних застосунків будь-якого напрямку.
2. Розробка серверної частини мобільних застосунків (найпопулярніший напрямок).
3. Ігри. Багато сучасних ігор для комп'ютерів (наприклад, World of Tanks) частково чи повністю написані на Python.
4. Вбудовані системи для різних пристроїв. Дуже часто Python використовують для написання внутрішніх платформ управління банкоматами.
5. Скрипти та плагіни до уже реалізованих програм для автоматизації процесів чи створення інших рішень.
6. Тестування (автоматизація цього процесу).
7. Машинне навчання. – основна мова для написання алгоритмів і аналітичних застосунків у сфері Machine Learning.

### **Бібліотеки Python**

Різні бібліотеки Python використовують для виконання конкретних завдань. Наприклад, Matplotlib підходить для відображення даних у двовимірній та тривимірній графіці. Pandas підходить для зручної роботи з даними. NumPy дозволяє створювати масиви та керувати ними. Requests використовується для веброзробки. OpenCV-Python відкриває можливості для обробки зображень з метою оптимізації систем «машинного зору».

### **Найвідоміші фреймворки для мови програмування Python**

Фреймворки Python допомагають створити зручне та функціональне середовище для розробки. У них міститься набір інструментів, модулів та бібліотек, корисних для виконання конкретних завдань. Це значно полегшує роботу: наприклад, дає змогу не витратити час на розписування дій, які

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

повторюються, а використати релевантний інструмент. Тож є можливість позбутися рутинних процесів та сконцентруватися на логіці проекту.

Серед найпопулярніших фреймворків для Python:

– Django – найстаріший та найвідоміший. Створений для реалізації великих інтерактивних проєктів;

– Pyramid – зручний у налаштуваннях, і дає можливість реалізувати складні нестандартні ідеї;

– Web2py – підходить в першу чергу для вебзастосунків і може використовуватись на будь-яких архітектурах.

### Популярні Python IDE

IDE або інтегровані середовища розробки – це програмне забезпечення, яке надає розробникам необхідні інструменти для написання, редагування, тестування та налаштування коду. Для розробки на Python найчастіше використовують IDE PyCharm, IDLE, Spyder та Atom.

### 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 віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

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

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

б) вибрати та обґрунтувати методикку 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

						<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			28

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.

КБПЗ-2024

					ВКРМ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

#### Маршрутизація при віртуалізації мережі

Як і у випадку фізичних мереж, маршрутизація є важливою частиною HNV. Існують два ключових аспекти, важливих для розуміння: маршрутизація пакетів між віртуальними підмережами й маршрутизація пакетів за межами віртуальної мережі.

#### Маршрутизація між віртуальними підмережами

У фізичній мережі підмережа є доменом рівня 2, у якому комп'ютери (віртуальні і фізичні) можуть зв'язуватися один з одним прямо без маршрутизації. В Windows при статичній конфігурації мережного адаптера для правильної маршрутизації трафіка можна встановити "шлюз за замовчуванням", що буде являти собою IP-адреса для відправлення всього трафіка, що виходить із конкретної підмережі. По суті це маршрутизатор для фізичної мережі. HNV використовує убудований маршрутизатор, що є частиною вузла, для формування розподіленого маршрутизатора віртуальної мережі. Це значить, що кожний вузол, зокрема віртуальний комутатор, діє як шлюз за замовчуванням для всього трафіка між віртуальними підмережами, що є частиною однієї мережі віртуальних машин. В Windows Server 2025 і Windows Server 2025 R2 використовується як шлюз за замовчуванням адреса – самий нижній запис для підмережі (наприклад, буде використовуватися адресу ".1" для префікса підмережі /24). Ця адреса резервується в кожній підмережі для шлюзу за замовчуванням, і віртуальні машини не можуть використовувати його у віртуальній підмережі.

HNV, що діє як розподілений маршрутизатор, забезпечує досить ефективний спосіб правильної маршрутизації всього трафіка в рамках мережі віртуальних машин, тому що кожний вузол може прямо направляти трафік у

					ВКРМ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

відповідний вузол без посередників. Це тим більше вірно, якщо дві віртуальні машини з однієї мережі, але з різних віртуальних підмереж віртуальних машин перебувають на одному фізичному вузлі. Як ви довідаєтеся пізніше в цьому розділі, пакету немає необхідності залишати фізичний вузол.

### **Маршрутизація за межами віртуальної мережі**

Розгортання більшості клієнтських мереж вимагає, щоб середовище віртуалізації мережі була пов'язана з ресурсами, не включеними в середовище віртуалізації мережі. Шлюзи віртуалізації мереж необхідні для забезпечення можливості зв'язку між цими двома середовищами. Сценарії, що вимагають використання шлюзу віртуалізації мережі, включають приватна хмара й гібридна хмара. В основному шлюзи віртуалізації мережі необхідні для віртуальної приватної мережі (VPN) і маршрутизації.

Шлюзи можуть мати різні фізичні конструктивні параметри. Вони можуть будуватися на Windows Server 2025 R2, вбудовуватися в стійковий комутатор верхнього рівня (TOR) або підсистему балансування навантаження, включатися в інші існуючі мережні пристрої або бути новим автономним мережним пристроєм.

### **Приватна хмара (маршрутизація)**

Великі підприємства можуть або не вирішуватися, або, з міркувань дотримання вимог, бути не в змозі перемістити деякі зі своїх служб і даних у загальнодоступну хмару постачальника послуг розміщення. Разом з тим підприємства хочуть за рахунок об'єднання ресурсів своїх центрів обробки даних у приватній хмарі одержувати переваги хмарних технологій, надавані віртуалізацією мережі. При розгортанні приватної хмари перекриття IP-адрес може не знадобитися, оскільки компанії, як правило, мають у своєму розпорядженні достатній внутрішній простір немаршрутизуємих адрес (наприклад, 10.x.x.x або 192.x.x.x).

Відзначимо в даному прикладі, що адреси клієнта у віртуальних підмережах представлені у вигляді 157.x, при тому що IP-адреси в немережевій віртуалізованій частини мережі (Corp Net) також представлені у вигляді 157.x. У

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

цьому випадку адреси типу АП для віртуальних підмереж у центрі реєстрації й обробки даних є IP-адресами виду 10.x. Таке розгортання дозволяє підприємству скористатися перевагами віртуалізації мережі для забезпечення гнучкості як при розміщенні віртуальних машин, так і перекрестно-підмережній динамічній міграції в структурі центра обробки даних. Це збільшує ефективність центра реєстрації й обробки даних, тим самим знижуючи як поточні витрати (OpEx), так і капітальні витрати (CapEx). У даному сценарії шлюз віртуалізації мережі забезпечує маршрутизацію між IP-адресами виду 10.x і 157.1.

### **Гібридна хмара (VPN типу " мережа-мережа")**

Ключова перевага віртуалізації мережі – те, що вона дозволяє досить легко розширити локальний центр обробки даних до центра обробки даних із застосуванням хмарних технологій на основі Windows Server 2025. Це називається моделлю гібридної хмари.

У даному сценарії внутрішня підмережа, наприклад підмережа, що містить веб-сервери, переміщається з мережі підприємства в центр реєстрації й обробки даних постачальника послуг розміщення хмари. Скориставшись перевагою *Використовуйте власну IP-адресу*, запропонованою постачальником послуг розміщення, підприємство позбулося від необхідності зміни параметрів мережі віртуальної машини веб-сервера або будь-якої іншої кінцевої точки мережі, прив'язаної до веб-серверу. Постачальник послуг розміщення забезпечує безпечний канал зв'язку за рахунок застосування шлюзу віртуалізації мережі. Адміністраторам підприємства потрібно лише настроїти свою локальну VPN з використанням необхідних IP-адрес. Для віртуальної машини веб-сервера її переміщення в хмару залишається непоміченим. Вона залишається в складі домену за рахунок застосування Active Directory (AD) і продовжує використовувати DNS-сервер підприємства. Віртуальна машина веб-сервера також продовжує взаємодіяти з іншими серверами підприємства, наприклад сервером SQL Server.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Шлюз віртуалізації мережі може підтримувати кілька тунелів VPN типу " мережа-мережа" (S2S). Відзначимо, що диспетчер віртуальних машин на схемі не показаний, але він необхідний для розгортання віртуалізації мережі.

### **Інкапсуляція пакета**

У віртуалізації мережі кожний віртуальний мережний адаптер пов'язаний із двома IP-адресами:

– Адреса клієнта (АК) – IP-адреса, призначувана клієнтом на основі інфраструктури своєї інтрамережі. Дана адреса дозволяє клієнтові обмінюватися мережним трафіком з віртуальною машиною безвідносно факту її переміщення в хмару загального користування або частка хмара. АК є видимим для віртуальної машини й доступним для клієнта.

– Адреса провайдеру (АП) – IP-адреса, призначувана адміністратором постачальника послуг або центра реєстрації й обробки даних на основі інфраструктури своєї фізичної мережі. АП з'являється в мережі у вигляді пакетів. Обмін ними виробляється із сервером, на якому розміщується віртуальна машина. АП є видимим у фізичній мережі, але не для віртуальної машини.

АК обслуговують топологію клієнтської мережі, що віртуалізується й втрачає прив'язки до фактичних адрес і топології базової фізичної мережі, як це реалізовано для АП.

На даній схемі віртуальні машини клієнта відправляють пакети даних у просторі АК, які переглядають інфраструктуру фізичної мережі через власні віртуальні мережі, або "тунелі". У наведеному вище прикладі дані тунелі можна зрівняти з "конвертами" навколо пакетів даних А і Б із зеленими індексами (адресами типу АП), які доставляються з вузла-джерела в лівій частині схеми на кінцевий вузол у правій частині схеми. Ключовим тут є те, яким образом дані вузли визначають "адреси відправлення" (АП), що відповідають АК А і Б, яким образом навколо пакетів формується "конверт" і як вузли призначення розвертають пакети й роблять правильну доставку вмісту на кінцеві віртуальні машини А і Б.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Ключові аспекти віртуалізації мереж пояснюються за допомогою простої аналогії:

– АК кожної віртуальної машини зіставляється з АП фізичного вузла. З одним АП може бути зв'язане декілька АК.

– Віртуальні машини відправляють пакети даних у просторах АК, які полягають в "конверт" і забезпечуються парою АП джерела й призначення на основі зробленого зіставлення.

– Зіставлення АК – АП повинні дозволяти вузлам робити диференціацію пакетів для різних клієнтських віртуальних машин.

У результаті механізм віртуалізації мереж зводиться до віртуалізації мережних адрес, використовуваних віртуальними машинами. У наступному розділі описується властиво механізм віртуалізації адрес.

### **Віртуалізація мереж за допомогою віртуалізації адрес**

Як механізм віртуалізації IP-адреси віртуалізація мережі підтримує Generic Routing Encapsulation (NVGRE).

– Generic Routing Encapsulation Цей механізм віртуалізації мережі використовує Generic Routing Encapsulation (NVGRE) як частина заголовка тунелю. У механізмі NVGRE пакет віртуальної машини інкапсулюється в інший пакет. Заголовок цього нового пакета забезпечується необхідними IP-адресами джерела й призначення типу АП на додаток до ID віртуального підмережі, що зберігається в ключовому полі заголовка GRE.

ID віртуальної підмережі дозволяє вузлам ідентифікувати клієнтську віртуальну машину для кожного окремого пакета навіть у випадку можливого перекриття АП і АК на пакетах. Це дозволяє всім віртуальним машинам одного вузла спільно використовувати єдиний АП.

Спільне використання АП впливає на масштабованість мережі. Кількість IP- і MAC-адрес, що вимагають запам'ятовування мережною інфраструктурою, може бути істотно знижено. Наприклад, якщо на кожному кінцевому вузлі є в середньому 30 віртуальних машин, кількість IP- і MAC-адрес, що вимагають запам'ятовування мережною інфраструктурою, знижується на коефіцієнт 30. ID

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

віртуальної підмережі, включені в пакети, також дозволяють робити просту кореляцію пакетів під фактичних клієнтів.

В Windows Server 2025 і більше пізніх версіях віртуалізація мережі повністю підтримує NVGRE; вона НЕ вимагає відновлення наявного або придбання нового мережного обладнання, наприклад мережних карт (мережних адаптерів), комутаторів або маршрутизаторів. Це досягається за рахунок того, що пакет NVGRE при провідній передачі є звичайним IP-пакетом у просторі АП, сумісному із сучасною мережною інфраструктурою.

В Windows Server 2025 високий пріоритет відданий роботі зі стандартами. Разом із ключовими партнерами галузі (Arista, Broadcom, Dell, Emulex, Hewlett Packard і Intel) корпорація Майкрософт опублікувала проект RFC, що містить опис використання загальної інкапсуляції маршрутів (GRE), що є існуючим стандартом IETF як протокол інкапсуляції для віртуалізації мереж з аббревіатурою GRE. Додаткові відомості див. у наступному проекті в мережі Інтернет: Віртуалізація мереж з використанням загальної інкапсуляції маршрутів (Network Virtualization using Generic Routing Encapsulation). У міру комерційного впровадження NVGRE вигода від реалізації даного методу ще більше зросте.

### **Приклад архітектури обслуговування одним екземпляром додатка декількох розгортань**

Представлено приклад розгортання двох клієнтів, що переміщуються в хмарі центра обробки даних з використанням взаємин адрес АК-АП, обумовлених політикою віртуалізації мережі.

До переходу до розміщення в службі IaaS, забезпеченої провайдером:

– Компанія А задіяла SQL Server (з ім'ям SQL) за IP-адресою 10.1.1.11 і веб-сервер (з ім'ям Web) за IP-адресою 10.1.1.12, що використовує свій SQL Server для транзакцій бази даних.

– Компанія Б задіяла SQL Server, що також має ім'я SQL і призначений IP-адреса 10.1.1.11, і веб-сервер, теж іменованій Web і що має IP-адреса 10.1.1.12, що використовує свій SQL Server для транзакцій бази даних.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

– Компанії А и Б перемістили свої сервери SQL Server і відповідні веб-сервери в те саме розміщення в службі IaaS, забезпеченої постачальником, де, по випадковому збігу, вони задіяли віртуальні машини SQL на вузлі 1 (Host 1) і віртуальні машини Web (IIS7) на вузлі 2 (Host 2). Всі віртуальні машини зберігають свої вихідні внутрімережні IP-адреси (свої АК).

Обом компаніям при виділенні віртуальних машин провайдером розміщення призначаються наступні ID віртуальної підмережі (VSID) і адреси типу АП:

– АП віртуальних машин компанії А: VSID – 5001, SQL – 192.168.1.10, Web – 192.168.2.20.

– АП віртуальних машин компанії Б: VSID – 6001, SQL – 192.168.1.10, Web – 192.168.2.20.

Постачальник розміщення створює параметри політики, що складаються із клієнтської віртуальної підмережі для компанії Б, що встановлює відповідність АК віртуальних машин компанії Б призначеним АП і VSID, і окремої клієнтської віртуальної підмережі для компанії А, що встановлює відповідність АК віртуальних машин компанії А призначеним АП і VSID. Далі провайдер застосовує дані параметри політики до Вузла 1 і Вузлу 2.

Коли віртуальна машина Web компанії А запитує на Вузлі 2 свій SQL Server за адресою 10.1.1.11, відбувається наступне:

Вузол 2, ґрунтуючись на своїх параметрах політики, переводить адреси у вигляді пакета по маршруті:

– Джерело: 10.1.1.12 (АК компанії А Web).

– Призначення: 10.1.1.11 (АК компанії А SQL).

Інкапсульований пакет містить:

– Заголовок GRE (OIM) з VSID: 5001.

– Зовнішнє джерело: 192.168.2.20 (АП компанії А Web).

– Зовнішнє призначення: 192.168.1.10 (АП компанії А SQL).

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

При одержанні пакета Вузол 1, ґрунтуючись на своїх параметрах політики, декапсулює пакет NVGRE:

- Зовнішнє джерело: 192.168.2.20 (АП компанії A Web).
- Зовнішнє призначення: 192.168.1.10 (АП компанії A SQL).
- Заголовок GRE (OIM) з VSID: 5001.
- Декапсульований пакет (вихідний пакет, відправлений з віртуальної машини Web компанії A) доставляється на віртуальну машину SQL компанії A:

- Джерело: 10.1.1.12 (АК компанії A Web).
- Призначення: 10.1.1.11 (АК компанії A SQL).
- Коли віртуальна машина SQL компанії A на Вузлі 1 відповідає на запит,

відбувається наступне:

Вузол 1, ґрунтуючись на своїх параметрах політики, переводить адреси у вигляді пакета по маршруті:

- Джерело: 10.1.1.11 (АК компанії A SQL).
- Призначення: 10.1.1.12 (АК компанії A Web).

Інкапсульований пакет містить:

- Заголовок GRE (OIM) з VSID: 5001.
- Зовнішнє джерело: 192.168.1.10 (АП компанії A SQL).
- Зовнішнє призначення: 192.168.2.20 (АП компанії A Web).

При одержанні пакета Вузол 2, ґрунтуючись на своїх параметрах політики, декапсулює даний пакет:

- Джерело: 192.168.1.10 (АП компанії A SQL).
  - Призначення: 192.168.2.20 (АП компанії A Web).
  - Заголовок GRE (OIM) з VSID: 5001.
  - Декапсульований пакет доставляється на віртуальну машину Web компанії A:
- Джерело: 10.1.1.11 (АК компанії A SQL).
  - Призначення: 10.1.1.12 (АК компанії A Web).

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

– Аналогічний процес для трафіка між Б Web і віртуальними машинами SQL використовує параметри політики HNV для корпорації Б. У результаті при використанні HNV віртуальні машини компанії Б и А взаємодіють так, ніби перебували у своїх інтрамережах компаній. При цьому вони не можуть взаємодіяти один з одним, навіть незважаючи на те, що використовують однакові IP-адреси.

– Роздільні адреси (АК і АП), параметри політики вузлів, а також перетворення адрес із АК в АП і назад для вхідні й вихідного трафіка віртуальних машин ізолюють ці комплекти серверів. Більше того, зіставлення віртуалізації й перетворення позбавляють архітектуру віртуальної мережі прив'язки до інфраструктури фізичної мережі. Хоча SQL і Web компанії А и SQL і Web компанії Б розташовуються в їх власних підмережах IP АК (10.1.1/24), їхнє фізичне розгортання відбувається на двох вузлах у різних підмережах АП: 192.168.1/24 і 192.168.2/24 відповідно. Суть у тім, що перекрестно-підмережне виділення віртуальних машин і динамічна міграція стають можливими із застосуванням віртуалізації мережі.

### **Архітектура віртуалізації мереж**

В Windows Server 2025 застосування політики віртуалізації мережі й віртуалізація IP виконувалися фільтром полегшеного доступу (LWF) специфікації інтерфейсу мережного драйвера (NDIS), що називається мережний віртуалізацією Windows (WNV). Фільтр WNV розташовувався нижче комутатора. У такої архітектури був побічний ефект: розширення комутатора могли бачити тільки трафік простору АК, але не простору АП.

В Windows Server 2025 R2 HNV є частиною віртуального комутатора, завдяки чому розширення можуть бачити обоє простору адрес: і АК, і АП.

Кожний мережний адаптер віртуальної машини при налаштуванні забезпечується IPv 4- або IPv 6-адресою. Існують АК, використовувані для зв'язку віртуальних машин один з одним, вони надходять в IP-пакетах із цих віртуальних машин. Віртуалізація мережі віртуалізує такі АК в АП, ґрунтуючись на політиках мережний віртуалізації.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

Віртуальна машина відправляє пакет з адресою джерела AK1, що віртуалізується на основі політики у фільтрі HNV у комутаторі. Особливий список керування доступом до віртуалізації мереж на основі VSID ізолює віртуальну машину від інших віртуальних машин, не включених у ту ж віртуальну підмережа або в той же домен маршрутизації.

### **Керування політикою віртуалізації мереж**

Платформа Windows надає API загального користування для керуючого ПЗ центрів обробки даних для керування віртуалізацією мережі. Диспетчер віртуальних машин є одним з таких продуктів для керування центром обробки даних. Керуюче ПЗ містить усе політики віртуалізації мережі. Оскільки диспетчер віртуальних машин повинен розпізнавати віртуальні машини й, що важливіше, забезпечувати віртуальні машини й у цілому клієнтські віртуальні мережі в центрі обробки даних, а також підтримувати архітектуру багатотенантного обслуговування, керування політикою віртуалізації мережі є природним розширенням організації мереж на основі політики.

### **Зведення**

Центри реєстрації й обробки даних на основі технології хмари можуть забезпечувати безліч переваг, наприклад поліпшену масштабованість і більше ефективне використання ресурсів. Реалізація цих потенційних переваг вимагає застосування технології, присвяченої принциповому рішенню проблем багатоклієнтської масштабованості в динамічному середовищі. Віртуалізація мережі створена для рішення цих проблем і підвищення робочої ефективності центрів обробки даних за рахунок поділу топологій віртуальної й фізичної мереж. Побудована на основі існуючого стандарту, віртуалізація мережі працює в середовищі сучасного центра обробки даних, а в міру впровадження встаткування, сумісного з NVGRE, кількість переваг, що здобуваються, буде збільшуватися. Із застосуванням віртуалізації мережі клієнти одержують можливість поєднувати свої центри обробки даних у приватна хмара або за допомогою технології гібридної хмари відносно легко розширювати свої центри обробки даних до середовища постачальника послуг розміщення.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

### 3.2 Розробка структурної схеми

#### Віртуалізація від Nuage

Мережні інфраструктури стають вузьким місцем при «переході» до віртуалізованих ІТ і хмарної моделі одержання ІТ-сервісів. Динамічне середовище віртуальних машин (ВМ) дуже погано співвідноситься зі статичними мережами, у яких використовуються протоколи (наприклад, VLAN і STP), що застосовувалися в них і п'ять, і десять років тому.

Розглянемо типове завдання по розгортанню нового додатка або впровадженню нового сервісу, що складає з декількох функціональних компонентів, розміщених у різних підмережах. Раніше для цього була потрібна установка нових серверів, що займало кілька тижнів. Сучасні системи віртуалізації серверних ресурсів дозволяють виділити необхідні новому додатку або сервісу віртуальні машини з потрібною продуктивністю за лічені хвилини.

Однак відповідна підготовка мережі займає набагато більше часу. Конфігурування різних типів мережних пристроїв часто відбувається вручну й вимагає залучення великої кількості фахівців: один займається локальною мережею, іншої – територіально розподіленою, третій – налаштуванням VLAN, четвертий – конфігуруванням міжмережних екранів і т.д. Як результат, всі переваги віртуалізації серверів у частині прискорення впровадження додатків зводяться на ні «повільністю» традиційних процедур налаштування мережі.

Вирішити проблему дозволяють нові технології віртуалізації мережі, які зараз активно розробляють і виводять на ринок багато виробників. Одне з найбільш зрілих рішень, що вже розгорнули у своїх мережах кілька закордонних компаній, пропонує Nuage Networks. Воно підтримує більшість основних гіпервізорів (KVM, Xen, ESXi), систем оркестрації (включаючи VMware, OpenStack, CloudStack) і працює поверх мережної інфраструктури, побудованої на встаткуванні будь-якого виробника.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Рішення Nuage Networks Virtualized Services Platform (VSP) складається із трьох основних елементів:

- каталогу віртуалізованих сервісів (Virtualized Services Directory, VSD);
- агентів віртуальної маршрутизації/комутації (Virtual Routing & Switching, VRS);
- контролерів віртуалізованих сервісів (Virtualized Services Controller, VSC).

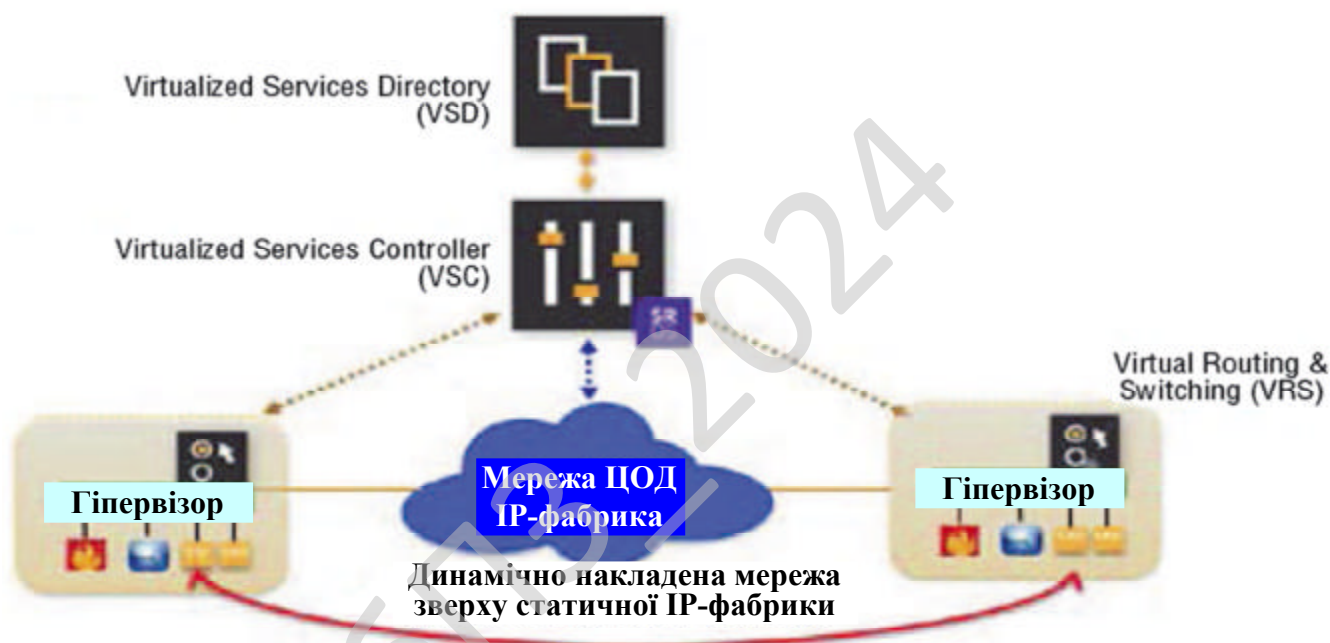


Рисунок 3.1 – Структурна схема системи

Каталог VSD забезпечує високий рівень абстракції. Він дозволяє описати мережні структури користувачів хмарної мережі й надати можливість адміністраторам самостійно управляти мережними ресурсами в заданих границях. Найважливішою функцією VSD є створення шаблонів мережних політик, не зв'язаних зі специфікою апаратного пристрою мережі. Надалі, при запуску конкретного екземпляра віртуалізованого додатка, на основі заданого шаблону створюється опис конфігурації його мережі. Наприклад, окремий

шаблон можна створити для додатків (віртуальних машин), що поміщаються в «демілітаризовану» зону, доступ у яку може здійснюватися тільки по TCP і тільки через протокольний порт 80.

Агенти VRS відповідають за керування віртуальними мережними інтерфейсами відповідно до заданих шаблонів (політиками) і працюють під керуванням найпоширеніших гіпервізорів. Вони інкапсулюють генеруємий віртуальними машинами трафік і передають його між кінцевими точками (гіпервізорами) по тунелях. Таким чином, поверх фізичної інфраструктури мережі формується накладена мережа типу «фабрика».

Агенти VRS одержують інструкції щодо того, що робити з конкретним трафіком, від контролера VSC, відповідального за цілісну роботу й подання загальної топології багатокористувальницької мережі. Оскільки цей контролер використовує ту ж операційну систему SROS, що й лінійка сервісних маршрутизаторів Alcatel-Lucent 7x50, на базі якої працюють ряд найбільших у світі IP-мереж, він здатний управляти повнофункціональною маршрутизацією (на рівнях L 2-L4) для тисяч користувальницьких підмереж.

При запуску віртуальної машини, що виконує роль одного з функціональних компонентів віртуалізованого додатка, контролер VSC за запитом від VRS одержує з бази даних VSD екземпляр шаблону (instance). Потім цей шаблон завантажується в контролер, що автоматично набудовує всі мережні елементи, які можуть бути задіяні в обслуговуванні даного додатка. Ручне переконфігурування окремих мережних пристроїв виключається.

Шаблон, у якому визначені мережна конфігурація й правила обробки трафіка конкретного додатка, залишається прив'язаний до відповідних віртуальних машин поза залежністю від їхніх переміщень. Вони можуть розташовуватися в одному ЦОД або в декількох, переміщатися між фізичними серверами й територіально віддаленими площадками – віртуальна мережа буде автоматично підлаштовуватися.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Прискорення інтеграції віртуалізованого додатка/сервісу в мережу, пояснює він, досягається за рахунок створення шаблону, у якому вказуються вимоги до мережі, включаючи конфігурацію підмереж, параметри якості обслуговування (Qo), налаштування безпеки та ін. Пізніше адміністратор може багаторазово тиражувати його при розгортанні екземплярів віртуалізованого додатка/сервісу.

На практичному семінарі фахівці Nuage Networks розгорнули демонстраційну мережу, до складу якої входили хости із двома типами гіпервізорів: ESXi (під керуванням VMware vCenter) і KVM (під керуванням OpenStack Havana). За допомогою засобів VSD було продемонстроване формування структури накладеної мережі (поділ її на зони, підмережі), після чого подібні дії були виконані із застосуванням системи OpenStack – відповідні зміни автоматично з'явилися в каталозі Nuage VSD.

Потім віртуальні машини були запущені на хостах з гіпервізорами KVM і ESXi (в останньому випадку через vCenter). Вони містилися в ті зони (підмережі), які забезпечували необхідні ним параметри обслуговування. Трафік між гіпервізорами різних типів (KVM і ESXi) передавався в режимі L3 – тобто з маршрутизацією.

Найбільш затребуваний сценарій – «живаючи» міграція віртуальних машин з одного хоста на іншій. Ця процедура була ініційована із системи керування vCenter, а мережа автоматично перенастроїлася для обслуговування VM на новому місці. Шаблон, що визначає правила обробки трафіка додатка, автоматично треба за віртуальною машиною при її переїзді. При цьому ніякого втручання адміністратора не потрібно. Більше того, віртуальну мережу можна розтягти по VPN через територіально розподілену мережу (WAN) до офісної мережі компанії. Ця функціональність теж була продемонстрована фахівцями Nuage Networks.

З огляду на той факт, що не всі підключені до мережі пристрої піддаються віртуалізації, рішення Nuage VSP передбачає механізми інтеграції апаратних

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

компонентів (таких, наприклад, як сервери баз даних, міжмережні екрани, різні шлюзи). Для сценаріїв, що не вимагають високої продуктивності, пропонується програмний модуль VRS Gateway. Для ситуацій з більшими обсягами трафіка розроблений апаратний шлюз 7850 VSG із продуктивністю більше 1 Тбіт/с, що підтримує інтерфейси 1GE, 10GE і 40GE.

### **Brocade: віртуалізуя фабрику й мережні функції**

З 2008 року в результаті покупки Foundry Networks у лінійці компанії Brocade з'явилися IP-рішення, і зараз у сферу її інтересів входять класичні IP-маршрутизатори L3, а також SDN, NFV і засобу оркестрації хмар (автоматизація виділення ресурсів у хмарі й керування ними). Останні реалізуються Brocade по SAN і LAN у рамках проекту OpenStack. Рішення, орієнтовані на ЦОД, мають стратегічне значення, були представлені як уже успішно експлуатовані в замовників продукти, так і нові розробки.

З появою в лінійці продуктів Ethernet взяли краще з миру FC, об'єднали ці інструменти з технологіями Ethernet і одержали Ethernet-Фабрики – віртуальний розподілений Ethernet-Комутатор. Фактично це новий підхід до побудови мережі передачі даних – з використанням одного логічного комутатора, керування яким здійснюється з єдиної точки, і при цьому забезпечується високий ступінь автоматизації. Зараз поряд з фабриками FC лінійка Brocade містить у собі Ethernet-Фабрики – основні її рішення для ЦОД.

Ethernet-Фабрика Brocade VCS реалізує віртуальний розподілений комутатор рівня доступу й агрегування. Вона практично не вимагає налаштування, підтримуючи «самоформування» і автовизначені пристроїв у будь-якій топології, при цьому керування нею здійснюється як одним віртуальним комутатором. Для запуску фабрики досить задати її домен і ID, а потім можна користуватися єдиною консоллю керування всіма комутаторами.

Для підвищення надійності підтримується топологія Full Mesh, завдяки чому працездатність системи зберігається навіть при виході з ладу одного комутатора. Крім того, фабрика готова до конвергенції (FC 16 / FCo 10 Гбіт/с),

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

підтримує три рівні балансування трафіка й здатна агрегувати (без попереднього налаштування) велика кількість каналів. Завдяки покадровому балансуванню забезпечується рівномірний розподіл навантаження між портами комутаторів. За допомогою протоколу ECMP здійснюється балансування між еквівалентними за вартістю шляхами, що з'єднують кінцеві точки (незалежно від числа транзитних вузлів). А балансування навантаження між шлюзами VRRP-E L3 (до чотирьох шлюзів) поліпшує масштабованість і надійність рішення.

Недавно з'явилася в Brocade технологія дає можливість створювати «віртуальні фабрики» усередині VCS. Виділення логічних фабрик для кожного користувача в рамках поділюваної фізичної фабрики на основі TRILL Fine-Grained Labels (IETF draft) дозволяє повторно задіяти VLAN. У рамках кожної фізичної фабрики підтримується до 8000 віртуальних фабрик VCS.

Завдяки інтеграції з VMware vCenter, Ethernet-Фабрика може одержувати інформацію про віртуальні машини, їх MAC- і IP-адресах, групах VLAN. Ця інформація автоматично заноситься в Ethernet-Фабрику, і за замовчуванням створюються «профілі портів», до яких можна застосовувати різні налаштування – наприклад, задавати якість обслуговування Qo. Система настроюється однократно – надалі фабрика сама застосовує задані політики по MAC-адресі, на якому б з портів він не був ідентифікований. Одна фабрика поєднує до 48 комутаторів.

В останніх версіях фабрики Brocade реалізований VXLAN-шлюз VCS (VTEP) для VMware NSX – міст між віртуальними й фізичними ресурсами. Його можна використовувати для інтеграції із класичними мережами, що не підтримують VXLAN. Повна підтримка VCS реалізована в комутаторах Brocade VDX серії 6740 і маршрутизаторах 8770. У серії 6740 ( SDN-ready) можна використовувати протокол OpenFlow 1.3. Відповідно до планів Brocade, контролер SDN з'явиться на ринку під кінець року. Він буде сполучати функції контролера й комутатора SDN, продовжуючи лінійку VDX 6740. Протокол OpenFlow підтримується також продуктами Brocade Fibre Channel.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

На думку аналітиків, у найближчій перспективі будуть розгортатися переважно гібридні рішення, що сполучають традиційну мережну інфраструктуру й SDN, тому більшість комутаторів SDN допускають програмне перемикання між SDN і комутацією L2/L3. Тим самим знижуються ризики впровадження нової технології. Brocade пропонує своє рішення. На відміну від інших вендорів компанія розробила комутатори й маршрутизатори з «гібридними портами», тобто вони можуть працювати в змішаному режимі, підтримуючи як традиційне керування, так і керування за допомогою контролера SDN. Такий режим зараз реалізується маршрутизаторами Brocade MLXe.

SDN уже знаходить практичне застосування. Так, в одного із замовників подібна система допомагає знизити негативний вплив атак DDo. Погрози виявляються за допомогою sFlow-RT (у режимі реального часу) і відправляються на колектор, де обробляються програмним забезпеченням централізованого керування мережею Brocade Network Advisor, здатним виступати також як контролер sFlow. Колектор сповіщає контролер, що потрібно використовувати правило OpenFlow DDo на маршрутизаторі MLXe (див. Рисунок 5) для протидії атаці DDo.

Поряд з SDN, у мережній галузі набирає популярність технологія NFV. Вона дозволяє за допомогою віртуалізації консолідувати різні мережні пристрої (міжмережне встаткування, комутатори, маршрутизатори, балансувальники навантаження й т.п.) на стандартних серверах x86, що веде до зменшення вартості й збільшенню «живучості» систем. Перераховані пристрої стають віртуальними (функціонують у вигляді VM), так що здобувати дороге «залізо» не потрібно.

Тим часом Brocade уже пропонує віртуальний маршрутизатор Vyatta vRouter з функціями концентратора VPN із шифруванням IPSec і міжмережного екрана. Це відносно недороге й гнучке рішення для невеликої мережі Ethernet дає можливість надавати мережні функції «за запитом» і реалізує маршрутизацію, міжмережний екран, VPN. Для розгортання таких VM можна використовувати

існуючі стандартні сервери x86, здатні успішно замінити спеціалізовані системи, у тому числі рішення для інфраструктури зв'язку. vRouter інтегрований зі стандартними оркестраторами (OpenStack).

Ще один віртуальний продукт Brocade – повнофункціональний балансувальник трафіка додатків ADX. Будучи ключовим компонентом L 4-L7 в NFV, він динамічно балансує трафік додатків і прискорює запуск сервісів. Для цілей централізованого керування передбачається підтримка OpenStack і API. Віртуальний ADX можна розгортати у вигляді VM на сервері з гіпервізором або без гіпервізора, виділяючи йому весь сервер.

Ці продукти дозволяють гнучко (за запитом) впроваджувати різні сервіси, однак пропускна здатність у них трохи нижче, ніж у фізичних пристроїв, – від 10 Мбіт/с до 3 Гбіт/с (обмеження залежить від ліцензії, а не від швидкодії сервера). Продуктова лінійка Brocade, включаючи віртуальні пристрої, підтримує оркестрацію за протоколом OpenStack (через оркестратори сторонніх вендорів). Компанія бере активну участь у розробці стандартів OpenStack і пропонує розширення Dynamic Network Resource Management (DNRM) у рамках проекту Neutron, щоб спростити побудову мультивендорних мереж.

Зі збільшенням обсягів мережного трафіка й кількості підключених до мережі пристроїв конфігурування мереж і керування ними перетворюється в складне завдання. Щоб неї полегшити, потрібні серйозні зміни в підходах до побудови, експлуатації мереж і керуванню ними. Мережна інфраструктура розвивається й еволюціонує повільно, відстаючи від бурхливого розвитку засобів віртуалізації серверів і систем зберігання, однак саме мережі повинні стати надійною й гнучкою основою для середовища хмарних обчислень. В останні рік-два в цій області намітився явний прогрес, а розробки в області NFV відкривають для замовників перспективи переходу від апаратних до віртуальних мережних рішень зі збереженням всіх функцій, властивим апаратним продуктам.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

### 3.3 Розробка функціональної схеми

Функціональна схема розробленої системи зображена на рисунку 3.2. Рішення задачі забезпечення своєчасності передачі інформації в віртуалізованій мережі Nuage Networks складається з чотирьох етапів:

- визначення множини  $\mathcal{N}_{\text{баз}}$  маршрутів передачі інформації;
- знаходження оптимальної множини маршрутів передачі цифрової інформації в телекомунікаційній мережі  $\mathcal{N}_{\text{об}}$ ;
- обчислення коефіцієнтів  $\tilde{k}_s$  розподілу інформаційного потоку і управління навантаженням віртуалізованої мережі Nuage Networks;
- створення та оновлення таблиці маршрутизації.

При рішенні задачі визначення множини  $\mathcal{N}_{\text{баз}}$  шляхів передачі інформації в віртуалізованій мережі Nuage Networks для ВЗ « $i$ » та « $j$ » з множини  $\mathcal{R}$  вузлів зв'язку спочатку необхідно знайти найкоротшу «відстань» (мінімальний час передачі інформаційних пакетів)  $T_{i,j \text{ min}}$  від джерела « $i$ » до адресата « $j$ » і множини  $S_j^{(i)}$  вузлів, найближчих ВЗ « $i$ » за напрямом руху потоку до « $j$ » (множина «вузлів-наступників») у порядку рівнів ієрархії дерева допустимих маршрутів множини  $U$ .

При рішенні поставленої задачі відомими алгоритмами пошуку найкоротших шляхів в більшості практичних випадків маємо проблему «зациклення» при передачі інформації в знайдених шляхах. Це призводить до збільшення часу передачі інформаційних пакетів, а деколи і до їх втрати. Уникнути «зациклення» при передачі інформації пропонується шляхом додання обмежень (умова постійної відсутності циклів), які надані у вигляді виразів:

$$T_{k,j} \leq T_{i,j \text{ min}}; \quad (3.1)$$

$$T_{k,j \text{ min}} \leq T_{i,k,j}, \quad k \in R, \quad (3.2)$$

де  $T_{k,j \min}$  – найкоротша «відстань» (мінімальний час передачі інформаційних пакетів) від вузла « $k$ » до адресата « $j$ »;  $T_{i,k,j}$  – «відстань» (час передачі інформаційних пакетів) від вузла « $i$ » до адресата « $j$ » через вузол « $k$ ».



Рисунок 3.2 – Функціональна схема системи

Рішення задачі пошуку множини шляхів, що виключають «цикли», складається з двох етапів: визначення найкоротшої «відстані»  $T_{i,j \min}$  від джерела « $i$ »

до адресата « $j$ »; знаходження множини  $S_j^{(i)}$  «вузлів-наступників» на маршрутах, що виключають «циклічність», для довільних джерел « $i$ » та адресатів « $j$ » за порядком множини  $U$  рівнів ієрархії дерева вибору допустимих маршрутів.

На першому етапі для визначення найкоротшої «відстані»  $T_{i,j \min}$  доцільно використати алгоритм розрахунку попередньої топології, який забезпечує вузли зв'язку інформацією про стан зв'язків для обчислення найкоротших шляхів до адресатів. Алгоритм створено на основі відомих алгоритмів стану зв'язків. На відміну від відомих, в цьому алгоритмі враховується ієрархічність побудови віртуалізованої мережі Nuage Networks (визначаються «відстані» від «вузла-джерела»  $i$  до «адресата»  $j$  відповідно до існуючих рівнів ієрархії), що надалі дозволить здійснити пропорційний (з урахуванням коефіцієнтів  $k_s$  і  $k_s^{(c)}$ ) розподіл інформації по знайдених маршрутах.

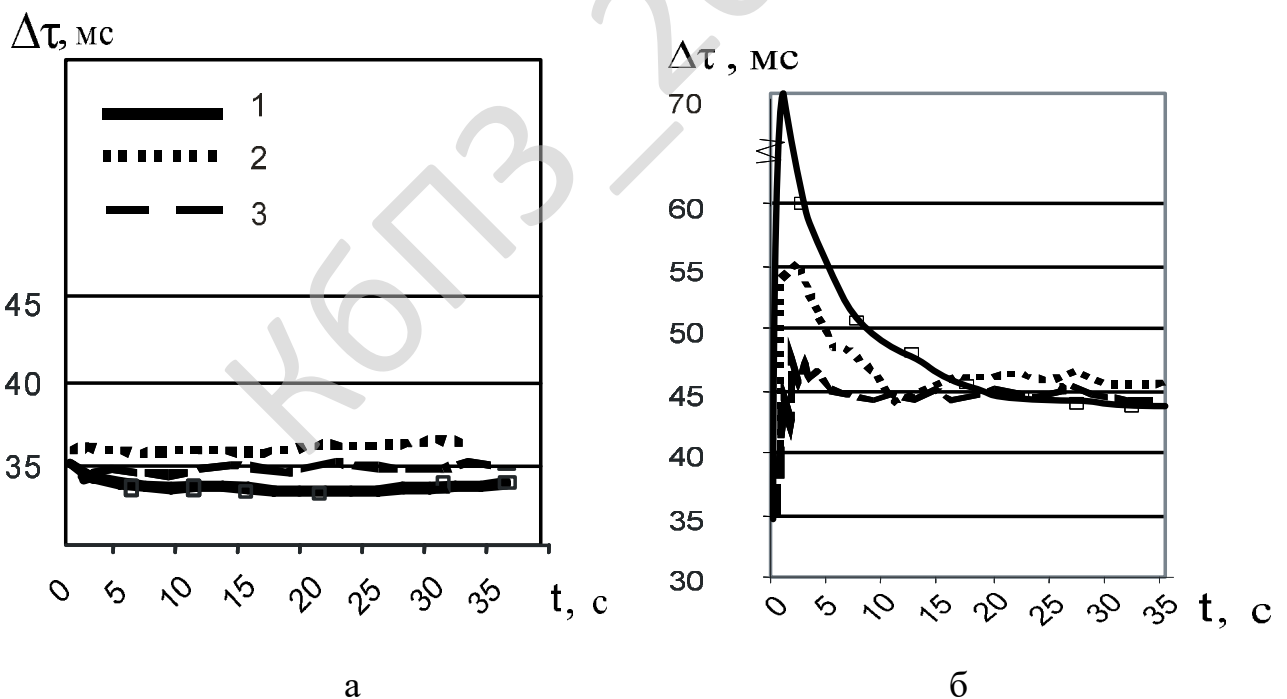


Рисунок 3.3 – Час затримки інформації: а – при статичному трафіку, б – при флуктуації трафіку від 3 до 10 Мбіт/с

На другому етапі для знаходження множини шляхів передачі інформації, що виключають «циклічність», використовується алгоритм розповсюдження попередньої топології множини шляхів. На відміну від відомих алгоритмів, в яких не враховується поточний стан ВЗ «i» обов'язкова синхронізація обміну службовими повідомленнями про стан зв'язків по всій мережі, в алгоритмі розповсюдження попередньої топології множини шляхів такої синхронізації підлягає тільки один перехід між сусідніми вузлами. Це значно спрощує роботу вузлів зв'язку і скорочує час збіжності алгоритмів. Сукупність алгоритмів розрахунку попередньої топології і розповсюдження попередньої топології множини шляхів є основою способу визначення множини шляхів передачі інформації.

Запропонований спосіб (крива 3, рисунок 3.3, а) за відсутністю флуктуацій трафіку зіставлений з аналогічними (крива 1 – спосіб Галлагера, крива 2 – модифікований дистанційно-векторний (MDVA)) за часом затримки передачі інформації, проте у декілька разів перевершує їх при різких флуктуаціях вхідного трафіку (рисунок 3.3, б).

Безпосереднє використання всієї знайденої множини  $N_{\text{баз}}$  шляхів передачі інформації розробленим способом не завжди є виправданим, особливо у разі високої пропускної спроможності декількох з наявних каналів зв'язку, здатних забезпечити виконання вимог (3.8), (3.9) при передачі інформації про повітряну обстановку. Розширення такої множини призводить до збільшення таблиць маршрутизації вузлів зв'язку, ускладнення процесу розподілу інформації і, як наслідок, до зниження достовірності передачі цифрової інформації. Тому виникає необхідність в знаходженні такої топології підмережі, тобто у виборі зі всієї знайденої множини  $N_{\text{баз}}$  шляхів деякої (оптимальної) сукупності  $N_{\text{об}}$  маршрутів, використання якої в умовах обмежень, що накладаються, дозволить забезпечити максимально можливу достовірність передачі інформації.

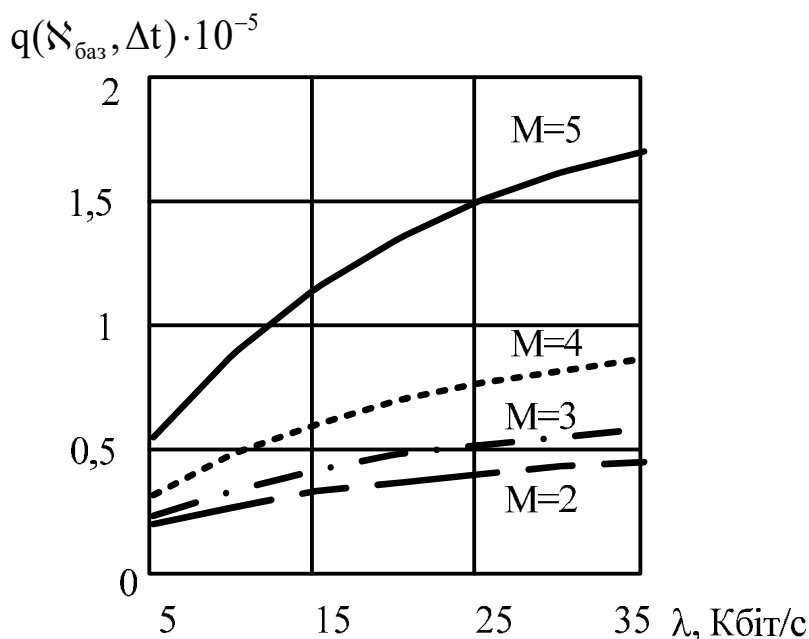


Рисунок 3.4 – Залежність ймовірності  $q(N_{\text{баз}}, \Delta t)$  спотворення інформаційних пакетів від інтенсивності  $\lambda$  вхідного потоку інформації

Одержані і наведені на рисунку 3.4 криві залежності ймовірності  $q(N_{\text{баз}}, \Delta t)$  від  $\lambda$ . Збільшення числа  $M$  використаних маршрутів, призводить до істотного (у 2,3...3,4 разів) збільшення ймовірності  $q(N_{\text{баз}}, \Delta t)$  спотворення інформації в процесі її передачі. Слід особливо відзначити, що для відомих методів розподілу характерна відсутність моніторингу поточного завантаження маршрутів, змін вхідного потоку інформації, а інколи і технічного стану каналів зв'язку. Принцип рівномірного завантаження вибраних  $M$  маршрутів, який використовується в таких методах, призводить до перевантаження одних і до неефективного використання інших маршрутів.

Для визначення початкового завантаження підмережі віртуалізованої мережі Nuage Networks пропонується проводити розрахунок коефіцієнтів  $k_s^{i,k,j}$  розподілу потоку інформації від джерела « $i$ » до адресата « $j$ » між «вузлами-наступниками» « $k$ » з  $M$ -мірної множини  $S_j^{(i)}$  залежно від значень «відстані» (часу передачі пакетів) між ВЗ на маршруті за співвідношенням

$$k_s^{i,k,j} = \left( 1 - \frac{(T_{k,j} + t_{i,k})}{\sum_{\xi \in S_j^{(i)}} (T_{\xi,j} + t_{i,\xi})} \right) / (|S_j^{(i)}| - 1), \quad k \in S_j^{(i)}, \quad (3.3)$$

де  $T_{k,j}$  – час передачі пакетів від «вузла-наступника» « $k$ » до адресата « $j$ » на маршруті ( $i, j$ );  $t_{i,k}$  – «відстань» від джерела « $i$ » до «вузла-наступника» « $k$ »;  $|S_j^{(i)}|$  – потужність множини  $S_j^{(i)}$ .

На рисунку 3.5 і 3.6 наведені відповідно залежності середнього часу  $T_{срд}$  доставки інформаційних пакетів в віртуалізованій мережі Nuage Networks і ймовірність  $Q^{(сч)}$  їхньої доставки за час, що не перевищує допустиме значення, від інтенсивності  $\lambda$  вхідного потоку інформації для повнозв'язного фрагменту віртуалізованої мережі Nuage Networks в умовах мінімальної ( $\rho_{min} = 16$  Кбіт/с), максимальної ( $\rho_{max} = 30$  Кбіт/с), середньої ( $\rho_z = 19$  Кбіт/с) пропускної спроможності каналів зв'язку (умови (I) – суцільні криві),  $\rho_{min} = 14$  Кбіт/с,  $\rho_{max} = 300$  Кбіт/с,  $\rho_z = 70$  Кбіт/с (умови (II) – пунктирні криві) при кількості маршрутів  $M=5$ . Параметром сімейства кривих є метод розподілу мережевого ресурсу в віртуалізованій мережі Nuage Networks («1» – статичний метод, «2» і «3» – відповідно відомий (MDVA) і адаптивний до умов початкового завантаження мережі методи розподілу).

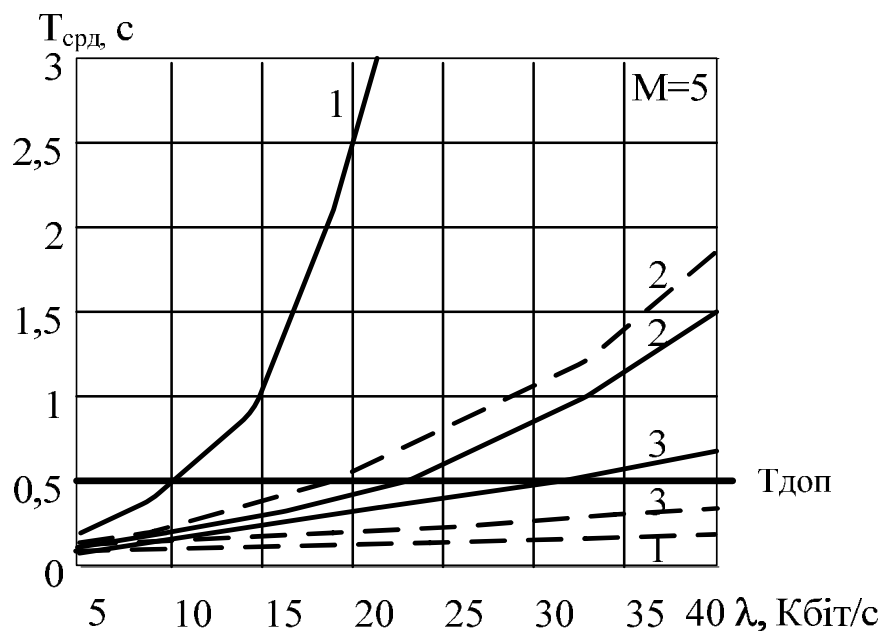


Рисунок 3.5 – Залежності середнього часу  $T_{срд}$  від інтенсивності  $\lambda$

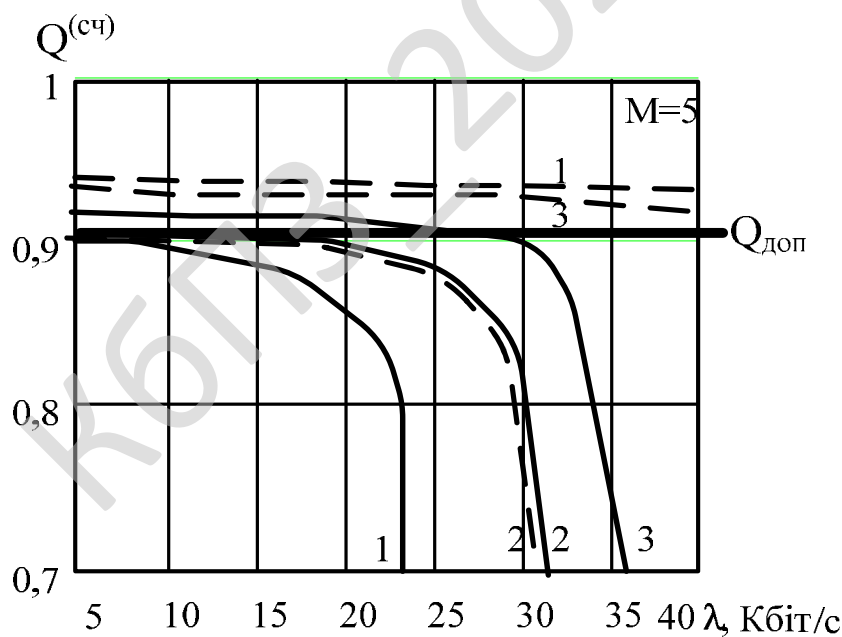


Рисунок 3.6 – Залежності ймовірності  $Q^{(сч)}$  від інтенсивності  $\lambda$

Аналіз залежностей показав, що у деяких ситуаціях адаптивний розподіл потоку інформації дозволяє «заощадити» 2 і більш маршрутів і тим самим додатково понизити ймовірність  $q(N_{баз}, \Delta t)$  порівняно з відомими методами управління, що

свідчить про необхідність оптимізації топології підмережі віртуалізованої мережі Nuage Networks. В роботі пропонується метод знаходження оптимальної множини маршрутів в віртуалізованій мережі Nuage Networks, який включає постановку завдання та процедуру знаходження оптимальної множини маршрутів в віртуалізованій мережі Nuage Networks. При цьому визначення початкового завантаження підмережі дозволяє найкращим чином розподілити потік пакетів, тим самим мінімізуючи  $T_{срд}$  і максимізуючи  $Q^{(сч)}$  в даних умовах.

В подальшому, для передачі інформації про повітряну обстановку та забезпечення «збалансування» загрузки при флуктуаціях трафіку виконується процедура розподілу інформаційних пакетів за оптимальною множиною маршрутів в віртуалізованій мережі Nuage Networks.

Проведемо оцінки ефективності методу забезпечення своєчасності передачі інформації в віртуалізованій мережі Nuage Networks при передачі інформації про повітряну обстановку і вірогідності отриманих результатів. В якості рекомендацій щодо практичного застосування методу забезпечення своєчасності передачі інформації в віртуалізованій мережі Nuage Networks при управлінні повітряним рухом розроблена структура процесу передачі інформації про повітряну обстановку в режимі реального часу. Врахована необхідність здійснювати комплекс заходів (адаптивне кодування, зменшення статистичної і психовізуальної надмірності, адаптивна маршрутизація та ін.), направлених на управління швидкістю передачі інформації про повітряну обстановку і зниження її інтенсивності. Для компенсації втрачених при передачі в каналах зв'язку інформаційних пакетів і забезпечення безперервності відтворення інформації про повітряну обстановку розроблено алгоритм компенсації втрачених інформаційних пакетів. Це дозволить використовувати для управління передачею цифрової інформації про повітряну обстановку протоколи транспортного рівня (RTP і UDP).

Оцінка ефективності роботи методу забезпечення своєчасності передачі інформації в віртуалізованій мережі Nuage Networks по відношенню до відомих методів показала ряд його переваг (ймовірність доведення інформації до одержувача

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

за час, що не перевищує допустиме значення, вище за аналогічну ймовірність відомих методів до 3 разів, середній час доставки інформаційних пакетів менше аналогічної характеристики відомих методів до 15 разів).

Для обґрунтування достовірності отриманих результатів математичного моделювання проведено імітаційне моделювання передачі інформації в повнозв'язній комп'ютерній мережі в умовах застосування бездротових каналів зв'язку (середня пропускна спроможність  $\rho_z = 19 \text{ Кбіт/с}$ ) при різних інтенсивностях вхідного потоку (у діапазоні  $\lambda = [5, \dots, 30] \text{ Кбіт/с}$ ).

За критерієм згоди Персона з рівнем значущості  $\alpha = 0,01$  показано, що число прийнятих за час  $T_{дон}$  інформаційних пакетів а також ймовірність  $Q_{експ}^{(сч)}$  можна вважати розподіленими за нормальним законом.

Одержані довірчі інтервали для оцінок математичного очікування  $Q_{експ}^{(сч)}$ , в які потрапляють «розрахункові» значення  $Q_{\lambda}^{(сч)}$  з довірчою ймовірністю 0,95. Високий ступінь збігу результатів імітаційного і математичного моделювання підтверджує достовірність математичної моделі, використаної для розрахунку ймовірності  $Q^{(сч)}$  доставки інформаційних пакетів за час, що не перевищує допустиме значення.

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

### 3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.7. 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).

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграми потоків даних містять чотири типи елементів:

- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).
- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.



Рисунок 3.7 – Діаграма взаємодії процесів

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

## 4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

### 4.1 Блок-схеми та опис алгоритмів функціонування системи

Під час роботи над магістерською дипломною роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>58</b>

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

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 віртуалізації мережі Nuage Networks.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу 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.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

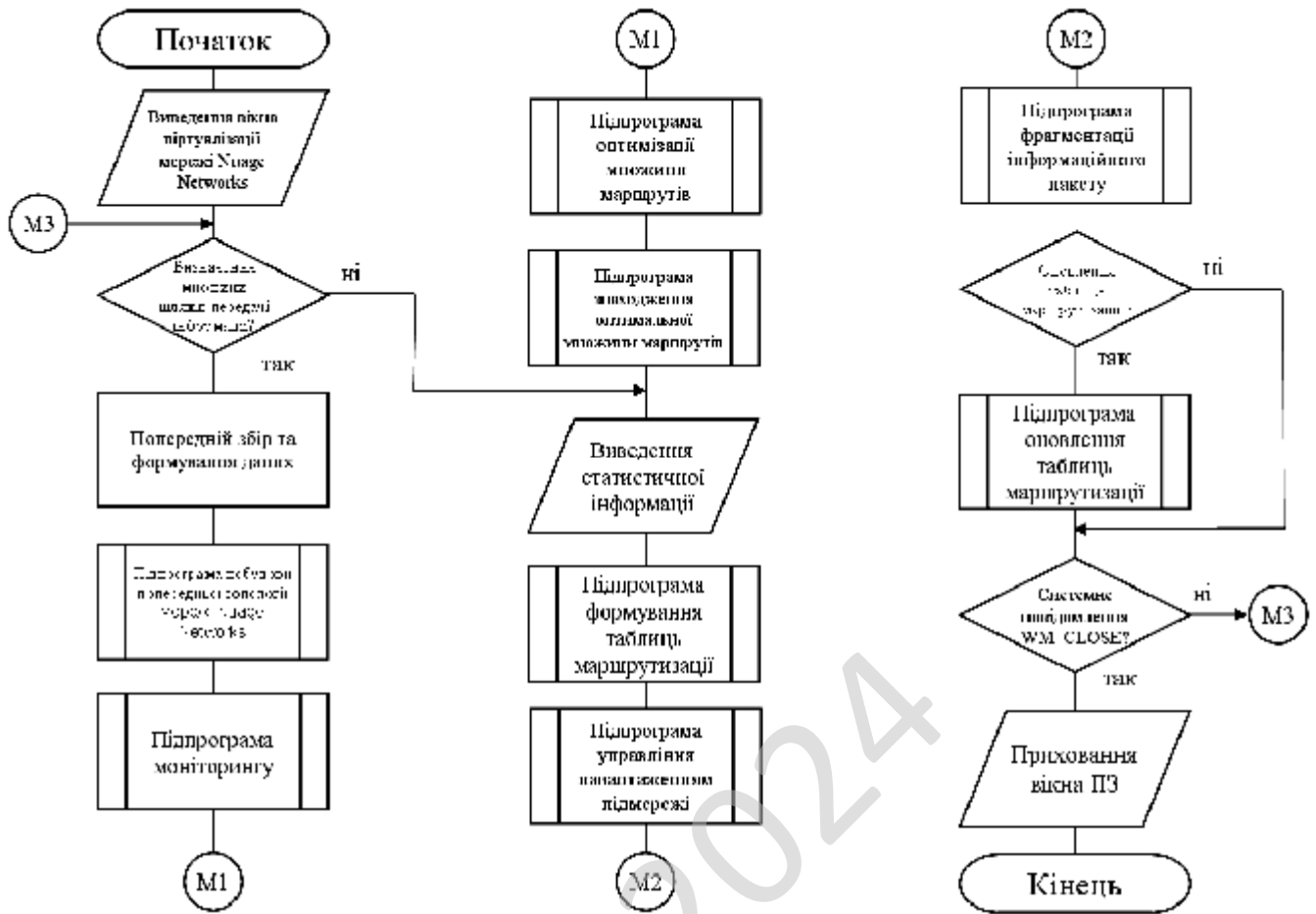


Рисунок 4.1 – Блок-схема основної програми

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки.

Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

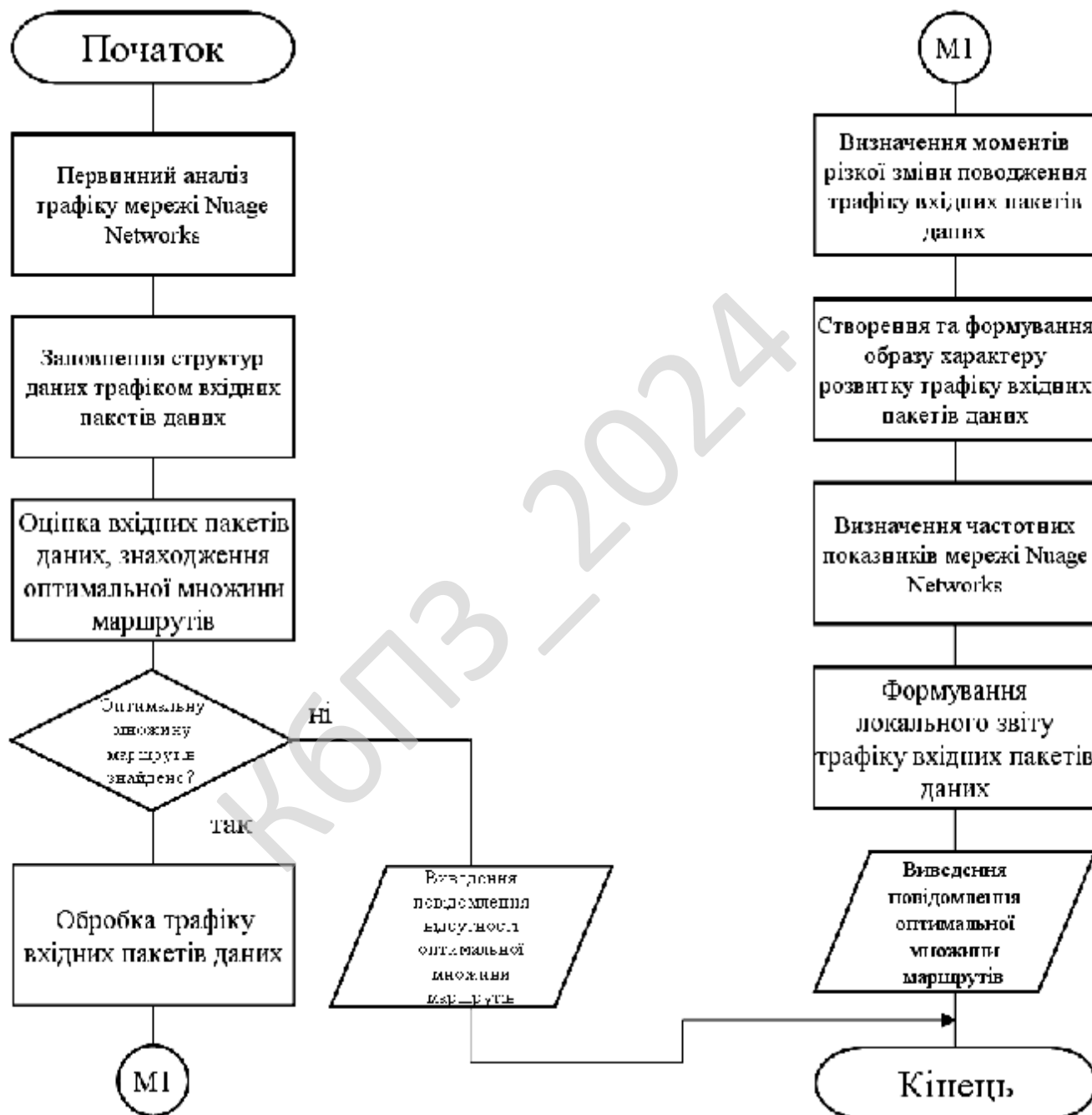


Рисунок 4.2 – Блок-схема роботи підпрограми



мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Асоціації це якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі

					ВКРМ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

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

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

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

Часто при моделюванні буває важливо вказати, скільки об'єктів може бути пов'язано допомогою одного примірника асоціації. Це число називається кратністю (Multiplicity) ролі асоціації та записується або як вираз, значенням якого є діапазон значень, або в явному вигляді.

Вказуючи кратність на одному кінці асоціації, ви тим самим говорите, що на цьому кінці саме стільки об'єктів повинно відповідати кожному об'єкту на протилежному кінці. Кратність можна задати рівною одиниці (1), можна вказати діапазон: "нуль або одиниця" (0..1), "багато" (0 .. \*), "одиниця або більше" (1 .. \*). Дозволяється також вказувати певне число (наприклад, 3). За допомогою списку можна задати і більш складні кратності, наприклад 0. . 1, 3..4, 6 .. \*, що означає "будь-яке число об'єктів, крім 2 і 5".

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з

						<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			66

класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого». Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбованим ромбиком.

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

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

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

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

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

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

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

Діаграма об'єктів в UML це діаграма, що відображає об'єкти та їх зв'язки в певний момент часу. Діаграма об'єктів може розглядатись як окремий випадок діаграми класів, на якій можуть бути представлені як класи, так і екземпляри (об'єкти) класів. Схожою за змістом є діаграма взаємодії (collaboration diagram).

Діаграми об'єктів не мають власної нотації. Оскільки діаграми класів можуть відображати об'єкти, то діаграма класів, на якій відображено лише об'єкти, та не відображено класи, може вважатись діаграмою об'єктів.

Діаграма об'єктів відображає об'єкти та зв'язки в певний момент роботи програми. Об'єкти можуть містити інформацію про власні значення а не про описання. Для відображення загальних шаблонів об'єктів та зв'язків, що можуть багаторазово створюватись під час роботи програми, слід використовувати діаграму взаємодії, яка може відображати характеристики об'єктів та зв'язків. Екземпляр діаграми взаємодії створює діаграму об'єктів.

Діаграма об'єктів не відображає еволюцію системи під час роботи. Натомість, слід використовувати діаграми взаємодії з повідомленнями, або діаграми послідовності.

Діаграма розгортання (deployment diagram) це діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

Система віртуалізації мережі Nuage Networks для автоматизації SDN (Software-Defined Networking) і SD-WAN (Software-Defined Wide Area Network)

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68



## Основні модулі та архітектура

### Модуль оркестрації:

```
import requests

class NuageOrchestrator:
    def __init__(self, vsd_url, username, password):
        self.vsd_url = vsd_url
        self.auth_token = self.authenticate(username, password)

    def authenticate(self, username, password):
        """Аутентифікація на Nuage VSD"""
        auth_endpoint = f"{self.vsd_url}/nuage/api/v5_0/auth"
        response = requests.post(auth_endpoint, auth=(username, password))
        if response.status_code == 200:
            return response.json()["token"]
        else:
            raise Exception("Authentication failed.")

    def create_policy(self, policy_name, description, priority):
        """Створення нової політики трафіку"""
        policy_endpoint = f"{self.vsd_url}/nuage/api/v5_0/policies"
        headers = {"Authorization": f"Bearer {self.auth_token}"}
        policy_data = {
            "name": policy_name,
            "description": description,
            "priority": priority
        }
        response = requests.post(policy_endpoint, headers=headers,
            json=policy_data)
        if response.status_code == 201:
            print(f"Policy '{policy_name}' created successfully.")
        else:
            print(f"Failed to create policy: {response.content}")

    def list_policies(self):
        """Отримання списку політик"""
        policy_endpoint = f"{self.vsd_url}/nuage/api/v5_0/policies"
        headers = {"Authorization": f"Bearer {self.auth_token}"}
        response = requests.get(policy_endpoint, headers=headers)
        if response.status_code == 200:
            return response.json()
```

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70



```

headers = {"Authorization": f"Bearer {self.auth_token}"}
response = requests.post(vrs_config_endpoint, headers=headers)
if response.status_code == 200:
    print(f"VRS configured on host {self.host_ip}.")
else:
    print(f"Failed to configure VRS: {response.content}")
def monitor_vrs(self):
    """Моніторинг стану VRS"""
    vrs_status_endpoint = f"http://{self.host_ip}/nuage/api/v5_0/vrs/status"
    headers = {"Authorization": f"Bearer {self.auth_token}"}
    response = requests.get(vrs_status_endpoint, headers=headers)
    if response.status_code == 200:
        return response.json()
    else:
        raise Exception("Failed to retrieve VRS status.")

```

### Приклад використання

```

# Створення політики
orchestrator = NuageOrchestrator("http://vsd-server-url", "admin", "password")
orchestrator.create_policy("HighPriorityTraffic", "Пріоритет для важливих
сервісів", 1)

# Застосування політики
controller = SDNController("http://vsc-server-url", orchestrator.auth_token)
controller.apply_policy("policy_id_123", "subnet_id_456")

# Моніторинг трафіку
traffic_data = controller.monitor_traffic("subnet_id_456")
print(f"Traffic stats: {traffic_data}")

```

## 4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму Camellia – блоковий шифр на основі мережі Фейстеля. У криптографії, Camellia – це симетричний ключ блоковий шифр із розміром блоку 128 біт і розмірами ключа 128, 192 і 256 біт. Він був розроблений спільно Mitsubishi Electric і NTT з Японії. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу NESSIE і Японським CRYPTREC

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

проект. шифр має рівні безпеки й можливості обробки, порівнянні з Advanced Encryption Standard.

Шифр був розроблений, щоб підходити як для програмних, так і для апаратних реалізацій, від недорогих смарт-карти для високошвидкісних мережних систем. Він є частиною криптографічного протоколу Transport Layer Security (TLS), призначеного для забезпечення безпеки зв'язки в комп'ютерній мережі, такий як Інтернет

Camellia – це шифр Фейстеля з 18 раундами (при використанні 128-бітних ключів) або 24 раундами (при використанні 192- або 256-бітних ключів). Кожні шість раундів застосовується шар логічного перетворення: так звана «FL-функція» або її зворотна. Camellia використовує чотири  $8 \times 8$ -бітних S-блоку із вхідними й вихідними афіними перетвореннями й логічними операціями. Шифр також використовує введення й вивід відбілювання клавiш. Шар дифузiя використовує лінійне перетворення на основі матриці з номером галузей 5.

#### **Аналіз безпеки**

Камелія вважається сучасним надійним шифром. Навіть при використанні параметра меншого розміру ключа (128 біт) вважається неможливим зламати його за допомогою атаки грубої сили на ключі за допомогою сучасних технологій. Немає відомих успішних атак, що значно послабляють шифр. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу NESSIE і Японським CRYPTREC проект. Японський шифр має рівні безпеки й можливості обробки, порівнянні із шифром AES/Rijndael.

Camellia – це блоковий шифр, який може бути повністю визначені мінімальними системами багатомірних багаточленів:

– Камелія (а також AES) S-блоки можуть бути описані системою 23 квадратних рівнянь в 80 членах.

– Розклад ключів можна описати 1120 рівняннями в 768 змінні з використанням 3328 лінійних і квадратичних членів.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

– Увесь блоковий шифр можна описати 5104 рівняннями в 2816 змінні з використанням 14 592 лінійних і квадратичних членів.

– Усього потрібно 6224 рівняння з 3584 змінними з використанням 17 920 лінійних і квадратичних членів.

– Кількість вільних членів становить 11 696, що приблизно таке ж число, що й для AES.

Теоретично, такі властивості можуть дозволити зламати Camellia (і AES) за допомогою алгебраїчної атаки, такий як розширена розріджена лінеаризація, у т Майбутнє за умови, що атака стане можливою.

Хоча Camellia запатентована, вона доступна за безоплатною ліцензією. Це дозволило шифру Camellia стати частиною проекту OpenSSL під ліцензією з відкритим вихідним кодом з листопада 2006 року. Це також дозволило йому стати частиною Mozilla Модуль NSS (Служби мережної безпеки).

Підтримка Camellia була додана в остаточний випуск Mozilla Firefox 3 в 2008 році (за замовчуванням відключене починаючи з Firefox 33 в 2014 році в дусі «Пропозиції по зміні стандартних наборів шифрів TLS, пропонованих браузерами», який був виключено з версії 37 в 2015 році). Pale Moon, відгалуження Mozilla / Firefox, продовжує пропонувати Camellia і розширив свою підтримку, включивши в нього набори Galois / Counter mode (GCM) із шифром, але вилучив GCM знову у випуску 27.2.0, пославшись на очевидну відсутність інтересу до них.

Пізніше, в 2008 році, група розробки релізу FreeBSD оголосила, що цей шифр також був включений в FreeBSD 6.4. Крім того, Йошисато Янагисава додав підтримку шифру Camellia у дисковий клас зберігання geli FreeBSD.

У вересні 2009 року GNU Privacy Guard додала підтримку Camellia у версії 1.4.10.

Veracrypt (відгалуження Truecrypt) включав Camellia як один з підтримуваних алгоритмів шифрування.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Крім того, різні популярні бібліотеки безпеки, такі як Crypto ++, Gnutls, mbed TLS і Openssl також включають підтримку Camellia.

26 березня 2013 р. було оголошено, що Camellia була знову обрана для включення в новий список рекомендованих шифрів для електронного уряду Японії як єдиний 128-бітний алгоритм блокового шифрування, розроблений у Японії. Це збігається з тим, що список CRYPTREC обновляється вперше за 10 років. Вибір був заснований на високій репутації Camellia у плані простоти придбання, а також характеристик безпеки й продуктивності, порівнянних з такими з Advanced Encryption Standard (AES). Камелія залишається незмінною у своєму повному втіленні. Неможлива диференціальна атака на Camellia з 12 раундами без шарів FL / FL дійсно існує.

### **Продуктивність**

S-блоки, використовувані Camellia, мають структуру, аналогічну S-блоку AES. У результаті можна прискорити реалізацію програмного забезпечення Camellia за допомогою наборів команд ЦП, розроблених для AES, таких як x86 AES-NI.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Процес розрахунків здійснюється в консольному додатку з передачею отриманих результатів у користувацький інтерфейс. На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання магістерської дипломної роботи. Розроблене програмне забезпечення віртуалізації мережі Nuage Networks складається з наступних функціональних блоків:

- Навігаційне меню: Файл; Гіпервізор; Налаштування; Довідка.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

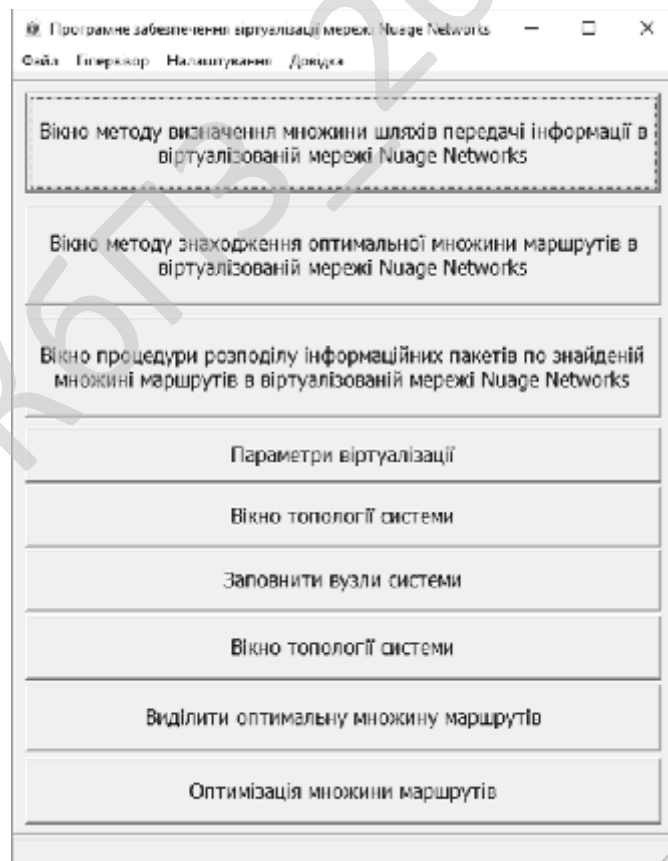


Рисунок 5.1 – Головне вікно розробленого ПЗ

					ВКРМ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

– Блоку функціональних кнопок ПЗ: Вікно методу визначення множини шляхів передачі інформації в віртуалізованій мережі Nuage Networks; Вікно методу знаходження оптимальної множини маршрутів в віртуалізованій мережі Nuage Networks; Вікно процедури розподілу інформаційних пакетів по знайдений множині маршрутів в віртуалізованій мережі Nuage Networks; Параметри віртуалізації; Вікно топології системи; Заповнити вузли системи; Вікно топології системи; Виділити оптимальну множину маршрутів; Оптимізація множини маршрутів.

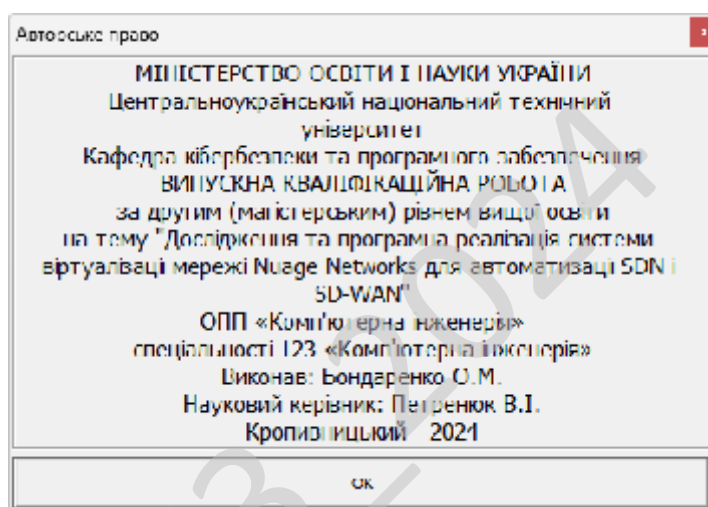


Рисунок 5.2 – Вікно розробника ПЗ

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;

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

- 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.

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

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

У цьому випадку формуються тестові варіанти, в яких:

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

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.
- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

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

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.
- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

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

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

Проводилось тестування чорної скриньки. Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме  $10^{10}$ . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чию поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

- Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).
- Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

						<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			79

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс;
- Сформулювати такі очікувані результати, які з високою ймовірністю є елементами набору ОТ.

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 – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно.

Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>80</b>

становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareestruvatisya), заплативши авторіві певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

КБПЗ\_2024

					VKPM-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

*Метою розробки є дослідження та програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.*

*Об'єктом дослідження є процес віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.*

*Предметом дослідження є методи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.*

*Методи дослідження базуються на методах віртуалізації 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 віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

– Розроблено вітчизняний продукт віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

## 7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

### 7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN можуть зацікавити аудиторію, подану на рисунку 7.1.

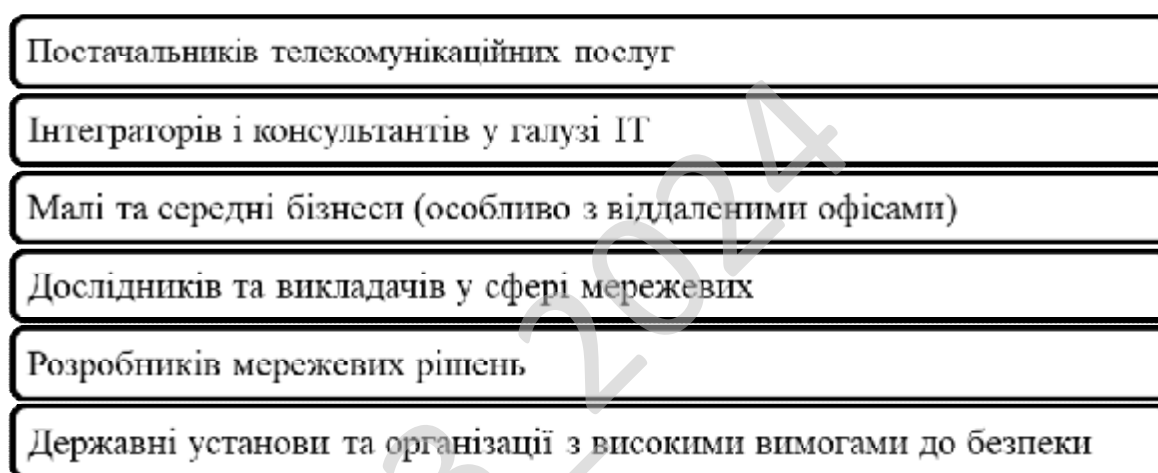


Рисунок 7.1 – Цільова аудиторія

Таке дослідження може також допомогти компаніям у кращому розумінні вартості й переваг використання SDN і SD-WAN, що сприятиме ефективнішому управлінню мережами.

### 7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості програмної реалізації системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN за допомогою методів експертних оцінок можна застосувати метод бальної оцінки.

Вибираємо основні критерії, за якими будемо оцінювати привабливість рішення, наприклад:

- ефективність автоматизації;
- масштабованість;
- безпека;
- зручність управління;
- вартість впровадження;
- сумісність з існуючою інфраструктурою.

До експертів долучаємо – мережових інженерів, спеціалістів з безпеки, представників бізнесу, кожен з яких має оцінити систему за обраними критеріями.

Таблиця 7.1 – Зведені результати опитування

Критерії	Експерт 1	Експерт 2	Експерт 3	Середній бал
Ефективність автоматизації	9	8	7	8.0
Масштабованість	8	9	8	8.3
Безпека	7	8	8	7.7
Зручність управління	9	8	9	8.7
Вартість впровадження	6	7	6	6.3
Сумісність	8	9	8	8.3

Кожен експерт оцінює критерії за бальною шкалою (від 1 до 10). Після цього збираються середні оцінки для кожного критерію.

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.

Далі ведемо розрахунок інтегрального показника привабливості. Застосовуємо вагові коефіцієнти до критеріїв залежно від їхньої важливості (визначені експертами).

Ефективність автоматизації – вага 0.2

Масштабованість – вага 0.15

Безпека – вага 0.2

Зручність управління – вага 0.1

Вартість впровадження – вага 0.15

Сумісність – вага 0.2

$$\text{Привабливість} = (8.0 \times 0.2) + (8.3 \times 0.15) + (7.7 \times 0.2) + (8.7 \times 0.1) + (6.3 \times 0.15) + (8.3 \times 0.2) = 7.93$$

На основі отриманих результатів можна зробити висновок про привабливість проєкту. Якщо інтегральний показник високий ( $>7$ ), це свідчить про перспективність впровадження системи Nuage Networks для автоматизації SDN та SD-WAN у поточному контексті. Такий метод дає об'єктивний підхід до оцінки на основі експертних знань та досвіду, що допоможе обґрунтувати рішення про впровадження або модифікацію системи.

### 7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN найкраще підходять такі методи як метод калькуляції витрат та метод порівняльної оцінки.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>85</b>

Метод калькуляції витрат полягає у детальному аналізі всіх витрат, пов'язаних із впровадженням системи. Він включає: витрати на ліцензії та програмне забезпечення, витрати на апаратне забезпечення, витрати на оплату праці, витрати на навчання персоналу, витрати на підтримку та технічне обслуговування. Після розрахунку загальної суми витрат цей метод надає чітке уявлення про вартість, але потребує детального обліку всіх компонентів.

Метод порівняльної оцінки базується на порівнянні витрат з аналогічними проектами або рішеннями. Важливо знайти компанії, які вже використовують Nuage Networks або інші системи SDN/SD-WAN, і порівняти їх витрати на впровадження. Основні аспекти порівняльної оцінки: вивчення витрат конкурентів або аналогічних організацій, аналіз ринку, щоб визначити середню вартість впровадження рішень SDN та SD-WAN, врахування особливостей, таких як специфіка поточної інфраструктури і потреби в масштабуванні. Цей метод корисний, коли є доступ до даних аналогічних проєктів і коли є потреба в швидкій оцінці.

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

Оцінка економічної ефективності впровадження програмної реалізації системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN може включати наступні розрахунки.

Впровадження системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN забезпечує значну економічну ефективність. За рік компанія може отримати чистий дохід у розмірі 103,000 USD з ROI близько 98.1%. Це підтверджує доцільність інвестицій у цю технологію, враховуючи зменшення витрат та збільшення доходів.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

Таблиця 7.2 – Економічна ефективність від реалізації

<b>ВИТРАТИ НА ВПРОВАДЖЕННЯ</b>	<b>ОПЕРАЦІЙНІ ВИТРАТИ</b>
Ліцензії на програмне забезпечення: 50,000 €	Річні витрати на підтримку та обслуговування: 10,000 €
Апаратне забезпечення (сервера, мережеве обладнання): 30,000 €	Зменшення витрат на ІТ-персонал (автоматизація процесів): -5,000 €
Навчання персоналу: 10,000 €	Зменшення витрат на зв'язок (завдяки оптимізації SD-WAN): -8,000 €
Послуги інтеграції: 15,000 €	
Загальні початкові витрати: 105,000 €	Річні операційні витрати: 10,000 € - 5,000 € - 8,000 € = -3,000 € (економія)
<b>ПЕРЕВАГИ ВІД ВПРОВАДЖЕННЯ</b>	<b>ЕКОНОМІЧНА ВИГОДА</b>
Зменшення часу на налаштування мережі: раніше 10 днів, тепер 2 дні. Це дозволяє обробляти більше запитів від клієнтів.	Потенційний дохід від нових клієнтів: 100,000 € на рік.
Покращення продуктивності: завдяки автоматизації мережевих процесів компанія може обробляти на 30% більше трафіку без збільшення витрат на апаратне забезпечення.	Чистий дохід = Економія + Збільшення доходів - Операційні витрати Чистий дохід = 3,000 + 100,000 = 103,000 € Оцінка повернення інвестицій (ROI) - 98.1%

### 7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Алгоритм просування проєкту програмної реалізації системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN. Цей алгоритм включає етапи, які допоможуть ефективно запуснути та просунути проєкт на ринку подано на

1. Дослідження ринку	Аналіз конкурентів: вивчення основних конкурентів у галузі SDN і SD-WAN, їхніх продуктів, цінних стратегій та ринкових позицій.
	Оцінка потреб цільової аудиторії: визначення потреб та проблем потенційних користувачів, щоб налаштувати продукт відповідно до їхніх вимог.
2. Розробка унікальної пропозиції	Формулювання унікальної торгової пропозиції (USP): визначення ключових переваг системи Nuage Networks, які відрізняють її від конкурентів (наприклад, масштабованість, безпека, простота управління).
	Визначення цільової аудиторії: сегментація ринку на основі типу бізнесу, розміру компанії та специфічних потреб у мережевих технологіях.
3. Створення маркетингової стратегії	Розробка контент-стратегії: створення корисного контенту (статті, вебінари, випадки успіху), що підкреслює переваги автоматизації SDN і SD-WAN.
	Вибір каналів просування: визначення ефективних каналів комунікації (соціальні мережі, email-маркетинг, професійні платформи).
4. Розробка демонстраційних версій та тестування	Створення безкоштовних пробних версій: надання можливості потенційним клієнтам спробувати продукт безкоштовно.
	Збір відгуків: отримання зворотного зв'язку від користувачів для вдосконалення продукту перед офіційним запуском.
5. Запуск рекламних кампаній	Цільові рекламні кампанії: використання онлайн-реклами (Google Ads, соціальні мережі) для залучення трафіку на веб-сайт продукту.
	Вебінари та конференції: організація заходів для демонстрації продукту та обговорення переваг автоматизації мереж.
6. Партнерство та співпраця	Співпраця з IT-компаніями та інтеграторами: укладення партнерських угод для розширення охоплення ринку та збільшення довіри до продукту.
	Участь у галузевих виставках і конференціях: активна участь у заходах для демонстрації продукту та налагодження контактів.
7. Моніторинг і аналіз ефективності	Відстеження ключових показників ефективності (KPI): аналіз показників, таких як трафік на сайт, конверсія, кількість лідів, продажі.
	Оптимізація кампаній: на основі отриманих даних коригування маркетингових стратегій для підвищення їхньої ефективності.
8. Післяпродажна підтримка та обслуговування	Надання технічної підтримки: забезпечення користувачів підтримкою після впровадження продукту.
	Збір відгуків та постійне вдосконалення продукту: активне залучення клієнтів до покращення продукту через зворотний зв'язок.

Рисунок 7.2 – Алгоритм просування проекту

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

Цей алгоритм допоможе систематично підходити до просування проекту програмної реалізації системи віртуалізації мережі Nuage Networks, забезпечуючи максимальну ефективність і досягнення поставлених бізнес-цілей. Використання всіх етапів дозволить не лише залучити нових клієнтів, але й підтримувати їхній інтерес і довіру до продукту.

## 7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту та шляхів реалізації проекту програмної реалізації системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN є критично важливими для забезпечення ефективності продажів і задоволення потреб клієнтів. І можлива лише завдяки наступним діям:

- проаналізуйте, які канали (прямий продаж, дистриб'ютори, онлайн-продажі) приносять найбільше доходу та якими є витрати на їх утримання.
- зрозумійте, які канали найкраще відповідають потребам вашої цільової аудиторії, зокрема для малих і середніх підприємств (МСП) та великих корпорацій.
- встановіть партнерські відносини з компаніями, які вже мають досвід у продажу ІТ-рішень, зокрема в галузі SDN і SD-WAN.
- оптимізуйте веб-сайт для електронної комерції, щоб клієнти могли легко купувати програмне забезпечення онлайн, і розгляньте можливість використання платформ для SaaS.
- впровадження системи управління взаємовідносинами з клієнтами (CRM) для автоматизації процесів продажу, управління лідами та збереження інформації про клієнтів.
- використання інструментів для автоматизації email-кампаній, сегментації аудиторії та збору аналітики для покращення комунікацій.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

– розробіть різноманітні комерційні пропозиції, які відповідають різним сегментам ринку (наприклад, пакети для малих підприємств, корпоративні рішення).

– надайте можливість потенційним клієнтам протестувати продукт безкоштовно на обмежений час для збільшення конверсії.

– забезпечте якісне навчання для клієнтів щодо використання продукту, а також ефективну технічну підтримку.

– активно збирайте зворотний зв'язок від клієнтів і впроваджуйте зміни, виходячи з їхніх потреб.

– використовуйте цифровий маркетинг для залучення уваги до вашого продукту, використовуючи платформи, які ваші клієнти найбільше використовують (LinkedIn, Facebook, Google Ads).

– створюйте цінний контент (статті, блоги, вебінари), щоб залучити потенційних клієнтів і продемонструвати експертизу у вашій галузі.

Забезпечення оптимізації каналів збуту та шляхів реалізації проєкту узагальнено представлено на рисунку 7.3.

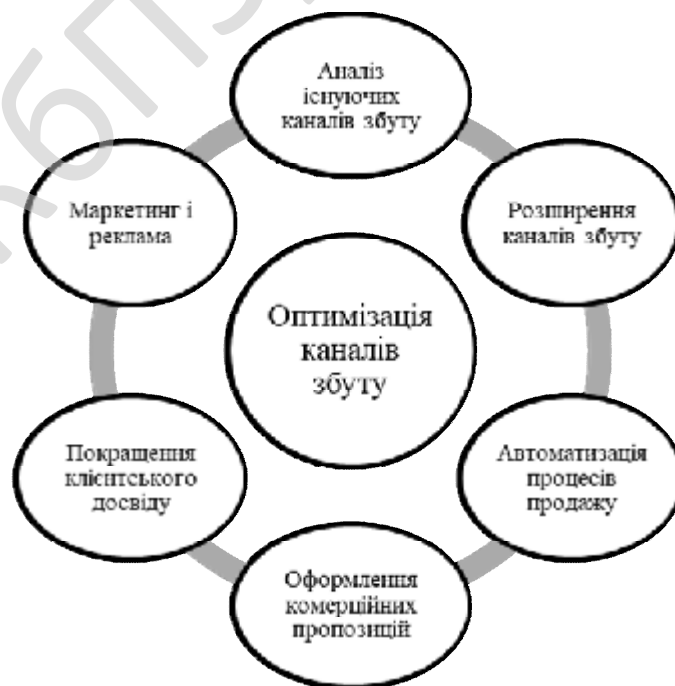


Рисунок 7.3 – Напрями оптимізації каналів збуту

Забезпечення оптимізації каналів збуту та шляхів реалізації проєкту програмної реалізації системи віртуалізації мережі Nuage Networks вимагає системного підходу, що включає аналіз, автоматизацію, покращення клієнтського досвіду та активну комунікацію. Використовуючи ці рекомендації, ви зможете значно підвищити ефективність продажів і задовольнити потреби ваших клієнтів.

### 7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN включають ряд стратегічних, технічних та організаційних аспектів наведені на рисунку 7.4.

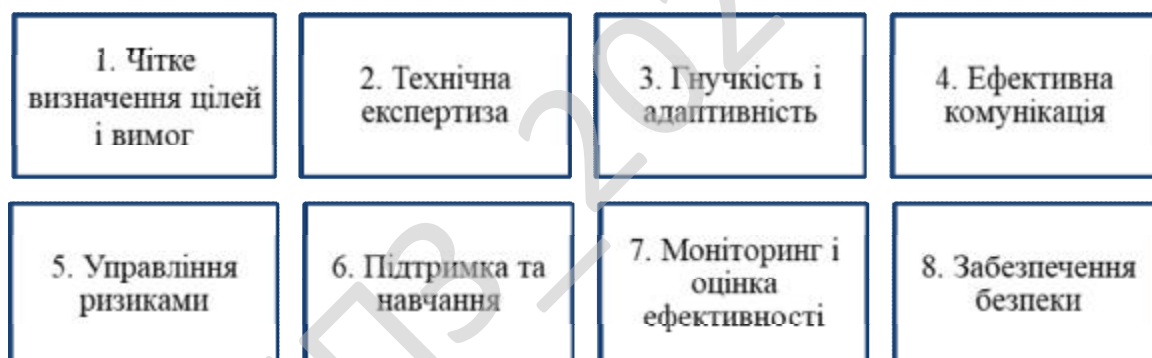


Рисунок 7.4 – Ключові фактори успіху проєкту

Успіх проєкту програмної реалізації системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN залежить від комплексного підходу, що включає чітке визначення цілей, технічну експертизу, гнучкість, ефективну комунікацію та підтримку. Врахування цих факторів допоможе забезпечити успішне впровадження і задоволення потреб клієнтів.

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

В охорону праці включають санітарно-гігієнічні, лікувально-профілактичні та організаційно-технічні системи правових і соціально-економічних заходів.

В кожній ІТ компанії є трудові відносини з працівниками. Згідно закону України “Про охорону праці” [1] кожна компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

- Інструкцій з охорони праці по кожній професії і загальні.
- Положення про охорону праці.
- Накази з охорони праці.
- Журнали реєстрації та інструктажу.

Роботодавець створює відділ який працює відповідно до типового положення, яку затверджується центральним органом виконавчої влади і забезпечує виконання вимог державної політики у сфері охорони праці.

За недотриманням вимог, керівники ІТ компаній можуть бути притягнуті до відповідальності, яка виглядає у виді накладання штрафу. Якщо в результаті порушення умов охорони праці є постраждалі працівники то керівні особи ІТ компаній притягуються до кримінальної відповідальності.

Законом України “Про охорону праці” регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [2], яким затверджено нормативно-правовий

					ВКРМ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [3]. Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів. Ці шкідливі фактори можуть привести до професійних захворювань. Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [3], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

## 8.2. Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машин (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т. ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат.

### 8.3 Аналіз умов праці на робочому місці фахівця

Робота програміста пов'язана з постійною роботою на ЕОМ, яка відбувається у кімнаті розмірами 4,4 м×6,2 м×2,9 м. Одна з її більших стін має шість двостулкових вікон, розмірами 2 м×1,8 м, які виходять на північний захід. Вікна розташовані рівномірно по всій довжині стіни. Підлога в кімнаті покрита

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

ліноліумом, всі стіни пофарбовані світло оранжевого кольору до висоти 2,8 м, а далі підвісна стеля. Уздовж стін розташовані комп'ютерні столи. На них розташовуються 2 персональні комп'ютери й інша оргтехніка (сканер принтери, телефони й ксерокс). Столи мають пластикове покриття. Габарити їхньої робочої поверхні 1255 мм×845 мм. Висота столів 760 мм. Висота стільців від рівня підлоги становить 430 мм.

Згідно НПАОП 0.00 – 1.28 – 10 «Правила охорони праці під час електронно-обчислювальних машин» площа повинна задовольняти умові – не менш 6 м<sup>2</sup> на одне робоче місце. Кратність повітрообміну в приміщенні вузла також регламентується ДСанПіН 3.3.2.007-98 [3], вона повинна становити 20 м<sup>3</sup>/годину на одне місце. Виконання даних вимог забезпечить підтримку в приміщенні вузла оптимального значення вологості й складу повітря.

Відповідно ДБН В.2.5 – 28 – 2006 [4] роботу програміста можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення вузла можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при сполученому висвітленні), повинен становити 0,5%, освітленість при штучному висвітленні повинна становити 300 лк.

За результатами виміру освітленості відділом охорони праці величина освітленості від системи загального штучного висвітлення лежить у межах 200-250 лк, що не відповідає вимогам, які пред'являються до приміщення.

Відповідно ДСанПіН 3.3.2.007-98 [3] рівні звукового тиску в робочому приміщенні не повинні перевищувати в октавних смугах із середньо геометричними частотами наступних значень, наведених у таблиці 8.1.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

У приміщенні перебувають наступні джерела шуму: електродвигуни внутрішнього вентилятора ЕОМ; працюючі принтери; працюючі дисководи.

Таблиця 8.1 – Допустимі спектри рівнів звукового тиску

Робоче місце	Рівень звукового тиску, дБ, в октавних смугах із середньгеометричними частотами, Гц								Рівень звуку і еквівалентний рівень звуку, дБА
	63	125	250	500	1000	2000	4000	8000	
Приміщення конструкторських бюро, програмістів обчислювальних машин, лабораторій для теоретичних робіт і опрацювання експериментальних даних, прийому хворих в медпунктах	71	61	54	49	45	42	40	38	50

Шум, вироблений вентилятором можна класифікувати як постійний, всі інші джерела шуму, як імпульсні. Відповідно паспорта на приміщення рівень звуку, Дб(А), обмірюваний за шкалою (А) шумоміра досяг величини 28,3 Дб(А) при роботі всього встаткування вузла, включаючи й ксерокс. Це дозволяє зробити висновок про відповідність рівня звуку в приміщенні вимогам нормативних актів.

Ергономічні вимоги до робочого місця працюючого з ВДТ ЕОМ і ПЕОМ нормуються НПАОП 0.00 – 1.28 – 10. Оптимальне положення тіла того, що працює забезпечується відповідною конструкцією робочого місця, а також регуляцією висоти робочої поверхні, сидіння, простори й підставки для ніг.

Даного місця програміста не мають регульованих параметрів. Відмінності реальних параметрів робочого місця від параметрів відповідні вимоги нормативного акту дані в таблиці 8.2.

У дужках зазначені реальні значення параметрів робочого місця; всі вони не відповідають параметрам, зазначеним у стандарті.

Таблиця 8.2 – Відмінності реальних параметрів робочого місця від параметрів відповідні вимоги нормативного акту

Ріст людини, см	Висота робочої поверхні мм,	Висота простору для ніг, мм	Висота робочого сидіння, мм
175	765(740)	655(600)	450(440)

Параметри мікроклімату можуть мінятися в широких межах, тоді як необхідною умовою життєдіяльності людини є підтримка сталості температури тіла завдяки властивості терморегуляції, тобто здатності організму регулювати віддачу тепла в навколишнє середовище. У приміщеннях, де встановлені комп'ютери, повинні дотримуватися певні параметри мікроклімату. У санітарних нормах ДСН 3.3.6.042 – 99 [4] встановлені величини параметрів мікроклімату, що створюють комфортні умови. Ці норми встановлюються в залежності від пори року, характеру трудового процесу і характеру виробничого приміщення (див. табл. 8.3).

Таблиця 8.3 – Параметри мікроклімату для приміщень, де встановлені комп'ютери

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні	22 – 24°C
	Відносна вологість	40 – 60%
	Швидкість руху повітря	до 0,1 м/с
Теплий	Температура повітря в приміщенні	23 – 25°C
	Відносна вологість	40 ... 60%
	Швидкість руху повітря	0,1 ... 0,2 м / с

## 8.4 Розробка заходів з поліпшення охорони праці

Провівши аналіз умов праці в розглянутому вище приміщенні, було отримано наступні результати:

- значення мікроклімату в приміщенні не перевищує норму;
- розрахунки розміру робочого місця на одного працівника відповідають нормі;
- рівень шуму в приміщенні не становить вище норми.

З вище перерахованих результатів можна зробити висновок, що основний вплив на продуктивність ІТ-спеціалістів є його психологічний стан. Тому є доцільним зменшити рівень стресу на робочому місці.

Рекомендовані наступні заходи: За потреби особливої концентрації уваги під час виконання робіт суміжні робочі місця операторів необхідно відділяти одне від одного перегородками висотою 1,5–2м. Конструкція робочого місця користувача персонального комп'ютера має забезпечити підтримання оптимальної робочої пози офісного працівника. Конструкція робочого столу має відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів. Висота робочої поверхні робочого столу має регулюватися в межах 680-800 мм, а ширина і глибина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600-1400мм, глибина – 800-1000мм). Робочий стіл повинен мати простір для ніг заввишки не менше ніж 600 мм, завширшки не менше ніж 500 мм, завглибшки (на рівні колін) не менше ніж 450 мм, на рівні простягнутої ноги не менше ніж 650 мм. Робочий стілець має бути підйомно-поворотним, регульованим за висотою, з кутом і нахилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край – заокругленим [5].

					ВКРМ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

## 8.5 Розрахункова частина

Заземлення – це з'єднання між електричним обладнанням і землею, завдяки якому утворюється нейтральна точка. Вона необхідна для захисту життя і здоров'я людини за рахунок передачі надлишку електроструму з мережі у ґрунт.

Крім того, заземлення сприяє створенню точки відліку для вимірювання електричного потенціалу в системі – це важливо для забезпечення правильної роботи обладнання, а також виявлення та усунення несправностей.

Початкові дані для розрахунку захисного заземлення:

- тип верхнього шару ґрунта – чорнозем,
- тип нижнього шару ґрунта – глина (питомий опір  $\rho_2 = 40 \text{ Ом}\cdot\text{м}$ ),
- умовна товщина верхнього шару ґрунта:  $H=0,5 \text{ м}$ .
- відстань між вертикальними заземлювачами  $A=1,7 \text{ м}$ .
- глибина закладення горизонтального контура заземлення  $t=0,6 \text{ м}$ .
- опір заземлювача, який нормується:  $R_{3Н} = 4 \text{ Ом}$ .

Необхідно визначити кількість вертикальних заземлювачів та довжину металевої смуги. Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток  $80 \times 50 \times 6 \text{ мм}$ , (згідно з ДСТУ 8769:2018 «Кутики сталеві гарячекатані нерівнополичні. Сортамент») довжиною  $L=1,7 \text{ м}$ , та горизонтальний електрод – металева смуга з перетином  $60 \cdot 5 \text{ мм}$ . Напруга –  $220/380 \text{ В}$ . Розрахункова схема розташування заземлюючих електродів – у ряд (рис. 8.1).

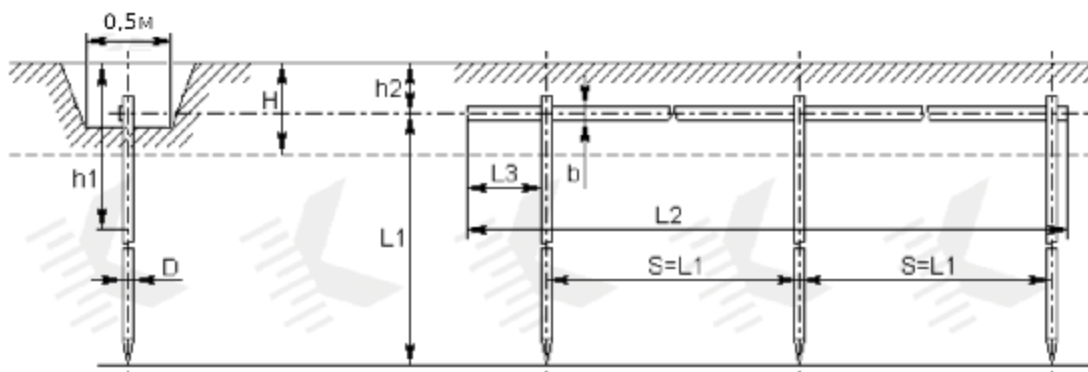


Рисунок 8.1 – Схема штучного заземлення

					ВКРМ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

Розрахунок проведемо за допустимим опором розтіканню струму заземлювача.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,6+1,7/2=1,45 \text{ м.}$$

Розрахунковий питомий опір ґрунта (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунта):

$$\rho = \psi \rho_2 = 1,36 \cdot 40 = 54,5 \text{ Ом} \cdot \text{м.}$$

де  $\psi = 1,36$  – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багат шаровому ґрунті [12];  $\rho_2 = 40 \text{ Ом} \cdot \text{м.}$  – табличне значення питомого опору нижнього шару ґрунта (глина) [12].

Еквівалентний діаметр вертикального електрода (кутка) [12]:

$$D_{\text{в}} = 0,95 \cdot K = 0,95 \cdot 65 = 59,85 \text{ мм.} = 0,062 \text{ м.}$$

де  $K = (80+50)/2 = 65 \text{ мм.}$  – середній розмір полиці металевих кутків.

Відношення  $A/L = 1,7/1,7 = 1$

Опір розтіканню електричного струму одного електрода вертикального заземлювача з урахуванням заглиблення заземлювача [8]:

$$\begin{aligned} R_0 &= 0,366 \cdot (\rho/L) \cdot [\lg(2L/D_{\text{в}}) + (1/2) \cdot \lg((4T+L)/(4T-L))] = \\ &= 0,366 \cdot (54,5/1,7) \cdot [\lg(2 \cdot 1,7/0,062) + (1/2) \cdot \lg((4 \cdot 1,45 + 1,7)/(4 \cdot 1,45 - \\ &1,7))] = 21,9 \text{ Ом.} \end{aligned}$$

Визначаємо коефіцієнт екранування вертикальних електродів  $K_{\text{ев}} = 0,8$  при орієнтовній кількості вертикальних електродів, яке дорівнює 5 [8].

Визначаємо необхідну кількість вертикальних електродів заземлювача (без врахування горизонтального заземлювача), при  $R_{\text{ЗН}} = 4 \text{ Ом.}$

$$T = K_0 \cdot (K_{\text{ев}} \cdot K_{\text{ЗР}}) = 21 \cdot 9 \cdot (0,8 \cdot 4) = 604,8 \approx 7 \text{ шт.}$$

Визначаємо довжину з'єднуючої смуги:

$$L_{\text{П}} = 1,05 \cdot A \cdot N = 1,05 \cdot 1,7 \cdot 7 = 12,22 \approx 12 \text{ м.}$$

Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору ґрунта  $K_{\text{П}}$  [8]:

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

$$R_{\Pi} = 0,366(\rho \cdot K_{\Pi} / L_{\Pi}) \lg(2 \cdot L_{\Pi}^2 / (B \cdot t)) =$$

$$= 0,366 \cdot (40 \cdot 5 / 12) \cdot \lg((2 \cdot 12^2) / (0,06 \cdot 0,6)) = 23,5 \text{ Ом.}$$

де  $K_{\Pi} = 5$  – табличне значення кліматичного коефіцієнта питомого опору ґрунта для відповідної кліматичної зони для з'єднуючої полоси [8]:

$B = 60 \text{ мм.} = 0,06 \text{ м.}$  – ширина з'єднуючої смуги (задана).

Загальний опір розтіканню електричного струму заземлювача [8]:

$$R = (R_0 \cdot R_{\Pi}) / (R_0 \cdot \eta_{\Pi} + N \cdot R_{\Pi} \cdot K_{ев}) =$$

$$= (21,9 \cdot 23,5) / (21,9 \cdot 0,75 + 9 \cdot 23,5 \cdot 0,8) = 3,48 \text{ Ом.}$$

де  $\eta_{\Pi} = 0,75$  – табличне значення коефіцієнта екранування з'єднуючої полоси [8].

Умова  $R \leq R_{3н}$  виконується ( $3,48 \leq 4$ ).

Оскільки при 7 вертикальних електродах  $R$  суттєво менше  $R_{3н}$ , зменшимо кількість вертикальних електродів  $N$  до 6 і виконаємо перерахунок. У результаті остаточно отримали:  $R = 3,98 \text{ Ом.}$  при кількості вертикальних електродів  $N = 6$ .

КБПЗ-2024

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

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 віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.
- Досліджена система віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.
- На основі отриманих результатів досліджень створена програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

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.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

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 Camellia.

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.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>103</b>

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бондаренко О.Д. Дослідження та програмна реалізація системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
2. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
3. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
4. 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.
5. 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.
6. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
7. 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.
8. 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.

9. 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.

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

11. 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.

12. 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.

13. 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.

14. 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.

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

20. 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.

21. 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.

22. 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.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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

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

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

36. 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.

					ВКРМ-123.24.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

37. 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.

38. 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.

39. 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.

40. 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).

41. 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

42. 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.

43. 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.

44. 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.

45. 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.

46. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

47. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

48. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

49. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

50. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

51. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

					<b>ВКРМ-123.24.0005.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

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

					<b>ВКРМ-123.24.0005.00.00.ТЗ</b>		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Бондаренко О.Д.				Літ.	Аркуш	Аркушів
Перевірів	Петренко В.І.						
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М		
Затв.	Смірнов О.А.						

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випуск 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 віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN.

## 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-123.24.0005.00.00.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 віртуалізації мережі Nuage Networks для автоматизації SDN і SD-WAN;
- 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.

					<b>ВКРМ-123.24.0005.00.00.ТЗ</b>	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-123.24.0005.00.00.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 повинні бути розглянуті шкідливі і небезпечні фактори при роботі з комп'ютером.

					ВКРМ-123.24.0005.00.00.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 – 110 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.

					<b>ВКРМ-123.24.0005.00.00.ТЗ</b>	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 віртуалізації мережі Nuage Networks для автоматизації SDN і SD-  
WAN*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 18

Літера: РП

Кропивницький – 2024 року

### Основна програма

```

import requests
import json

from backup_restore import BackupRestore
from device_management import DeviceManagement
from routing_automation import RoutingAutomation
from network_segmentation import NetworkSegmentation
from network_services_orchestration import NetworkServicesOrchestration
from sdwan_scaling import SDWANScaling
from user_management import UserManagement
from network_monitoring import NetworkMonitoring
from configuration_automation import ConfigurationAutomation
from system_integration import SystemIntegration
from analytics_reporting import AnalyticsReporting
from security_policy_management import SecurityPolicyManagement

class NuageNetworksAPI:
    def __init__(self, base_url, username, password, enterprise):
        self.base_url = base_url
        self.username = username
        self.password = password
        self.enterprise = enterprise
        self.headers = {'Content-Type': 'application/json'}
        self.session = requests.Session()
        self.session.headers.update(self.headers)

    def authenticate(self):
        # Аутентифікація до Nuage Networks
        url = f"{self.base_url}/me"
        auth_data = {
            "username": self.username,
            "password": self.password,
            "enterprise": self.enterprise
        }
        response = self.session.post(url, data=json.dumps(auth_data))
        if response.status_code == 200:
            # Збереження токена аутентифікації
            token = response.json()['token']
            self.headers['Authorization'] = f'Bearer {token}'
            self.session.headers.update(self.headers)
        else:
            raise Exception(f"Authentication failed: {response.status_code}
{response.text}")

    def create_domain(self, name, description, domain_type="L3"):
        # Створення домену SDN
        url = f"{self.base_url}/domains"
        domain_data = {
            "name": name,
            "description": description,
            "type": domain_type
        }
        response = self.session.post(url, data=json.dumps(domain_data))
        if response.status_code == 201:
            return response.json()
        else:
            raise Exception(f"Domain creation failed: {response.status_code}
{response.text}")

    def create_l2_domain(self, name, description):
        # Створення L2 домену
        return self.create_domain(name, description, domain_type="L2")

    def create_vport(self, domain_id, name, description, type="VM"):
        # Створення віртуального порту (vPort)

```

```

url = f"{self.base_url}/domains/{domain_id}/vports"
vport_data = {
    "name": name,
    "description": description,
    "type": type
}
response = self.session.post(url, data=json.dumps(vport_data))
if response.status_code == 201:
    return response.json()
else:
    raise Exception(f"VPort creation failed: {response.status_code}
{response.text}")

def list_domains(self):
# Отримання списку всіх доменів
url = f"{self.base_url}/domains"
response = self.session.get(url)
if response.status_code == 200:
    return response.json()
else:
    raise Exception(f"Fetching domains failed: {response.status_code}
{response.text}")

def list_vports(self, domain_id):
# Отримання списку vPort для домену
url = f"{self.base_url}/domains/{domain_id}/vports"
response = self.session.get(url)
if response.status_code == 200:
    return response.json()
else:
    raise Exception(f"Fetching vPorts failed: {response.status_code}
{response.text}")

def create_policy(self, domain_id, policy_name, description):
# Створення політики безпеки
url = f"{self.base_url}/domains/{domain_id}/policies"
policy_data = {
    "name": policy_name,
    "description": description
}
response = self.session.post(url, data=json.dumps(policy_data))
if response.status_code == 201:
    return response.json()
else:
    raise Exception(f"Policy creation failed: {response.status_code}
{response.text}")

def attach_policy_to_vport(self, domain_id, vport_id, policy_id):
# Прив'язка політики до vPort
url =
f"{self.base_url}/domains/{domain_id}/vports/{vport_id}/attach_policy"
attach_data = {
    "policy_id": policy_id
}
response = self.session.post(url, data=json.dumps(attach_data))
if response.status_code == 200:
    return response.json()
else:
    raise Exception(f"Attaching policy to vPort failed:
{response.status_code} {response.text}")

def configure_qos(self, vport_id, qos_settings):
# Налаштування QoS для vPort
url = f"{self.base_url}/vports/{vport_id}/qos"
response = self.session.put(url, data=json.dumps(qos_settings))
if response.status_code == 200:
    return response.json()
else:

```

```

        raise Exception(f"QoS configuration failed: {response.status_code}
{response.text}")

    def delete_domain(self, domain_id):
# Видалення домену
        url = f"{self.base_url}/domains/{domain_id}"
        response = self.session.delete(url)
        if response.status_code == 204:
            return "Domain deleted successfully"
        else:
            raise Exception(f"Domain deletion failed: {response.status_code}
{response.text}")

    def monitor_network(self, domain_id):
# Моніторинг мережі для домену
        url = f"{self.base_url}/domains/{domain_id}/monitor"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Network monitoring failed: {response.status_code}
{response.text}")

    def deploy_sdwan(self, branch_name, site_id, vpn_settings):
# Розгортання SD-WAN на віддаленій філії
        url = f"{self.base_url}/sdwan/branches"
        branch_data = {
            "branch_name": branch_name,
            "site_id": site_id,
            "vpn_settings": vpn_settings
        }
        response = self.session.post(url, data=json.dumps(branch_data))
        if response.status_code == 201:
            return response.json()
        else:
            raise Exception(f"SD-WAN deployment failed: {response.status_code}
{response.text}")

    def configure_sdwan_qos(self, branch_id, qos_settings):
# Налаштування QoS для SD-WAN філії
        url = f"{self.base_url}/sdwan/branches/{branch_id}/qos"
        response = self.session.put(url, data=json.dumps(qos_settings))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"SD-WAN QoS configuration failed:
{response.status_code} {response.text}")

    def scale_sdwan(self, branch_id, additional_bandwidth):
# Масштабування SD-WAN за рахунок додавання смуги пропускання
        url = f"{self.base_url}/sdwan/branches/{branch_id}/scale"
        scale_data = {
            "additional_bandwidth": additional_bandwidth
        }
        response = self.session.put(url, data=json.dumps(scale_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"SD-WAN scaling failed: {response.status_code}
{response.text}")

    def get_sdwan_status(self, branch_id):
# Отримання статусу SD-WAN для віддаленої філії
        url = f"{self.base_url}/sdwan/branches/{branch_id}/status"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:

```

```

        raise Exception(f"SD-WAN status fetch failed: {response.status_code}
{response.text}")

# Основна функція для запуску автоматизації
if __name__ == "__main__":
    base_url = "https://nuage.example.com/api/v1"
    username = "admin"
    password = "password"
    enterprise = "Enterprise"

# Ініціалізація об'єкта Nuage Networks API
    nuage_api = NuageNetworksAPI(base_url, username, password, enterprise)

# Аутентифікація
    nuage_api.authenticate()

# Створення нового домену SDN
    domain = nuage_api.create_domain("TestDomain", "This is a test SDN domain")

# Створення віртуального порту (vPort)
    vport = nuage_api.create_vport(domain_id=domain['ID'], name="TestVPort",
description="VPort for testing")

# Налаштування політики безпеки
    policy = nuage_api.create_policy(domain_id=domain['ID'],
policy_name="TestPolicy", description="Security policy for testing")

# Прив'язка політики до vPort
    nuage_api.attach_policy_to_vport(domain_id=domain['ID'],
vport_id=vport['ID'], policy_id=policy['ID'])

# Налаштування QoS для vPort
    qos_settings = {"bandwidth": 1000, "latency": 50, "jitter": 5}
    nuage_api.configure_qos(vport_id=vport['ID'], qos_settings=qos_settings)

# Розгортання SD-WAN на віддаленій філії
    vpn_settings = {"encryption": "AES-256", "redundancy": "high"}
    sdwan_branch = nuage_api.deploy_sdwan(branch_name="Branch1", site_id="1001",
vpn_settings=vpn_settings)

# Масштабування SD-WAN філії
    nuage_api.scale_sdwan(branch_id=sdwan_branch['ID'],
additional_bandwidth=500)

# Перевірка статусу SD-WAN
    status = nuage_api.get_sdwan_status(branch_id=sdwan_branch['ID'])

    print("SD-WAN Status:", status)

# Приклад використання нових модулів
backup = BackupRestore(base_url, nuage_api.session)
backup.backup_domain(domain_id=domain['ID'])

devices = DeviceManagement(base_url, nuage_api.session)
devices.discover_devices()

routing = RoutingAutomation(base_url, nuage_api.session)
routing.create_dynamic_route(domain_id=domain['ID'], destination="10.0.0.0/24",
next_hop="192.168.1.1")

segmentation = NetworkSegmentation(base_url, nuage_api.session)
segmentation.create_vlan(domain_id=domain['ID'], vlan_id=10, name="VLAN_10")

services = NetworkServicesOrchestration(base_url, nuage_api.session)
services.deploy_vnf(domain_id=domain['ID'], vnf_name="Firewall",
config={"rules": "default"})

scaling = SDWANScaling(base_url, nuage_api.session)
scaling.scale_sdwan(branch_id="branch_01", additional_bandwidth=1000)

```

```
# Приклад використання нових модулів
user_mgmt = UserManagement(base_url, nuage_api.session)
user_mgmt.create_user(username="newuser", password="securepass", role="admin",
enterprise="Enterprise")

monitor = NetworkMonitoring(base_url, nuage_api.session)
monitor.monitor_traffic(domain_id=domain['ID'])

analytics = AnalyticsReporting(base_url, nuage_api.session)
analytics.collect_network_stats(domain_id=domain['ID'])
```

КБПЗ\_2024

```
import requests
import json

class UserManagement:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def create_user(self, username, password, role, enterprise):
        # Створення нового користувача
        url = f"{self.base_url}/users"
        user_data = {
            "username": username,
            "password": password,
            "role": role,
            "enterprise": enterprise
        }
        response = self.session.post(url, data=json.dumps(user_data))
        if response.status_code == 201:
            return response.json()
        else:
            raise Exception(f"User creation failed: {response.status_code}
{response.text}")

    def assign_role(self, user_id, role):
        # Призначення ролі користувачу
        url = f"{self.base_url}/users/{user_id}/roles"
        role_data = {
            "role": role
        }
        response = self.session.put(url, data=json.dumps(role_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Role assignment failed: {response.status_code}
{response.text}")

    def delete_user(self, user_id):
        # Видалення користувача
        url = f"{self.base_url}/users/{user_id}"
        response = self.session.delete(url)
        if response.status_code == 204:
            return "User deleted successfully"
        else:
            raise Exception(f"User deletion failed: {response.status_code}
{response.text}")
```

```
import requests
import json

class NetworkMonitoring:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def monitor_traffic(self, domain_id):
        # Моніторинг трафіку для домену
        url = f"{self.base_url}/domains/{domain_id}/traffic"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Traffic monitoring failed: {response.status_code}
{response.text}")

    def monitor_qos(self, vport_id):
        # Моніторинг QoS для vPort
        url = f"{self.base_url}/vports/{vport_id}/qos/monitor"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"QoS monitoring failed: {response.status_code}
{response.text}")

    def get_event_logs(self, domain_id):
        # Отримання логів подій для домену
        url = f"{self.base_url}/domains/{domain_id}/logs"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Fetching logs failed: {response.status_code}
{response.text}")
```

## Файл configuration\_automation.py

```
import requests
import json

class ConfigurationAutomation:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def deploy_template(self, template_id, domain_id):
        # Розгортання шаблону конфігурації для домену
        url = f"{self.base_url}/domains/{domain_id}/deploy_template"
        template_data = {
            "template_id": template_id
        }
        response = self.session.post(url, data=json.dumps(template_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Template deployment failed: {response.status_code}
{response.text}")

    def configure_routing(self, domain_id, routing_protocol):
        # Налаштування маршрутизації для домену
        url = f"{self.base_url}/domains/{domain_id}/routing"
        routing_data = {
            "protocol": routing_protocol
        }
        response = self.session.put(url, data=json.dumps(routing_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Routing configuration failed:
{response.status_code} {response.text}")

    def apply_configuration_change(self, domain_id, change_data):
        # Автоматична зміна конфігурації
        url = f"{self.base_url}/domains/{domain_id}/apply_change"
        response = self.session.put(url, data=json.dumps(change_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Configuration change failed:
{response.status_code} {response.text}")
```

## Файл system\_integration.py

```
import requests
import json

class SystemIntegration:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def integrate_nms(self, nms_url, nms_key):
        # Інтеграція з системою управління мережею (NMS)
        url = f"{self.base_url}/integrations/nms"
        nms_data = {
            "nms_url": nms_url,
            "nms_key": nms_key
        }
        response = self.session.post(url, data=json.dumps(nms_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"NMS integration failed: {response.status_code}
{response.text}")

    def integrate_virtualization(self, platform, credentials):
        # Інтеграція з системами віртуалізації (VMware, OpenStack)
        url = f"{self.base_url}/integrations/virtualization"
        platform_data = {
            "platform": platform,
            "credentials": credentials
        }
        response = self.session.post(url, data=json.dumps(platform_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Virtualization integration failed:
{response.status_code} {response.text}")

    def integrate_cloud_services(self, cloud_service, api_key):
        # Інтеграція з хмарними сервісами
        url = f"{self.base_url}/integrations/cloud"
        cloud_data = {
            "service": cloud_service,
            "api_key": api_key
        }
        response = self.session.post(url, data=json.dumps(cloud_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Cloud service integration failed:
{response.status_code} {response.text}")
```

## Файл analytics\_reporting.py

```
import requests
import json

class AnalyticsReporting:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def collect_network_stats(self, domain_id):
        # Збір статистики мережі для домену
        url = f"{self.base_url}/domains/{domain_id}/stats"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Fetching network stats failed:
{response.status_code} {response.text}")

    def generate_performance_report(self, domain_id):
        # Генерація звіту про продуктивність мережі
        url = f"{self.base_url}/domains/{domain_id}/performance_report"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Generating performance report failed:
{response.status_code} {response.text}")

    def visualize_data(self, domain_id, metric):
        # Візуалізація мережевих показників
        url = f"{self.base_url}/domains/{domain_id}/visualize?metric={metric}"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Data visualization failed: {response.status_code}
{response.text}")
```

## Файл security\_policy\_management.py

```
import requests
import json

class SecurityPolicyManagement:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def create_security_policy(self, domain_id, policy_name, rules):
        # Створення політики безпеки з правилами
        url = f"{self.base_url}/domains/{domain_id}/security_policies"
        policy_data = {
            "name": policy_name,
            "rules": rules
        }
        response = self.session.post(url, data=json.dumps(policy_data))
        if response.status_code == 201:
            return response.json()
        else:
            raise Exception(f"Security policy creation failed:
{response.status_code} {response.text}")

    def audit_security(self, domain_id):
        # Аудит безпеки мережі
        url = f"{self.base_url}/domains/{domain_id}/security_audit"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Security audit failed: {response.status_code}
{response.text}")

    def log_user_actions(self, user_id):
        # Логування дій користувачів
        url = f"{self.base_url}/users/{user_id}/logs"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Fetching user logs failed: {response.status_code}
{response.text}")
```

## Файл backup\_restore.py

```
import requests
import json

class BackupRestore:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def backup_domain(self, domain_id):
        # Створення резервної копії домену
        url = f"{self.base_url}/domains/{domain_id}/backup"
        response = self.session.post(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Backup creation failed: {response.status_code}
{response.text}")

    def restore_domain(self, domain_id, backup_id):
        # Відновлення домену з резервної копії
        url = f"{self.base_url}/domains/{domain_id}/restore"
        restore_data = {
            "backup_id": backup_id
        }
        response = self.session.post(url, data=json.dumps(restore_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Restore failed: {response.status_code}
{response.text}")

    def schedule_backup(self, domain_id, schedule):
        # Планове резервне копіювання
        url = f"{self.base_url}/domains/{domain_id}/schedule_backup"
        schedule_data = {
            "schedule": schedule
        }
        response = self.session.post(url, data=json.dumps(schedule_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Scheduling backup failed: {response.status_code}
{response.text}")
```

## Файл device\_management.py

```
import requests
import json

class DeviceManagement:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def discover_devices(self):
        # Автоматичне виявлення пристроїв у мережі
        url = f"{self.base_url}/devices/discover"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Device discovery failed: {response.status_code}
{response.text}")

    def update_device_firmware(self, device_id, firmware_version):
        # Оновлення прошивки пристрою
        url = f"{self.base_url}/devices/{device_id}/update_firmware"
        firmware_data = {
            "version": firmware_version
        }
        response = self.session.put(url, data=json.dumps(firmware_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Firmware update failed: {response.status_code}
{response.text}")

    def get_device_inventory(self):
        # Інвентаризація пристроїв
        url = f"{self.base_url}/devices/inventory"
        response = self.session.get(url)
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Fetching device inventory failed:
{response.status_code} {response.text}")
```

## Файл routing\_automation.py

```
import requests
import json

class RoutingAutomation:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def create_dynamic_route(self, domain_id, destination, next_hop):
        # Створення динамічного маршруту
        url = f"{self.base_url}/domains/{domain_id}/routing"
        route_data = {
            "destination": destination,
            "next_hop": next_hop,
            "type": "dynamic"
        }
        response = self.session.post(url, data=json.dumps(route_data))
        if response.status_code == 201:
            return response.json()
        else:
            raise Exception(f"Dynamic route creation failed:
{response.status_code} {response.text}")

    def create_static_route(self, domain_id, destination, next_hop):
        # Створення статичного маршруту
        url = f"{self.base_url}/domains/{domain_id}/routing"
        route_data = {
            "destination": destination,
            "next_hop": next_hop,
            "type": "static"
        }
        response = self.session.post(url, data=json.dumps(route_data))
        if response.status_code == 201:
            return response.json()
        else:
            raise Exception(f"Static route creation failed:
{response.status_code} {response.text}")

    def update_route(self, domain_id, route_id, new_next_hop):
        # Оновлення маршруту
        url = f"{self.base_url}/domains/{domain_id}/routing/{route_id}"
        route_data = {
            "next_hop": new_next_hop
        }
        response = self.session.put(url, data=json.dumps(route_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Route update failed: {response.status_code}
{response.text}")
```

## Файл network\_segmentation.py

```
import requests
import json

class NetworkSegmentation:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def create_vlan(self, domain_id, vlan_id, name):
        # Створення VLAN сегментації
        url = f"{self.base_url}/domains/{domain_id}/vlans"
        vlan_data = {
            "vlan_id": vlan_id,
            "name": name
        }
        response = self.session.post(url, data=json.dumps(vlan_data))
        if response.status_code == 201:
            return response.json()
        else:
            raise Exception(f"VLAN creation failed: {response.status_code}
{response.text}")

    def create_vxlan(self, domain_id, vxlan_id, name):
        # Створення VXLAN сегментації
        url = f"{self.base_url}/domains/{domain_id}/vxlans"
        vxlan_data = {
            "vxlan_id": vxlan_id,
            "name": name
        }
        response = self.session.post(url, data=json.dumps(vxlan_data))
        if response.status_code == 201:
            return response.json()
        else:
            raise Exception(f"VXLAN creation failed: {response.status_code}
{response.text}")

    def delete_vlan(self, domain_id, vlan_id):
        # Видалення VLAN
        url = f"{self.base_url}/domains/{domain_id}/vlans/{vlan_id}"
        response = self.session.delete(url)
        if response.status_code == 204:
            return "VLAN deleted successfully"
        else:
            raise Exception(f"VLAN deletion failed: {response.status_code}
{response.text}")
```

## network\_services\_orchestration.py

```

import requests
import json

class NetworkServicesOrchestration:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def deploy_vnf(self, domain_id, vnf_name, config):
        # Розгортання віртуальної мережевої функції (VNF)
        url = f"{self.base_url}/domains/{domain_id}/vnf"
        vnf_data = {
            "vnf_name": vnf_name,
            "config": config
        }
        response = self.session.post(url, data=json.dumps(vnf_data))
        if response.status_code == 201:
            return response.json()
        else:
            raise Exception(f"VNF deployment failed: {response.status_code}
{response.text}")

    def update_vnf(self, domain_id, vnf_id, new_config):
        # Оновлення конфігурації VNF
        url = f"{self.base_url}/domains/{domain_id}/vnf/{vnf_id}"
        config_data = {
            "config": new_config
        }
        response = self.session.put(url, data=json.dumps(config_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"VNF update failed: {response.status_code}
{response.text}")

    def delete_vnf(self, domain_id, vnf_id):
        # Видалення VNF
        url = f"{self.base_url}/domains/{domain_id}/vnf/{vnf_id}"
        response = self.session.delete(url)
        if response.status_code == 204:
            return "VNF deleted successfully"
        else:
            raise Exception(f"VNF deletion failed: {response.status_code}
{response.text}")

```

Файл `sdwan_scaling.py`

```
import requests
import json

class SDWANScaling:
    def __init__(self, base_url, session):
        self.base_url = base_url
        self.session = session

    def scale_sdwan(self, branch_id, additional_bandwidth):
        # Масштабування SD-WAN шляхом додавання смуги пропускання
        url = f"{self.base_url}/sdwan/branches/{branch_id}/scale"
        scale_data = {
            "additional_bandwidth": additional_bandwidth
        }
        response = self.session.put(url, data=json.dumps(scale_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"SD-WAN scaling failed: {response.status_code}
{response.text}")

    def reduce_sdwan_bandwidth(self, branch_id, bandwidth_reduction):
        # Зменшення смуги пропускання для SD-WAN
        url = f"{self.base_url}/sdwan/branches/{branch_id}/scale"
        scale_data = {
            "reduce_bandwidth": bandwidth_reduction
        }
        response = self.session.put(url, data=json.dumps(scale_data))
        if response.status_code == 200:
            return response.json()
        else:
            raise Exception(f"Reducing bandwidth failed: {response.status_code}
{response.text}")
```