

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 аналізу та
консолідації розподіленої інформації віртуалізованих
об’єднаних комунікацій у мережі”

Виконав здобувач вищої освіти
II курсу, групи КН-23М
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Цішевський М.С.
« ____ » _____ 2024 р.

Керівник проекту
кандидат технічних наук
_____ Буравченко К.О.
« ____ » _____ 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 122 "Комп'ютерні науки"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

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

Цішевському Михайлу Станіславовичу

(прізвище, ім'я, по батькові)

- | | | | | | | | | | | | |
|--|---|--|----------------------------|---|--|--|--|--|---------------------|--|--|
| 1. Тема роботи | <i>Дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі</i> | | | | | | | | | | |
| 2. Керівник роботи | <i>Буравченко Костянтин Олегович, канд. техн. наук</i>
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 18-13 від 07.08.2024 року | | | | | | | | | | |
| 3. Строк подання студентом роботи до захисту | <i>2.12.2024 р.</i> | | | | | | | | | | |
| 4. Мета та завдання випускної кваліфікаційної роботи: | <i>Метою розробки є дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі</i> | | | | | | | | | | |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | <table border="1"><tr><td><i>1. Призначення та область використання.</i></td><td><i>6. Наукова новизна.</i></td></tr><tr><td><i>2. Перегляд аналогічних існуючих систем.</i></td><td><i>7. Маркетингове та економічне обґрунтування ІТ-проєкту.</i></td></tr><tr><td><i>3. Опис і обґрунтування проектних рішень.</i></td><td><i>8. Заходи з охорони праці та техніки безпеки.</i></td></tr><tr><td><i>4. Етапи програмування системи.</i></td><td><i>9. Висновки.</i></td></tr><tr><td><i>5. Впровадження системи в промислову експлуатацію</i></td><td></td></tr></table> | <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 ІТ-проєкту.</i> | <i>3. Опис і обґрунтування проектних 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> | |
| <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 ІТ-проєкту.</i> | | | | | | | | | | |
| <i>3. Опис і обґрунтування проектних 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

Цішевський М.С. Дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Метою розробки є дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Об'єктом дослідження є процес аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Предметом дослідження є методи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Методи дослідження базуються на методах аналізу та консолідації розподіленої інформації, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

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'ютерні науки, віртуалізовані об'єднані комунікації

ABSTRACT

Tsyshevskiy M.S. Research and software implementation of the system of analysis and consolidation of distributed information of virtualized unified communications in the network. 122 Computer Science. 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 system of analysis and consolidation of distributed information of virtualized unified communications in the network.

The purpose of the development is the research and software implementation of the system of analysis and consolidation of distributed information of virtualized unified communications in the network.

The object of research is the process of analysis and consolidation of distributed information of virtualized unified communications in the network.

The subject of research is methods of analysis and consolidation of distributed information of virtualized unified communications in the network.

Research methods are based on methods of analysis and consolidation of distributed information, methods of mathematical statistics, and methods of software development.

The result of the work is the software implementation of the system of analysis and consolidation of distributed information of virtualized unified communications in the network.

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 science, virtualized unified communications

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	5
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	8
1.1 Призначення системи.....	8
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	18
2.3 Розгорнута постановка завдання	23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	24
3.1 Опис функціонування системи	24
3.2 Розробка структурної схеми.....	28
3.3 Розробка функціональної схеми	42
3.4 Розробка діаграми процесів.....	47
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	49
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	49
4.2 Захист розробленого програмного забезпечення.....	63
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	68
6 НАУКОВА НОВИЗНА	72

						ВКРМ-122.24.0016.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Цішевський М.С.				Дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі	Літ.	Аркуш	Аркушів
Перев.	Буравченко К.О.					М	1	97
Н.контр.	Коваленко А.С.					ЦНТУ КН-23М		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	73
7.1	Визначення цільової аудиторії кінцевого готового продукту	73
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	74
7.3	Вибір методу оцінки вартості ПЗ	75
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	76
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	76
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	80
7.7	Визначення ключових факторів успіху конкретного проєкту.....	80
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	81
8.1	Вступ.....	81
8.2	Аналіз умов праці на робочому місці ІТ-фахівця	82
8.3	Розробка заходів з умов поліпшення охорони праці	86
8.4	Розрахункова частина	87
9	ОСНОВНІ ВИСНОВКИ.....	90
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	92

КБПЗ - 2024

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АТМУ	–	автоматизована телефонна мережа установи
БД	–	база даних
КМЗ	–	корпоративна мережа зв'язку
ЛОМ	–	локальна обчислювальна мережа
ОЗП	–	оперативно-запам'ятовувальний пристрій
ПАТМ	–	пристрій автоматизованої телефонної мережі
ПЗ	–	програмне забезпечення
ПП	–	програмний продукт
СПД	–	система передачі даних
СУБД	–	система управління БД
ТфОП	–	телефонний оператор
AVVID	–	архітектурна модель фірми Cisco Systems
CNG	–	Comfort Noise Generator. Генератор комфортного шуму
DSP	–	Digital Signal Processor. Процесор цифрової обробки сигналів
DTMF	–	Dual Tone Multi-Frequency. Багаточастотна система кодування цифр номера
GK	–	Gatekeeper. Гейткіпер . Виконує функції керування зоною мережі H.323
GW	–	Gateway. Шлюз. Апаратно-програмний комплекс, що забезпечує обмін даними між мережами різних типів
H.323	–	Рекомендація ІТУ-Т, що визначає системи мультимедійного зв'язку в мережах з пакетною комутацією
H.248	–	Протокол керування транспортним шлюзом
IP	–	Internet Protocol. Протокол міжмережної взаємодії
LPC	–	Linear Prediction Coding. Кодування з лінійним прогнозуванням
MCU	–	Multipoint Control Unit. Пристрій керування конференцією

MG	– Media Gateway. Транспортний шлюз
MGCP	– Media Gateway Control Protocol. Протокол керування шлюзами
MP	Multipoint processor. Процесор для обробки інформації користувачів при централізованих конференціях
OSI	– Open System Interconnection. Взаємодія відкритих систем
PPP	– Point-to-Point Protocol. Протокол двостороннього зв'язку
RADIUS	Remote Authentication Dial-In User Service. Протокол автентифікації й авторизації абонентів, а також обліку обсягу наданих їм послуг
RAS	Registration Admission and Status. Протокол взаємодії термінального встаткування з gatekeeper. Входить у сімейство протоколів H.323
RSVP	– Resource Reservation Protocol. Протокол резервування ресурсів
RTCP	Real-time Transport Control Protocol. Протокол контролю транспортування інформації в реальному часі
SIP	– Session Initiation Protocol. Протокол ініціювання сеансів зв'язку
TAPI	Telephony Applications Programming Interface. Інтерфейс для програмування телефонних додатків
TCP	Transmission Control Protocol. Протокол керування передачею (даних) Основний транспортний протокол у стеці протоколів TCP/IP.
TCP/IP	– Transmission Control Protocol/Internet Protocol. Стек протоколів, що забезпечують організацію зв'язку між комп'ютерами в мережі Інтернет
UDP	– User Datagram Protocol. Протокол передачі дейтаграмм користувача.
VoIP	– Voice over Internet Protocol. Технологія, що дозволяє використовувати IP-мережу для передачі мовної інформації

ВСТУП

Актуальність теми. Віртуалізовані рішення для реалізації об'єднаних комунікацій уже широко поширені на підприємствах, однак медіашлюзи – спеціалізовані апаратні пристрої, що виконують роль посередників між різними комунікаційними технологіями, – дотепер залишалися осторонь. Тепер і в цьому сегменті спостерігається тенденція до переходу на програмні засоби, а виходить, і до віртуалізації.

Крім скорочення витрат на апаратне забезпечення, віртуалізація володіє рядом інших переваг: насамперед це значне поліпшення доступності систем. Сучасні рішення для резервування й аварійного відновлення після збоїв (Backup/Disaster Recovery) при необхідності забезпечують безперебійний перехід на резервні системи й дозволяють скоротити час простою до мізерно малих значень – одне з найважливіших вимог для телефонії. Однак використання виділених апаратних пристроїв зв'язку нівелює вигоду, одержувану від високої доступності віртуальних телекомунікаційних рішень.

Тільки віртуалізовані медіашлюзи дозволяють успішно завершити реалізацію об'єднаних комунікацій (Unified Communications, UC) за допомогою програмних засобів, а виходить, створити повністю віртуалізовану інфраструктуру.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

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

– Огляд існуючих систем аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

– Дослідження системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

– Програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Об'єктом дослідження є процес аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Предметом дослідження є методи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Методи дослідження базуються на методах аналізу та консолідації розподіленої інформації, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

– Розроблено вітчизняний продукт аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

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

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

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 аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2024

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Не у всіх випадках можна легко й просто здійснити консолідацію всіх серверних додатків. У сфері факс-серверів, засобів уніфікованої передачі повідомлень (Unified Messaging) і об'єднаних комунікацій часто використовуються контролери у вигляді внутрішніх знімних карт або зовнішніх апаратних пристроїв. Однак комунікація всі частіше здійснюється не через телефонні інтерфейси, а за допомогою каналів на базі IP – як правило, у вигляді технології SIP Trunking. Ці з'єднання встановлюються прямо – за посередництвом провайдеру телефонних послуг або шляхом інтеграції SIP в існуючу телекомунікаційну інфраструктуру. Оскільки для обох варіантів не потрібно спеціалізованих апаратних інтерфейсів, мова йде про ідеальних кандидатів для віртуалізації.

Щоб краще зрозуміти концепцію віртуального шлюзу, корисно зробити невеликий екскурс у минуле й згадати про традиційні схеми підключення до телекомунікаційних мереж і послуг. Протягом багатьох років технологія ISDN формувала стабільну основу для телефонних і факсових сервісів. Традиційні комунікаційні контролери зв'язують зовнішні телефонні інтерфейси з локальними додатками на базі IP – звичайно мова йде об факс-серверах. Всі частіше контролери застосовуються як сполучну ланку з рішеннями для IP-телефонії (наприклад, із системою об'єднаних комунікацій Microsoft Lync Server).

Усе більше підприємств переводять свої телефонні служби із традиційних офісних АТМ на сучасні комунікаційні рішення. Однак радикальний перехід – це скоріше виключення. Як правило, користувачі переводяться на нову систему поетапно, протягом деякого перехідного періоду, коли обидві технології застосовуються паралельно. Цікавий спосіб реалізації цієї стратегії – установка

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

медіашлюзів між зовнішньою телефонною лінією й офісною АТМ. Цей шлюз повинен додатково оснащуватися внутрішніми телефонними інтерфейсами, які стосовно АТМ є транковими лініями.

В ідеалі при реалізації цього сценарію конфігурація офісної АТМ залишається незачепленою. При використанні систем рівня Microsoft Lync шлюз автоматично вирішує, куди йому перенаправляти вхідний дзвінок. Якщо набраний номер записаний в Active Directory як користувач Lync, розмова передається на сервер Lync через SIP. Всі інші дзвінки як і раніше направляються через ISDN на стару офісну АТМ.

1.2 Область застосування

Після успішного завершення локальної міграції все спілкування співробітників усередині підприємства повинне здійснюватися винятково через IP. Навіть якщо в користуванні залишається невелика кількість аналогових кінцевих пристроїв, їх можна інтегрувати за допомогою аналогових телефонних адаптерів (SIP-АТА).

Тепер до створення віртуального шлюзу залишається лише один крок: як тільки зовнішня комунікація буде повністю переведена на SIP, необхідність у телефонних портах зникне – для всіх видів зв'язку будуть використовуватися мережні інтерфейси. У цьому випадку функції шлюзу на всі сто відсотків реалізуються за допомогою програмних засобів, а виходить, вони готові до віртуалізації. Описаний варіант шлюзу є класичним прикордонним контролером сеансів корпоративного рівня (Enterprise Session Border Controller, E-SBC) і здатний виконувати безліч завдань, у тому числі:

- перехід з публічних у частки IP-мережі із захистом за допомогою брандмауера;
- реєстрація/автентифікація SIP-транків;

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

– забезпечення взаємодії між різними варіантами протоколу SIP у ролі двостороннього користувальницького агента SIP (Back to Back User Agent, B2BUA);

– перетворення між UDP, TCP або TLS, а також між зашифрованою й незашифрованою передачею голосових даних (SRTP/RTP);

– відправлення й прийом факсів на вибір або через інтегрований програмний модем за протоколом T.30, або за допомогою протоколу T.38 на базі IP;

– захист від збоїв (Failover) за рахунок створення множинних резервних екземплярів;

– розподіл навантажень по колу (Round Robin).

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

3CX WebMeeting

Попит на послуги хмарних АТМ постійно росте, особливо серед невеликих компаній. Однак, використання віртуальних АТМ має ряд недоліків: клієнти одержують обмежений набір функцій, а інтегратори залежать від творців віртуальних АТМ – вони не володіють реєстраційними даними користувачів, не можуть контролювати якість сервісу й обмежені в можливостях технічної підтримки. Сформована ситуація створює ризики як для інтеграторів, так і для кінцевих користувачів.

Представляємо 3CX Phone System – віртуальну АТМ для установки на хостингу, створену спеціально для інтеграторів. 3CX пропонує інтеграторам повний контроль над обліковими даними користувачів і якістю сервісу. Замість обмеженого набору базових функцій АТМ, запропонуєте клієнтові повноцінну віртуальну 3CX Phone System з усім функціоналом, що вже встигли оцінити користувачі. Можливості об'єднаних комунікацій, зручні клієнти для смартфонів і персональних комп'ютерів, підтримка провідних виробників ІР телефонів – і все це з підтримкою від всесвітньо відомого виробника. Тепер інтегратори готові запропонувати клієнтам повноцінну корпоративну хмарну АТМ, розмістивши її на своїх серверах.

Система хмарної телефонії 2600hz KAZOO

Телекомунікаційний проект KAZOO молодій компанії за назвою 2600hz (США) уже встиг зібрати кілька галузевих нагород і використовується багатьма великими операторами зв'язку.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

KAZOO – відкрита (open-source) платформа віртуальної хмарної телефонії, на основі якої можна будувати сервіси віртуальних АТМ, віртуальні мобільні мережі й інші масштабні хмарні телекомунікаційні рішення операторського класу.

Фактично, KAZOO – це перша відкрита (open-source, MPL) масштабна хмарна платформа віртуальної телефонії [4]. KAZOO – функціонально багата, розподілена, відказостійка, масштабована, високопродуктивна, керована через API платформа віртуальних АТМ, призначена для операторів зв'язку, які хочуть надавати розширені послуги IP-телефонії й віртуальних офісів, але не хочуть платити мільйони доларів за рішення «від лідерів ринку».

KAZOO використовує трохи компонент:

– SIP-проксі сервер Kamailio (із власним модулем db_kazoo, призначеним для звертання до даних KAZOO). Сервери Kamailio забезпечують реєстрацію SIP-пристроїв, передачу даних presence і розподіл навантаження між серверами FreeSWITCH (за допомогою стандартного модуля dispatcher). Модуль db_kazoo, спеціально розроблений авторами системи для kamailio, забезпечує взаємодія з KAZOO через AMQP.

– IP-АТМ FreeSWITCH (із власним модулем mod_kazoo, за допомогою якого здійснюється інтеграція FreeSWITCH в Erlang-Оточення). Відома IP-АТМ FreeSWITCH використовується в проєкті як оброблювач дзвінків, при цьому все керування дзвінками лягає на плечі додатків KAZOO, написаних на Erlang. Спеціальний модуль FreeSWITCH перетворює IP-АТМ у так званий Erlang C-Node, прямо взаємодіючий з вузлами Erlang.

– NoSQL СУБД BigCouch (так, усі знають що цей проєкт умер, рано або пізно відбудеться перехід на CouchDB), написану на Erlang високопродуктивну, масштабовану, розподілену СУБД.

– Сервер повідомлень AMQP RabbitMQ, також написаний на Erlang високопродуктивний messaging-сервер.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

– Балансувальник навантаження NAPROXY (використовується для розподілу доступу до серверів СУБД).

– Платформу й мову програмування Erlang, на якій написана логіка системи: компоненти eCallMgr і додатка Whistle apps.

Компоненти активно взаємодіють між собою й формують цільну хмарну VoIP-платформу.

Кожна віртуальна АТМ (account) усередині KAZOO «з коробки» підтримує наступні функції:

- виклик пристрою;
- виклик користувача (будуть викликані всі його пристрої);
- голосові меню;
- голосову пошту;
- маршрутизацію за часом;
- перенапрямок дзвінків;
- прийом і відправлення (через API або e-mail) факсів;
- запис розмов;
- виклик зовнішніх скриптів (є режим сумісності з Twilio);
- групи дозвона;
- перехоплення викликів;
- паркування викликів;
- переадресацію;
- hot-desking (присвоєння пристроїв);
- інтерком;
- DND;
- черги дзвінків (простий ACD);
- керування CallerID;
- конференції meet-me;
- сервери конференцій (з введенням номера кімнати);

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

– можливість підключення «своїх» транків (наприклад, шлюзу FXO з гаряче улюбленим «аналоговим» номером).

Оператор може обмежувати клієнтові кількість використовуваних транків різної спрямованості, надавати номерну ємність, стежити за витратою засобів, і т.д. і т.п.

Так як керування станцією виробляється через API (докладніше нижче), система може бути оснащена зовсім довільним «особистим кабінетом», що реалізує практично будь-яку логіку керування, від «майстра установки» до інтерфейсу з перетаскуванням піктограм (прийнятим у деяких сучасних IP АТМ). До складу системи входить демонстраційний інтерфейс, на жаль, що слабо підходить для комерційного використання.

Компоненти KAZOO можуть (і повинні) розміщатися на різних вузлах, а кластери можуть формувати географічно розподілені вузли, зв'язані через WAN. При наявності високошвидкісних каналів зв'язку, елементи кластера можуть бути рознесені між різними площадками для додаткової відказостійкості.

Компоненти системи відмінно працюють у віртуальному оточенні й можуть розміщатися в приватних, публічних або гібридних IaaS-Хмарах, що істотно спрощує розгортання й експлуатацію.

KAZOO можна поставити на одному сервері (щоб «подивитися»), однак навіть лабораторне оточення для розробки краще мати у вигляді мінімального кластера з 7 вузлів, так як дуже великий обсяг роботи в проекті пророблений саме з метою забезпечення роботи розподіленого оточення.

Практично всі (за єдиним виключенням, у вигляді не має fallback-шляху зв'язку Kamailio-RabbitMQ) компонента системи можуть бути багаторазово продубльовані як з метою підвищення продуктивності, так і з метою забезпечення надійності й високої доступності. Декілька проксі-серверів Kamailio балансують трафік між серверами FreeSWITCH, які управляються серверами eCallMgr. Множинні сервери додатків Whistle apps взаємодіють із серверами eCallMgr через AMQP і обмінюються даними із серверами BigCouch,

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

«закритими» NPROXY. Сервіси API надаються серверами Whistle apps і можуть бути сховані за балансуєчими проху-серверами.

Все спілкування із системами зберігання й зовнішніми скриптами відбувається за протоколом HTTP, що дозволяє домогтися відказостійкості всього комплексу (використовуючи Раа-платформи або багаторівневі відказостійкі ферми серверів зі скриптами).

Про масштабованість почасти написано в попередньому абзаці, досить додати лише те, що продуктивність, фактично, налаштовується додаванням ресурсів. Накладні витрати при горизонтальному масштабуванні мінімальні, так побудована архітектура.

Висока продуктивність системи почасти реалізується за рахунок гарної масштабованості, однак всі використовувані компоненти самі по собі дуже швидкі:

– Kamailio здавна відомий своєю високою продуктивністю [5] і широко використовується операторами зв'язку по усьому світі.

– FreeSWITCH відмінно почуває себе у віртуальному оточенні й, при правильній інсталяції, здатний обробляти тисячі одночасних дзвінків на сервер.

– RabbitMQ написаний на Erlang, добре масштабується сам по собі й здатний показувати дуже високу продуктивність [6].

– СУБД BigCouch також написана на Erlang і була обрана авторами системи, зокрема, за показану продуктивність [7].

– «Бізнес-логіка» системи, компоненти eCallMgr і Whistle apps, написані на Erlang, самі по собі працюють дуже швидко. Автори дуже добре думали, перш ніж писати код, і вибір інструмента й структура ядра були не випадкові. У результаті саме повільне що відбувається усередині логічного шару – це запис системних журналів.

Унікальною якістю платформи KAZOO є її повнофункціональний REST API. Саме за допомогою цього API клієнтські додатки управляють сутностями усередині віртуальних АТМ, а адміністратори можуть змінювати системні налаштування.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Основні сутності, для яких кінцевим користувачам надаються API:

– Accounts (аккаунти, орендарі). Властиво, віртуальні АТМ. Кожний account ідентифікується своїм SIP-realm (доменним ім'ям) і унікальним ID. Аккаунти можуть бути вкладені друг у друга, що дозволяє будувати агентські схеми надання послуг.

– Users (користувачі). Фізичні особи, власники пристроїв (devices), використовуваних для дзвінків.

– Devices (пристрою). Фактичні прикінцеві пристрої, на які можуть бути спрямовані виклики. Пристрою належать користувачам і можуть викликатися як окремо, так і разом.

– Callflows (сценарії викликів). Фактично, dialplan-и. Для callflow призначаються номери, будь-який виклик номера, фактично, приводить до виконання того або іншого callflow-сценарію.

– Menus (голосові меню). IVR-меню, що припускають уведення тим, то дзвонить, цифр.

– Media (медіа-ресурси). Медіа-файли або TTS.

– Faxboxes (факсові ящики). Використовуються для прийому й відправлення факсів.

– VMBoxes (ящики голосової пошти). Використовуються для прийому й зберігання голосової пошти (яка може відправлятися на e-mail).

– CDRs (запису CDR). Дають можливість одержувати інформацію про дзвінки, що відбулися.

– Resources (ресурси зовнішніх підключень), локальні для account-ів.

– Temporal rules (правила маршрутизації за часом).

– Lists (списки номерів або регулярних виражень).

– Metaflows (сценарії DTMF-команд під час розмови).

– Limits (обмеження кількості транків).

– Click-to-call (ініціація викликів через API).

– Webhooks (веб-виклики при ініціації, відповіді, завершенні розмов).

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

- Queues (черги ACD).
- Agents (даних операторів ACD).

Адміністраторам через API, зокрема, надається доступ до системних налаштувань і компонентів:

- System configs (елементи конфігурації системи).
- Resources (ресурси зовнішніх підключень), глобальні для системи.
- Hangups (статистика кодів завершення розмов), призначений для моніторингу й виявлення аномалій.

На основі KAZOO можливо порівняно швидке створення сервісів на зразок Grasshopper, RingCentral, SendHub, віртуальних АТМ різній складності і ємності. Зрозуміло, щоб зробити повноцінний сервіс однієї KAZOO недостатньо, але як основу рішення вона підходить зовсім чудово й дозволяє закласти в архітектуру досить пристойний ріст і одержати широкі функціональні можливості.

KAZOO не проста. Це – досить складне рішення для тих, кому вже потрібний масштаб, що традиційно коштує дуже й дуже дорого. Відсутність ліцензійних платежів істотно знижує вартість входу в бізнес віртуальної телефонії, особливо під час кризи.

Проект дуже швидко розвивається. Не за горами підтримка SMS, поліпшена документація, інтерфейс realtime-sockets і інші дуже корисні й зручні функції. Компанія 2600hz одержує нагороди й регулярно звітує про успішні проекти й інтеграції. Майбутнє продукту бачиться світлі й райдужним, автори дуже намагаються на радість співтовариству підтримувати open-source версію, за що їм низький уклін і велике спасибі.

Проект KAZOO – унікальна open-source хмарна платформа операторської віртуальної телефонії, що дозволяє операторам зв'язку надавати своїм замовникам сучасні послуги віртуальних АТМ, а також будувати інші складні телекомунікаційні рішення (наприклад, віртуальні мобільні мережі).

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

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

Швидкість виконання коду Python

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

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

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

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Використання Python

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

Застосування мови програмування Python:

1. BitTorrent – протокол для обміну даними.
2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.
3. Blender – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.
4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.
5. World of Tanks.
6. Вільна енциклопедія Вікіпедія.
7. Пошукова система Google.
8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.
9. YouTube – популярне відеосховище.
10. ...

Версії Python

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

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

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

Інструкція машинної мови існує для кожної операції, яку процесор здатний виконати – є інструкція для додавання чисел, є інструкція для віднімання чисел і т.д. Увесь набір інструкцій, який центральний процесор може виконати, відомий як набір інструкцій процесора.

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

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

Отримати

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

Декодувати

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

Виконати

Останній крок циклу – виконати операцію.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

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

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

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

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

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

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

1. Компілятор.
2. Інтерпретатор.

Завантаження Python

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

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Середовище програмування для Python

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

Текстові редактори та інтегровані середовища програмування для Python:

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

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

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

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

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

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

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

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

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

						ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			22

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускні кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

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 методіку побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Віртуальні шлюзи підтримують функції гнучкого масштабування за рахунок виділення відповідних ресурсів. Від числа виділених процесорних ядер залежить продуктивність шлюзу, а виходить, і кількість паралельно використовуваних каналів. Якщо паралельних з'єднань більше ста, їх бажано розподілити між декількома системами. Для успішної передачі голосових даних за допомогою протоколу RTP потрібно забезпечити тверді параметри передачі даних у реальному часі, тому необхідно стежити за тим, щоб паралельно функціонуючі системи не гальмували роботу віртуальної машини, на якій виконується шлюз.

Сучасні середовища віртуалізації пропонують для цих цілей цілий ряд налаштувань, за допомогою яких можна здійснити необхідне тонке налагодження систем. У виняткових випадках, коли це неможливо або вимоги до якості роботи паралельних з'єднань особливо високі, цей же програмний шлюз можна встановлювати на фізичні серверні системи й експлуатувати без віртуалізації.

Віртуальні шлюзи дуже прості в установці – досить завантажити відповідний інсталяційний образ, і вже через десять хвилин шлюз готовий до первинного налаштування. Потім треба виконати резервне копіювання віртуальної машини, щоб у випадку відмови хост-системи її можна було швидко активувати на іншому хості. Тим самим підтримується високий рівень доступності, що традиційні апаратні комунікаційні контролери забезпечити не здатні. Використовуючи віртуальні телекомунікаційні сервіси для підтримки ключових бізнес-процесів, підприємства хочуть бути впевнені в їхній якості, масштабованості й надійності. Проект інтелектуальної мультисервісної мережі й голосової платформи націлений на створення для цього необхідних умов.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Широко обговорювані в останні роки хмарні комунікаційні сервіси стають реальністю. Їхнє застосування дозволяє підвищити ефективність бізнес-процесів по обслуговуванню клієнтів і збільшити продажі, що, поряд із гнучкістю й зручністю, є стимулом для переходу до нової моделі. Однак щоб бути визнаними замовниками, вони повинні ґрунтуватися на надійній операторській інфраструктурі. Насамперед, це стосується хмарних сервісів з інтегрованою телефонією, від якої традиційно очікують високого рівня доступності.

Складні послуги, попит на які росте, знаходять застосування при рішенні таких завдань, як розширення вики продажів, підвищення якості обслуговування, лояльності й задоволеності клієнтів. Наприклад, підключившись до хмарного ЦОВ, територіально розподілені компанії одержують нові можливості для більше продуктивної роботи з потоками дзвінків, поліпшення різних параметрів бізнесу, обслуговування клієнтів і використання ресурсів.

Для обслуговування зростаючої клієнтської бази розгорнута масштабна інтелектуальна мультисервісна мережа (ІМС) і впроваджена вдосконалена система моніторингу. Завдяки наявному резерву продуктивності інтелектуальної голосової й хмарної платформи, пропоновані сервіси можуть бути адаптовані для компанії будь-якого розміру, що дозволить підтримувати високу якість сервісу при швидкому росту числа користувачів і збільшенні обсягу послуг, а також допоможе справлятися з піковими навантаженнями.

На базі вдосконаленої ІМС надається широкий спектр послуг зв'язку: віртуальна ІР-АТМ, ЦОВ, багатоканальні телефонні номери, номери « 8-800», телефонізація офісів і об'єднання філій під одним номером, послуги місцевого, міжміського й міжнародного зв'язку – однак для їхнього використання потрібно організувати взаємодію з телефонними мережами загального користування (ТфОП). Внутрікорпоративні комунікації реалізуються переважно за допомогою SIP, а з'єднання із ТфОП задіюються лише при обробці вхідних і вихідних місцевих, міжміських і міжнародних дзвінків.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Однак через збільшення кількості дзвінків з SIP-телефонів і шлюзів треба було збільшити число сполучних ліній із ТфОП у сегменті мережі – до 120 потоків Е1. Цифрові потоки Е1 використовуються для організації високоякісних голосових каналів, взаємодії систем VoIP з телефонними мережами операторів традиційного фіксованого й мобільного зв'язку. VoIP-шлюзи й SIP-телефони для SMB вибираються з урахуванням вимог до числа портів і схеми організації зв'язку в клієнта.

Віртуальна АТМ – це георозподілений кластер, доповнений кластером автентифікації й сигнальними серверами.

У частині телефонії SIP це вісім розподілених комутаторів SoftSwitch, що працюють як єдина система. Якщо один з комутаторів виходить із ладу або стає недоступним, забезпечувані ним з'єднання губляться, але кожної з абонентів може миттєво ініціювати повторний виклик. ВАТМ реалізована на платформі OpenSIPS і підтримує СОРМ власної реалізації, що повністю враховує специфіку хмарних телекомунікаційних сервісів

Для забезпечення безперервності бізнесу необхідне резервування компонентів інфраструктури. Це дозволило перетворити центральний технічний вузол у єдиний комплекс, розподілений по трьох площадках ЦОД з повним взаємним резервуванням всіх ключових підсистем, і забезпечити резервування каналів зв'язку між окремими елементами цього вузла, а також між центральним і регіональним вузлами.

За рахунок територіального рознесення вузлів зв'язку й організації дубльованих з'єднань між ними підвищила пропускну здатність, керованість, масштабованість і катастрофостійкість інфраструктури. Крім того, була розширена функціональність хмарної платформи, що підтримує телефонію (SIP-кластер), і поліпшені Web-Інтерфейс особистого кабінету з керуванням налаштуваннями послуг, мережа передачі даних і транспортне приєднання до операторів.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Всі системи в ЦОД функціонують у режимі повного взаємного резервування, для них реалізовані гаряче резервування конфігурацій і плани аварійного відновлення. Аварії й збої на одному з компонентів інфраструктури опорно-транспортної мережі не позначаються на її функціонуванні й не впливають на працездатність мережі в цілому. Наприклад, компоненти кластера АТМ (географічно розподілені комутатори SoftSwitch) і сигнальні сервери ОКС-7 перебувають на різних площадках ЦОД і працюють у режимі автоматичного взаємного резервування. Якщо один з комутаторів цієї єдиної АТМ виходить із ладу, дзвінки перерозподіляються між що залишилися. Відмова кожного із сигнальних серверів теж не впливає на якість сервісів – логіка спрацьовується на іншому сервері.

Типове рішення являє собою три площадки. Між трьома площадками – офісом компанії й двома географічно розподіленими ЦОД – створене відказостійке кільце SDH і організоване з'єднання «кожний з кожним» (full mesh), тобто перший вузол з'єднується із третім через другий без його участі, що ще більше підвищує катастрофостійкість. Рівень надійності орендованих площадок у ЦОД, де перебувають 60-70 серверів оператора, відповідає, відповідно до заяви оператора, Tier III.

У центральному офісі розташоване некритичне для бізнесу встаткування, а також розгорнуті тестового середовища для розробки й перевірки нового функціонала. Основну частину серверного парку становить устаткування SuperMicro – багатопроцесорні (2 і більше) сервери в корпусі 2U+. Для рішення деяких завдань застосовуються сервери HP. До складу встаткування входять, крім іншого, багатодискові системи зберігання даних.

На перерахованих площадках виконані пірінгові приєднання до мереж операторів зв'язку й провайдерів послуг передачі даних. Триває співробітництво з операторами зв'язку по резервуванню сигнальних каналів і їхньому підключенню до географічно розподіленого встаткування SDH. Таке резервування підвищує надійність сервісу й дозволяє територіально

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

розподіленим компаніям, що пред'являють високі вимоги до безперервності бізнесу, задіяти сервіси оператора в ключових бізнес-процесах.

Для підвищення надійності сервісів використовується повноцінна кластеризація з безліччю вузлів і балансуванням навантаження – програмної (засобами OpenSIPS і Web-додатків) і апаратної (на рівні SDH, компонентів мереж передачі даних, серверів і комутаторів). Бази даних, у яких утримуються налаштування ВАТМ, файли CDR, статистика, журнали, дані для білінгу й автентифікації абонентів, функціонують у режимі реплікації, а навантаження на них розподіляється виходячи із критичності додатків.

3.2 Розробка структурної схеми

Система централізованого моніторингу й керування територіально розподіленою мережею контролює роботу кластера АТМ, сервера конфігурацій, інтегрованої системи запобігання шахрайства (антифрод) і системи інтелектуального керування маршрутизацією трафіку. Кореляція інформації системи моніторингу з даними від бізнес-додатків (ВАТМ, CRM і ЦОВ), використовуваних оператором у власних бізнес-процесах, наприклад з довжиною черги звернень користувачів, дозволяє виявляти аномалії у функціонуванні мережі й оперативно реагувати на них.

Програмно-апаратні комплекси й телекомунікаційна інфраструктура, використовувані для надання послуг, створювалися з метою забезпечення безперервності зв'язку й відказостійкості сервісів при різному навантаженні, у тому числі при відновленні сервісів і проведенні планових регламентних робіт. У цьому випадку робота більшої частини простих сервісів (дзвінки, комутація) не зупиняється. Підприємства, для яких безперервність бізнесу життєво важлива, одержали адекватне технічне рішення.

У якості одного з інструментів використовується система моніторингу й відстеження статусів різноманітних сервісів комп'ютерної мережі, серверів і

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

мережного встаткування, що може перевіряти доступність стандартних сервісів без установки якого-небудь програмного забезпечення на хост-системі, а встановлювані програмні агенти дозволяють одержувати розширені дані – про завантаження процесора, використанні мережі, дисковому просторі й т.д. Крім того, програма підтримує моніторинг через SNMP.

Для спостереження й контролю за станом обчислювальних вузлів і служб застосовується також програма для моніторингу функціонування комп'ютерних систем і мереж з відкритим кодом. Вона сповіщає адміністратора, якщо якісь служби припиняють роботу. Для моніторингу ВАТМ у регіонах використовуються їхні власні тригери.

За допомогою бізнес-правил, що налаштовуються, утиліта дозволяє програмувати різні дії для тих або інших ситуацій, аж до блокування окремих користувачів/абонентів при сплесках трафіку (система антифрод) з повідомленням системного адміністратора по різним каналам зв'язку, у тому числі й через SMS-шлюз.

Захист від DDoS реалізована на рівні компонентів мереж передачі даних: інформація про IP-адреси зловмисників регулярно оновлюється за допомогою автоматизованих сценаріїв на підставі даних, одержуваних із загальнодоступних джерел.

З певною періодичністю плани аварійного відновлення (DR) мережі перевіряються на придатність і актуальність після внесення змін в архітектуру мережі, збільшення її масштабів і появи нових компонентів інформаційних систем. Перевірка плану по відновленню критичних систем проводиться в години найменшого навантаження. Резервне копіювання здійснюється залежно від типів даних і вимог до їхньої цілісності.

Канали для передачі голосової інформації у всіх містах присутності оператора резервуються, що дозволяє перенаправляти трафік у випадку аварії по іншому маршруті. Вузол кожної філії має не менш двох крапок приєднання до місцевих мереж телефонії. Крім того, для збільшення пропускної здатності

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

мережі передачі даних при обробці внутрімережних потоків даних і поліпшення взаємодії з іншими операторами зв'язку були замінені мережні комутатори й граничні маршрутизатори.

Обробку всіх додаткових видів послуг п'ятого класу здійснює центральний комутатор, взаємодіючий з регіональними комутаторами четвертого класу й мультиплексорами. Сигнальний трафік ОКС-7 управляється кластером з лінк-серверів. Централізована й база даних білінгу, що забезпечує миттєвий доступ до інформації про всі зроблені дзвінки.

Віртуальна АТМ

Принцип функціонування хмарної АТМ, що є одним з передових рішень у рамках ІР-телефонії, проста: компанія використовує віртуальну телефонну станцію, «розміщену» на сервері провайдера, причому функціонал АТМ можна настроїти під потреби свого бізнесу. Передача даних здійснюється за допомогою SIP- і VoIP-протоколів. Перші потрібні для передачі мультимедіа (відео), а другі – для голосового зв'язку. Всі дані зберігаються на хмарному сервері організації, що надає послугу.

Із цього випливають особливості підключення послуги: не потрібно прокладати додаткові проведення й здобувати складне дороге встаткування, досить лише стабільного доступу до Мережі. Таким чином, за невелику плату організація одержує всі можливості офісної міні-АТМ, а витрати на послуги зв'язку, тим часом, скорочуються в рази.

Можливості хмарної телефонії

Сучасний офіс бідує не тільки в якісному телефонному зв'язку, але й у додаткових функціях, які надає хмарна АТМ:

– Багатоканальний номер дозволяє приймати необмежене число дзвінків одночасно. Якщо всі оператори зайняті, клієнт «займає місце» у віртуальній черзі.

– Додаткові номери для всіх співробітників – простий і зручний спосіб зв'язку усередині компанії.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

– Голосове меню дозволяє істотно знизити навантаження на секретаря або операторів. Клієнт одержує можливість набору додаткового номера й самостійно попадає до потрібного фахівця.

– Інтелектуальна переадресація – це не тільки звичайні функції перенаправку дзвінків у випадках, «якщо зайнято» або «якщо немає відповіді», але й налаштування алгоритмів з'єднання залежно від часу доби, дня тижня й інших факторів.

– Інтеграція із сайтом, у першу чергу, містить у собі популярний сервіс «зворотний дзвінок», що дозволяє користувачам зв'язуватися з операторами прямо зі сторінок онлайн-ресурсу натисканням однієї клавіші.

– Складання деталізованих звітів про кількість і тривалість розмов.

– Відеодзвінки для багатьох стають заміною «живому» спілкуванню з колегами при географічній далекості філій або співробітників.

– Конференцзв'язок – зручне проведення групових переговорів і нарад на будь-яких відстанях.

– Голосова пошта дозволяє клієнтам залишати звукові повідомлення, які потім приходять на e-mail організації у вигляді аудіофайлу.

– Запис телефонних розмов, як показує практика, – необхідний сервіс для компаній, що займаються продажами, обслуговуванням і роботою із клієнтами. Це й постійний контроль над якістю виконання своїх обов'язків операторами, і можливість швидкого рішення проблем у випадку конфліктів і претензій.

– Електронний факс приймає повідомлення й документи так само, як і звичайний. Єдине розходження – вони надходять на e-mail у вигляді файлу й при необхідності друкуються на принтері.

– «Чорний список» обмежує небажані дзвінки.

Переваги віртуальної АТМ

Хмарна телефонна станція, крім зручності у використанні, має й інші, не менш важливі переваги. У першу чергу, ця висока якість ІР-телефонії, значно перевищуючі показники стільникового зв'язка й аналогових каналів.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Але, мабуть, найважливішим для більшості є економічна перевага хмарних АТМ: поза залежністю від розміру компанії їхня організація не вимагає більших витрат. Не потрібні проведення й дороге встаткування, монтаж і технічна підтримка системи, спрощується процедура одержання прямого міського номера. Налаштування віртуальної АТМ із нуля займають 5-30 хвилин часу й не вимагає спеціальних пізнань. Витрати на дзвінки теж істотно зменшуються – приміром, міжміський зв'язок обходиться до 70% дешевше, міжнародна – до 80%, а внутрішні дзвінки й зовсім стають безкоштовні.

Хмарна АТМ популярна серед компаній всіх розмірів і форматів. Індивідуальних підприємців і маленьких колективів залучає можливість швидко почати роботу. При цьому співробітники можуть виконувати свої обов'язки в будь-якій крапці земної кулі. У розпорядженні компанії буде прямий міський номер, що викликає довіра клієнтів і цілодобово доступний для дзвінків і повідомлень. При необхідності можна обійтися й без реального офісу – фактичне місце розташування фірми з віртуальної АТМ надійно приховано від що дзвонять.

При розширенні організації стають помітні й інші достоїнства ІР-АТМ: наприклад, легке збільшення числа каналів виклику й внутрішніх номерів. Кілька філій легко поєднуються в мережу із загальним номером і зручною переадресацією до співробітника з будь-якого регіону.

Великі компанії за допомогою хмарної АТМ одержують можливість створення повноцінного call-центра при невеликих витратах, з повним контролем статистики дзвінків, легенею зміною налаштувань і записом всіх розмов.

Вибір постачальника будь-якої послуги вкрай важливий, а тим більше, коли мова йде про провайдер телефонного зв'язку. Підключення до ІР-телефонії й установка віртуальної АТМ – справа нескладний і недовге, а от підбор оператора, що пропонує умови, що влаштовують, може затягтися. Приведемо кілька критеріїв для вибору хмарної станції.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

В Інтернеті чимало відкликань від уже використовують послугу хмарної телефонної станції, тому, якщо у вас ще немає досвіду подібної діяльності, прочитайте уважно про всі сильні й слабкі сторони продукції тих або інших провайдерів.

Звичайно, у кожної компанії своя сфера діяльності й своїх вимог до провайдера і його послуг, тому сліпо опиратися на чийсь досвід використання тої або іншої хмарної АТМ не можна. Потрібно, виходячи із цілей вашої конкретної організації, вибрати саме той спектр послуг, які будуть необхідні вам. Інші можуть бути більш-менш корисним доповненням, але «базовий» пакет потрібно відбирати найбільше ретельно.

Для рекламних, юридичних або ріелторських агентств, навіть невеликих, украй важлива послуга «віртуальний секретар», у великій компанії, де офіси найчастіше перебувають у різних містах, а те й країнах, не обійтися без внутрішньої корпоративної мережі. Власники інтернет-магазинів повинні задуматися про організацію call-центра, а банкам і медичним центрам буде корисно підключення послуги «гарячої лінії».

Отже, після того, як ви визначилися із провайдером і вибрали потрібний пакет послуг, залишається ще одна немаловажна процедура – підключення до того або іншого тарифу. У кожного постачальника є тариф-«база», що містить у собі, крім прямого багатоканального міського номера, інтелектуальне голосове меню й голосову пошту, можливість занесення в «чорний список», кілька внутрішніх ліній, сховище для записів розмов. Базовий тариф може бути розширений у будь-який момент. Деякі компанії пропонують можливість налаштування індивідуального тарифу, адаптованого під всі ваші вимоги, однак така послуга, як ви розумієте, буде не з дешевих.

Багато які аналітики ринку послуг зв'язки сходяться в тому, що IP-телефонія поступово витісняє аналогову й через кілька років майже не залишиться компаній, що не використовують віртуальні АТМ. Це пояснюється тим, що подібні рішення дозволяють легко управляти вхідними й вихідними

						ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			33

дзвінками, організувати call-центри й скорочувати витрати на зв'язок як великим організаціям, так малому й середньому бізнесу. Спробуйте самі й переконаєтеся в тому, що хмарні АТМ набагато вигідніше й простіше у використанні, ніж застаріваючі аналогові системи.

Розглянемо структурну схему системи. Почнемо з розгляду пристроїв складових системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Гібридні мережні пристрої. Як правило, використовують маршрутизатори, які містять як порти для передачі даних (Ethernet), так і голосові порти (FXS або FXO) – фактично вони є гібридом маршрутизатора й адаптера або шлюзу. Гібридний мережний пристрій може застосовуватися для створення корпоративної системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі, з'єднуючи офісні УПАТМ через мережу Інтернет.

При такій схемі організації прийдеться мати втрати на придбання шлюзів, а також на створення служб експлуатації. При цьому звичайний провайдер послуг мережі Інтернет не зможе гарантувати надання необхідної смуги пропускання, тому що по каналу передаються й дані, і мова. Щоб уникнути подібних проблем доцільніше скористатися послугами ITSP-провайдера. Провайдери системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі гарантують не тільки доступ до шлюзового встаткування, але й необхідну смугу пропускання.

ITSP надають можливості для викликів наступних типів сценаріїв:

– "Від телефону до телефону". Виклик іде зі звичайного телефонного апарата до комплекту встаткування ITSP і через мережу Інтернет доходить до іншого ITSP, що здійснює зворотні перетворення.

– "Від телефону до комп'ютера". Для організації переважніше встановити на кожне автоматизоване робоче місце локальної мережі програмне забезпечення системи аналізу та консолідації розподіленої інформації віртуалізованих

об'єднаних комунікацій у мережі й мікрофон (більшість сучасних ПК уже мають звукові адаптери й акустичні системи) і підключити локальну мережу до ITSP.

– "Від комп'ютера до телефону". Шлюз в IP-системі виконує значне функціональне навантаження. Шлюз призначений для перетворення аналогових мовних і службових сигналів у цифрову послідовність, організації із цієї послідовності пакетів глобальної мережі Інтернет і передачі її в мережу, прийому пакетів і відновлення кодової послідовності цифрових, мовних і службових сигналів і їхнього перетворення в аналогову форму, а також рішення інших завдань, пов'язаних з організацією інтерфейсів, генеруванням і детектуванням сигналів абонентської сигналізації, керуванням режимами телефонних переговорів і деяких інших.

Шлюзи можуть установлюватися на серверах інтернет-провайдерів, міських телефонних станціях, АТМ для установ, серверах локальних обчислювальних мереж, веб-серверах компаній, що потребують в організації голосових гарячих ліній, служб технічної підтримки, діалогових довідкових служб і т.д. Також шлюзи можуть входити до складу маршрутизаторів.

Залежно від схеми організації зв'язку архітектура шлюзу може мінятися, можуть додаватися й інтегруватися деякі функції, виконувані шлюзом, додаватися або мінятися інтерфейси. Однак головні завдання шлюзу – забезпечення якісного дуплексного телефонного спілкування абонентів у режимі пакетної передачі й комутації цифрових сигналів – зберігаються поза залежністю від варіанта.

Варто помітити, що розглянуті вище базові схеми можуть комбінуватися. Можливі різноманітні способи організації IP-телефонного зв'язку з використанням шлюзів, розміщених у функціонально різних точках мережі. Однак у кожному разі шлюз у кожному з варіантів з'єднання буде ключовим елементом.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35



Рисунок 3.1 – Структурна схема системи

Крім терміналів користувачів, які підключаються до мереж аналогової телефонії, існують також і сервери керування. До серверів керування відносяться:

- SIP-сервер (він же SIP-Proxy). Веде список підключених до нього й зареєстрованих клієнтів для того, щоб брати участь у пошуку абонента для з'єднання й керування сеансом з'єднання.

- Проксі для вихідних з'єднань (outbound proxy). Потрібний для обходу клієнтом обмежень, що накладаються використанням трансляції адрес (NAT).

- STUN-сервер. Спеціальний сервер, що дозволяє клієнтові визначити використовуваний тип трансляції адрес для того, щоб спробувати обійти його обмеження без використання проксі для вихідних з'єднань.

- DNS (Domain Name Service). Відомий протокол, що в SIP застосовується для знаходження SIP-сервера для заданого домену шляхом публікації спеціальних DNS-записів у зоні цього домену.

Окремим типом сервера є SIP-агент. Це програма, що безпосередньо займається прийомом і здійсненням дзвінків. Вона містить у собі кодеки й взаємодіє з кінцевим користувачем. SIP-агент убудований у кожному SIP-терміналі – будь то програмний телефон, шлюз або апаратний VoIP-телефон. SIP-агенти взаємодіють між собою прямо (у випадку використання технології SIP Peer to Peer) або через SIP-сервери й проксі для вихідних з'єднань.

						ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			36

Помітимо, що сам протокол SIP голос не передає. Але він використовується для установки сеансу зв'язку й керування ім. Самі голосові дані передає протокол RTP. Причому працюють вони паралельно, але по різних портах: SIP координує сесію, а RTP передає голос.

Без використання RTP SIP не може "передавати" голос, а без SIP – RTP не зможе встановити сеанс зв'язку.

Типи погроз у мережах системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі

Існує кілька основних типів погроз, що представляють небезпеку в мережах системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі:

– Прослуховування. У момент передачі конфіденційної інформації про користувачів (ідентифікаторів, паролів) або конфіденційних даних по незахищених каналах існує можливість прослуховування й зловживання ними в корисливих цілях зловмисником.

– Маніпулювання даними. Дані, які передаються по каналах зв'язку, у принципі можна змінити.

– Підміна даних про користувача відбувається у випадку спроби видачі одного користувача мережі за інший. При цьому виникає ймовірність несанкціонованого доступу до важливих функцій системи.

– Відмова в обслуговуванні (denial of service – DoS) є однією з різновидів атак порушників, у результаті якої відбувається вивід з ладу деяких вузлів або всієї мережі. Вона здійснюється шляхом переповнення системи непотрібним трафком, на обробку якого йдуть всі системні ресурси. Для запобігання даної погрози необхідно використовувати засіб для розпізнавання подібних атак і обмеження їхнього впливу на мережу.

Базовими елементами в області безпеки є:

- автентифікація;
- цілісність;

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

– активна перевірка.

Застосування розширених засобів автентифікації допомагає зберегти в недоторканності вашу ідентифікаційну інформацію й дані. Такі засоби можуть ґрунтуватися на інформації, що користувач знає (пароль).

Цілісність інформації – це здатність засобу обчислювальної техніки або автоматизованої системи забезпечувати незмінність інформації в умовах випадкового й (або) навмисного перекручування (руйнування). Під погрозою порушення цілісності розуміється будь-яка навмисна зміна інформації, що зберігається в обчислювальній системі або передаваній з однієї системи в іншу. Коли зловмисники навмисно змінюють інформацію, говориться, що цілісність інформації порушена. Цілісність також буде порушена, якщо до несанкціонованої зміни приводить випадкова помилка програмного або апаратного забезпечення.

І, нарешті, активна перевірка означає перевірку правильності реалізації елементів технології безпеки й допомагає виявляти несанкціоноване проникнення в мережу й атаки типу DoS. Активна перевірка даних діє як система раннього оповіщення про різні типи неполадок і, отже, дозволяє вжити попереджуючі заходи, поки не нанесений серйозний збиток.

Особливості системи безпеки в системі аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі

У системі аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі повинні забезпечуватися два рівні безпеки: системний і викличний.

Для забезпечення системної безпеки використовуються наступні функції:

– Запобігання неавторизованого доступу до мережі шляхом застосування поділюваного кодового слова. Кодове слово одночасно обчислюється по стандартних алгоритмах на ініціюючій і кінцевій системах, і отримані результати порівнюються. При встановленні з'єднання кожна із двох систем системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

у мережі спочатку ідентифікує іншу систему; у випадку принаймні одного негативного результату зв'язок переривається.

– Списки доступу, у які вносяться всі відомі шлюзи системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

– Запис відмов у доступі.

– Функції безпеки інтерфейсу доступу, включаючи перевірку ідентифікатора й пароля користувача з обмеженням доступу по читанню/запису, перевірку прав доступу до спеціального WEB-серверу для адміністрування.

– Функції забезпечення безпеки виклику, включаючи перевірку ідентифікатора й пароля користувача (необов'язково), статус користувача, профіль абонента.

При встановленні зв'язку шлюзу з іншим шлюзом своєї зони виробляється необов'язкова перевірка ідентифікатора й пароля користувача. Користувач у будь-який час може бути позбавлений права доступу.

Дійсно, при розробці протоколу IP не приділялося належної уваги питанням інформаційної безпеки, однак згодом ситуація мінялася, і сучасні додатки, що базуються на IP, містять досить захисних механізмів. А рішення в області системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі не можуть існувати без реалізації стандартних технологій автентифікації й авторизації, контролю цілісності й шифрування й т.д. Для наочності розглянемо ці механізми в міру того, як вони задіюються на різних стадіях організації телефонної розмови, починаючи з підняття слухавки й закінчуючи сигналом відбою:

– Телефонний апарат. В системі аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі, перш ніж телефон пошле сигнал на встановлення з'єднання, абонент повинен увести свій ідентифікатор і пароль на доступ до апарата і його функцій. Така автентифікація

дозволяє блокувати будь-які дії сторонніх і не турбуватися, що чужі користувачі будуть дзвонити в інше місто або країну за ваш рахунок.

– Установлення з'єднання. Після набору номера сигнал на встановлення з'єднання надходить на відповідний сервер керування дзвінками, де здійснюється цілий ряд перевірок з погляду безпеки. У першу чергу засвідчує дійсність самого телефону – як шляхом використання протоколу 802.1x, так і за допомогою сертифікатів на базі відкритих ключів, інтегрованих в інфраструктуру системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі. Така перевірка дозволяє ізолювати несанкціонованно встановлені в мережі IP-телефони, особливо в мережі з динамічною адресацією. Однак автентифікацією телефону справа не обмежується – необхідно з'ясувати, чи надано абонентові право дзвонити по набраному їм номеру. Це не стільки механізм захисту, скільки міра запобігання шахрайства. Якщо інженерів організації не можна користуватися міжміським зв'язком, то відповідне правило відразу записується в систему керування дзвінками, і з якого би телефону не здійснювалася така спроба, вона буде негайно припинена. Крім того, можна вказувати маски або діапазони телефонних номерів, на які має право дзвонити той або інший користувач. У випадку ж з IP-телефонією проблеми зі зв'язком, подібні до перевантажень ліній в аналоговій телефонії, неможливі: при грамотному проектуванні мережі з резервними з'єднаннями або дублюванням сервера керування дзвінками відмова елементів інфраструктури системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі або їхнє перевантаження не робить негативного впливу на функціонування мережі.

– Телефонна розмова. В системі аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі рішення проблеми захисту від прослуховування передбачалося із самого початку. Високий рівень конфіденційності телефонного зв'язку забезпечують перевірені алгоритми й протоколи (DES, 3DES, AES, IPSec і т.п.) при практично повній відсутності витрат на організацію такого захисту – всі необхідні механізми (шифрування,

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

контролю цілісності, хешування, обміну ключами й ін.) уже реалізовані в інфраструктурних елементах, починаючи від IP-телефону й закінчуючи системою керування дзвінками. При цьому захист може з однаковим успіхом застосовуватися як для внутрішніх переговорів, так і для зовнішніх (в останньому випадку всі абоненти повинні користуватися IP-телефонами). Однак із шифруванням зв'язаний ряд моментів, про які необхідно пам'ятати, впроваджуючи інфраструктуру VoIP. По-перше, з'являється додаткова затримка внаслідок шифрування/дешифрування, а по-друге, ростуть накладні витрати в результаті збільшення довжини переданих пакетів.

– Невидимий функціонал. Дотепер ми розглядали тільки ті небезпеки, яким піддана традиційна телефонія і які можуть бути усунуті впровадженням системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі. Але перехід на протокол IP несе із собою ряд нових погроз, які не можна не враховувати. На щастя, для захисту від цих погроз уже існують рішення, технології й підходи, які добре зарекомендували себе. Більшість із них не вимагає ніяких фінансових інвестицій, будучи вже реалізованими в мережному встаткуванні, що і лежить в основі будь-якої інфраструктури системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі. Найпростіше, що можна зробити для підвищення захищеності телефонних переговорів, коли вони передаються по тій же кабельній системі, що й звичайні дані, – це сегментувати мережу за допомогою технології VLAN для усунення можливості прослуховування переговорів звичайними користувачами. Гарні результати дає використання для сегментів системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі окремого адресного простору. І, звичайно ж, не варто скидати з рахунків правила контролю доступу на маршрутизаторах (Access Control List, ACL) або міжмережних екранах (firewall), застосування яких ускладнює зловмисникам завдання підключення до голосових сегментів.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

– Спілкування із зовнішнім миром. Якої би переваги IP-телефонія не надавала в рамках внутрішньої мережі, вони будуть неповними без можливості здійснення й прийому дзвінків на міські номери. При цьому, як правило, виникає завдання конвертації IP-трафіка в сигнал, переданий по телефонній мережі загального користування (ТфОП). Вона вирішується за рахунок застосування спеціальних голосових шлюзів (voice gateway), що реалізують і деякі захисні функції, а сама головна з них – блокування всіх протоколів системі аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі (H.323, SIP і ін.), якщо їхнього повідомлення надходять із неголосового сегмента. Для захисту елементів голосової інфраструктури від можливих несанкціонованих впливів можуть застосовуватися спеціалізовані рішення – міжмережні екрани (ММЕ), шлюзи прикладного рівня (Application Layer Gateway, ALG) і прикордонні контролери сеансів (Session Border Controller). Зокрема, протокол RTP використовує динамічні порти UDP, відкриття яких на міжмережному екрані приводить до появи зяючої діри в захисті. Отже, міжмережний екран повинен динамічно визначати використовувані для зв'язку порти, відкривати їх у момент з'єднання й закривати по його завершенню. Інша особливість полягає в тому, що ряд протоколів, наприклад, SIP, інформацію про параметри з'єднання розміщає не в заголовку пакета, а в тілі даних. Тому пристрій захисту повинне бути здатне аналізувати не тільки заголовок, але й тіло даних пакета, отримуючи з нього всі необхідні для організації голосового з'єднання відомості. Ще одним обмеженням є складність спільного застосування динамічних портів і NAT.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема розробленого програмного забезпечення.

З функціональної схеми ми бачимо, що основний функціональний модуль включає в себе наступні функціональні підблоки:

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

- IP-телефонія.
- Селективний зв'язок.
- Відеоконференція.
- Формування вхідної інформації.
- База даних абонентів мережі спеціального зв'язку.
- Білінг.
- Виведення статусу ПЗ.
- Довідник.

Дані, які поступають з основного функціонального модуля шифруються алгоритмом шифрування ДСТ 28147-89, та перетворюються згідно протоколу H.323, після чого передаються до IP-мережі.

Робота з програмою починається з введення інформаційного вікна й активізації системи меню. Робота програми здійснюється по діалоговому і подійному режиму, при цьому по діалогом розуміється надання користувачу декількох альтернатив і обробка його вибору. У діалогову систему входять головне меню з відповідними спливаючими підменю а також діалогові вікна. Під подіями розуміються процеси, що активізуються користувачем (наприклад – натискання функціональних клавіш), а також програмні події – одержання з'єднання з абонентом або закінчення з'єднання з абонентом. На підставі даних подій активізуються процедури контролю допустимості даних.

Головне вікно програми призначене для запуску основних процедур програми і завершення роботи з програмою.

Модуль роботи з довідниками містить у собі два довідники:

- Довідник – Довідкова система.
- Довідник – Інструкція користувача.

Модуль «Формування вхідної інформації» призначений для введення первинних даних і перегляду раніше занесених. Даний модуль реалізує задачі обліку телефонних номерів, забезпечуючи введення номерів та їх легенд, ранжування за важливістю та їх знищення.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

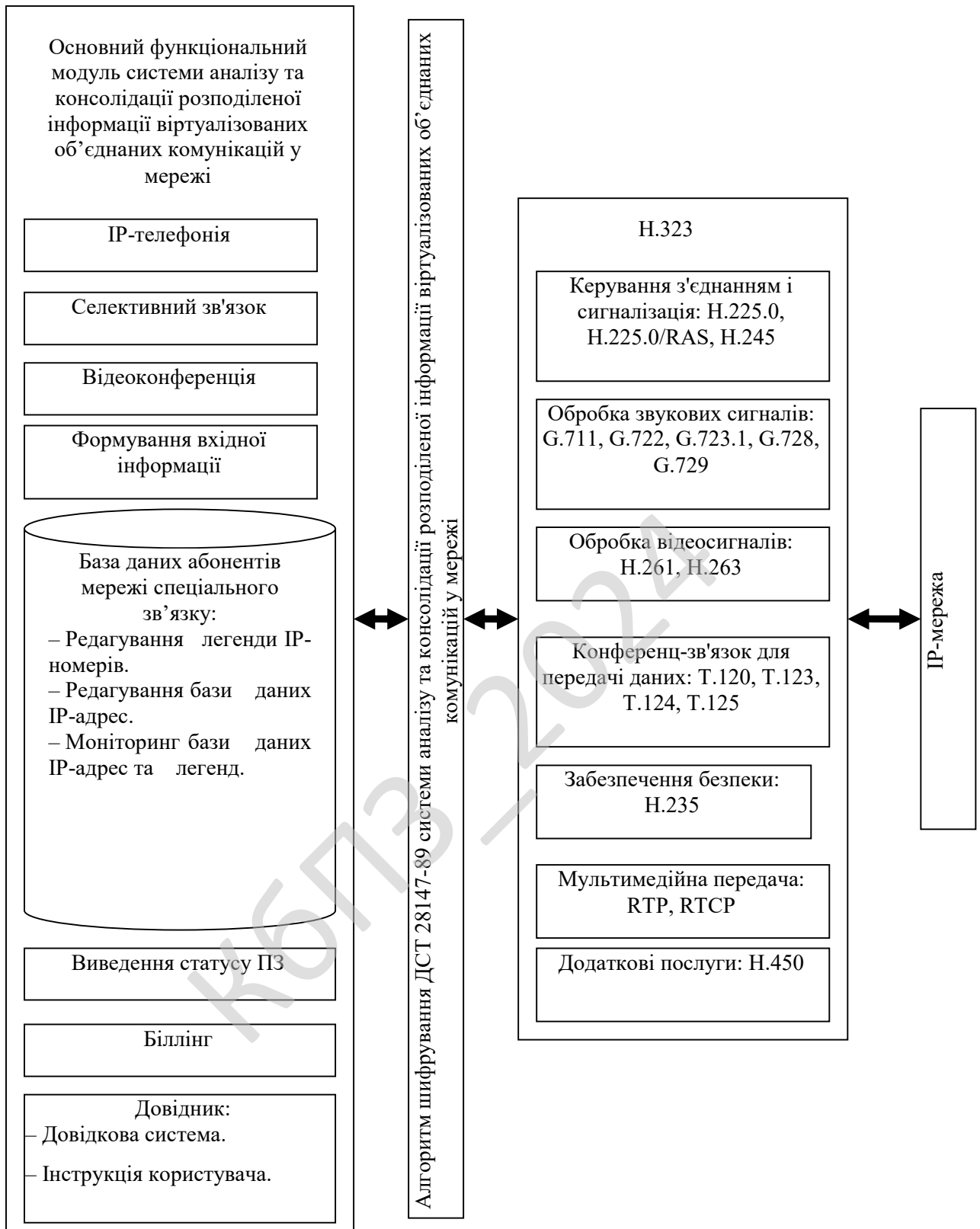


Рисунок 3.2– Функціональна схема розробленої системи

У комп'ютерних системах користувачі для введення, перегляду та редагування інформації бази даних (IP-адрес та відповідних легенд) можуть застосовувати форми. Основні переваги використання форм наступні:

– При введенні даних у поля-форми, додаток може зчитувати словник даних сервера й автоматично перевірити допустимість даних відповідно до правил цілісності.

– Поле введення у формі може представляти список допустимих значень, з яких користувачі можуть легко вибрати потрібне.

– Область форми може виводити шаблон, що відповідає поточної виведеної у формі запису.

– Командні кнопки у формі можуть виконувати дії, зв'язані з виведеної у формі поточною записом.

Біллінгова система – автоматизована система розрахунків з абонентами за надані послуги. Керування й статистика може бути доступна з будь-якої точки мережі через Веб-інтерфейс.

Стандарт H.323 визначає широкі вимоги для багатьох різних протоколів, які становлять повний стек протоколів H.323.

Стек H.323 складають 7 груп протоколів:

- керування й сигналізація;
- обробка звукових сигналів;
- обробка відеосигналів;
- конференц-зв'язок;
- передача мультимедійної інформації;
- забезпечення інформаційної безпеки;
- додаткові послуги;

1. Керування з'єднанням і сигналізація:

– H.225.0: протоколи сигналізації й пакетування мультимедійного потоку (використовує підмножину протоколу сигналізації Q.931).

– H.225.0/RAS: процедури реєстрації, допуску й стану.

– H.245: протокол керування для мультимедіа.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

2. Обробка звукових сигналів:

- G.711: імпульсно-кодова модуляція тональних частот.
- G.722: кодування звукового сигналу 7 кГц в 64 кбіт/с.
- G.723.1: мовні кодери на дві швидкості передачі для організації мультимедійного зв'язку зі швидкістю передачі 5.3 і 6.3 кбіт/с.
- G.728: кодування мовних сигналів 16 кбіт/с за допомогою лінійного пророкування з кодуванням сигналу порушення з малою затримкою.
- G.729: кодування мовних сигналів 8 кбіт/с за допомогою лінійного пророкування з алгебраїчним кодуванням сигналу порушення сполученої структури.

3. Обробка відеосигналів:

- H.261: відеокодеки для аудіовізуальних послуг зі швидкістю 64 кбіт/с.
- H.263: кодування відеосигналу для передачі з малою швидкістю.

4. Конференц-зв'язок для передачі даних:

- T.120: це стек протоколів (який включає T.123, T.124, T.125) для передачі даних між окінцевими пунктами. Він може використовуватися для різних додатків в області спільної роботи (Collaboration Work), такий як колективне редагування растрових зображень, спільне використання додатків і спільна організація документів. В T.120 застосовується багаторівнева архітектура, подібна моделі OSI.

5. Мультимедійна передача:

- RTP: транспортний протокол реального часу.
- RTCP: протокол керування передачею в реальному часі.

6. Забезпечення безпеки:

- H.235: забезпечення безпеки й шифрування для мультимедійних терміналів мережі H.323.

7. Додаткові послуги:

- H.450.1: узагальнені функції для керування додатковими послугами в H.323.
- H.450.2: переклад з'єднання на телефонний номер третього абонента.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

- Н.450.3: переадресація виклику.
- Н.450.4: утримання виклику.
- Н.450.5: паркування виклику (park) і відповідь на виклик (pick up).
- Н.450.6: повідомлення про виклик, що надійшов, у стані розмови.
- Н.450.7: індикація повідомлення, що очікує.
- Н.450.8: служба ідентифікації імен.
- Н.450.9: служба завершення з'єднання для мереж Н.323.

Модуль роботи з базою даних абонентів мережі спеціального зв'язку включає в себе наступні підмодулі:

- Редагування легенди ІР-номерів.
- Редагування бази даних ІР-адрес.
- Моніторинг бази даних ІР-адрес та легенд.

Призначення даного модуля є пошук і перегляд інформації з телефонних даних адрес абонентів корпорації, а також їх легенд.

Інформаційною базою даного модуля є таблиці: Значення ІР-номерів та Легенда ІР-номерів. Дані в інформаційну базу заносяться за допомогою спеціальних форм, що викликаються з головного меню програми.

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

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерської дипломної роботи, наведена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

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

чому буде показано взаємодію системи. 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.

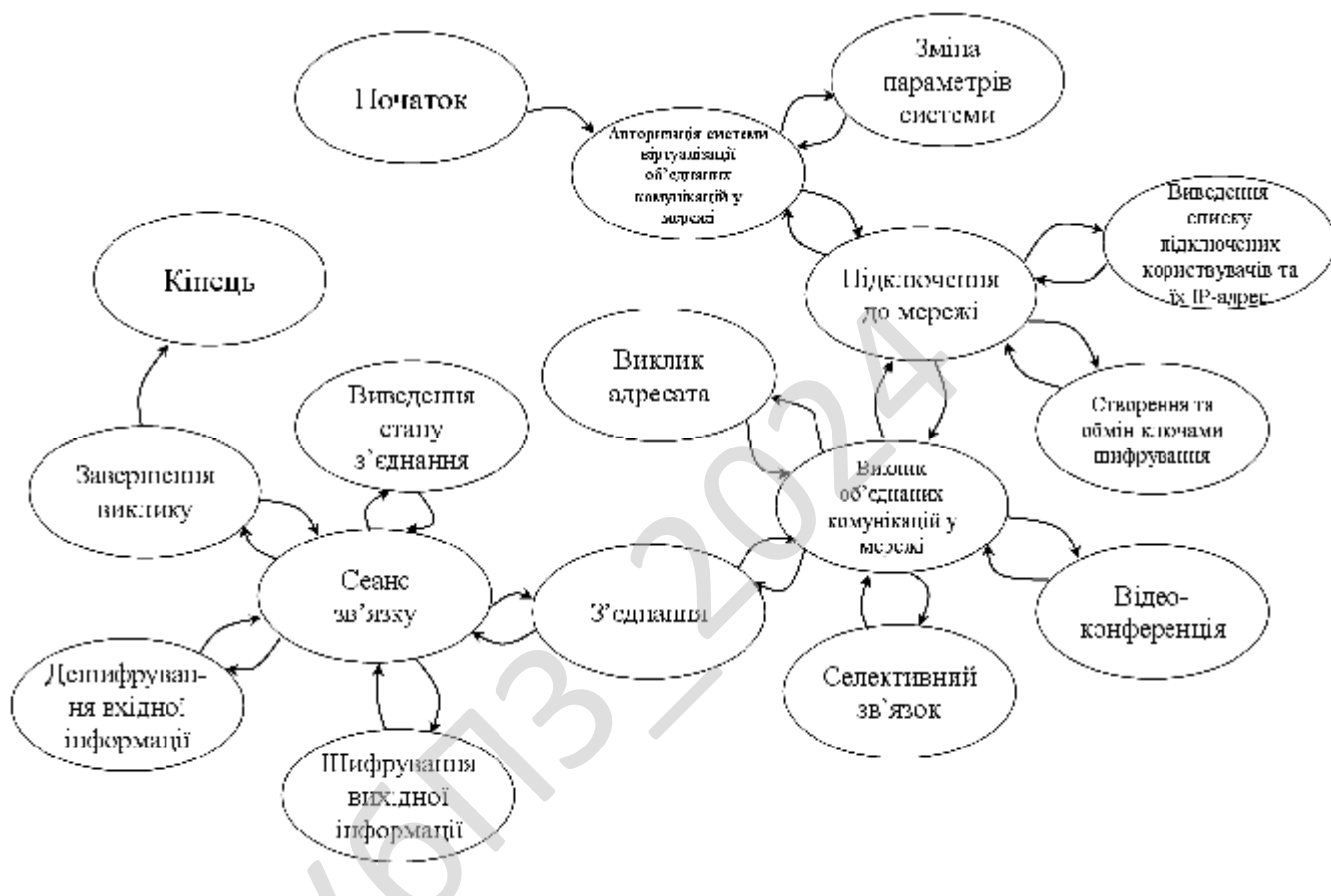


Рисунок 3.3 – Діаграма взаємодії процесів

- Сховища даних (репозиторії).
- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.

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

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем.

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

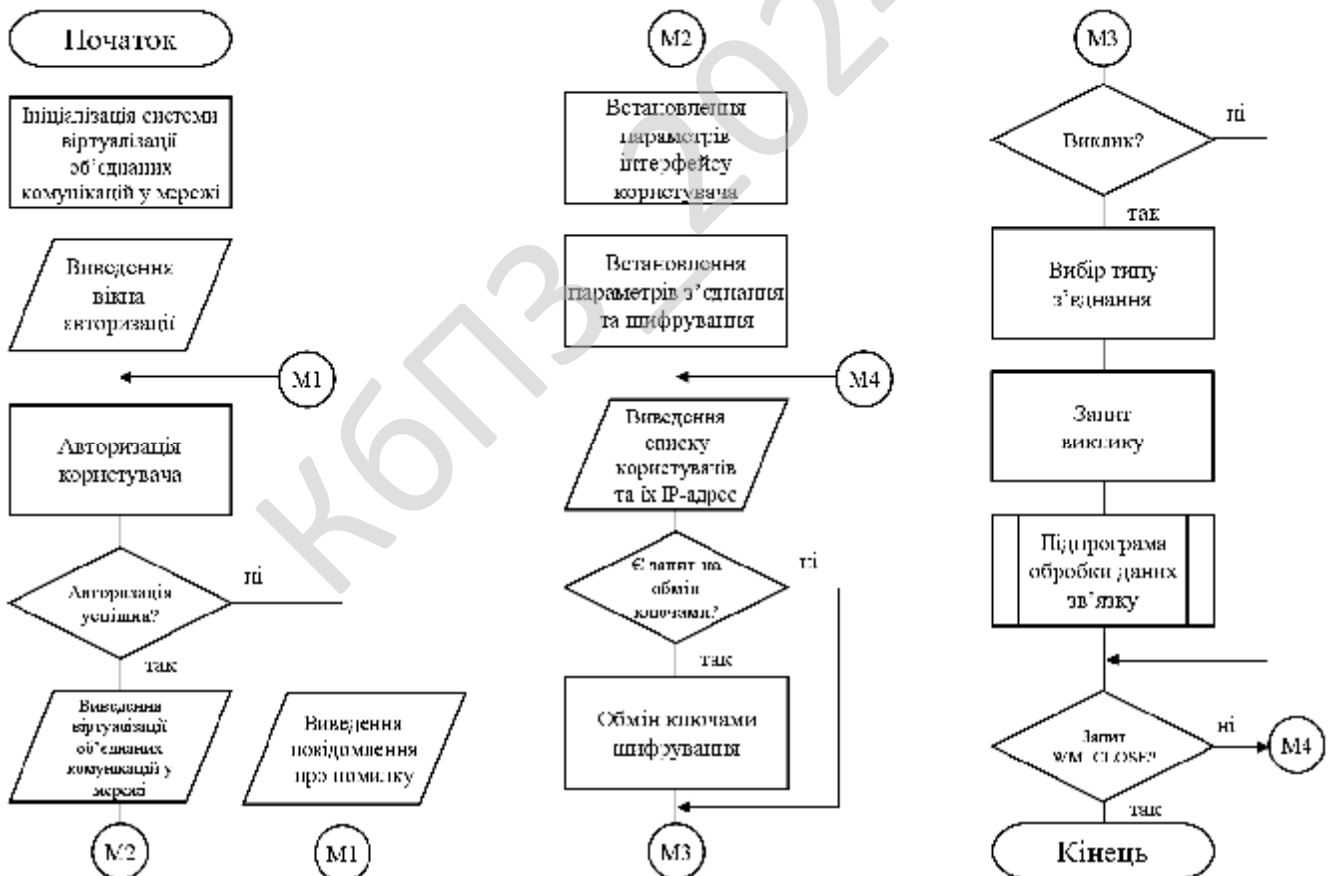


Рисунок 4.1 – Блок-схема основної програми

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.

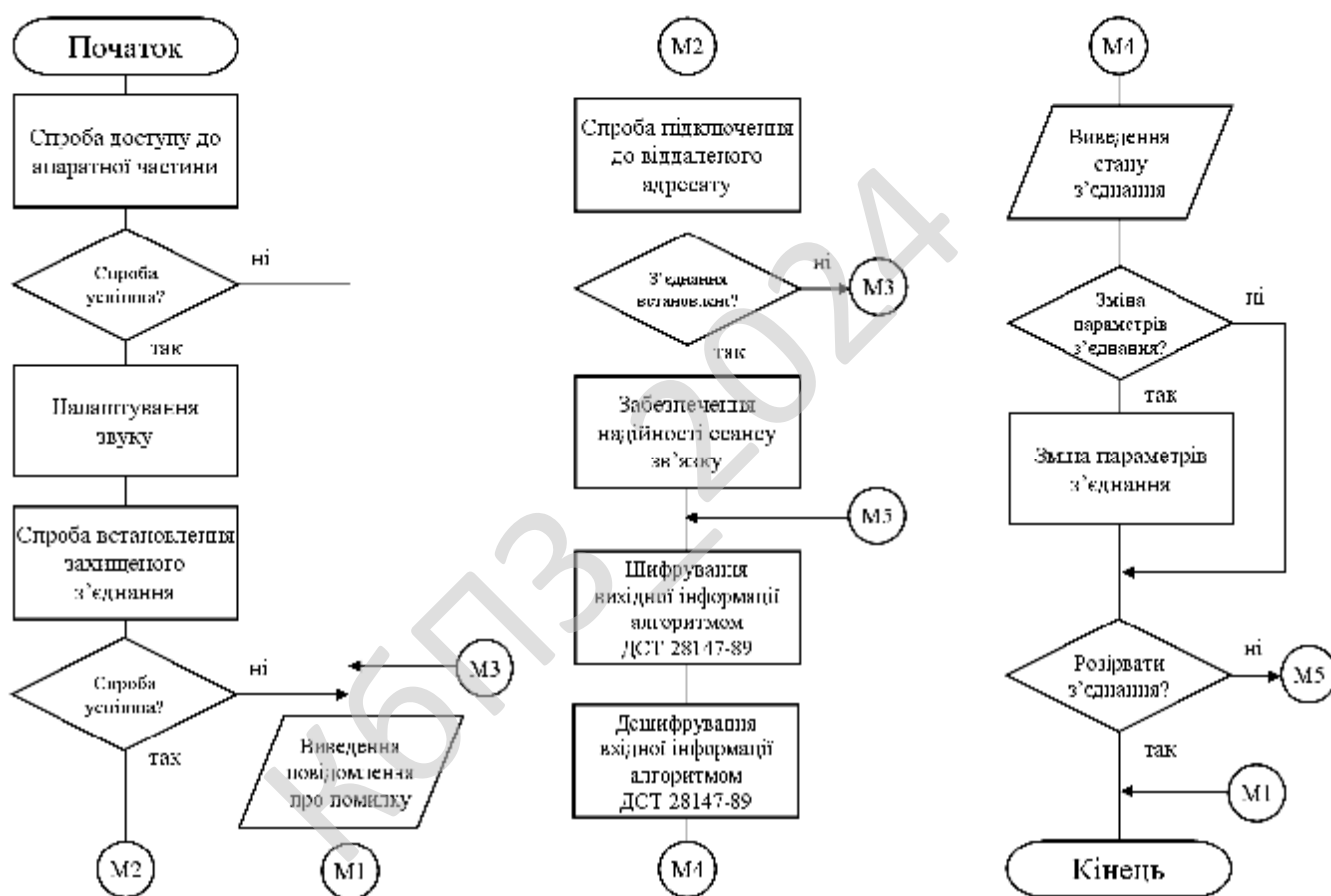


Рисунок 4.2 – Блок-схема роботи підпрограми

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

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма

буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

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

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.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

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

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.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

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

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

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

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

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

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

Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

(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.

Реальний безперервний сигнал володіє спектром, основна частина енергії якого зосереджена в обмеженій смузі частот. Це зумовлене тим, що прилади, що формують і перетворюють повідомлення і сигнали, а також канали зв'язку мають кінцеву смугу пропускання.

Функція часу з обмеженим по ширині спектром повністю визначається своїми миттєвими значеннями, відрахованими через інтервали часу:

$$\Delta t = 1/2F_m, \quad (4.1)$$

де F_m – найвища частота спектру сигналу.

Це положення складає зміст теореми Котельникова. Теорема дискретизації, або, як її ще називають, теорема Котельникова, теорема Уїтекера, формулюється в такий спосіб: безперервна функція $X(t)$ з обмеженим спектром, тобто не має у своєму спектрі:

$$F\{X(t)\} = \int_{-\infty}^{\infty} X(t) \cdot e^{-j2\pi ft} dt, \quad (4.2)$$

складових із частотами, що лежать за межами смуги $f \in (-F_m, F_m)$, повністю визначається послідовністю своїх обчислень у дискретні моменти часу $X(t_i)$, що виходять із кроком $\Delta t < 1/F_m$. Іншими словами безперервна детермінована функція часу $X(t)$, що має обмежений спектр, може бути розкладена в ряд по ортогональним функціям часу виду:

$$\Psi_i(t) = \frac{\sin 2F_m \pi(t - i\Delta t)}{2\pi F_m (t - i\Delta t)}, \quad (4.3)$$

з коефіцієнтами, рівними значенням функції $X(i\Delta t)$. Цей розклад, що називається рядом Котельникова, має наступний вигляд: $X(t) = \sum_{i=-\infty}^{\infty} X(i\Delta t)\Psi_i(t)$.

Функція відрхунків має наступні властивості:

1. Вона дорівнює 1 при $t = i\Delta t$ і рівна 0 в усіх інших точках $t = j\Delta t; j \neq i$.
2. Її спектр рівномірний в межах смуги частот від $-F_{\min}$ до F_{\max} але рівний 0 поза цією смугою.

3. Вона має властивості ортогональності:

$$\int_{-\infty}^{\infty} \Psi_i(t)\Psi_j(t) dt = \begin{cases} 0 & \text{при } i \neq j \\ \frac{1}{2F_{\max}} & \text{при } i = j \end{cases}$$

Якщо функція $f(t)$ з обмеженим спектром розглядається на кінцевому інтервалі часу T , то число дискрет, що передаються, буде дорівнювати: $T/\Delta t = 2F_m T$. Величину $F_m T$ називають базою сигналу.

Доказ сформульованої теореми ґрунтується на однозначній відповідності між сигналами й відповідними їм спектрами. Іншими словами, якщо сигнали однакові, те й відповідні їм спектри також однакові. І, навпаки, якщо спектри двох сигналів однакові, той і відповідний сигнали також однакові.

Опис технологічного процесу обробки даних. Технологічний процес складається з двох основних етапів – збір і облік даних по IP-адресам й формування відповідної легенди та власне сама робота програми у режимі IP-телефонії та селективного зв'язку. Вони можуть виконуватися в будь-який календарний момент часу і включають операції введення, з'єднання і ін. Операції мають програмне виконання, підлегле єдиній алгоритмічній схемі.

Програма реалізована в середовищі Python. Робота з програмою починається з виведення інформаційного вікна й активізації системи меню й здійснюється в діалоговому і по-дійному режиму. При цьому під діалогом розуміється надання користувачу декількох альтернатив і обробка його вибору. У діалогову систему входять головне меню з відповідними спливаючими підменю а також діалогові вікна.

Під подіями розуміються процеси, що активізуються користувачем (наприклад – натискання функціональних клавіш), а також програмні події – з'єднання по IP-технології.

Програма складається з наступних основних модулів.

Основна процедура – конфігурація середовища оточення, формування основного екрана програми, створення системи головного меню і відповідних підменю, активізація меню. Процедура обробки головного меню – запуск відповідної процедури. Процедура введення даних – забезпечення введення інформації у бази даних IP-адрес та легенд до них, контроль за допустимістю значень, забезпечення введення даних шляхом вибору зі списку.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Допоміжні процедури і функції – реалізація запитів, повідомлень, формування списків вибору IP-адрес та легенд, а також контроль за даними, що вводяться. Усі модулі в програмі зв'язані між собою за даними, що аналізуються на вході і виробляються на виході. Дані в модулі надходять через діалог з користувачем, параметри і документи інформаційної бази. Передача даних від одного модуля до іншого здійснюється тільки через збережені документи.

Дані через діалог можуть бути отримані прямим і непрямим способом. Прямий спосіб реалізується шляхом їхнього введення за шаблоном чи по запиту конкретних значень реквізитів. Непрямий спосіб – шляхом чи меню логічних (альтернативних) запитів – «так», «ні». При непрямому способі дані, що надходять у модуль, заздалегідь передбачені алгоритмом, але зовні виглядають в обліку відомими фразами.

Параметри (мова) – вхідні дані, отримані у виді конкретних значень, переданих в оперативній пам'яті суміжним модулям (функціям).

Обґрунтування інформаційного забезпечення – перелік первісних даних. Під вхідною інформацією розуміється вся інформація, необхідна для вирішення задачі і розташована на різних носіях: первинних документах, машинних носіях, у пам'яті персонального комп'ютера. Вхідною інформацією для розроблювальної в дипломному проєкті корпоративної системи зв'язку з використанням IP-технологій є мова або відеозображення.

Щоб передати мову через телефонну мережу, мовну інформацію потрібно перетворити в аналоговий електричний сигнал. При переході до цифрових мереж зв'язку виникла необхідність перетворити аналоговий електричний сигнал у цифровий формат на передавальній стороні, тобто закодувати, і перевести назад в аналогову форму, тобто декодувати, на прийомній стороні [5-8].

Процес перетворення аналогового мовного сигналу в цифрову форму називають аналізом або цифровим кодуванням мови, а зворотний процес відновлення аналогової форми мовного сигналу – синтезом або декодуванням мови.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Ціль будь-якої схеми кодування – одержати таку цифрову послідовність, що вимагає мінімальної швидкості передачі й з якої декодер може відновити вихідний мовний сигнал з мінімальними змінами.

При перетворенні мовного сигналу в цифрову форму, мають місце два процеси – дискретизація, тобто формування дискретних у часі відрахунків амплітуди сигналу, і квантування, тобто дискретизація отриманих значень за амплітудою. Ці дві функції виконуються т.зв. аналого-цифровими перетворювачами (АЦП), які розміщуються в сучасних АТС на платі абонентських комплектів, а у випадку передачі мови по IP-мережах – у терміналі користувача (комп'ютері або IP-телефоні).

Процес аналого-цифрового перетворення одержав, стосовно до систем зв'язку, назву імпульсно-кової модуляції (ІКМ).

Щоб знизити необхідну швидкість передачі біт, застосовують нелінійний (логарифмічний) закон квантування, тобто квантуванню піддається не амплітуда сигналу, а її логарифм. У цьому випадку має місце процес «стиску» динамічного діапазону сигналу, а при відновленні сигналу відбувається зворотний процес.

Сьогодні застосовуються два основні різновиди ІКМ: з кодуванням по m -закону й по A -закону. У результаті стиску сигнал з амплітудою, що кодується 12-13 бітами, описується всього вісьма бітами. Розрізняються ці різновиди ІКМ деталями процесу стиску (m -закон кодування переважніше використовувати при малій амплітуді сигналу й при малому відношенні сигнал/шум). У Північній Америці використовується кодування по m -закону, а в Європі – по A -Закону. Тому при міжнародному зв'язку в багатьох випадках потрібне перетворення m -закону в A -закон, відповідальність за яке несе країна, у якій використовується m -закон кодування.

В обох випадках кожний відлік кодується 8 бітами, або одним байтом, який можна вважати звуковим фрагментом. Для передачі послідовності таких фрагментів необхідна пропускна здатність каналу, рівна 64 Кбіт/с. Оскільки ІКМ була першою стандартною технологією, що одержала широке застосування в

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

цифрових системах передачі, пропускна здатність каналу, рівна 64 Кбіт/с, стала всесвітнім стандартом для цифрових мереж всіх видів, причому – стандартом, що забезпечує передачу мови з дуже гарною якістю. Однак така висока якість передачі мовного сигналу (що є еталоном при оцінці якості інших схем кодування) досягнута в системах ІКМ за рахунок явно надлишкова, при сучасному рівні технології, швидкості передачі інформації [3].

Щоб зменшити властиву ІКМ надмірність і знизити вимоги до смуги пропускання, послідовність чисел, отримана в результаті перетворення мовного аналогового сигналу в цифрову форму, піддається математичним перетворенням, що дозволяють зменшити необхідну швидкість передачі. Ці перетворення «сирого» цифрового потоку в потік меншої швидкості називають «стиском» (а часто – кодуванням, розглядаючи ІКМ як якусь відправну точку для подальшої обробки інформації). Існує безліч підходів до «стиску» мовної інформації; всі їх можна розділити на три категорії: кодування форми сигналу (waveform coding), кодування вихідної інформації (source coding) і гібридне кодування, що представляє собою сполучення двох підходів.

Кодування форми сигналу. Імпульсно-кодова модуляція, по суті, і являє собою схему кодування форми сигналу. Однак цікавлять більш складні алгоритми, що дозволяють знизити вимоги до смуги пропускання. Розглянуті методи кодування форми сигналу використовують ту обставину, що між випадковими значеннями декількох послідовних обчислень існує деяка залежність. Це дозволяє з досить високою точністю пророчити значення будь-якого відліку на основі значень декількох попередніх йому обчислень.

При побудові алгоритмів кодування названа закономірність використовується двома способами. По-перше, є можливість змінювати параметри квантування залежно від характеру сигналу. У цьому випадку крок квантування може змінюватися, що дозволяє певною мірою згладити протиріччя між зменшенням числа біт, необхідних для кодування величини відліку при збільшенні кроку квантування, і звуженням динамічного діапазону кодера,

неминучим без адаптації.

Деякі алгоритми передбачають зміну параметрів квантування приблизно в рамках вимовних складів, а деякі змінюють крок квантування на основі аналізу статистичних даних про амплітуду сигналу, отриманих за відносно короткий проміжок часу. По-друге, існує підхід, називаний диференціальним кодуванням або лінійним проорокуванням. Замість того, щоб кодувати вхідний сигнал безпосередньо, кодують різницю між вхідним сигналом і «передвіщеною» величиною, обчисленою на основі декількох попередніх значень сигналу. Описаний метод називається лінійним проорокуванням, тому що він використовує тільки лінійні функції попередніх обчислень. Найпростішою реалізацією останнього підходу є так звана дельта-модуляція (ДМ), алгоритм якої передбачає кодування різниці між сусідніми обчисленнями сигналу тільки одним інформаційним бітом, забезпечуючи передачу, по суті, тільки знака різниці.

Алгоритмом, побудованим на описані вище принципах, є алгоритм адаптивної диференціальної імпульсно-кової модуляції (АДІКМ) (G.726). Алгоритм передбачає формування сигналу помилки проорокування і його наступне адаптивне квантування.

При досить гарних характеристиках алгоритму, АДІКМ практично не застосовується для передачі мови по мережах з комутацією пакетів, тому що цей алгоритм дуже чутливий до втрат цілих блоків відліку, що відбуваються при втратах пакетів у мережі. У таких випадках порушується синхронізація кодера й декодера, що приводить до катастрофічного погіршення якості відтворення мови навіть при малій імовірності втрат [1-7].

Система аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі використовує Python та різні модулі для обробки даних у віртуалізованому середовищі. Основна це забезпечити обробку та аналіз даних у масштабованому середовищі з можливістю інтеграції з іншими компонентами мережі.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Розглянемо частину вихідного коду.

```
import threading
import queue
import time
import random

# Ініціалізація черги для зберігання розподілених повідомлень
message_queue = queue.Queue()

# Модуль симуляції отримання розподіленої інформації
def data_acquisition(node_id):
    while True:
        data = f"Node {node_id} data: {random.randint(0, 100)}"
        message_queue.put(data)

        # Симуляція часу передачі
        time.sleep(random.uniform(0.5, 2))

# Модуль аналізу отриманої інформації
def data_analysis():
    analyzed_data = []

    while True:
        if not message_queue.empty():
            data = message_queue.get()

# Аналіз отриманих даних
            analyzed_result = f"Analyzed {data}"
            analyzed_data.append(analyzed_result)

# Виведення проаналізованих даних
            print(analyzed_result)

# Симуляція часу аналізу
            time.sleep(1)

# Обчислення середнього значення для контролю ефективності
            if len(analyzed_data) >= 10:
                avg_value = sum([int(item.split(':')[1]) for item in analyzed_data]) /
                len(analyzed_data)
                print(f"Average value after 10 analyses: {avg_value}")
```

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

```

analyzed_data.clear()

# Модуль керування потоками для розподіленої обробки
def main_system():
    nodes = []

# Ініціалізація віртуалізованих вузлів
    for node_id in range(1, 6):
node_thread = threading.Thread(target=data_acquisition, args=(node_id,))
node_thread.start()
nodes.append(node_thread)

# Запуск модуля аналізу
    analysis_thread = threading.Thread(target=data_analysis)
analysis_thread.start()

# Очікування завершення обробки
for node_thread in nodes:
    node_thread.join()
analysis_thread.join()

# Запуск системи
if __name__ == "__main__":
    main_system()

```

Опис архітектури

1. `data_acquisition` модуль отримання розподіленої інформації, який симулює отримання даних з різних вузлів мережі
2. `data_analysis` модуль, що аналізує отримані дані і виконує консолідацію на основі розрахунків та аналізу середніх значень
3. `main_system` — основний модуль, що керує потоками для паралельної обробки інформації від кількох вузлів та аналізує ці дані

Розрахунки:

- Середнє значення проаналізованих даних визначає ефективність роботи системи.
- Кожен вузол мережі генерує дані у випадковому порядку, що симулює реальну динаміку роботи мережі.
- Модуль аналізу забезпечує обчислення середніх значень для контролю

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

стабільності отриманих даних.

Такий підхід гарантує масштабованість та надійність системи аналізу й консолідації, що підтверджується регулярним контролем середніх значень аналізованих даних.

4.2 Захист розробленого програмного забезпечення

Дані у програмному забезпеченні захищаю за допомогою MISTY1. MISTY1 – блоковий алгоритм шифрування, створений для компанії Mitsubishi Electric криптологом Міцуру Мацуї. Назва є аббревіатурою Mitsubishi Improved Security Technology. Алгоритм був розроблений в 1995-1996 рр. Відомі також дві модифікації алгоритму MISTY1: MISTY2 і KASUMI

Шифр став переможцем на Європейському конкурсі NESSIE. У результаті аналізу алгоритму експерти зробили вивід, що ніяких серйозних уразливостей даний алгоритм не має (переважно, завдяки вкладеним мережам Фейстеля, що суттєво утрудняє криптоаналіз). У нього високий запас криптостійкості, алгоритм має високу швидкість шифрування й досить ефективний для апаратної реалізації.

Алгоритм був розроблений на основі теорії «підтвердженої безпеки» проти диференціального й лінійного криптоаналізу. Цей алгоритм був спроектований, щоб протистояти криптоатакам, відомим на момент створення.

З моменту публікації MISTY1 було проведено багато досліджень, щоб оцінити його рівень безпеки.

Диференціальний і неможливий диференціальний криптоаналіз високого порядку ефективно застосовується до блокових шифрів з малим ступенем. Найкращі результати для обох варіантів були отримані для 5-рівневого алгоритму MISTY1 без FL функцій. Саме FL функції й широкобітні AND/OR операції в сильно утрудняють використання диференціального криптоаналізу, що не заважає проведенню в цьому напрямку всі нових досліджень і досягненню усе більш близьких до розв'язку результатів.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

Параметри вихідних даних

MISTY1 – це шифр на основі вкладених мереж Фейстеля з варьюємим числом раундів. Рекомендоване використання 8-раундової версії, але може використовуватися будь-яка кількість раундів, кратне 4-м. Розмір блоку вихідного тексту – 64 біта, розмір ключа – 128 біт.

Для роботи алгоритму також попередньо виконується процедура розширення ключа, яка для 8-мі раундів обчислює 1216 бітів ключової інформації з 128-бітного ключа шифрування.

Структура алгоритму

Для задоволення вимогам конкурсу NESSIE, а також для задоволення завдання мультиплатформеності, в алгоритмі MISTY1 використовувалися наступні методи шифрування:

- Логічні операції.
- Арифметичні операції.
- Операції зрушення.
- Таблиці перестановок.

Як говорилося вище, алгоритм MISTY1 заснований на «вкладених» мережах Фейстеля. Спочатку блок вихідного тексту розбивається на два 32-бітних субблоки, після чого виконується r раундів наступних перетворень[1]:

- У кожному непарному раунді обое субблоки обробляються операцією FL
- Над оброблюваним субблоком виконується операція FO.
- Результат цих операцій накладається логічною операцією «, що виключає або» (XOR) на неопрацьований субблок.
- Субблоки міняються місцями. Після заключного раунду обое субблоки ще раз обробляються операцією FL.

Операція FL

Оброблюваний 32-бітний субблок розбивається на два 16-бітних фрагмента, до яких застосовуються операції, де:

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

- L і R – вхідні значення лівого й правого фрагментів відповідно;
- L' і R' – вихідні значення;
- i – фрагменти j -го підключа i -го раунду для функції FL (процедура розширення ключа докладно описана далі);
- i – побітові логічні операції «і» і «або» відповідно.

Операція FO

Саме ця функція є вкладеною мережею Фейстеля. Тут, як і раніше, виконується розбивка вхідного значення на два 16-бітних фрагмента, що проходять 3 раунду наступних перетворень:

- На лівий фрагмент операцією XOR накладається фрагмент ключа, де k – номер раунду функції FO.
- Лівий фрагмент обробляється операцією FI.
- На лівий фрагмент накладається операцією XOR значення правого фрагмента.
- Фрагменти міняються місцями.

Після третього раунду операції FO на лівий фрагмент накладається операцією XOR додатковий фрагмент ключа.

Операція FI

Дана операція також представляє собою третій рівень вкладеності мережі Фейстеля. На відміну від двох верхніх рівнів, дана мережа є незбалансованою: оброблюваний 16-бітний фрагмент ділиться на дві частини: 9-бітну ліву й 7-бітну праву. Потім виконуються 3 раунду перетворень, що впливають:

- Ліва частина зазнає обробці S-box. 9-бітна частина (в 1-м і 3-м раундах) обробляється таблицею S_9 , а 7-бітна (в 2-м раунді) – таблицею S_7 . Дані таблиці описані нижче.
- На ліву частину операцією XOR накладається поточне значення правої частини. При цьому, якщо праворуч 7-бітна частина, вона доповнюється нулями ліворуч, а в 9-бітної частини віддаляються ліворуч два біти.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

– У другому раунді на ліву частину операцією XOR накладається фрагмент ключа раунду, а на праву – фрагмент. В інших раундах ці дії не виконуються.

– Ліва й права частини міняються місцями.

Для оптимального розв'язку завдання мультиплатформеності, таблиці S7 і S9 алгоритму MISTY1 можуть бути реалізовані як за допомогою обчислень, так і безпосередньо таблицями.

Розширення ключа

Для 8 раундів алгоритму результатом процедури розширення ключа буде наступний набір ключових значень:

- 20 фрагментів ключа (), кожний з яких має розмір по 16 бітів;
- 32 16-бітних фрагмента ();
- 24 7-бітних фрагмента (при $k=4$, тобто в 4-м раунді функції FO, операція FI не виконується);
- 24 9-бітних фрагмента.

Виконується дане обчислення в такий спосіб:

1. 128-бітний ключ ділиться на 8 фрагментів ... по 16 бітів кожний.
2. Формуються значення: у якості використовується результат обробки значення функцією FI, яка в якості ключа (тобто сукупності необхідних 7- і 9-бітних фрагментів) використовує значення (якщо індекс n фрагмента ключа перевищує 8, то замість нього використовується індекс $n-8$).

Необхідні фрагменти розширеного ключа «набираються» у міру виконання перетворень із відповідних масивів і згідно з відповідними таблицями 16-бітний фрагмент ділиться на 7-бітний фрагмент і 9-бітний .

Розшифрування

Розшифрування проводиться виконанням тих же операцій, що й при зашифруванні, але з наступними змінами:

– фрагменти розширеного ключа використовуються у зворотній послідовності,

– замість операції FL використовується зворотна їй операція – FLI.

Схеми виконання функції FLI і процедури розшифрування наведено на малюнках 6 і 7 відповідно:

Методи аналізу

Як говорилося на початку розділу, диференціальний і неможливий диференціальний аналізи виявилися ефективні лише до версій шифру з меншою кількістю раундів і без операції FL [2][3]. Проте, на даний момент цей напрямок аналізу, особливе використання слабких ключів, найбільше перспективно, тому що наближене до реальних можливих допущень при використанні алгоритму.

Так само, ученим з Японії був проведений інтегральний аналіз повного алгоритму, використовуючи відкритих текстів зі складністю обчислення, рівної [4].

Лінійний аналіз дав результати тільки для 7-раундової версії шифру, і також без операції FL[5].

Так як MISTY1 створювався, у тому числі, з розрахунку на апаратну реалізацію, має сенс диференціальний аналіз, заснований на використанні атаки по помилках обчислень, що в цьому випадку наближене до реальності.

Висновок

Таким чином, була докладно описана структура алгоритму шифрування MISTY1 і розглянуті методи його аналізу, найбільш прагматичні напрямки дослідження. Далі має бути створення програмної реалізації для більш детального розгляду алгоритму й набір статистичних даних для повного дослідження й пошуку оптимального підходу до аналізу MISTY1.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання магістерської дипломної роботи. Розрахунок проводиться через консоль, а результати передаються до графічного інтерфейсу. Розроблене програмне забезпечення системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі складається з наступних функціональних блоків:

- Навігаційне меню: Сеанс зв'язку; Шифрування; Параметри; Довідка.
- Підрозділи: IP-адреса; Користувачі; IP-івиклик користувача.
- Вікно виведення результату роботи системи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ.

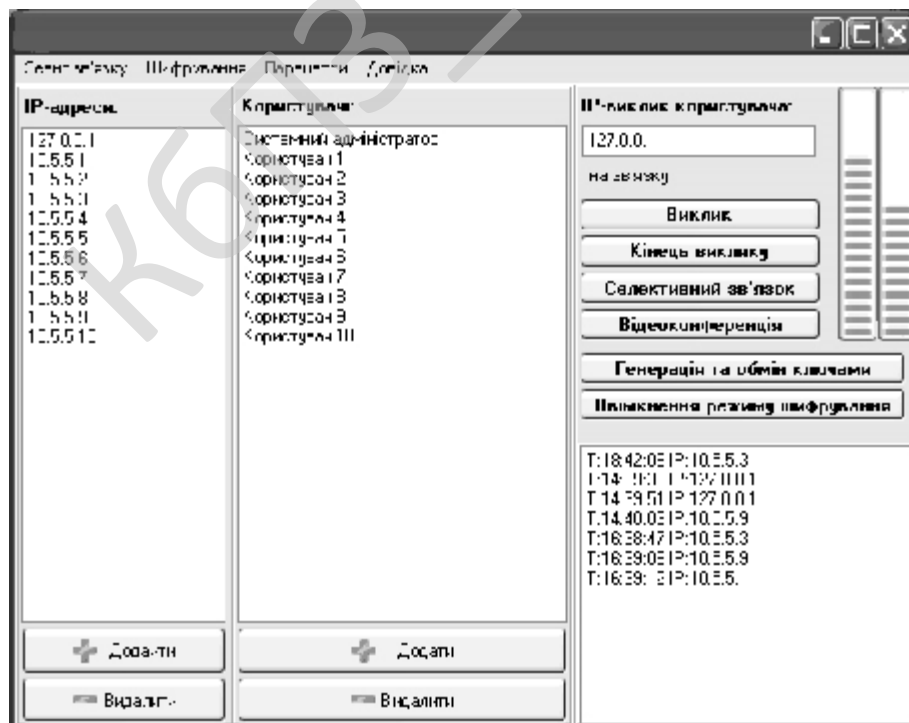


Рисунок 5.1 – Головне вікно розробленого ПЗ

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

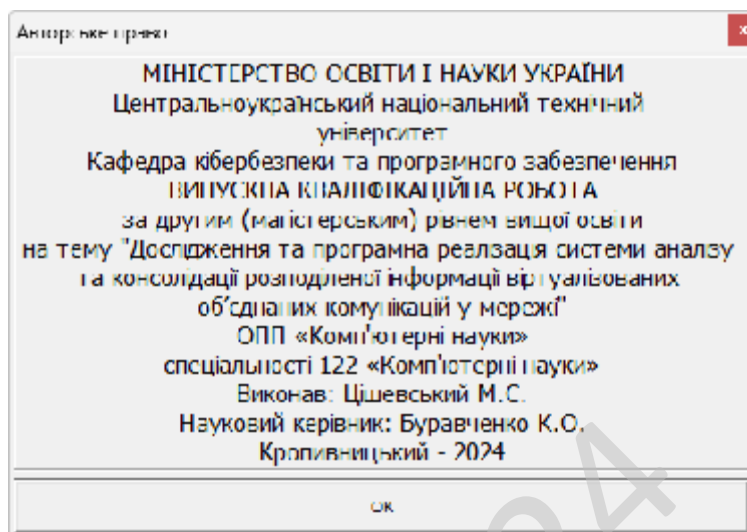


Рисунок 5.2 – Вікно розробника ПЗ

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

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

Проводилась оцінка:

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

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

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

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

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

У цьому випадку формуються тестові варіанти, в яких:

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

Недоліки тестування "білої скриньки":

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

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

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.
- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.
- При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

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

Обрано умови розповсюдження – Freeware.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку.

Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

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

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

КБПЗ-2024

					VKPM-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Метою розробки є дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Об'єктом дослідження є процес аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Предметом дослідження є методи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Методи дослідження базуються на методах аналізу та консолідації розподіленої інформації, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

– Розроблено вітчизняний продукт аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій можуть зацікавити досить широкий спектр категорій спеціалістів та організацій (рисунок 7.1).

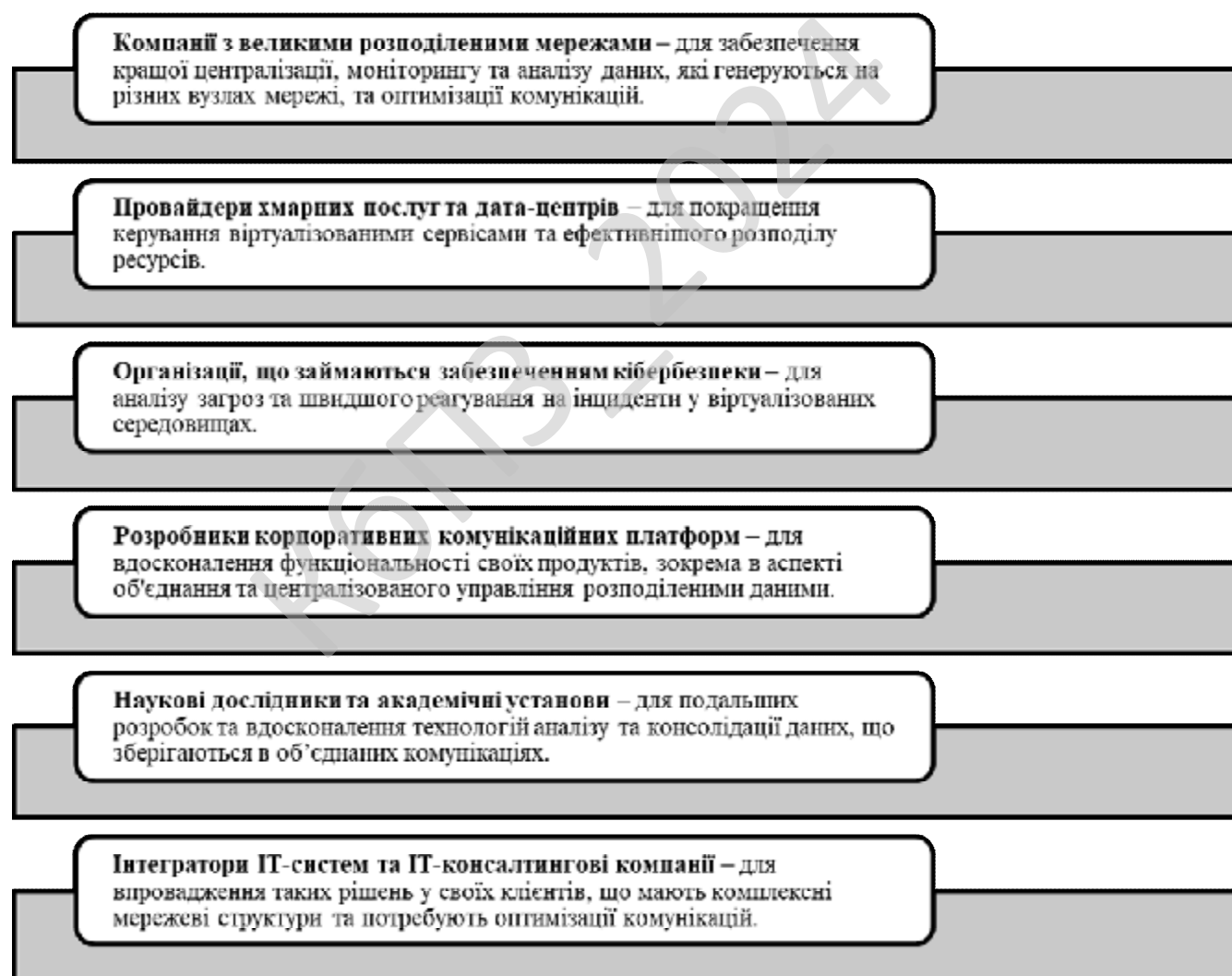


Рисунок 7.1 – Цільова аудиторія

Така система дозволить значно підвищити ефективність управління даними, комунікаціями та забезпечення безпеки в організаціях, які активно використовують віртуалізацію та об'єднані комунікації.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Для оцінки привабливості програмної реалізації системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій можна застосувати методи експертних оцінок, зокрема метод середніх оцінок. Розглянемо застосування методу середніх оцінок.

Перший крок – вибір експертів, які мають досвід у сферах віртуалізації, мережевих комунікацій та аналітики даних. Для цього розрахунку запросимо п'ять експертів. Після цього разом з експертами визначаються критерії оцінки, які впливають на привабливість системи (рисунок. 7.2).

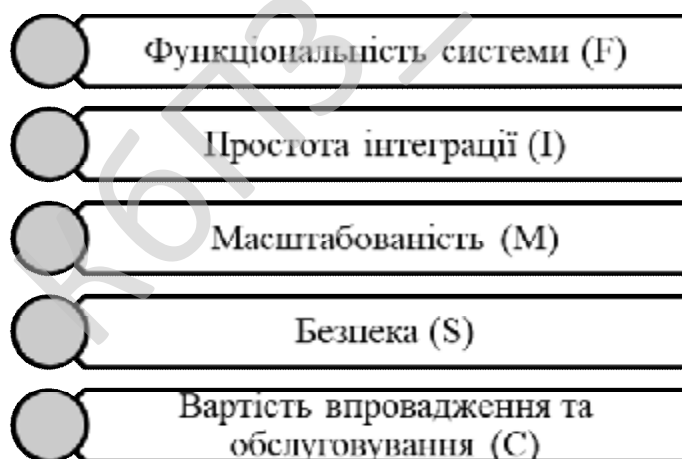


Рисунок 7.2 – Критерії оцінки

Експерти оцінюють кожен критерій за шкалою від 1 до 10, де 1 – низька привабливість, а 10 – висока привабливість. Результати зводимо в таблицю 7.1.

Таблиця 7.1 – Зведені результати експертних оцінок

Критерій	Експерт 1	Експерт 2	Експерт 3	Експерт 4	Експерт 5	Середня оцінка
F (Функціональність)	9	8	10	9	8	8.8
I (Простота інтеграції)	7	8	8	7	9	7.8
M (Масштабованість)	10	9	9	10	9	9.4
S (Безпека)	8	9	10	9	8	8.8
C (Вартість)	6	7	6	7	5	6.2

З урахуванням вагових коефіцієнтів, визначених на основі значимості критеріїв (функціональність – 0.3, масштабованість – 0.25, безпека – 0.2, простота інтеграції – 0.15, вартість – 0.1), обчислюємо інтегральну оцінку.

$$I=(8.8 \times 0.3)+(7.8 \times 0.15)+(9.4 \times 0.25)+(8.8 \times 0.2)+(6.2 \times 0.1)=8.61.$$

Інтегральна оцінка 8.61 вказує на високу привабливість системи з урахуванням встановлених критеріїв. За необхідності, компанія може провести додатковий аналіз чутливості результатів залежно від змін вагових коефіцієнтів, щоб врахувати специфічні умови або пріоритети при прийнятті остаточного рішення щодо реалізації системи.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації такої системи варто використати метод аналогії. Цей метод базується на оцінці вартості схожих реалізованих проєктів, що мають подібну архітектуру або масштаб. Якщо в організації вже реалізовувались подібні проєкти для віртуалізованих мереж, метод аналогії може стати зручним способом оцінки.

						ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			75

Ключові етапи проведення оцінки: аналіз схожих проєктів (визначаються проєкти з подібним обсягом роботи, технологіями та інфраструктурою), корекція оцінки (враховуються можливі відмінності, як-от розширення функціональності чи складність інтеграції з іншими системами), оцінка з урахуванням додаткових витрат (додаються коригуючі коефіцієнти для врахування ризиків або специфічних вимог).

Перевагами цього методу є швидкість (оскільки метод базується на даних попередніх проєктів, це дає можливість швидко отримати приблизну оцінку) та економічність (відсутність потреби в детальних розрахунках трудовитрат знижує витрати на підготовку оцінки).

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

Економічна ефективність від впровадження системи аналізу та консолідації розподіленої інформації у віртуалізованих об'єднаних комунікаціях може бути значною, і її можна оцінити через показники скорочення витрат, збільшення продуктивності, а також поліпшення ефективності роботи мережі (рисунок 7.3). Таким чином, впровадження системи принесе економію підприємству та підвищення ефективності роботи.

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Для успішного просування проєкту з програмної реалізації системи аналізу та консолідації розподіленої інформації у віртуалізованих об'єднаних комунікаціях можна розробити покроковий алгоритм, який охоплює аналіз ринку, створення УТП (унікальної торгової пропозиції), підготовку маркетингових матеріалів та стратегій, залучення потенційних клієнтів і зворотний зв'язок (рисунок 7.4).

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Зниження витрат на управління мережею

Завдяки централізованому аналізу та консолідації даних, зменшуються витрати на адміністрування, оскільки система дозволяє автоматично відслідковувати стан вузлів, виявляти проблеми та формувати звіти.

Зниження витрат на зберігання даних

Консолідація інформації дозволяє оптимізувати зберігання, видаляючи дублікати і неактуальні дані, що знижує витрати на сховище.

Зменшення простоїв завдяки моніторингу та швидкому реагуванню

Система аналізу швидше ідентифікує проблеми, що дозволяє скоротити простої на 30%.

Зниження витрат на безпеку

Завдяки аналітиці загроз і централізованому контролю зменшується кількість інцидентів безпеки.

Підвищення продуктивності співробітників

Автоматизація та консолідація даних дають співробітникам доступ до актуальної інформації швидше, що підвищує їхню продуктивність на 10%.

Рисунок 7.3 – Економічна ефективність від реалізації проєкту для клієнта

Аналіз ринку та конкурентів	<ul style="list-style-type: none"> дослідження потреб ринку та визначення основних конкурентів. виявлення прогалин у продуктах конкурентів, які можна заповнити в межах вашого проекту. визначення цільових сегментів: компанії з великими мережами, провайдери хмарних послуг, компанії в галузі кібербезпеки.
Розробка унікальної торгової пропозиції (УТП)	<ul style="list-style-type: none"> сформулюйте ключові переваги системи: централізований моніторинг, безпека, зниження витрат, масштабованість. створіть УТП, яка виділить вашу систему серед конкурентів, наприклад, акцент на покращену швидкість аналізу даних і зниження простоя на 30%.
Розробка маркетингових матеріалів	<ul style="list-style-type: none"> підготуйте технічні презентації, аналітичні звіти про ефективність системи, кейс-стаді з розрахунками економічної вигоди. розробіть обмежену версію системи для демонстрації її можливостей. створіть відео з оглядом функціонування, а також навчальні вебінари для потенційних користувачів.
Визначення каналів просування	<ul style="list-style-type: none"> таргетована реклама в соціальних мережах (LinkedIn, Facebook) та Google Ads. дублікація статей на платформах типу Medium, CIO.com, а також на форумах і в профільних групах для обміну досвідом. представляйте продукт на профільних конференціях, таких як Cisco Live, Gartner IT Symposium.
Просування через партнерства та колаборації	<ul style="list-style-type: none"> співпраця з компаніями, які займаються кібербезпекою або надають хмарні послуги, для обміну потенційними клієнтами. партнерство з інтеграторами ІТ-систем та ІТ-консалтинговими компаніями для просування системи у клієнтів, які потребують оптимізації мережних рішень.
Залучення перших клієнтів та проведення пілотних проєктів	<ul style="list-style-type: none"> виберіть кілька компаній для проведення пілотних впроваджень, що дозволить продемонструвати реальні результати. проведіть об'ємні впровадження як уживаючи ціною або на умовах пробного використання, щоб клієнти могли оцінити переваги системи.
Збір зворотного зв'язку та оптимізація системи	<ul style="list-style-type: none"> отримуйте зворотний зв'язок від користувачів пілоотної версії для вдосконалення функціоналу системи. додавайте нові функції та налаштування відповідно до вимог користувачів, щоб підвищити цінність продукту.
Розширення клієнтської бази та масштабування продажів	<ul style="list-style-type: none"> використовуйте успішні кейси пілотних проєктів для створення портфоліо клієнтів. масштабуйте продажі через розширення маркетингових кампаній, залучення нових партнерів і просування позитивного досвіду використання.
Оцінка ефективності стратегії просування та внесення корективів	<ul style="list-style-type: none"> регулярно оновлюйте КРІ (наприклад, кількість залучених клієнтів, кількість проведених демо-версій, відсоток конверсії з потенційних клієнтів). оптимізуйте канали просування та методи взаємодії на основі аналізу показників.

Рисунок 7.4 – Алгоритм просування проєкту

Цей алгоритм дозволяє поступово просувати проєкт, адаптуючи його відповідно до відгуків ринку, та формувати лояльну клієнтську базу для стабільного розвитку.

1. Оптимізація каналів збуту

- **Прямі продажі через спеціалізовані команди**
залучіть команду спеціалістів з продажів, які зможуть фокусуватися на особистій взаємодії з великими підприємствами або корпораціями, що мають складні мережеві інфраструктури. Такий підхід дозволяє побудувати довгострокові відносини з клієнтами і надати їм індивідуальні рішення.
- **Партнерські програми та співпраця з інтеграторами**
важливо налагодити партнерства з компаніями-інтеграторами ІТ-систем, провайдерами хмарних послуг, та постачальниками мережевого обладнання, що вже працюють з цільовими клієнтами. Інтегратори можуть включати вашу систему до своїх рішень як додаткову послугу для своїх клієнтів, що полегшує доступ до нових ринків.
- **Участь у спеціалізованих конференціях та виставках**
присутність на конференціях і виставках, таких як Gartner IT Symposium або Cisco Live, дозволить показати систему безпосередньо потенційним клієнтам та продемонструвати її переваги. Ви можете також запустити демо-версії на місці для залучення інтересу та поширення матеріалів серед цільової аудиторії.
- **Онлайн-канали: вебінари, блог-пости та відео-контент**
використовуйте вебінари та відео-контент для роз'яснення роботи системи, її можливостей і результатів впровадження. Також варто створити блог, де публікуватимуться аналітичні статті про використання вашої системи в різних галузях.
- **Сегментована реклама та цифровий маркетинг**
орієнтуйте рекламні кампанії на вузькі сегменти, що відповідають вашій аудиторії (наприклад, менеджери з ІТ, фахівці з мережевої інфраструктури), та використовуйте Google Ads, LinkedIn та Facebook для таргетованих оголошень.

2. Оптимізація шляхів реалізації проєкту

- **Запровадження гнучкої системи ціноутворення**
розробіть модель, що дозволить адаптувати систему під різні масштаби бізнесу. Наприклад, можна пропонувати ліцензію з оплатою за користування (SaaS-модель) для малих компаній та корпоративну ліцензію для великих замовників. Такий підхід дозволить охопити ширше коло потенційних клієнтів.
- **Пілотні проєкти та тестові періоди**
пропонуйте компаніям тестовий період використання вашої системи, щоб вони могли випробувати її функціонал без зобов'язань. Деякі клієнти можуть бути більш схильні до покупки після успішного тестування та оцінки її впливу на ефективність.
- **Вдосконалення інтеграційних можливостей**
оскільки система буде працювати в існуючій інфраструктурі компаній, забезпечте її максимальну інтеграцію з популярними інструментами управління мережами, платформами для моніторингу або аналізу даних. Це дозволить спростити впровадження та забезпечити зручне використання для компаній з розподіленими інформаційними системами.
- **Автоматизація процесів впровадження та технічна підтримка**
запровадьте автоматизовані інструменти для налаштування та інтеграції системи з інфраструктурою замовника. Додайте можливості дистанційного моніторингу та обслуговування, що зменшить час на впровадження і дозволить підтримувати систему без значних додаткових ресурсів.
- **Розробка кейсів та документування успішних реалізацій**
зберіть дані з впровадження системи в різних організаціях, створіть детальні кейс-стаді з розрахунками економічного ефекту та підвищення ефективності. Це дозволить більш наочно продемонструвати вигоди системи потенційним клієнтам і зміцнити репутацію.

3. Залучення зворотного зв'язку для оптимізації подальших продажів

- збирайте відгуки та рекомендації від перших клієнтів для вдосконалення функціоналу.
- використовуйте зворотний зв'язок для коригування УТП і маркетингової стратегії, щоб зробити продукт ще привабливішим для потенційних клієнтів.
- впроваджуйте зміни в систему на основі зворотного зв'язку, що підвищать її адаптивність та відповідність потребам ринку.

Рисунок 7.5 – Оптимізація каналів збуту та шляхів реалізації

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту та шляхів реалізації проекту програмної системи аналізу та консолідації розподіленої інформації у віртуалізованих об'єднаних комунікаціях важливо сфокусуватися на найефективніших каналах та підходах для залучення цільової аудиторії (рисунок 7.5).

Ці кроки допоможуть побудувати стратегію для масштабування проекту та забезпечення його довготривалого успіху, адаптуючись до потреб цільової аудиторії й покращуючи конкурентоспроможність системи.

7.7 Визначення ключових факторів успіху конкретного проекту

Для успіху проекту з розробки та впровадження системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій важливо звернути увагу на кілька ключових факторів (на рис. 7.6).

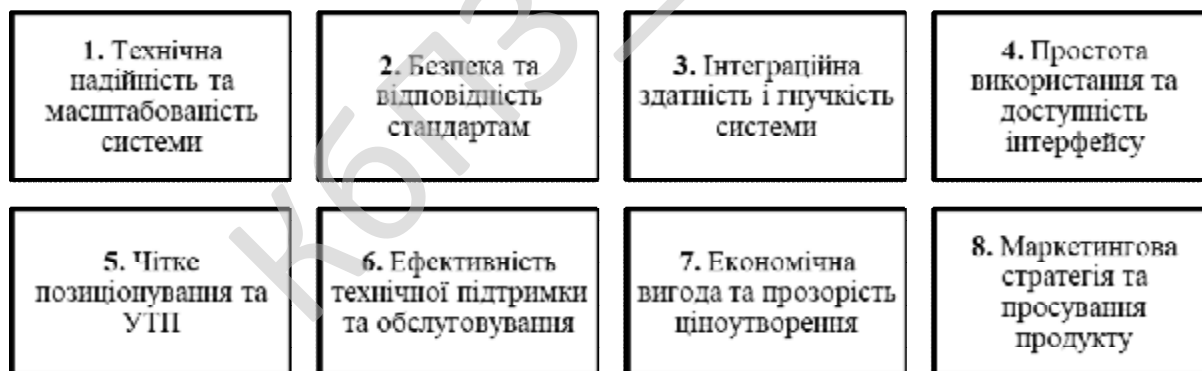


Рисунок 7.6 – Ключові фактори успіху проекту

Ці фактори створюють базу для успіху та стійкого розвитку продукту, дозволяючи не лише реалізувати, але й ефективно просувати та підтримувати систему, підвищуючи її цінність для користувачів.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Найявний в даний час в нашій країні комплекс розроблених організаційних заходів та технічних засобів захисту, накопичений передовий досвід роботи ряду обчислювальних центрів показує, що є можливість домогтися значно більших успіхів у справі усунення впливу на працюючих небезпечних і шкідливих виробничих факторів. Проте стан умов праці та його безпеки в ряді обчислювальних центрів (ОЦ) та підприємств ще не задовольняють сучасним вимогам. Оператори ЕОМ, оператори підготовці даних, програмісти та інші працівники ОЦ та підприємств ще стикаються з впливом таких фізично небезпечних і шкідливих виробничих факторів, як підвищений рівень шуму, підвищена температура зовнішнього середовища, відсутність або недостатня освітленість робочої зони, електричний струм, статична електрика і інші.

Багато працівників ОЦ та підприємств пов'язані з впливом таких психофізичних факторів, як розумова перенапруга, перенапруження зорових і слухових аналізаторів, монотонність праці, емоційні перевантаження. Вплив зазначених несприятливих факторів призводить до зниження працездатності, викликане розвиваються втому. Поява і розвиток втоми пов'язане зі змінами, які виникають під час роботи в центральній нервовій системі, з гальмівними процесами в корі головного мозку. Наприклад сильний шум викликає труднощі з розпізнаванням колірних сигналів, знижує швидкість сприйняття кольору, гостроту зору, зорову адаптацію, порушує сприйняття візуальної інформації, зменшує на 5 – 12% продуктивність праці. Тривала дія шуму з рівнем звукового тиску 90 дБ знижує продуктивність праці на 30 – 60%.

Медичні обстеження працівників ОЦ та підприємств показали, що крім зниження продуктивності праці високі рівні шуму призводять до погіршення

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

слуху. Тривале перебування людини в зоні комбінованого впливу різних несприятливих факторів може призвести до професійного захворювання. Аналіз травматизму серед працівників ВЦ показує, що в основному нещасні випадки відбуваються від впливу фізично небезпечних виробничих факторів при заправці носія інформації на обертний барабан при знятому кожусі, під час співробітниками невластивих їм робіт. На другому місці випадки, пов'язані з дією електричного струму.

8.2 Аналіз умов праці на робочому місці ІТ-фахівця

На робочому місці ІТ-фахівця (або програміста) виникають небезпечні та шкідливі для безпечної життєдіяльності фактори:

- підвищений рівень шуму;
- несприятливі мікрокліматичні умови;
- недостатній рівень освітленості;
- шкідливі речовини;
- підвищений рівень електромагнітних випромінювань радіочастот;
- висока напруга електричної мережі;
- статична електрика та інші.

Робота програміста супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, вивчення умов праці на робочому місці програміста є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів. Робоче місце, добре пристосоване до трудової діяльності інженера, правильно і доцільно організоване, щодо простору, форми, розміру забезпечує йому зручне положення при роботі і високу продуктивність праці при найменшому фізичному

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

і психічному напруженні.

Нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Для постійних робочих місць, якими є робочі місця ІТ-фахівців, встановлені оптимальні параметри мікроклімату, а за неможливості їх дотримання використовують допустимі параметри. Робота ІТ-фахівця за важкістю відноситься до Іа (роботи, що виконуються сидячи і не потребують фізичного напруження) та Іб (роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням) категорій. В таблиці 8.1. наведені оптимальні параметри мікроклімату в приміщеннях.

Таблиця 8.1 – Параметри мікроклімату для приміщень з ПК

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

Виміряні за допомогою приладів температура та вологість у приміщеннях праці ІТ-фахівців повинні відповідати зазначеним у таблиці для теплового періоду року. Слід зазначити, що для нормалізації параметрів мікроклімату слід використовувати у приміщеннях кондиціонування повітря, або забезпечити подачу свіжого повітря системами вентиляції. Норми подачі свіжого повітря наведені у таблиці 8.2.

Таблиця 8.2 – Норми подачі свіжого повітря в приміщення

Характеристика приміщення	Об'ємна витрата свіжого повітря, що подається в приміщення, м ³ на одну людину в годину
Об'єм до 20 м ³ на людину	Не менше 30
20... 40 м ³ на людину	Не менше 20
Більше 40 м ³ на людину	Може біти використана природна вентиляція

Створення сприятливих умов праці і правильне естетичне оформлення робочих місць на виробництві має велике значення як для полегшення праці, так і для підвищення його привабливості, позитивно впливає на продуктивність праці. Забарвлення приміщень і меблів повинні сприяти створенню сприятливих умов для зорового сприйняття, гарного настрою. У службових приміщеннях, у яких виконується одноманітна розумова робота, що вимагає значної нервової напруги і великого зосередження, забарвлення повинно бути спокійних тонів – малонасичені відтінки холодного зеленого або блакитного кольорів.

При розробці оптимальних умов праці програміста необхідно враховувати освітленість. Раціональне освітлення робочого місця є одним з найважливіших факторів, що впливають на ефективність трудової діяльності людини, що попереджають травматизм і професійні захворювання. Правильно організоване освітлення створює сприятливі умови праці, підвищує працездатність і продуктивність праці. Освітлення на робочому місці програміста повинно бути таким, щоб працівник міг без напруги зору виконувати свою роботу. Стомлюваність органів зору залежить від ряду причин: недостатність освітленості; надмірна освітленість; неправильний напрям світла. Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці

може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань. [2]

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців ІТ-індустрії. Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців ІТ-індустрії і стандартів підприємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів. Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців ІТ-індустрії. Всі наведені заходи щодо вдосконалення охорони праці фахівців ІТ-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці підприємства. Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

Для більшого розуміння, пропозиції щодо підвищення працездатності ІТ-фахівців, розіб'ємо на декілька категорій:

1. Середовище і розпорядок праці. Для мінімізації негативних ефектів, що пов'язані з перевтомленням ІТ-фахівців, потрібно чітко прописати і реалізувати графік періодів праці-відпочинку, щоб фахівець міг можливість переключити увагу, дати можливість відпочити очам, мозку, елементарно, встати розім'яти ноги. Також потрібно зробити максимально комфортними умови мікроклімату у офісному приміщенні, де працюють ІТ-фахівці. Мається на увазі встановлення і експлуатація, коли виникає необхідність, кондиціонерів, опалення, та системи вентиляції, задля попередження перегрівання, переохолодження ІТ-фахівців, і подальшої неможливості ними виконувати свої функції. Також, за можливості, нами пропонується введення практики віддаленої праці ІТ-фахівцями, якщо

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

роботодавець не може забезпечити оптимальні і безпечні умови в офісному приміщенні, або якщо фахівця вони не влаштовують із певних причин.

2. Фізичні і психоемоційні чинники. Першим і найважливішим чинником, що впливає на працездатність ІТ-фахівців є робоче місце, і саме тому, роботодавець має забезпечити максимальний його комфорт і безпеку. Гарантією цих факторів може слугувати сертифікація меблів, що використовуються на підприємстві ІТ-галузі. Тому нами пропонується закупівля тільки меблів, які пошли сертифікацію на відповідність. Під психоемоційними чинниками ми розуміємо гарне самопочуття фахівців, позитивний настрій, гарний психологічний клімат у колективі, тощо. Задля того, щоб психоемоційні чинники мали максимально позитивний ефект, керівництву слід поводити заходи, які сприятимуть укріпленню і покращенню міжособистісних стосунків у колективі, таких як психологічні тренінги, тимблдінг, спортивні змагання і естафети. Також, сюди можна віднести розробку і впровадження системи мотивації працівників, як фінансової, так моральної і адміністративної.

8.3 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень.

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга) [9].

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

8.4 Розрахункова частина

Система освітлення робочого місця користувача ПК має відповідати наступним вимогам (рис. 8.1).

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 3 м, довжина – 4.4 м, висота – 3 м.

У зазначеному приміщенні працює 2 людей.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F = E \cdot S \cdot K \cdot Z / n,$$

де: F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк, $E = 300$ Лк;

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

S – площа освітлюваного приміщення.

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку $K = 1,5$);

Z – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку $Z = 1,1$);

n – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{\text{стін}}$) і стелі ($\rho_{\text{стелі}}$), значення коефіцієнтів дорівнюють:

$$\rho_{\text{стін}} = 50\% \text{ і } \rho_{\text{стелі}} = 50\%.$$

Обчислимо індекс приміщення за формулою:

$$i = S / (h \cdot (A + B)),$$

де: S – площа приміщення, $S = 13,2 \text{ м}^2$;

h – розрахункова висота підвісу, $h = 3 \text{ м}$. (співпадає з висотою стелі, оскільки лампи освітлення закріплюються на стелі);

A – ширина приміщення, $A = 3 \text{ м}$;

B – довжина приміщення, $B = 4,4 \text{ м}$.

Підставимо всі значення у формулу та визначимо індекс приміщення: $i = 1,1$.

Знаючи індекс приміщення, за знаходимо $n = 0,46$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом ламп) [8]. Підставимо всі значення у формулу, визначимо світловий потік: $F = 14204 \text{ Лм}$.

Для розрахунку дудемо використовувати стельові світлодіодні панелі Призма-72 6400К, світловий потік яких $F_{\text{л}} = 7200 \text{ Лм}$.

Число ламп визначається по формулі:

$$N = F / F_{\text{л}}$$

де: F – світловий потік, $F_{\text{л}}$ – світловий потік однієї лампи.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88



Рисунок 8.1 – Вимоги до системи освітлення робочого місця користувача ПК

Підставимо всі значення у формулу та визначемо індекса приміщення:

$$N = 14204 / 7200 = 1,9 \text{ шт.}$$

Приймаємо необхідну кількість *світлодіодних світильників* 2 шт.

Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз умов праці, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

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 аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.
- Досліджена система аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.
- На основі отриманих результатів досліджень створена програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

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

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

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 MISTY1.

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

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

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Цішевський М.С. Дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Cisco Voice Over IP. Student Guide – Cisco Systems Inc, 2003.
3. Cisco IP Telephony. Student Guide. Version 3.3. – Cisco Systems Inc, 2002.
4. Robert Padjen – Cisco AVVID and IP Telephony. Design & Implementation – Syngress Publishing Inc, 2001.
5. Paul J. Fong – Configuring Cisco Voice Over IP. Second Edition – Syngress Publishing Inc, 2001.
6. ITU-T Recommendation G.723.1. Dual Rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kbit / sec. – 1996.
7. ITU-T Recommendation G.729. Speech codec for multimedia telecommunications transmitting at 8 / 13 kbit / s. – 1996.
8. ITU-T Recommendation H.225.0. Call signaling protocols and media stream packetization for packet-based multimedia communication systems. -Geneva, 1998.
9. ITU-T Recommendation H.245. Control protocol for multimedia communication. -Geneva, 1998.
10. ITU-T Recommendation H.248. Gateway control protocol. – Geneva, 2000.
11. ITU-T Recommendation H.320. Narrow-band Visual Telephone Systems and Terminal Equipment. – 1996.
12. ITU-T Recommendation H.321. Adaptation of H.320 Visual Telephone Terminals to B-ISDN Environments. – 1996.
13. ITU-T Recommendation H.322. Visual Telephone Systems and Terminal

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

Equipment for Local Area Networks which Provide a Guaranteed Quality of Service. – 1996.

14. ITU-T Recommendation H.323. Packet based multimedia communication systems. – Geneva, 1998.

15. ITU-T Recommendation H.324. Terminal for Low Bit Rate Multimedia Communications. -1996.

16. ITU-T Recommendation Q.931. ISDN User-Network Interface Layer 3 Specification for Basic Call Control. – 1993.

17. RFC 2705. Media Gateway Control Protocol (MGCP) Version 1.0. M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett. October 1999.

18. RFC 2865. Remote Authentication Dial In User Service (RADIUS). C.Rigney, S. Willens, A. Rubens, W. Simpson. June 2000.

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

20. 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.

21. Smirnov, O., Neskrodieva, T., Fedorov, E., Rudakov, K., Neskrodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

22. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland)* Volume 22, Issue 16, 6223, 2022.

23. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». *SN Computer Science*, Vol 2, 337, 2021.

<https://doi.org/10.1007/s42979-021-00739-w>

					BKPM-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

24. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

25. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

26. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

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

28. 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.

29. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

30. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

31. 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.

32. 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.

33. 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.

34. 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.

35. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.

36. Zhurakovskiy, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.

37. 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.

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. 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.

45. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated

					БКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

46. 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.

47. 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.

48. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

49. 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.

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

51. 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.

					ВКРМ-122.24.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.24.0016.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Цішевський М.С.				<i>Дослідження та програмна реалізація системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі</i>	Літ.	Аркуш	Аркушів
Перевірів	Буравченко К.О.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-23М			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 18-13 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 аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі.

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;
- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;

					ВКРМ-122.24.0016.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи аналізу та консолідації розподіленої інформації віртуалізованих об'єднаних комунікацій у мережі;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-122.24.0016.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Python.

					ВКРМ-122.24.0016.00.00.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 повинен бути розглянутий аналіз умов праці на робочому місці ІТ-фахівця.

					ВКРМ-122.24.0016.00.00.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 – 97 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

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

					ВКРМ-122.24.0016.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти
_____ Буравченко К.О.

*Дослідження та програмна реалізація
системи аналізу та консолідації розподіленої інформації віртуалізованих
об'єднаних комунікацій у мережі*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 18

Літера: РП

Кропивницький – 2024 року

Основна програма

```

import threading
import queue
import time
import random
import logging

# Ініціалізація черги для зберігання розподілених повідомлень
message_queue = queue.Queue()

# Налаштування журналу для запису подій у систему
logging.basicConfig(filename="system.log", level=logging.INFO)

# Модуль симуляції отримання розподіленої інформації
def data_acquisition(node_id, data_type):
    data_types = {"sensor": "Sensor Data", "log": "Log Data", "control": "Control Signal"}
    while True:
        data = f"Node {node_id} {data_types[data_type]}: {random.randint(0, 100)}"
        message_queue.put(data)
        logging.info(f"Data from Node {node_id}: {data}")
        time.sleep(random.uniform(0.5, 2))

# Модуль аналізу отриманої інформації
def data_analysis():
    analyzed_data = []
    while True:
        if not message_queue.empty():
            data = message_queue.get()
            analyzed_result = analyze_data(data)
            analyzed_data.append(analyzed_result)
            print(analyzed_result)
            logging.info(f"Analyzed data: {analyzed_result}")
            time.sleep(1)
            if len(analyzed_data) >= 10:
                avg_value = calculate_average(analyzed_data)
                print(f"Average value: {avg_value}")
                logging.info(f"Average value calculated: {avg_value}")
                analyzed_data.clear()

# Функція аналізу даних
def analyze_data(data):
    if "Sensor" in data:
        return f"Processed {data} - Valid Sensor Reading"
    elif "Log" in data:
        return f"Processed {data} - System Log"
    elif "Control" in data:
        return f"Processed {data} - Control Command"
    else:
        return f"Processed {data} - Unknown Type"

# Функція для обчислення середнього значення
def calculate_average(analyzed_data):
    values = [int(item.split(":")[-1]) for item in analyzed_data if "Processed" in item]
    return sum(values) / len(values) if values else 0

# Модуль симуляції розподілу навантаження
def load_balancer():
    while True:
        # Розподіл навантаження між вузлами
        nodes = [f"Node-{i}" for i in range(1, 6)]
        chosen_node = random.choice(nodes)
        print(f"Load directed to {chosen_node}")
        logging.info(f"Load directed to {chosen_node}")
        time.sleep(2)

```

```

# Модуль моніторингу ресурсів
def resource_monitor():
    while True:
        cpu_usage = random.randint(10, 90)
        memory_usage = random.randint(100, 500)
        print(f"CPU usage: {cpu_usage}%, Memory usage: {memory_usage}MB")
        logging.info(f"Resource usage - CPU: {cpu_usage}%, Memory: {memory_usage}MB")
        time.sleep(3)

# Модуль для керування запитами на контроль
def control_requests():
    while True:
        request_type = random.choice(["Start", "Stop", "Pause"])
        print(f"Control request received: {request_type}")
        logging.info(f"Control request: {request_type}")
        time.sleep(4)

# Модуль для взаємодії з іншими системами
def system_integration():
    while True:
        external_system = random.choice(["System A", "System B", "System C"])
        print(f"Integration with {external_system} in progress")
        logging.info(f"Integration with {external_system} initiated")
        time.sleep(5)

# Модуль резервування даних
def data_backup():
    while True:
        print("Data backup in progress")
        logging.info("Data backup started")
        time.sleep(10)

# Модуль керування потоками для розподіленої обробки
def main_system():
    nodes = []
    for node_id in range(1, 6):
        for data_type in ["sensor", "log", "control"]:
            node_thread = threading.Thread(target=data_acquisition, args=(node_id,
            data_type))
            node_thread.start()
            nodes.append(node_thread)
        analysis_thread = threading.Thread(target=data_analysis)
        analysis_thread.start()
        load_balancer_thread = threading.Thread(target=load_balancer)
        load_balancer_thread.start()
        monitor_thread = threading.Thread(target=resource_monitor)
        monitor_thread.start()
        control_thread = threading.Thread(target=control_requests)
        control_thread.start()
        integration_thread = threading.Thread(target=system_integration)
        integration_thread.start()
        backup_thread = threading.Thread(target=data_backup)
        backup_thread.start()
    for node_thread in nodes:
        node_thread.join()
    analysis_thread.join()
    load_balancer_thread.join()
    monitor_thread.join()
    control_thread.join()
    integration_thread.join()
    backup_thread.join()

# Запуск системи
if __name__ == "__main__":
    main_system()

```

Файл encryption.py

```
from cryptography.fernet import Fernet

# Генерація ключа шифрування
def generate_key():
    return Fernet.generate_key()

# Функція для шифрування повідомлення
def encrypt_message(key, message):
    cipher_suite = Fernet(key)
    encrypted_message = cipher_suite.encrypt(message.encode())
    return encrypted_message

# Функція для розшифрування повідомлення
def decrypt_message(key, encrypted_message):
    cipher_suite = Fernet(key)
    decrypted_message = cipher_suite.decrypt(encrypted_message).decode()
    return decrypted_message

# Приклад використання
if __name__ == "__main__":
    key = generate_key()
    message = "Test message for encryption"
    encrypted = encrypt_message(key, message)
    decrypted = decrypt_message(key, encrypted)

    print(f"Original message: {message}")
    print(f"Encrypted message: {encrypted}")
    print(f"Decrypted message: {decrypted}")
```

Файл auth.py

```
import hashlib
import os

# Імітація бази даних користувачів
user_db = {
    "admin": hashlib.sha256("admin_pass".encode()).hexdigest(),
    "user": hashlib.sha256("user_pass".encode()).hexdigest()
}

# Функція для аутентифікації користувача
def authenticate(username, password):
    hashed_password = hashlib.sha256(password.encode()).hexdigest()
    if username in user_db and user_db[username] == hashed_password:
        return True
    return False

# Функція для авторизації доступу до ресурсів
def authorize(username, resource):
    if username == "admin":
        return True
    if username == "user" and resource != "admin_panel":
        return True
    return False

# Приклад використання
if __name__ == "__main__":
    username = "user"
    password = "user_pass"

    if authenticate(username, password):
        print(f"{username} authenticated")
        if authorize(username, "dashboard"):
            print(f"{username} authorized to access dashboard")
        else:
            print(f"{username} not authorized to access dashboard")
    else:
        print(f"Authentication failed for {username}")
```

Файл data_transfer.py

```
import requests
import random
import json

# Модуль для передачі даних за різними протоколами
def send_data_http(url, data):
    headers = {'Content-Type': 'application/json'}
    response = requests.post(url, data=json.dumps(data), headers=headers)
    return response.status_code

def send_data_mqtt(client, topic, data):
    client.publish(topic, json.dumps(data))
    print(f>Data sent to MQTT topic {topic}")

# Прогнозування на основі поточних даних
def predict_future(data):
    # Простий приклад прогнозування (наприклад, усереднене значення)
    average = sum(data) / len(data)
    future_value = average * random.uniform(0.9, 1.1) # Прогноз з певним
    відхиленням
    return future_value

# Приклад використання
if __name__ == "__main__":
    data = [random.randint(10, 100) for _ in range(10)]
    future_value = predict_future(data)
    print(f>Predicted future value: {future_value}")

# Приклад передачі даних
url = "http://example.com/api/data"
data_to_send = {"sensor_value": random.randint(10, 100)}
status = send_data_http(url, data_to_send)
print(f>HTTP request status: {status}")
```

Файл `big_data_processing.py`

```
from pyspark import SparkContext, SparkConf

# Налаштування для обробки великих даних
conf = SparkConf().setAppName("BigDataProcessing").setMaster("local")
sc = SparkContext(conf=conf)

# Функція для обробки великих масивів даних
def process_large_data(data):
    rdd = sc.parallelize(data)
    result = rdd.map(lambda x: (x, x ** 2)).collect() # Простий приклад обробки
    return result

# Приклад використання
if __name__ == "__main__":
    large_data = [i for i in range(1000)]
    processed_data = process_large_data(large_data)
    print(f"Processed large data: {processed_data[:10]}") # Виведемо лише перші
10 результатів
```

КБПЗ_2024

Файл monitoring.py

```
import time
import psutil

# Модуль для моніторингу використання ресурсів системи в реальному часі
def monitor_resources():
    while True:
        cpu_usage = psutil.cpu_percent(interval=1)
        memory_info = psutil.virtual_memory()
        disk_usage = psutil.disk_usage('/')

        print(f"CPU Usage: {cpu_usage}%")
        print(f"Memory Usage: {memory_info.percent}%")
        print(f"Disk Usage: {disk_usage.percent}%")

        time.sleep(5)

# Приклад використання
if __name__ == "__main__":
    monitor_resources()
```

КБПЗ_2024

Файл logging_system.py

```
import logging

# Налаштування журналу з різними рівнями логування
logging.basicConfig(
    filename="detailed_system.log",
    format='%asctime)s - %(levelname)s - %(message)s',
    level=logging.DEBUG
)

# Приклади логування
def log_events():
    logging.debug("This is a debug message")
    logging.info("This is an info message")
    logging.warning("This is a warning message")
    logging.error("This is an error message")
    logging.critical("This is a critical message")

# Приклад використання
if __name__ == "__main__":
    log_events()
    print("Log messages have been written to the file")
```

КБПЗ_2024

Файл auto_recovery.py

```
import threading
import time
import random
# Модуль для симуляції стану вузлів
nodes_status = {f"Node-{i}": "healthy" for i in range(1, 6)}
# Функція для моніторингу та автоматичного відновлення вузлів
def monitor_and_recover_nodes():
    while True:
        for node, status in nodes_status.items():
            if status == "unhealthy":
                print(f"{node} is down. Attempting recovery...")
                recovery_success = random.choice([True, False])
# Симуляція успішного відновлення
                if recovery_success:
                    nodes_status[node] = "healthy"
                    print(f"{node} has been successfully recovered.")
                else:
print(f"Recovery failed for {node}. Retrying...")
                    time.sleep(5)
# Функція для симуляції збоїв вузлів
def simulate_node_failures():
    while True:
        failing_node = random.choice(list(nodes_status.keys()))
        nodes_status[failing_node] = "unhealthy"
        print(f"{failing_node} has failed.")
        time.sleep(random.randint(10, 30))

# Запуск моніторингу та відновлення вузлів
if __name__ == "__main__":
    monitor_thread = threading.Thread(target=monitor_and_recover_nodes)
    simulate_thread = threading.Thread(target=simulate_node_failures)

    monitor_thread.start()
    simulate_thread.start()

    monitor_thread.join()
    simulate_thread.join()
```

Файл resource_optimization.py

```
import psutil
import time
import random

# Функція для моніторингу ресурсів і динамічної оптимізації
def optimize_resources():
    while True:
        cpu_usage = psutil.cpu_percent(interval=1)
        memory_info = psutil.virtual_memory().percent
        disk_info = psutil.disk_usage('/').percent

# Простий приклад оптимізації
    if cpu_usage > 80:
        print("CPU usage high. Scaling down workloads...")
    if memory_info > 80:
        print("Memory usage high. Clearing cache...")
    if disk_info > 80:
        print("Disk usage high. Initiating cleanup...")

    time.sleep(10)

# Симуляція зміни навантаження на систему
def simulate_workload():
    while True:
        load_factor = random.uniform(0.5, 1.5)
        print(f"Simulating workload with load factor: {load_factor}")
        time.sleep(15)

# Запуск оптимізації ресурсів
if __name__ == "__main__":
    optimization_thread = threading.Thread(target=optimize_resources)
    workload_thread = threading.Thread(target=simulate_workload)

    optimization_thread.start()
    workload_thread.start()

    optimization_thread.join()
    workload_thread.join()
```

Файл cloud_integration.py

```
import boto3
import time

# Підключення до AWS S3
def connect_to_s3():
    s3 = boto3.client('s3')
    return s3

# Функція для завантаження файлів у хмару (AWS S3)
def upload_to_s3(s3_client, file_name, bucket):
    try:
        s3_client.upload_file(file_name, bucket, file_name)
        print(f"File {file_name} uploaded to {bucket}")
    except Exception as e:
        print(f"Error uploading file: {e}")

# Функція для резервування даних у хмарі
def backup_data_to_cloud(s3_client, file_name):
    while True:
        upload_to_s3(s3_client, file_name, 'my-backup-bucket')
        print(f"Backup of {file_name} completed")
        time.sleep(60) # Резервування кожну хвилину

# Запуск резервування у хмарі
if __name__ == "__main__":
    s3_client = connect_to_s3()
    file_name = "data.txt" # Наприклад, файл для резервування
    backup_data_to_cloud(s3_client, file_name)
```

Файл distributed_database.py

```
from cassandra.cluster import Cluster

# Підключення до Cassandra
def connect_to_cassandra():
    cluster = Cluster(['127.0.0.1'])
    session = cluster.connect('my_keyspace')
    return session

# Створення таблиці у Cassandra
def create_table(session):
    session.execute("""
CREATE TABLE IF NOT EXISTS nodes_data (
    node_id text,
    data_type text,
    value int,
    PRIMARY KEY (node_id, data_type)
)
""")
    print("Table 'nodes_data' created")

# Вставка даних у розподілену базу
def insert_data(session, node_id, data_type, value):
    session.execute("""
INSERT INTO nodes_data (node_id, data_type, value)
VALUES (%s, %s, %s)
""", (node_id, data_type, value))
    print(f>Data inserted: {node_id}, {data_type}, {value}")

# Запит даних з бази
def query_data(session, node_id):
    rows = session.execute(f"SELECT * FROM nodes_data WHERE
node_id='{node_id}'")
    for row in rows:
        print(row)

# Запуск системи
if __name__ == "__main__":
    session = connect_to_cassandra()
    create_table(session)
    insert_data(session, "Node-1", "sensor", 75)
    query_data(session, "Node-1")
```

Файл auto_update.py

```
import subprocess
import time

# Функція для автоматичного оновлення системи
def auto_update():
    while True:
        print("Checking for updates...")
        result = subprocess.run(['git', 'pull'], capture_output=True, text=True)
        if "Already up to date" in result.stdout:
            print("System is up to date")
        else:
            print("Updates installed. Restarting system...")
            restart_system()
            time.sleep(3600) # Перевіряти оновлення щогодини

# Функція для перезапуску системи
def restart_system():
    print("Restarting system...")
    time.sleep(5) # Симуляція перезапуску

# Запуск функції автоматичного оновлення
if __name__ == "__main__":
    auto_update()
```

КБПЗ_2024

Файл real_time_backup.py

```
import os
import shutil
import time

# Функція для резервування даних у реальному часі
def real_time_backup(source_directory, backup_directory):
    while True:
        for file_name in os.listdir(source_directory):
            full_file_name = os.path.join(source_directory, file_name)
            if os.path.isfile(full_file_name):
                shutil.copy(full_file_name, backup_directory)
                print(f"Backup completed for {file_name}")
        time.sleep(60) # Резервування кожну хвилину

# Запуск резервування у реальному часі
if __name__ == "__main__":
    source_directory = "./data"
    backup_directory = "./backup"

    if not os.path.exists(backup_directory):
        os.makedirs(backup_directory)

    real_time_backup(source_directory, backup_directory)
```

КБПЗ_2024

Файл ai_load_balancer.py

```
import random
import time

# Симуляція стану вузлів
nodes = {
    "Node-1": {"load": 20, "capacity": 100},
    "Node-2": {"load": 30, "capacity": 100},
    "Node-3": {"load": 50, "capacity": 100},
    "Node-4": {"load": 10, "capacity": 100},
    "Node-5": {"load": 60, "capacity": 100},
}

# Функція для оцінки продуктивності вузлів на основі навантаження
def evaluate_node_performance(node_data):
    performance_score = (node_data["capacity"] - node_data["load"]) /
node_data["capacity"]
    return performance_score

# Функція для розподілу навантаження на основі продуктивності вузлів
def ai_load_balancer(task_load):
    best_node = None
    best_score = -1

    for node_id, node_data in nodes.items():
        performance_score = evaluate_node_performance(node_data)
        if performance_score > best_score:
            best_score = performance_score
            best_node = node_id

    if best_node:
```

Файл version_control.py

```
import os
import shutil
import time

# Директорія для зберігання версій
version_directory = "./versions/"

# Функція для створення нової версії файлу
def create_versioned_copy(file_path):
    if not os.path.exists(version_directory):
        os.makedirs(version_directory)

    timestamp = time.strftime("%Y%m%d-%H%M%S")
    versioned_file_name = f"{os.path.basename(file_path)}-{timestamp}"
    versioned_file_path = os.path.join(version_directory, versioned_file_name)

    shutil.copy(file_path, versioned_file_path)
    print(f"Created versioned copy: {versioned_file_path}")

# Функція для відновлення файлу до певної версії
def restore_version(file_path, versioned_file_name):
    versioned_file_path = os.path.join(version_directory, versioned_file_name)
    if os.path.exists(versioned_file_path):
        shutil.copy(versioned_file_path, file_path)
        print(f"Restored {file_path} from version {versioned_file_name}")
    else:
        print("Version not found")

# Симуляція створення версій і відновлення файлу
if __name__ == "__main__":
```

Файл auto_scaling.py

```
import random
import time

# Симуляція стану системи
system_load = 50# Відсоток використання системних ресурсів
nodes = 5# Кількість вузлів

# Функція для автоматичного масштабування системи
def auto_scaling():
    global system_load, nodes

    while True:
        print(f"Current system load: {system_load}% with {nodes} nodes")

# Якщо навантаження перевищує 70%, додаємо вузли
    if system_load > 70:
        nodes += 1
        print(f"Added new node. Total nodes: {nodes}")

# Якщо навантаження нижче 30%, зменшуємо кількість вузлів
    elif system_load < 30 and nodes > 1:
        nodes -= 1
        print(f"Removed a node. Total nodes: {nodes}")

# Симуляція зміни навантаження
    system_load = random.randint(20, 80)
    time.sleep(5)

# Запуск автоматичного масштабування
if __name__ == "__main__":
    auto_scaling()
```