

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 реалізації AVB-  
технології у мережі”

КБПЗ - 2024

Виконав здобувач вищої освіти  
II курсу, групи КІ-23М  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Дудзінський А.М.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

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

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2024 року

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

Дудзінському Артему Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи реалізації AVB-технології у мережі

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

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

3. Строк подання студентом роботи до захисту 2.12.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи реалізації AVB-технології у мережі

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Маркетингове та економічне обґрунтування IT-проєкту.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

## 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

**Дудзінський А.М. Дослідження та програмна реалізація системи реалізації AVB-технології у мережі. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи реалізації AVB-технології у мережі.

Метою розробки є дослідження та програмна реалізація системи реалізації AVB-технології у мережі.

Об'єктом дослідження є процес реалізації AVB-технології у мережі.

Предметом дослідження є методи реалізації AVB-технології у мережі.

Методи дослідження базуються на методах теорії кодування, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи реалізації AVB-технології 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'ютерна інженерія, AVB-технологія, мережа

## ABSTRACT

**Dudzinsky A.M. Research and software implementation of the AVB technology implementation system in the network. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the system of implementing AVB technology in the network.

The purpose of the development is the research and software implementation of the AVB technology implementation system in the network.

The object of research is the process of implementing AVB technology in the network.

The subject of research is methods of implementing AVB technology in the network.

Research methods are based on coding theory methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the AVB technology implementation system 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 OS Windows 10/11.

The program was developed in the Python environment.

**Keywords:** computer engineering, AVB technology, network

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	18
2.3 Розгорнута постановка завдання .....	21
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	23
3.1 Опис функціонування системи .....	23
3.2 Розробка структурної схеми.....	26
3.3 Розробка функціональної схеми .....	34
3.4 Розробка діаграми процесів.....	37
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	39
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	39
4.2 Захист розробленого програмного забезпечення.....	50
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	53
6 НАУКОВА НОВИЗНА .....	57

						ВКРМ-123.24.0010.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Дудзінський А.М.				Дослідження та програмна реалізація системи реалізації AVB-технології у мережі	Літ.	Аркуш	Аркушів
Перев.	Петренко В.І.					М	1	84
Н.контр.	Коваленко А.С.				ЦНТУ КІ-23М			
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ .....	58
7.1	Визначення цільової аудиторії кінцевого готового продукту .....	58
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	58
7.3	Вибір методу оцінки вартості ПЗ .....	60
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	61
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ .....	62
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ .....	63
7.7	Визначення ключових факторів успіху конкретного проєкту.....	65
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	66
8.1	Вступ.....	66
8.2	Аналіз умов праці .....	67
8.3	Техніка безпеки та протипожежна профілактика .....	69
8.4	Розрахункова частина .....	72
8.5	Висновки до розділу.....	75
9	ОСНОВНІ ВИСНОВКИ.....	76
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	78

КБПЗ-2024

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>2</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

DS – DirectShow мультимедійне розширення Microsoft;

TrFl – Трансформ-фільтр;

RnFl – Рендер-фільтр;

SourceFilter – Фільтр джерела;

IC (Intelligent Connect) – інтелектуальне з'єднання;

ООП – об'єктно-орієнтоване програмування;

DSP – MS Direct Show Components and class to write Multimedia Applications – бібліотека компонентів для розробки мультимедійних програм;

VWINDOW – вертикальна синхронізація;

HWINDOW – горизонтальна синхронізація;

PLL – Цикл Блокування Стадії;

USB – Universal Serial Bus – універсальна послідовна шина;

URL – universal resource locator – локатор ресурсів Інтернет;

HTML – HyperText Markup Language – мова розмітки гіпертекстових документів;

ОС – операційна система;

ПЕОМ – персональна електронно-обчислювальна машина;

КС – комп'ютерна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** Audio Video Bridging дозволяє усунути потенційні проблеми при перекладі професійного аудіо-відео контенту (АВ-контенту) в мережу Ethernet. Процес переходу на комп'ютерні технології торкнути практично весь медіаринок. За останні кілька десятиліть від традиційної телефонії майже нічого не залишилося. Лише в деяких будинках ще збереглася проводка у вигляді «локшини», але на всіх магістральних каналах провайдери якщо не перейшли, то активно здійснюють перехід на VoIP. Аналогічний процес спостерігається й у сфері телебачення – всі частіше ця послуга надається як IP-TV через Інтернет, по мережах Ethernet і IP, що відкриває доступ до більше різноманітного контенту.

Те ж відбувається й з відеоспостереженням. Колись для цього будувалися окремі мережі й інфраструктура. Тепер повсюдно використовуються IP-відеокамери, а сигнал від них передається по комп'ютерній мережі на записуючий і відтворюючий пристрій. Чому ж настільки популярний перехід із пропріетарних, закритих технологій на Ethernet? Причина в уніфікації. А що дає уніфікація? Відповідь проста – економію.

Одна перспективна область, де перехід на використання комп'ютерної мережі ще не відбувся, – це професійні аудіовідеосистеми. У них застосовуються досить закриті технології, власні протоколи й, природно, спеціалізована інфраструктура. Крім студій звукозапису, ці системи розгортаються на більших об'єктах, де необхідно встановити встаткування для звукового оповіщення, доставки контенту, показу відео. Мова йде про стадіони, концертні комплекси, конференц-зали, парках, готелях, аеропортах і студіях віщання.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи реалізації АВВ-технології у мережі.

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

					ВКРМ-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

- Огляд існуючих систем реалізації AVB-технології у мережі.
- Дослідження системи реалізації AVB-технології у мережі.
- Програмна реалізація системи реалізації AVB-технології у мережі.

*Об'єктом дослідження є процес реалізації AVB-технології у мережі.*

*Предметом дослідження є методи реалізації AVB-технології у мережі.*

*Методи дослідження базуються на методах теорії кодування, методах математичної статистики, методах розробки програмного забезпечення.*

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

- Удосконалено метод реалізації AVB-технології у мережі.
- Розроблено вітчизняний продукт реалізації AVB-технології 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 реалізації AVB-технології 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.

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 реалізації AVB-технології у мережі, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Професійні аудіовідеосистеми – це зовсім особливий мир. Мало хто з фахівців з ІТ розбирається в АВ настільки ж добре, як, наприклад, у мережах. У цих рішень досить багато нюансів. Перша й сама головний – якість доставки аудіо- і відеоконтента. Технологія Audio Video Bridging (AVB) дозволяє усунути потенційні проблеми, які виникають при перекладі професійного АВ-контенту в мережу Ethernet: забезпечити синхронізацію, резервування смуги, рівномірний потік даних і т.д. По оцінках аналітиків, у цього ринку великий потенціал – близько 2 млрд доларів (при поточному обсязі 100 млн доларів).

### Особливості аудіо-відео

Професійні аудіовідеосистеми (Audio Video System, AVS) повинні забезпечувати студійна якість звуку й відео, при цьому вони висувають високі вимоги до рівня затримки, що не повинна перевищувати 2 мс. Найчастіше використовується кодек без стиску, для кожного каналу виділяється окремий кабель, з'єднання організується за принципом «крапка – крапка», зв'язок однобічний, напівдуплексний. При цьому процес комутації підключень досить трудомісткий, тому що кожний виробник реалізує в устаткуванні, що випускається їм, свої рознімання й свої стандарти передачі (SDI для відео й S/PDIF для звуку).

У результаті при побудові систем AVS на базі традиційних технологій доводиться використовувати велику кількість пристроїв. Всі вони зв'язані між собою безліччю кабелів, а для комутації потрібно досить складне й дороге встаткування. Причому це постійна комутація, тобто з того самого мікрофона звук виводиться на ті ж самі колонки – устаткування не можна налаштовувати динамічно.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Як зміниться картина при використанні мережі Ethernet? Кількість необхідних кабелів і встаткування різко скоротиться. Так що ж заважає перевести АВ-сигнал у мережу IP? Адже давно вже існують технології передачі аудіо-відео-контенту по мережі: VoIP, IP-TV, різні методи кодування, оцифровки й розбивки на кадри. Однак не можна просто взяти Ethernet-комутатори й поставити їх у мережу АВ: стандартна технологія Ethernet не дозволяє забезпечити ні припустиму (не більше 2 мс) затримку при проходженні сигналу через всю мережу, ні синхронізацію за часом і рівномірний потік даних, ні гарантовану пропускну здатність і автоматичне виявлення пристроїв. Саме для рішення цих проблем, без чого неможливо перевести весь АВ-трафік на Ethernet, і був розроблений стандарт AVB.

## 1.2 Область застосування

Насправді AVB – це ціла група стандартів IEEE:

- 802.1BA описує загальну архітектуру системи AVB, у тому числі автоматичне виявлення й завдання параметрів пристроїв;
- 802.1AS відповідає за синхронізацію;
- 801.Qav розширює 802.1Q у частині обслуговування чутливих вчасно потоків;
- 801.Qat доповнює 802.1Q протоколом резервування потоків.

Для початку познайомимося з використовуваними термінами. Всі кінцеві пристрої підрозділяються на Talker, Listener і Endpoint. Talker, або мовцем пристрій, передає контент у мережу. Listener, або слухач, цей контент одержує, або слухає. Endpoint, або властиво кінцева крапка, являє собою будь-який пристрій, що може приймати й/або передавати трафік AVB. Таким чином, кінцева крапка може бути як передавачем, так і приймачем або обома одночасно. Крім цього, вводиться поняття потоку, під яким мається на увазі потік

					ВКРМ-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

медіаданих від передавача й до приймача. Стандарт AVB передбачає можливість віщання з одного мікрофона на кілька відтворюючих пристроїв.

Як приклад розглянемо просту мережу, до якої підключено два джерела звуку (гітара й барабани) і два споживачі звуку (2 колонки). Будь-які музичні пристрої – і гітара, і барабани – видають аналогові сигнали. Щоб аналоговий сигнал потрапив у мережу, його потрібно перевести в цифровий вид. А для передачі оцифрованого сигналу по мережі спеціальні пристрої – експандери й конвертери – повинні інкапсулювати його до протоколу AVB. Відповідно, кожний аналоговий пристрій постачений експандером, що саме й конвертує звук у цифровий вид. На виході встановлені інші два експандера, що конвертують цифровий звук назад в аналоговий формат.

Щоб експандери «знали», на який кінцевий пристрій направляти сигнал, необхідний сервер керування. З його допомогою оператор системи АВ робить налаштування підключених до мережі пристроїв. Крім того, система керування відповідає за виявлення в мережі підключених пристроїв.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи реалізації AVB-технології у мережі, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>8</b>

## 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

У сучасних комплексних системах безпеки (КСБ) величезну роль грають системи відеомоніторингу. Це ока всієї системи безпеки. На відміну від інших підсистем КСБ, системи відеоспостереження представляють користувачеві інформацію про стан безпеки на об'єкті не у вигляді тої або іншої комбінації мерехтливих світлодіодів, а у звичному для сприйняття візуальному виді. Саме системи відеоспостереження є найбільш інформативними для користувача й саме вони забезпечують можливість персоналу служби безпеки оперативно приймати рішення залежно від обстановки на охоронюваному об'єкті, проводити розслідування по всіх тривожних ситуаціях. Але у відеосистем є й інше призначення, на справжній момент цілком не реалізоване, котре тепер є вкрай важливим і перспективним. Системи відеоспостереження можуть бути не тільки очами, але й мозком комплексних систем безпеки. Цю перспективу варто мати на увазі вже зараз, і, вибираючи систему відеоспостереження, корисно поцікавитися не тільки її актуальними можливостями й функціональними даними, але й перспективами розвитку, адже по існуючі на ринку охоронних систем практиці, ви не будете платити додаткові гроші за нові функції системи, їх установлять вам безкоштовно в якості UpGrade. Таким чином, купуючи систему відеоспостереження, ви купуєте не тільки комп'ютер, відеокамери й програмний продукт, але й перспективи розвитку цієї системи.

#### Шляхи розвитку комплексних систем безпеки

Якщо оцінити перспективи подальшого розвитку як центральної апаратури КСБ, так і її складовий компонент, то ми прийдемо до наступних

					ВКРМ-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

выводів. Системи пожежної, охоронної сигналізація, системи обмеження доступу працюють із дуже обмеженим обсягом інформації (відбулася спрацювання якого-небудь датчика або ні, чи правильно лічений номер з електронного ключа або немає й т.д.). Вся інформація, що циркулює в таких системах, має чітко певну специфіку, зв'язану винятково із прямим функціональним призначенням цих систем.

Всі ці системи, безумовно, будуть розвиватися, ставати надійніше, буде збільшуватися число прикінцевих пристроїв, що підключаються до однієї системи, оптимізуватися алгоритми керування виконавчими пристроями (устаткування блокування ліфтів, відкриття або закриття дверей, керування системою вентиляції) і т.д. Однак всі ці системи й зараз непогано працюють і цілком адекватно справляються з тими завданнями, які перед ними коштують. Навряд чи варто очікувати якого-небудь фундаментального прориву в цій області.

Розвиток цих систем не припинилося й не може припинитися, але воно може бути тільки екстенсивним.

Подивимося ж, що відбувається із системами відеоспостереження.

Відразу визначимося, що мова в даній статті йде винятково про цифрові комп'ютерні системи відеоспостереження. Саме вони переживають зараз момент революційних перетворень, що переводять ці системи на зовсім інший рівень розв'язуваних ними завдань. Перспективи ж аналогових систем відеоспостереження вже очевидний – це музей.

Щомісяця на ринку з'являються нові відеокамери, могутніші й більш швидкодійні комп'ютери, удосконалюється апаратура передачі відеоданих на відстань. Безумовно, така ситуація не може не відбиватися на системах відеоспостереження. Всі ці технічні нововведення, так чи інакше, у них застосовуються, але застосування нових відеокамер або комп'ютерів має відношення до тому, про що ми говорили вище – до екстенсивного розвитку систем.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Основний напрямок систем відеоспостереження – це впровадження в них новітніх методів аналізу відеоінформації.

Ми бачимо, що механізми аналізу, закладені в системи охоронної, пожежної сигналізації, системи контролю доступу цілком відповідають обсягам і специфіці вступник у ці системи інформації. У тім же, що стосується систем відеоспостереження ситуація, зложилася кардинально інша. Через ці системи проходить величезна кількість інформації, але механізми її обробки вкрай примітивні й не справляються з інформаційними потоками. Якщо система відеоспостереження бачить на охоронюваній території людини і єдина інформація, що вона може донести до персоналу служби безпеки, це інформація об наявність невідомих і неідентифікуємих змін в охоронюваній зоні, то невідповідність між наявною в системи інформацією й виробленими на основі цієї інформації висновками – очевидно навіть непрофесіоналові.

Таким чином, сама інформаційно насичена підсистема КСБ виявляється найменш здатною до обробки вступник інформації. Можливості величезні, а результати поки більш ніж скромні. Не дивно, що область обробки відеоінформації – ця та область, прогрес у якій за останні роки найбільш помітний. Мало того це та область, що буде надалі визначальній у розвитку комплексних систем безпеки.

Завдання-максимум у розвитку таких методів математичного аналізу, застосовуваних у системах відеоспостереження – очевидна. Уже на сучасному рівні розвитку техніки, система відеоспостереження, за допомогою спеціально розроблених відеокамер і об'єктивів може одержувати візуальну інформацію, недоступну людського ока, а в результаті подальшого вдосконалювання математичного апарата, система повинна стати свого роду аналогом людського мозку в тому, що стосується сприйняття й аналізу візуальної інформації.

Для завдань забезпечення безпеки не так вуж чи важливо розв'язний у принципі це завдання-максимум, важливо те, що в процесі успішних або неуспішних спроб її рішення з'являються цікаві й потрібні практичні розробки,

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

які можуть приносити й уже приносять відчутну допомогу в справі забезпечення безпеки.

Природно, що механізми аналізу, програмувальні в системах охоронного телебачення повинні бути багаторазово складніше, ніж аналогічний математичний апарат у системах охоронної сигналізації. Розгляд специфічної проблематики розпізнавання образом не є предметом справжньої статті, ми лише перелічимо основні практичні наробітки, які вже пройшли випробування й довели свою застосовність для завдань, розв'язуваних охоронним телебаченням. Також ми спробуємо окреслити коло перспективних розробок, які повинні бути реалізовані найближчим часом.

Розглянемо більш докладно основні перспективи розвитку відеосистем. Як ми вже визначили, воно буде відбуватися в методах аналізу накопичується інформаційною системою. Інформація – це відеознімки, відеофрагменти (маленькі фільми), які фіксує система, тобто практично, те ж саме, що сприймає людське око й аналізує мозок. Інформація ця вкрай різноманітна й неоднорідна, частина її є надлишковою, частина просто не має відносини до завдань забезпечення безпеки.

Виходячи із цього, система відеоспостереження сучасного рівня повинна класифікувати, на що варто звернути увагу, а на що ні, а на наступному етапі аналізу приймати рішення на що варто відреагувати, на що ні, і якщо реакція необхідна, те який саме вона повинна бути.

Таким чином, перше завдання, що коштує перед аналізом відеоінформації – це завдання відсікання інформаційних шумів. Тобто система не повинна реагувати на зміну природних умов (дощ, сніг і т.д.) на зміну освітленості, пов'язане зі зміною часу доби, мінливою хмарністю. У цей час існують і успішно застосовуються в системах відеоспостереження безліч ефективних алгоритмів відсікання шумовий (надлишкової) інформації.

На другому етапі система, визнавши зміну в зоні видимості нешумовим і зафіксувавши його, продовжує аналіз інформації на більше високому рівні. Якщо

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

завдання фільтрації відеоінформації є загальною для всіх систем і практично не залежить від типу об'єкта, на якому система застосовується, то другий рівень аналізу пов'язаний з розпізнаванням і категорюванням тривожних ситуацій на об'єкті, і методи такого аналізу цілком і повністю визначаються специфікою оперативних завдання, розв'язуваних на конкретному об'єкті. Систему можна запрограмувати так, що вона буде реагувати тільки на появу в зоні видимості людини й залишати без уваги кішок, що пробігають, або собак, можна запрограмувати так, що вона буде реагувати тільки на всі проїжджаючі автомобілі або тільки на вантажний транспорт. Більше тонкі методи аналізу, що перебувають поки в процесі розробки й тестування, можуть розпізнавати конкретну людину або номери проїжджаючих машин.

На третьому етапі система буде аналізувати траєкторії зафіксованого об'єкта. На практиці нерідко зустрічаються ситуації, коли знаходження людей у зоні охоронюваного об'єкта є ситуацією штатної й не потребуючої реакції служби безпеки, але є деякі особливості руху людей по території, які є підозрілими. Приміром, ходить охоронець по території по заздалегідь заданому маршруті – це нормально, а якщо він зупинився, то на цю ситуацію варто звернути увагу. Крім цього аналіз траєкторій дозволяє з високою точністю визначити швидкість спостережуваного предмета, що особливо на об'єктах з обмеженням швидкісного режиму. Іншим немаловажним застосуванням методів обчислення й аналізу траєкторій є можливість на їхній основі управляти поворотними пристроями й трансфокаторами відеокамер і в автоматичному режимі стежити за всіма переміщеннями в зоні охоронюваного об'єкта.

Резюмуючи вищевикладене можна сказати, що основні напрямки аналізу відеоінформації наступні:

1. Адекватне відсікання інформаційних шумів.
2. Усе більше точна й тонка класифікація тривожних змін у зоні охоронюваного об'єкта

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13





## **Kwikset**

Компанія представила 5 нових ригельних замків з убудованим чипом Z-Wave 500 серії, що дозволяє розширити зону дії бездротового зв'язку й підвищити надійність шифрування. У замку відсутня замкова шпара, що можна розглядати як додатковий захід безпеки, тому що виключається можливість його розкриття або злому.

Замість ключа використовується ємнісна сенсорна панель (працює на чотирьох батарейках AA), на якій уводиться код безпеки.

Також серед новинок були представлені:

– Комплект Z-Wave Smart Lock Conversion Kit для модернізації механічних замків (різних виробників) до рівня смарт-замків з інтернет-підключенням.

– П'яти-кнопковий замок у новому виконанні, що закривається в одне торкання й зроблене повністю з металу.

– Замки SmartCode 914 і SmartCode 916 з оновленим дизайном як доповнення до традиційних версій.

Всі продукти будуть доступні до кінця 2016 року, про ціни виробник поки не повідомляє.

## **Nortek Security and Control**

Компанія представила свій перший бездротовий датчик. Новинку можна встановити у вже наявні підключені димові оповіщувачі для їхньої взаємодії з панелями аварійної сигналізації 2GIG GC2 і GC3. Перетворювач 2GIG Smoke Ring встановлюється прямо між одним з димових оповіщувачів у мережі й стіною або стелею.

Традиційні взаємозалежні димові оповіщувачі сконструйовані таким чином, що якщо один пристрій визначив дим, звуковий сигнал передається пристроями по ланцюжку, щоб сповістити все приміщення про небезпеку. Однак багато пристроїв не оснащені можливостями такої взаємодії й не можуть прямо передавати сигнал на панель. 2GIG Smoke Ring може допомогти це виправити.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

## **Panasonic**

Компанія представила модернізовану систему для спостереження за будинком “зроби сам”. Суть відновлення полягає в можливості підключення трьох “розумних” термостатів Honeywell з підтримкою Wi-Fi.

Система побудована навколо хаба, що взаємодіє з датчиками й іншими пристроями через стандарт бездротового доступу DECT-ULE з ультра-низьким енергоспоживанням і зоною дії більше 300 метрів. Хаб також оснащений Wi-Fi, завдяки чому користувачі можуть управляти системою дистанційно.

До складу лінійки Panasonic входять датчики й розумні розетки з підтримкою технології бездротового зв'язку DECT, камера Wi-Fi з мікрофоном для будинку й зовнішня камера. Для керування можна також використовувати бездротову слухавку.

**URC:** постачальник рішень для домашньої автоматизації представив свою першу систему безпеки, до якої можна підключати мультимедійні системи й системи домашнього керування. Новинка включає хаб C100 HomeSet, яким можна управляти через смартфон. За бажанням можна встановити панель дистанційного керування R100 HomeSet Remote Control для керування пристроями з підтримкою Z-Wave, продуктами Nest, аудіо системами Sonos і іншими пристроями у вісьмох кімнатах.

Додаток HomeSet сумісний з додатками інших виробників систем безпеки. Камери мають дозвіл 1080p, кут огляду 130 градусів, ІЧ підсвічування й підтримують технологію широкого динамічного діапазону. Відео з камер відображається на РК дисплеї панелі R100.

## **Yi Technology**

Компанія із Сієтла запускає свою другу камеру відеоспостереження для будинку з підтримкою Wi-Fi Yi Home Camera 2.

За словами виробника, новинка включає карту пам'яті Micro SD на 32Гб і має значні поліпшення в порівнянні з раніше випущеною моделлю.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

По-перше, нова модель записує відео з дозволом 1080p і оснащена запатентованими технологіями запису й аналітики, що забезпечують високу якість запису відео й звуку. Крім цього, кут огляду камери був збільшений до 130 градусів і додані функції широкого динамічного діапазону, корекції перекручувань об'єктива, виявлення рухів і нічного бачення (ІЧ підсвічування).

Не можна не відзначити нову технологію аудіокодування, що дозволяє забезпечити висока якість звуку й знизити навколишній шум. Частота дискретизації двостороннього звукозапису становить до 48 КГц. Пристрій може визначити звук плачучої дитини й відправити користувачеві повідомлення на мобільний.

YI Home Camera 2 уміє розпізнавати силует людини й відрізнити його від інших об'єктів, що дозволяє знизити число помилкових спрацьовувань.

Виробник також пропонує відеореєстратори й екшн-камери.

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

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

Python – скриптова мова програмування з досить простим синтаксисом. Для розуміння достатньо порівняти принципи написання найпростішої програми, яка виводить на екран текстове повідомлення. Саме тому мова програмування Python більш доступна для новачків, а професіонали встигли адаптувати її для вирішення великої кількості завдань. Це мультиплатформне рішення, тому

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

знання Python дає можливість працювати у різних сферах: від розробки мобільних застосунків до ігрової індустрії та штучного інтелекту.

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

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

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

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

Мова програмування Python використовується у наступних сферах:

1. Розробка програмних застосунків будь-якого напрямку.

					ВКРМ-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19



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

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

### **Популярні Python IDE**

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

### **2.3 Розгорнута постановка завдання**

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи реалізації AVB-технології 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 методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

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

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

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

КБПЗ\_2024

					VKPM-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

#### Синхронізація

У великому приміщенні, наприклад у концертному залі, колонки, що відтворюють звук, перебувають у різних його частинах, і тому виникає проблема синхронізації. Необхідно зробити так, щоб звук у ці колонки надходив у те саме час, інакше та, що перебуває ближче до джерела, буде відтворювати його раніше, ніж розташована, скажемо, за трьома комутаторами, кожний з яких вносить певну затримку при проходженні сигналу. Одночасність відтворення забезпечує протокол синхронізації.

Як він працює? По-перше, всі пристрої обмінюються певними пакетами, перевіряючи, що кожне з них підтримує протокол 802.1AS (це стосується всіх пристроїв, сертифікованих по стандарті AVB). Далі, автоматично або шляхом налаштування вручну, у мережі вибирається так званий гранд-майстер, що стає єдиним і самим головним еталоном часу в мережі. Це може бути будь-який мережний пристрій, у цьому випадку ми вибрали Viamp Tesira. Воно може «брати» час як з мережі (з будь-якого мережного пристрою по протоколі NTP), так і від якогось зовнішнього джерела (наприклад, від установлених рядом атомних годин).

Цей гранд-майстер повідомляє всі інші пристрої, скільки зараз часу, при цьому протокол ураховує, що пакети, що йдуть до конкретного експандера, передаються по конкретному шляху. Кожний пристрій AVB на шляху проходження пакета записує в пакеті 802.1AS внесену їм затримку: умовно говорячи, комутатор указує, що на вхідному інтерфейсі 100 Мбіт/с така-те затримка, при проходженні трафіку між двома його портами затримка така-то, на вихідному гігабітному інтерфейсі, відповідно, затримка така-то. Всі затримки

					ВКРМ-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

підсумуються й відбиваються в пакеті 802.1AS. Таким чином, у пакеті, що надійшов на кінцевий пристрій, утримується фактична затримка при передачі по мережі.

Як бачимо, пристрою не тільки синхронізують годинники й одержують інформацію про точний час, але й визначають затримку, внесену будь-якими мережними вузлами. Знаючи точний час і величину затримки на кожному етапі, можна розрахувати так званий презентаційний час, коли повинен бути відтворений сигнал на всіх одержувачах. Воно визначається виходячи із самого гіршого варіанта проходження сигналу від джерела до пристрою відтворення. Відповідно, звук з гітари відправляється на далеку колонку раніше, ніж на ближню, тому на обидві колонки він приходить одночасно.

Це одне із самих головних досягнень групи стандартів AVB: забезпечення синхронізації за часом у мережі Ethernet при передбачуваних затримках, завдяки чому досягається синхронне відтворення звуку й відео.

### **Резервування потоків**

Що відбувається після визначення затримок у мережі? Коли оператор АВ-системи вказує, що звук від гітари й барабанів повинен надходити на обидві колонки, сервер керування інформує відповідні експандери про те, що їм потрібно передавати звук такому-то й такому-то кінцевому встаткуванню (його експандерам). Потім у дію вступає протокол резервування потоків (Stream Reservation Protocol, SRP).

Відповідно до протоколу, кожне джерело звуку надсилає запит, що називається Talker Advertise – дослівно «оголошення мовця». У відповідь експандери, яким ці потоки призначаються, відправляють повідомлення про готовність Listener ready. Воно знову проходить весь шлях по мережі до первісних експандерів, звідки береться звук. Мережні комутатори відслідковують всі чинені дії й при зворотному проходженні пакетів з інформацією про готовність слухачів кожний з них резервує необхідну пропускну здатність для даного конкретного потоку.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Це одне з найважливіших додавань у стандарт, тому що воно дозволяє гарантувати смугу для трафіку. Причому мова йде не про гарантію засобами QoS, а про тверде виділення необхідної пропускної здатності. До 75% пропускної здатності інтерфейсу може бути віддане під AVB, а 25% залишено для іншого трафіку, щоб AVB не займав весь канал.

Якщо смуги не вистачає, наприклад, комутатор уже й так обслуговує досить багато потоків AVB на певному інтерфейсі, то цей комутатор пошле спеціальне повідомлення в обидва боки про неможливість організувати потік.

У даному прикладі аудіопотоки йдуть прямо від джерел до пристроїв відтворення, але на практиці вони, швидше за все, будуть проходити через систему мікшування.

### **Зіткнення пріоритетів**

Якщо який-небудь користувач починає завантажувати з Інтернету файл великого обсягу й завантаження займає значну пропускну здатність якого-небудь інтерфейсу, то смугу, виділену для потоків AVB, цей трафік не зможе використовувати. Однак при перегляді, наприклад, потокового відео або у випадку проведення ресурсномісткої телеконференції конкуруючий за смугу трафік теж має високий пріоритет.

Відповідно, комутатор з підтримкою DiffServ буде намагатися відправити й потік AVB, і потік IP-TV якнайшвидше, з тим самим пріоритетом. Відсутність розподілу відправлення пакета в часі може привести до переповнення буферів. Щоб цього не відбувалося, у рамках групи стандартів AVB прийнятий ще один стандарт – FQTSS.

Він забезпечує свого роду формування трафіку (shaping), але не зовсім звичайне – на основі кредитних буферів. FQTSS розподіляє пакети за часом, тобто забезпечує рівномірний, плавний потік пакетів AVB, що виключає виникнення різниці в затримці, дозволяючи уникнути джиттера.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

### 3.2 Розробка структурної схеми

#### Налаштування AVB на комутаторах

AVB споконвічно створювався таким чином, щоб для його підтримки було потрібно внести лише мінімальні зміни в стандарт Ethernet, у пристрої, що беруть участь в організації мережі Ethernet, і в налаштування встаткування.

Формально, для того щоб настроїти AVB на комутаторах, необхідно виконати налаштування наступних компонентів:

- VLAN;
- протоколу резервування смуги;
- шейпінга;
- синхронізації й т.д.

На перший погляд, це складно, але на комутаторах Extreme Networks протокол AVB настроюється однією командою `enable AVB ports all`, за якої ховається багато внутрішніх команд, тобто це якась автоналаштування, що дозволяє комутатору самостійно здійснювати синхронізацію, резервування пропускної здатності й інші дії.

Щоб комутатор Ethernet міг підтримувати AVB, у ньому повинна бути реалізована апаратна й програмна підтримка. У комутаторах Extreme апаратна підтримка забезпечується встановленими в них чип-сетами, а програмна – операційною системою. У цей час Extreme Networks – єдина компанія, що випускає комутатори корпоративного рівня з підтримкою AVB і одночасно POE і POE+. Для використання AVB ніяких додаткових ліцензій здобувати не потрібно.

POE й POE+ потрібні для живлення експандерів, тому що колонки, мікрофони й інші системи можуть розташовуватися де завгодно. У величезному концертному залі колонка може перебувати досить далеко від розетки, і її підключення до мережі Ethernet через експандер вимагає підведення до нього електрики. Щоб спростити інфраструктуру, у тому числі електричну, для живлення таких пристроїв краще задіяти POE.

					ВКРМ-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

У цей час AVB підтримується фіксованими комутаторами серій 440 (комутатори початкового рівня для доступу), 460 (комутатор рівня агрегації) і X-670 (48-портовий комутатор з можливістю установки портів 40 Gigabit Ethernet). Найближчим часом буде реалізована підтримка AVB у комутаторах Black Diamond X-8.

### **Планування мережі**

Що входить до складу готового рішення? Насамперед потрібні комутатори з підтримкою AVB і пристрою для конвертації звуку (для цифрового або аналогового форматів) в Ethernet з підтримкою AVB. При цьому протоколи AVB повинні підтримуватися кожним задіяним пристроєм, тобто потрібна наскрізна підтримка AVB. У кожному разі потрібне грамотне планування всієї мережі.

Залежно від розв'язуваних завдань необхідно розрахувати кількість потоків і необхідну пропускну здатність. Так, на один звуковий канал (моно) потрібно близько 6 Мбіт/с, якщо це незжатиий звук. Знаючи кількість потоків, можна визначити, які з'єднання будуть потрібні (100 Мбіт/с, 1, 10, 40 Гбіт/с).

Якщо, наприклад, у мережі є 100 пристроїв, на яких необхідно відтворювати той самий контент, і на кожній пристрій направляти окремий потік, то ці потоки займуть всю доступну ємність. Однак, задіявши багатоадресне розсилання, можна на один потік підключити навіть 500 пристроїв.

Крім того, мережа Ethernet накладає обмеження на дальність передачі. У випадку мідної проводки вона становить 100 м, а щоб затримка не перевищувала 2 мс, у мережі не повинне бути більше 7 транзитних вузлів (для каналів 100 Мбіт/с). Коли в мережі використовуються гігабітні з'єднання, при тих же 7 транзитних вузлах затримка буде становити менш 1 мс.

У даній роботі пропонується використовувати технологію Audio Video Bridging для систем відеоспостереження.

Структура системи мережного відеоспостереження з використанням технології Audio Video Bridging (AVB):

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

- периферійне встаткування (відеокамери й тепловізори, відеокодери , відеореєстратори, кожухи й поворотні пристрої, прожектори);
- мережа передачі дані системи відеоспостереження (радіоканали, оптоволокло, кручена пари), активне й пасивне мережне встаткування (СКС, комутатори, маршрутизатори)
- серверне встаткування системи відеоспостереження (система зберігання інформації);
- програмне забезпечення для аналізу відеоінформації, керування камерами й роботи з відеоархівами;
- автоматизовані робочі місця інтегрованої системи відеоспостереження: центральний пункт спостереження й керування (ЦПСК), додаткові пункти спостереження й керування (ДПСК);

### **Устаткування системи відеоспостереження**

#### **Камери**

Мережні камери можна класифікувати залежно від можливості їхнього використання усередині й поза приміщенням.

Мережні камери (як внутрішні, так і вуличні) підрозділяються на фіксовані, фіксовані купольні, PTZ-камери (поворотні з можливістю трансфокації) і купольні PTZ-камери.

Залежно від умов експлуатації камери забезпечуються захисними кожухами, що охороняють від впливу навколишнього середовища (пилу, вологи, опадів, високих і низьких температур), вандалізму або псування.

Також виділяють камери з підвищеною розв'язною здатністю – HDTV/мегапіксельні камери. Мегапіксельні мережні камери забезпечують дозвіл, як мінімум у три рази переважаючий дозвіл аналогових CCTV-камер.

До окремого класу мережних камер ставляться тепловізійні мережні камери. Завдяки роботі в інфрачервоному діапазоні випромінювання тепловізійні камери можуть працювати в повній темряві, без додаткового висвітлення в режимі цілодобового спостереження.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>28</b>



Вибір фізичного середовища для підключення залежить від декількох факторів:

- розташування місця установки камери;
- можливості прокладки кабелю;
- відстані камери від найближчої можливої точки підключення до мережі (найближчого комутатора)
- вимоги до якості й розв'язної здатності зображення, одержуваного з камери (дозвіл, кількість кадрів у секунду).

Особливості підключення з використанням провідних технологій:

– Кабель кручена пара, стандарт IEEE 802.3 (Ethernet): максимальна довжина кабелю становить 100м; швидкість підключення 100/1000 Мбіт/с; є можливість забезпечення живлення камери через кабель передачі даних за технологією PoE. Перевага: висока швидкість передачі, дешевина технології. Недоліки: невелика дальність.

– Мідний кабель одна/дві пари проводів, стандарт SHDSL: максимальна довжина кабелю до 0,5км при швидкості 15 Мбіт/с і до 5 км при швидкості до 1Мбіт/с. Перевага: дешевина технології, більші дальності. Недоліки: швидкість сильно залежить від товщини провідників у мідній парі; необхідність використання додаткового встаткування (модемів).

– Багатомодовий оптоволоконний кабель 1000 BASE-SX: максимальна довжина кабелю становить 550 м швидкість підключення до 1000 Мбіт/с. Перевага: висока швидкість, більші дальності. Недоліки: трудомісткість прокладки, вартість, необхідність використання додаткового встаткування (медіаконверторів).

– Одномодовий оптоволоконний кабель 1000 BASE-LX: максимальна довжина кабелю становить 5 км швидкість підключення до 1000 Мбіт/с. Перевага: висока швидкість, більші дальності. Недоліки: трудомісткість прокладки, висока вартість робіт і кабелю, необхідність використання додаткового встаткування (медіаконверторів).

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Особливості підключення з використанням бездротових каналів зв'язку:

Використання бездротових технологій для побудови систем охоронного відеоспостереження – це швидке, економічне й гнучке рішення при організації відео спостереження на великій площі, при цьому відпадає необхідність у прокладці кабелю. Також широкополосні бездротові канали даних є єдиною можливістю підключення камер до мережі передачі даних, де прокладка кабелів заборонена, утруднена або просто неможлива (переправи, мости, залізничні колії, захищені будинки й т.д.).

Бездротові системи широкополосного доступу, використовувані для побудови каналів зв'язку в системах відеоспостереження, повинні задовольняти наступним вимогам:

- Підтримка технології адаптивного маркерного доступу до радіосередовища (адаптивного поліпшення) по засобах організації дуплексних каналів зв'язку з тимчасовим поділом (TDD).

- Висока смуга пропускання й пакетна продуктивність.

- Стійкість системи до багатопроменевого перевідбиття сигналу в умовах складної портової забудови.

- Установка пріоритету передачі відеотрафіку (QoS).

- Можливість передачі безлічі відеопотоків по одному радіоз'єднанню.

- Можливість передачі відео Multicast-віщанням.

- Використання убудованих сучасних алгоритмів автентифікації, авторизації (WPA/WPA2) і шифрування трафіку (AES) для запобігання несанкціонованого доступу.

Виробники, що рекомендуються: Infinet Wireless, Proxim Wireless.

### **Система керування відеоспостереженням**

Система керування відеоспостереженням складається з:

- Серверів керування.

- Мережного сховища.

- АРМ керування й моніторингу відеоспостереження.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

– ПО для керування системою відеоспостереження.

Вибір параметрів серверів керування й визначення розмірів мережного сховища інформації залежить від наступних факторів:

- Кількість камер.
- Необхідність у безперервному записі або записі при виявленні події.
- Кількість годин у день, відведених для запису.
- Частота кадрів у секунду.
- Дозвіл зображення.
- Типи стиску відео: Motion JPEG, MPEG-4, H.264.
- Зона спостереження: складність зображення (наприклад, сіра стіна або ліс), висвітлення й кількість рухів (офіс або переповнена станція метро).
- Тривалість зберігання даних.
- Необхідності резервування.
- Кількості АРМ операторів відеоспостереження

Основним призначенням серверів з ПО керування системою відеоспостереження є:

- Запис відеозображення й звуку в реальному часі
- Аналіз і відтворення відеозображення
- Пошук і високошвидкісне перемотування відеоархіву
- Підтримка функцій відеоаналітики й керування подіями (адаптивний детектор об'єктів, детектор залишених предметів, детекція саботажу, детектори перетинання й напрямку, лічильник об'єктів по напрямку руху й лічильник об'єктів, що перетинають задану лінію)
  - Гнучка система відображення інформації з великої кількості камер на моніторах АРМ операторів спостереження
  - Інтеграція з іншими системами забезпечення безпеки морського порту (охоронно-пожежною сигналізацією, СКД і ін. промисловими системами).
  - Підтримка стандарту ONVIF для адміністрування й керування широким набором периферійних IP-пристроїв;

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

## Опис розподіленої мережної системи відеоспостереження

Периферійне встаткування підключається по радіоканалах, оптоволокну або кручений парі до мережі передачі дані системи відеоспостереження. Потоки відеоінформації з периферійного встаткування надходять на сервери керування відеоспостереженням через які інформація відправляється в мережне сховище даних.

Сервери керування підключені до Центрального пункту спостереження й керування через локальну мережу системи відеоспостереження (радіоканали, кручена пари, оптичний кабель). З будь-якого сервера можна бачити й локальні камери, і камери, підключені до інших серверів при наявності відповідних прав доступу.

Система відеоспостереження забезпечує безперервний відеозапис, відображення й відеодетекцію руху для кожної з підключених відеокамер. Перегляд відеоархіву здійснюється без переривання процесу запису.

При виявленні несанкціонованого проникнення на територію морського порту або виникнення позаштатної ситуації:

- включається система звукового або мовного оповіщення;
- автоматично передається сигнал тривоги й відеозображення з місця події групі швидкого реагування;
- автоматично передається сигнал тривоги відповідальним особам і відповідальному керівникові морського на стільниковий телефон по GSM/GPRS каналі.

### Переваги

- Простота інтегрування з існуючими аналоговими системами відеоспостереження – існуючі аналогові камери підключаються до серверів через медіаконвертери.
- Створення додаткових інформаційних каналів морського порту: резервні канали мережі відеоспостереження можуть бути використані Замовником для розширення інформаційної мережі;

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

- Простота розширення системи: додаткові камери можуть бути підключені з мінімальними тимчасовими й працезатратами;
- Висока надійність системи: при порушенні зв'язку між локальними серверами, кожний сервер продовжує функціонувати в автономному режимі.

### 3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно. Для того, щоб зрозуміти, як функціонально працює система наведені відповідно наступні алгоритми:

- Опис алгоритму збільшення частоти кадрів АВ-контенту на основі піксельної компенсації руху та подальше придушення ефекту тремтіння кадру у VideoStream.

- Опис алгоритмів придушення тремтіння кадру у VideoStream з використанням Image-Based Rendering.

Наведемо спершу опис цих алгоритмів.

Розроблений алгоритм дозволяє збільшувати частоту кадрів VideoStream в довільне ціле число разів. Перетворення частоти проводиться за рахунок збільшення загального числа кадрів VideoStream – додавання нових (інтерпольованих) кадрів. Нові кадри вставляються у вихідну відео послідовність таким чином, що число інтерпольованих кадрів між будь-якими двома сусідніми вихідними кадрами одне і те ж і пропорційно кратності.

Розглянемо як він працює. Одна ітерація алгоритму починається з обчислення карти відео векторів далі проводиться сегментація даних та обчислення інтерпольованих кадрів. З правого боку описано основну формулу алгоритму.

Придушення ефекту тремтіння кадру у VideoStream з використанням Image-Based Rendering працює наступним чином.

Ефект тремтіння кадру виникає через неякісну лінію зв'язку з мережею

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Інтернет. Спочатку проходить визначення яскравості пікселя з обчисленням наведеним у формулі.

Далі проходить розрахунок схеми розташування локальних сусідів пікселя та схеми розташування глобальних сусідів пікселя з обчисленням координат поточного пікселя на інтерпольованому кадрі з врахуванням вектора руху у наведеній формулі.

Основне призначення системи це обробка АВ-контенту застосовуючи алгоритм піксельної компенсації руху.

Функціональна схема розробленої системи зображена на рисунку 3.2. З рисунку видно, що розроблена система працює на алгоритмі збільшення частоти кадрів АВ-контенту на основі піксельної компенсації руху та подальше придушення ефекту тремтіння кадру у VideoStream.

Програмний продукт, забезпечує універсальну систему обробки вхідного потоку, забезпечує гарантовано стійкій відео сигнал з мінімальною кількістю шумів та з максимальною різкістю зображення яка ґрунтується на нових алгоритмах обробки відеозображень та частково модернізованих існуючих.

Розглянемо її, спочатку ми отримуємо вхідні дані – цифровий потік відео інформації як він є, тобто неякісний, нечіткий. Він має N кількість кадрів.

До ОС Windows 10 він потопляє через підсистему драйверів (WINDOWS DRIVER MODEL).

Спочатку до драйверу відео пристрою потім до блоку управління WDM пристроями далі до WDM драйверу операційної системи.

Після проходження такої ланки розроблене ПЗ може його обробляти що і проходить.

Розроблене ПЗ приймає АВ-контент та застосовує наступні алгоритми:

– Збільшення частоти кадрів VideoStream на основі піксельної компенсації руху.

– Відстеження точкових особливостей у VideoStream, зміна різкості та придушення ефекту тремтіння кадру.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

- Проведення просторово-часового придушення шуму у VideoStream.
- Стиск АВ-контенту на основі алгоритму Video Codec Multi-Scale Retinex (MSR).

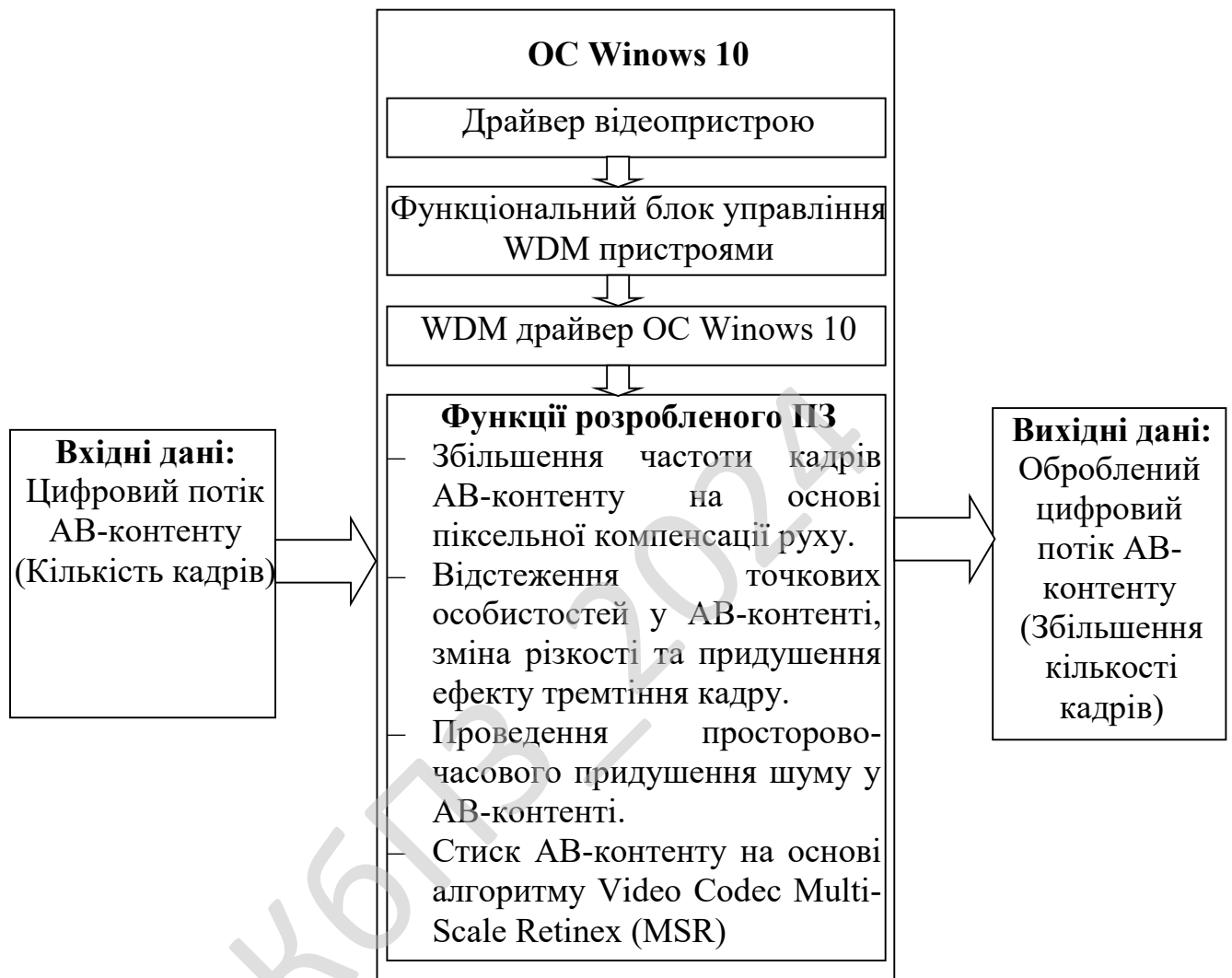


Рисунок 3.2 – Функціональна схема системи

В кінцевому випадку на виході ми отримуємо оброблений АВ-контент зі збільшеною кількістю кадрів  $N+A$ .

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

### 3.4 Розробка діаграми процесів

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

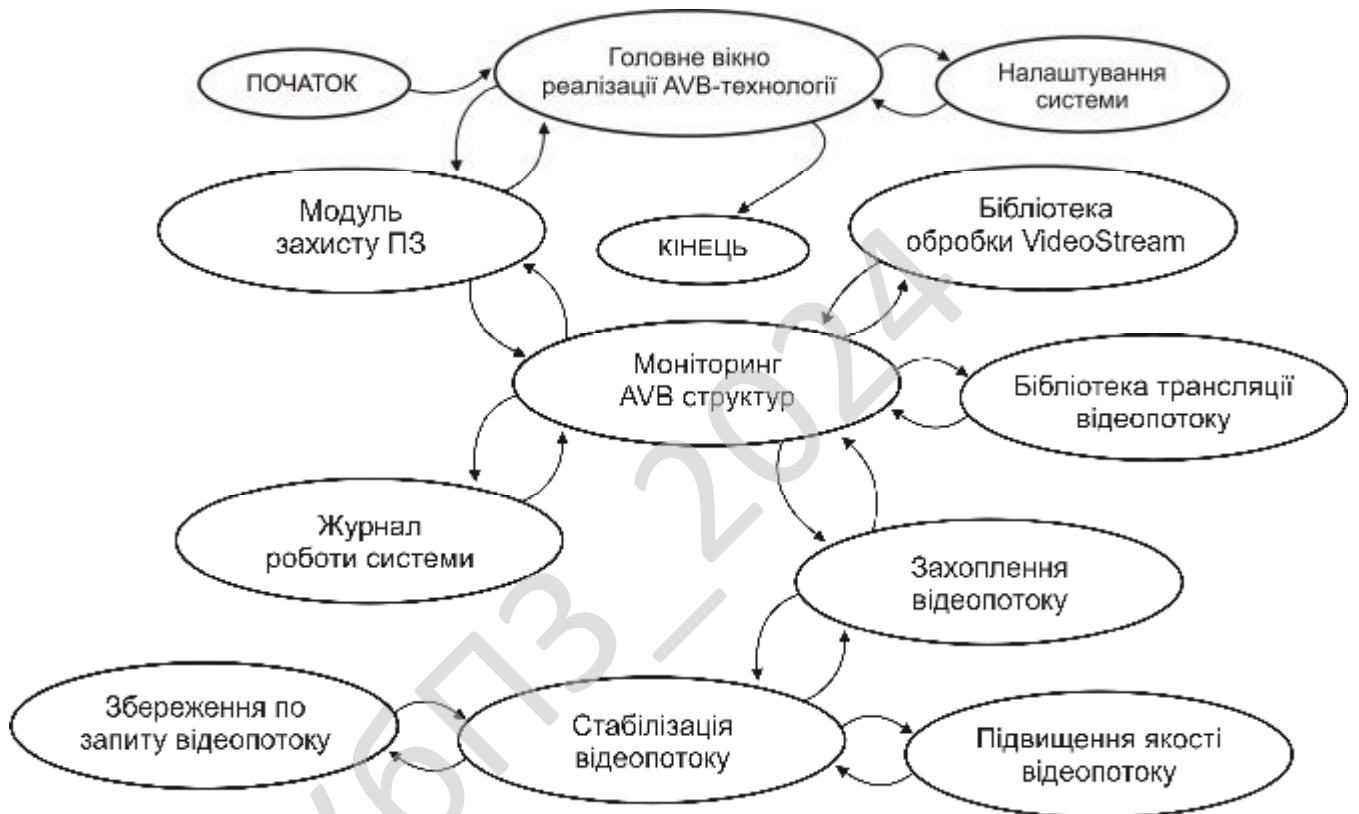


Рисунок 3.3 – Діаграма взаємодії процесів

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.

– 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.

КБПЗ – 2024

					ВКРМ-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

## 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.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

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

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю реалізації AVB-технології 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.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

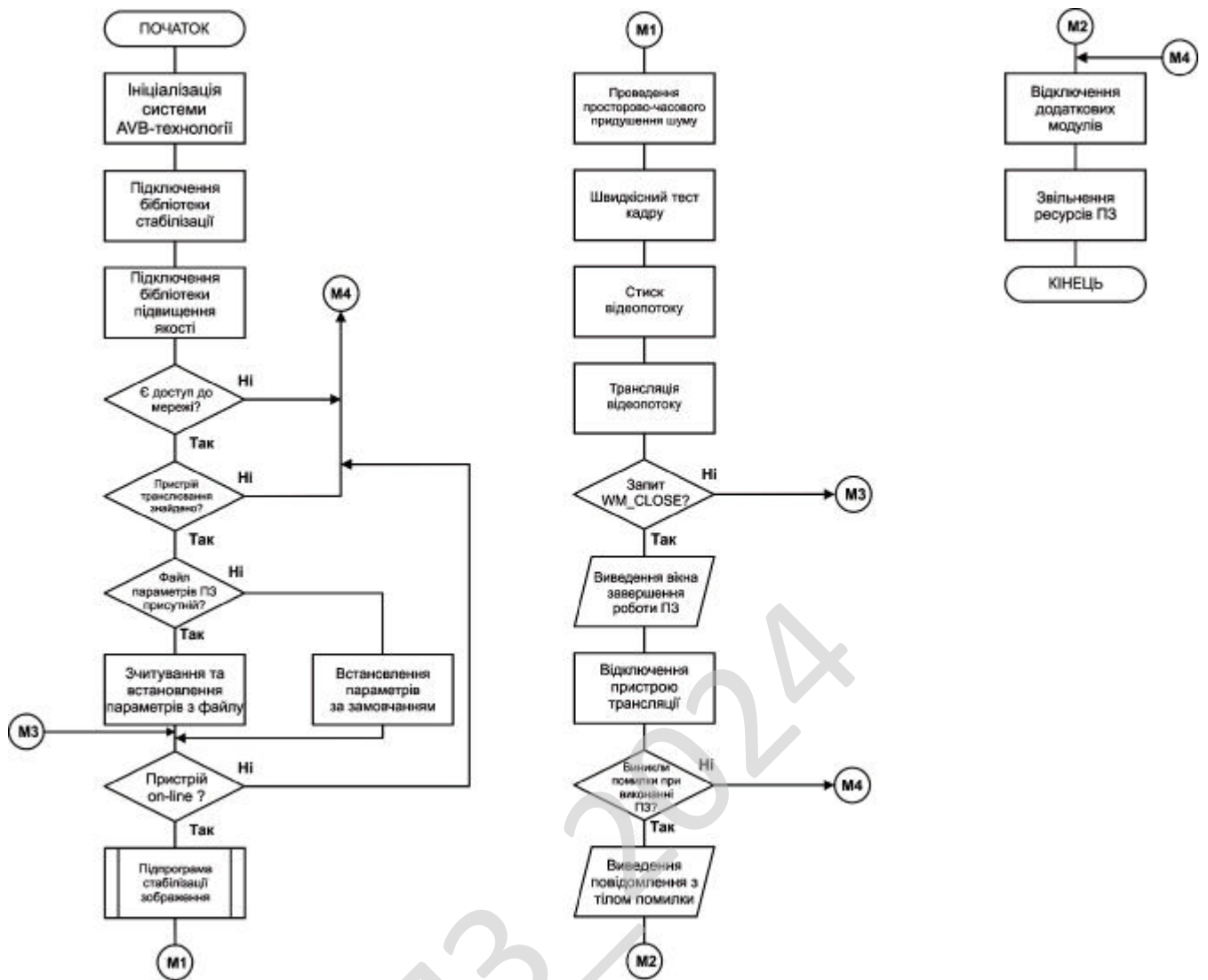


Рисунок 4.1 – Блок-схема основної програми

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.



(елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються.

Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

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

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

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

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.

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

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

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

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

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

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

- Керівникам проектів, які керують розподілом завдань і контролем за проектом.
- Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.
- Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.
- Програмістам які реалізують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

Також при розробці магістерської дипломної роботи було використано наступні підходи UML: Діаграма класів; Діаграма компонент; Діаграма об'єктів; Діаграма розгортання.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

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

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

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

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

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

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

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

					ВКРМ-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

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

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

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

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

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

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

					ВКРМ-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46



багаторазово створюватись під час роботи програми, слід використовувати діаграму взаємодії, яка може відображати характеристики об'єктів та зв'язків. Екземпляр діаграми взаємодії створює діаграму об'єктів.

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

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

AVB (Audio Video Bridging) – це набір стандартів, які дозволяють передавати аудіо- і відеодані по Ethernet-мережах з низькою затримкою та високою надійністю. Для реалізації AVB-технології у мережі необхідно розробити програмне забезпечення, яке керує передачами, враховує затримки та забезпечує необхідний рівень якості обслуговування (QoS). Розглянемо архітектуру та основні модулі такої системи.

Архітектура системи складається з наступних основних модулів:

1. Network Configuration Module конфігурує мережу для забезпечення підтримки AVB-трафіку.
2. Traffic Shaping and Scheduling Module забезпечує управління чергою пакетів та їх розподіл з урахуванням пріоритетів.
3. Stream Reservation Protocol (SRP) Module керує резервуванням ресурсів для кожного аудіо/відео потоку.
4. Synchronization Module забезпечує синхронізацію годинників усіх пристроїв у мережі для мінімізації затримок.

import socket

						<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			<b>48</b>

```

import time
import threading

class AVBSystem:
    def __init__(self, network_interface):
        self.network_interface = network_interface
        self.srp_module = SRPModule()
        self.traffic_module = TrafficShapingModule()
        self.sync_module = SynchronizationModule()

    def configure_network(self):
        self.sync_module.synchronize_clocks()
        self.traffic_module.setup_qos(self.network_interface)
        self.srp_module.setup_streams()

    def start_streaming(self, stream_data):
        self.srp_module.reserve_resources(stream_data)
        self.traffic_module.start_transmission(stream_data)

class SynchronizationModule:
    def synchronize_clocks(self):
        # Виконується синхронізація годинників по мережі
        pass

class TrafficShapingModule:
    def setup_qos(self, interface):
        # Налаштовується якість обслуговування (QoS) для інтерфейсу
        pass

    def start_transmission(self, stream_data):
        # Починається передача даних з урахуванням черги та пріоритету
        pass

class SRPModule:
    def setup_streams(self):
        # Налаштування потоків для AVB
        pass

    def reserve_resources(self, stream_data):
        # Резервування ресурсів для кожного потоку
        pass

```

```

# Приклад використання системи
avb_system = AVBSystem("eth0")
avb_system.configure_network()

stream_data = {
    "id": 1,
    "type": "audio",
    "bandwidth": 512, # Кількість кбіт/с
    "priority": 1
}

avb_system.start_streaming(stream_data)

```

Розрахунки та дані для підтвердження правильності проектних рішень.

Для визначення пропускної здатності мережі та ефективного резервування ресурсів необхідно врахувати такі параметри:

1. Пропускна здатність кожного AVB-потіку (наприклад, аудіо-потік може потребувати 512 кбіт/с, відео-потік до 10 Мбіт/с).
2. Затримка, яка має бути меншою ніж 2 мс для аудіо-потоків і до 5 мс для відео.
3. Синхронізація годинників у мережі має бути з точністю до мікросекунд.

Також важливо перевірити та відрегулювати параметри QoS на мережевому обладнанні. Це може бути зроблено за допомогою налаштувань DSCP (Differentiated Services Code Point) або використання окремих черг для пріоритетного трафіку.

#### 4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Threefish – в криптографії симетричний блоковий криптоалгоритм, розроблений автором Blowfish та Twofish, американським криптографом Брюсом Шнайером 2008 року для використання в геш-функції Skein і як універсальну заміну наявним блоковим шифрам. Основними принципами розробки шифру були: мінімальне використання пам'яті, необхідна для використання в геш-

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50



```

    {33, 8, 18, 57, 21, 12, 32, 54},
    {34, 43, 25, 60, 44, 9, 59, 34},
    {28, 7, 47, 48, 51, 9, 35, 41},
    {17, 6, 18, 25, 43, 42, 40, 15},
    {58, 7, 32, 45, 19, 18, 2, 56},
    {47, 49, 27, 58, 37, 48, 53, 56},
};
// D - раунд, j - індекс в таблиці циклічного зсуву
void mix (int j, int d) {
    unsigned long long rotl;
    y [0] = x [0] + x [1];
    rotl = R16 [d% 8] [j];
    y [1] = (x [1] << rotl) | (x [1] >> (64 - rotl));
    y [1] ^ = y [0];
}
</Source>

```

Процедура розшифрування обернена процедурі зашифрування і містить зворотну функцію DEMIX.

Кожен з 72 раундів Threefish-256 і Threefish-512 має чотири MIX перетворення, Threefish-1024 – вісім звернень до MIX функції.

### Безпека

За заявою авторів, алгоритм має більш високий рівень безпеки, ніж AES. Існує атака на 25 з 72 раундів Threefish, в той час як для AES – на 6 з 10. Threefish має показник фактора безпеки 2.9, в свою чергу, AES всього 1.7 [3]

Для досягнення повної дифузії, шифру Threefish-256 досить 9 раундів, Threefish-512 – 10 раундів і Threefish-1024 – 11 раундів. Виходячи з цього, 72 і 80 раундів відповідно в середньому, забезпечать кращі результати, ніж існуючі шифри. [4]

У той же час, алгоритм має набагато простішу структуру і функцію перетворення, проте виконання 72-80 раундів, на думку дослідників, забезпечує необхідну стійкість. Вживаний розмір ключа від 256 до 1024 біт зводить нанівець можливість повного перебору паролів при так званій атаці грубою силою (brute force attack) на сучасному обладнанні.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання магістерської дипломної роботи. Розроблене програмне забезпечення реалізації AVB-технології у мережі складається з наступних функціональних блоків: Навігаційне меню (Відеопотік, Фільтри AVB-технології, Мережа, Налаштування, Довідка); Вікно виведення відеоінформації; Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Функціональних кнопок ПЗ (On/Off стабілізацію, On/Off стиску відеопотоку підвищення, On/Off підвищення якості зображення, Налаштування, Налаштування джерела відеосигналу, Авторське право).



Рисунок 5.1 – Головне вікно розробленого ПЗ

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>53</b>

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.

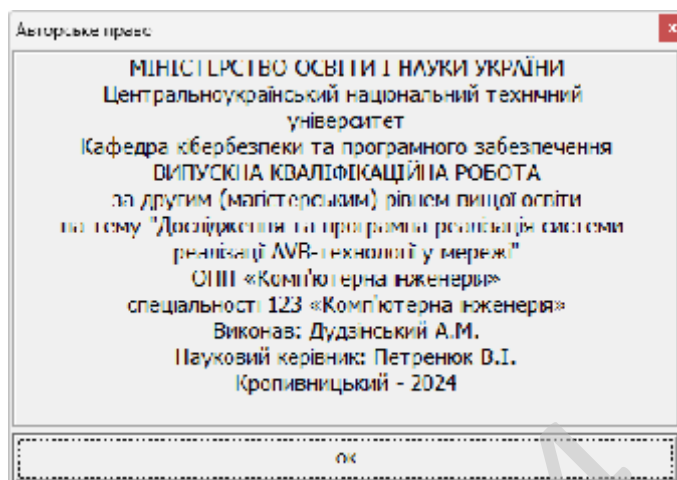


Рисунок 5.2 – Вікно розробника ПЗ

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.

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.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Обрано умови розповсюдження – Shareware.

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

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

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

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

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

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи реалізації AVB-технології у мережі.

*Метою розробки є дослідження та програмна реалізація системи реалізації AVB-технології у мережі.*

*Об'єктом дослідження є процес реалізації AVB-технології у мережі.*

*Предметом дослідження є методи реалізації AVB-технології у мережі.*

*Методи дослідження базуються на методах теорії кодування, методах математичної статистики, методах розробки програмного забезпечення.*

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

- Удосконалено метод реалізації AVB-технології у мережі.
- Розроблено вітчизняний продукт реалізації AVB-технології у мережі, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

## 7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

### 7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи AVB-технології у мережі можуть зацікавити кілька категорій фахівців і організацій (рисунок 7.1).

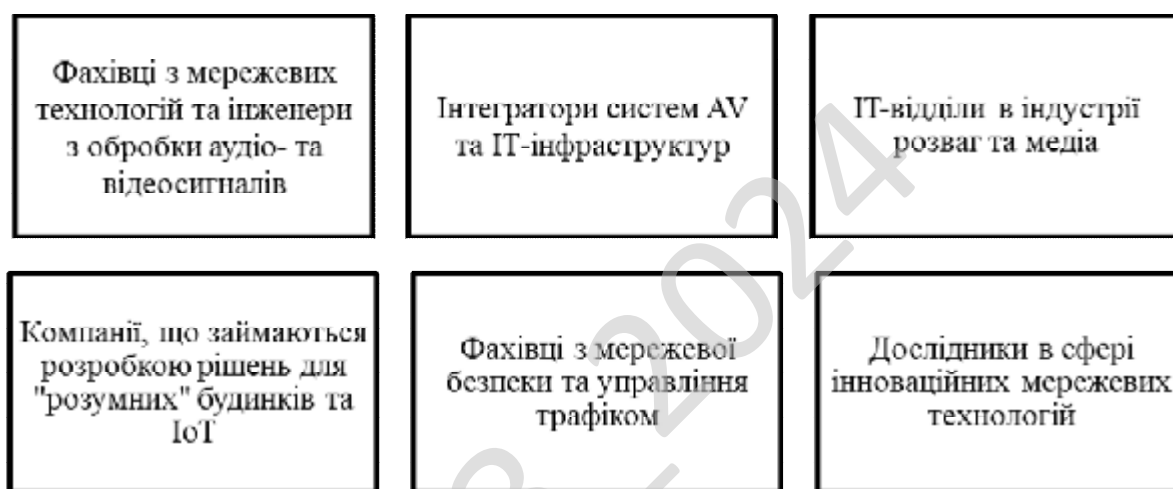


Рисунок 7.1 – Цільова аудиторія

Такий спектр потенційної аудиторії свідчить про багатогранність застосувань AVB-технології, яка може стати корисним доповненням до сучасних рішень у сфері передачі аудіо- та відеоданих.

### 7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи AVB-технології у мережі за допомогою методів експертних оцінок передбачає залучення групи експертів з різних сфер (інженерів, менеджерів, фахівців з IT та AV технологій),

які оцінюють потенційну цінність, рентабельність та ефективність впровадження AVB-технології.

Зупиняємось на методі експертної оцінки (метод бальної оцінки (де експерти виставляють бали кожному критерію) або метод ранжування (де експерти оцінюють критерії (рисунок 7.2) за пріоритетністю).

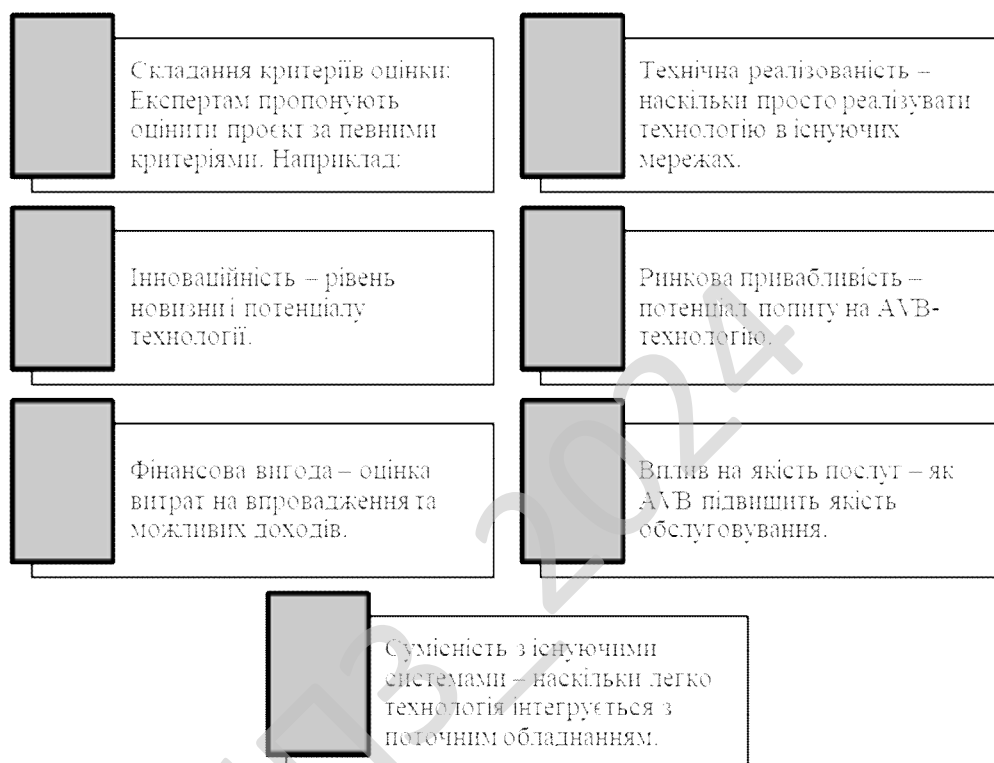


Рисунок 7.2 – Критерії експертної оцінки

Наступним етапом кожен експерт оцінює проєкт за 5-бальною шкалою (1 – дуже низька привабливість, 5 – дуже висока привабливість) для кожного критерію. Оцінка експертів оформлюється в таблиці 7.1.

Після оцінки експертами середній бал за кожним критерієм дозволяє визначити сильні та слабкі сторони AVB-технології. Наприклад, високий бал за критерієм «Вплив на якість послуг» (5.0) свідчить про суттєве покращення якості передачі даних, тоді як бал 3.7 за «Фінансову вигоду» може вказувати на значні початкові інвестиції, що можуть відлякувати потенційних інвесторів.



Суть методу експертної оцінки - злучаються експерти, які на основі свого досвіду дають оцінку вартості проєкту. Метод може дати хорошу приблизну оцінку навіть без детальних даних, але при цьому суб'єктивний і точність якого залежить від кваліфікації експертів. Застосовується на ранніх етапах проєкту або коли немає доступу до детальних даних для розрахунків. Підходить для первинної оцінки, щоб визначити порядок вартості, перш ніж розпочинати детальний аналіз.

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

Економічна ефективність від впровадження системи реалізації AVB-технології у мережі може бути оцінена на основі різних економічних вигод і оптимізації, які AVB-технологія приносить підприємству. Пропонуємо варіант розрахунку ефективності та обґрунтування економічні вигоди такого впровадження (табл. 7.2).

Таблиця 7.2 – Основні показники впровадження проєкту

Обсяг інвестицій у впровадження AVB-технології: \$100,000.

Плановий термін окупності: 3 роки.

Зниження операційних витрат за рахунок оптимізації використання мережі та зменшення потреб у додатковому обладнанні для передачі аудіо- та відеоконтенту: \$20,000 на рік.

Покращення якості обслуговування клієнтів (наприклад, стабільність передачі сигналу, менша затримка), що дозволяє підвищити лояльність клієнтів та залучити нових. Це може привести до збільшення доходів на 10% (наприклад, на \$30,000 щороку).

Зниження витрат на ремонт і підтримку (наприклад, на обслуговування звичайного AV-обладнання): \$5,000 на рік.

## Продовження таблиці 7.2

Розрахунок економічної ефективності:

Загальна річна економія:

Зниження операційних витрат: \$20,000.

Збільшення доходів від підвищення якості: \$30,000.

Зниження витрат на ремонт і підтримку: \$5,000.

Загальна економія за рік = \$20,000 + \$30,000 + \$5,000 = \$55,000.

Розрахунок чистого доходу (Net Income):

Чистий дохід за рік = Загальна економія за рік – Обсяг інвестицій / Плановий термін окупності.

Чистий дохід за рік = \$55,000 – (\$100,000 / 3) = \$55,000 – \$33,333 ≈ \$21,667.

Коефіцієнт економічної ефективності (ROI):

$ROI = (\text{Сума чистого доходу за рік} / \text{Обсяг інвестицій}) \times 100\%$ .

$ROI = (\$21,667 / \$100,000) \times 100\% \approx 21.67\%$ .

Термін окупності:

Обсяг інвестицій / Загальна річна економія = \$100,000 / \$55,000 ≈ 1.82 роки.

Інвестиції в AVB-технологію окупляться приблизно за 1.82 роки, а щорічний чистий дохід складе близько \$21,667. Коефіцієнт економічної ефективності (ROI) – 21.67%, що свідчить про вигідність такого проекту. Економічна ефективність досягається за рахунок зменшення витрат, підвищення якості та збільшення доходів від нових клієнтів.

## 7.5 Пропозиція алгоритму просування проекту розробки ПЗ

Просування проекту програмної реалізації системи AVB-технології у мережі потребує комплексного підходу, що включає маркетингові, технічні та комунікаційні заходи. Пропонуємо алгоритм процесу просування такого проекту (рис. 7.3).

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

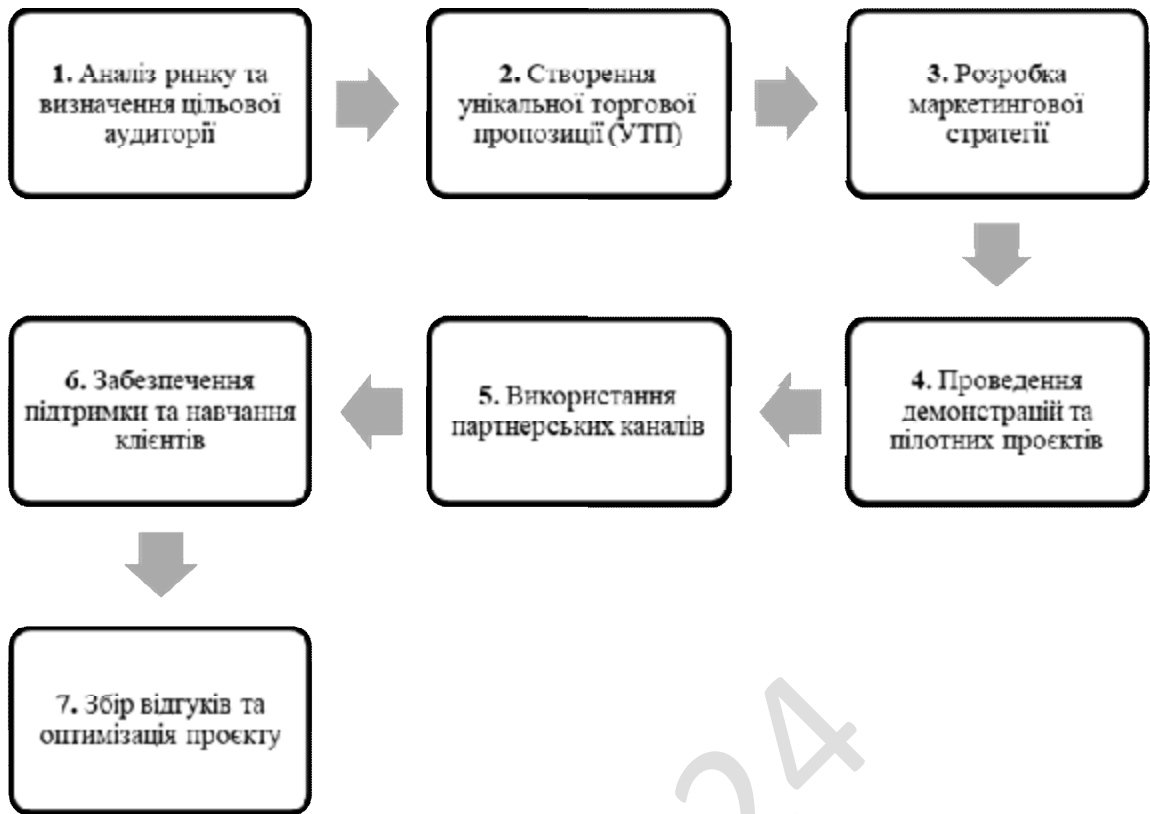


Рисунок 7.3 – Алгоритм просування проєкту

Такий алгоритм просування дозволить поступово залучити цільову аудиторію, показати переваги AVB-технології та підвищити впізнаваність проєкту на ринку. Завдяки демонстраціям, партнерським програмам та зворотному зв'язку, компанія зможе вдосконалювати продукт і задовольняти потреби клієнтів, що сприятиме сталому зростанню та успішному впровадженню AVB-рішень.

### 7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту та шляхів реалізації проєкту з AVB-технологією пропонуємо скористатись пропозиціями зведеними в рисунок 7.4.

- Оцінка існуючих каналів:** проаналізувати ефективність поточних каналів збуту за критеріями, такими як досяжність аудиторії, витрати, конверсія. Виявити найбільш рентабельні канали та оптимізувати витрати на менш ефективні.
- Сегментація ринку:** поділити клієнтів на групи (корпорації, медіакорпорації, освітні установи), щоб адаптувати канали збуту для кожного сегмента та знайти оптимальні шляхи до них.
- Прямий продаж та онлайн-канали:** створити збалансовану комбінацію прямих продажів через власний сайт, онлайн-платформи та партнерські маркетингові канали.
- Вебінари та онлайн-презентації:** запровадити онлайн-демонстрації можливостей AVB-систем, які можна просувати через соціальні мережі, розсилки та партнерські канали для залучення нових клієнтів.
- Співпраця з інтеграторами AV-обладнання:** знайти партнерів серед компаній, які вже постачають AV-рішення, для сумісного продажу програмного забезпечення з їхнім обладнанням.
- Дистрибуторські угоди:** розширити доступ до регіональних ринків через дистрибуторів або агентів, які мають досвід роботи в AV-сфері.
- CRM система для управління клієнтами:** впровадити CRM для відстеження циклу продажів, обробки лідів та персоналізації комунікацій з клієнтами.
- Автоматизація маркетингових процесів:** використовувати інструменти автоматизації для розсилок, реклами та обробки звернень, щоб знизити витрати часу на рутинні процеси.
- Післяпродажне обслуговування:** створити відділ технічної підтримки для забезпечення швидкого реагування на питання клієнтів та підтримання їх лояльності.
- Навчальні програми та підтримка:** забезпечити доступ до навчальних матеріалів і консультацій для клієнтів та партнерів, що допоможе їм краще зрозуміти та використовувати технологію.
- Локалізація продукту:** адаптувати програмне забезпечення та маркетингові матеріали до мовних та культурних особливостей регіонів, де планується просування.
- Аналіз місцевих регуляцій:** забезпечити відповідність продукту місцевим стандартам і вимогам, щоб спростити процес виходу на нові ринки.
- Моніторинг ключових показників:** регулярно аналізувати показники ефективності кожного каналу збуту та оптимізувати інвестиції в ті, що приносять найкращі результати.
- Тестування нових каналів:** запускати експерименти з новими каналами збуту, зокрема контент-маркетингом або відеомаркетингом, для оцінки їх потенціалу.

Рисунок 7.5 – Ключові фактори успіху проекту

Такі заходи дозволять підвищити ефективність збуту, адаптувати стратегії до ринкових умов та залучити більше клієнтів для реалізації AVB-технології.

### 7.7 Визначення ключових факторів успіху конкретного проєкту

Для успішної реалізації проєкту AVB-технології у мережі існує ряд ключових факторів (рисунок 7.6).

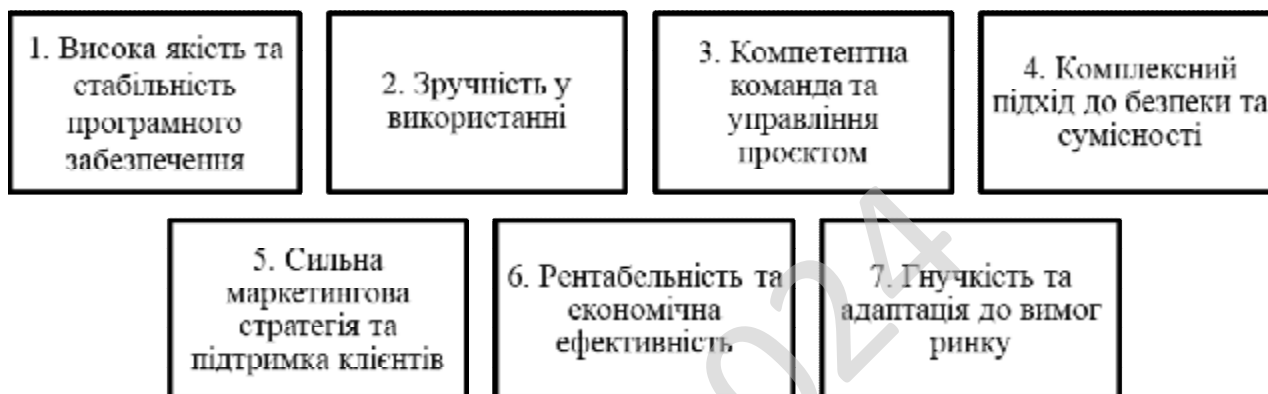


Рисунок 7.6 – Ключові фактори успіху проєкту

Ці фактори дозволять створити стійке та конкурентоспроможне рішення, яке відповідатиме високим вимогам користувачів і ринку, що сприятиме успіху проєкту AVB-технології в мережі.

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Незважаючи на стрімке розповсюдження комп'ютерних мереж, з'ємні носії інформації не втрачають актуальності у ряді сфер професійної діяльності.

Стрімкий розвиток галузі інформаційних технологій (ІТ) не тільки у сфері управління виробництвом, в банківській системі, бізнесі, системі освіти, але також на транспорті, сфері обслуговування призвів до того, що десятки мільйонів людей у всьому світі виявились втягнутими у взаємодію людини з комп'ютером. Природно виникає запитання: настільки безпечною є ця взаємодія для людини? Адже відома аксіома про те, що будь-яка взаємодія людини та засобів праці двостороння.

Впровадження комп'ютерних технологій принципово змінило характер праці різних категорій фахівців. Працівники, використовують комп'ютерну техніку, на своєму досвіді оцінили її величезні можливості. Одночасно виникла певна безтурботність при її експлуатації.

Недотримання вимог безпеки призводить до того, що й через кілька днів роботи за комп'ютером співробітник починає відчувати певний дискомфорт: в нього виникає головний біль і різь у власних очах, з'являються почуття виснаження й дратівливості. В окремих людей порушується сон, погіршується зір, занедужують руки, шия, попереки тощо.

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

- недостатню площу і обсяг виробничого приміщення;
- недотримання вимог, мікроклімату на робочих місцях;
- низький рівень освітленості у приміщеннях і на робочих поверхнях апаратури;

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

- підвищений рівень низькочастотних магнітних полів від моніторів;
- порушення вимог організації робочих місць;
- недотримання вимог до режимам праці та відпочинку;
- надмірне виробничу навантаження працівників;
- відсутність навичок зниження впливу психоемоційного напруги.

Відповідно до ст.14 Закону «Про охорони праці» [1] на роботодавця покладено обов'язок забезпечити: безпеку працівників при експлуатації устаткування; застосування коштів індивідуальної захисту працівників; відповідні вимоги охорони праці, умови праці в кожному робоче місце; дотримання режиму праці та відпочинку працівників; навчання безпечним методам і прийомам виконання; інструктаж з охорони праці; організацію контролю над станом умов праці в робочих місць; проведення атестації робочих місць в умовах праці.

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

## 8.2 Аналіз умов праці

Приміщення розташовано на третьому поверсі п'ятиповерхового будинку. У приміщенні розташовано 3 робочих місць з комп'ютерами (далі ПК). Відповідно до норм «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2] площа, що відводиться для робочого місця з комп'ютером повинна бути не менше 6 м<sup>2</sup>, об'єм не менше 20 м<sup>3</sup>. Розміри даного приміщень складають: довжина – 6 м, ширина – 4,5 м, висота – 3,5 м, тобто загальна фактична площа складає 27 м<sup>2</sup>. Необхідна площа на 3 робочих місця із установленими ПК складає 18 м<sup>2</sup>, що не перевищує фактичну. Обсяг кабінету на одного працюючого складає 31,5м<sup>3</sup>, отже відповідає нормі ДСанПіН 3.3.2-007-98 – не менше 20 м<sup>3</sup> [2].

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

При роботі з ПК людина може піддатися впливу шкідливих та небезпечних факторів. Під шкідливими виробничими факторами розуміють фактори, тривалий вплив яких викликає розвиток професійних захворювань. Небезпечні виробничі фактори – вплив яких на працюючого викликає травму, тобто пошкодження організму. Шкідливі і небезпечні чинники, з якими стикається бібліограф при роботі з ПК, приведені в таблиці 8.1.

Таблиця 8.1 – Перелік шкідливих та небезпечних виробничих факторів

Найменування факторів	Можливі джерела їх виникнення	Характер дії
Небезпека ураження електричним струмом	Мережа живлення	Небезпечний
Пожежонебезпечність приміщень	Наявність матеріалів, що згорають і джерел запалення (електроапаратура)	Небезпечний та шкідливий
Іонізація повітря	Статична електрика випромінювання	Шкідливий
Підвищений рівень шуму	Шум створюється перетворювачем напруги ЕОМ, її технічною периферією, а також людьми, що працюють в приміщенні	Шкідливий
Несприятлива освітленість	Недостатнє штучне і природне освітлення	Шкідливий
Незадовільні параметри мікроклімату	Незадовільний стан системи опалення і вентиляції	Шкідливий
Психофізіологічні напруження	Монотонність праці, перенапруженість зорових аналізаторів, розумова напруженість, незручність і статичність пози	Шкідливий

По категорії вибухо- і пожежонебезпеки, згідно дане приміщення відноситься до категорії В – пожежонебезпечне, тому що присутні тверді матеріали, що горять, такі як дерев'яні столи, папір і інше. Виходячи з категорії

пожежонебезпеки і поверховості будинку, ступінь вогнестійкості будівлі II. Згідно з ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» [13] ЕОМ повинні розташовуватись в будівлі не менше ніж II ступню вогнестійкості.

По ступені небезпеки поразки людей електричним струмом відділ, згідно, класифікується як приміщення з підвищеною небезпекою, тому що не виключена можливість одночасного дотику людини до маючих з'єднання з землею конструкціям будинку, з одного боку, і до металевих корпусів електроустаткування, що можуть виявити під напругою – з іншого.

Для забезпечення вищевказаних оптимальних метеорологічних умов у помешканні передбачена система опалення (загальне парове) в холодному періоді, та вентиляція і кондиціонування в теплий період року, згідно ДБН2.5–67–2013 «Опалення, вентиляція та кондиціонування» [4]. При виконанні замірів параметрів мікроклімату, значення їх відповідали оптимальним та допустимим параметрам відповідно до ДСанПіН3.3.2.007–98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно – обчислювальних машин» [2].

Припустимий рівень іонізації повітря помешкання відповідно до СН 21.52-80 повинен складати 1500 – 3000 один./м<sup>3</sup>.

Нормування освітлення здійснюється відповідно до ДБН В.2.5 – 28 – 2006 «Природне та штучне освітлення». [5]

Відділ забезпечений комбінованим освітленням. В темний час доби передбачається загальне і/або місцеве рівномірне штучне, а в світлий – бокове одностороннє природне освітлення два віконних прорізи.

### **8.3 Техніка безпеки та протипожежна профілактика**

Відповідно ДБН В 1.1-7-2016 «Пожежна безпека об'єктів будівництва» [13] будинок можна віднести до II групи по ступені вогнестійкості й до категорії Д по ступені пожежонебезпеки.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Від розподільного щита по праву й ліву сторони встановлені кондиціонери, зовнішня електропроводка, поміщена в ізолюваний кабель. Висота проводки становить 2,2 м від рівня підлоги, її кріплення здійснюється за допомогою металевих власників. Біля кожного стола організований розподільний щит, розташований на текстолітовій пластинці, закріпленої на стіні на рівні 1 м від підлоги. Усього до складу входять п'ять розеток і дві клеми заземлення. Всі обчислювальні машини з'єднані із клемми заземлення. Чотири з п'яти розеток забезпечують подачу напруги 220 V, а одна, забезпечує подачу напруги в 36 В. Про це є відповідні написи на кожному розподільному щиті.

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

- евакуаційних шляхів;
- так званих «курилок»;
- місць складування продукції та сировини;
- стоянки транспорту.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Також встановлюється порядок роботи та технічного обслуговування:

- вентиляційного устаткування;
- засобів пожежогашіння і захисту від загорянь;
- нагрівальних приладів;
- електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними викладачами.

Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електро-установки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об'єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань.

Відповідно НПАОП 40.1-1.21-98 “Правил безпечної експлуатації електроустановок споживачів” [6], приміщення можна віднести до приміщень без підвищеної небезпеки, оскільки це приміщення, сухе, з нормальною температурою й ізолюючими підлогами, що не має заземлених металоконструкцій.

Персональні ЕОМ можна віднести до першого класу електротехнічних виробів по способі захисту людини від поразки електричним струмом, оскільки

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

їхні корпуси зроблені з ізолюючої пластмаси й кожен пристрій має заземлення. Відповідно правилам пристрою електроустановок ЕОМ можна віднести до електроустановок з робочою напругою до 1000 В.

Однієї з достовірних причин пожежі в приміщенні з обчислювальною технікою може бути коротке замикання, що спричиняє спалах електропроводки. Для його попередження вся обчислювальна техніка, а також інші електричні пристрої повинні бути обладнані плавкими запобіжниками, а на вході електромережі повинен бути передбачений автомат захисту. Не слід користуватися електричними подовжувачами й трійниками, що не мають сертифікатів відповідності вимогам безпеки.

Необхідно передбачити наявність у межах досяжності первинних засобів гасіння пожежі (вогнегасників) для локалізації вогню власними засобами до приїзду команди пожежної охорони. Повинен бути розроблений план екстреної евакуації персоналу при виникненні загоряння. Кількість евакуаційних виходів повинне бути не менш двох. Допускається використання одного евакуаційного виходу, якщо відстань найбільш віддаленого робочого місця до цього виходу не перевищує 25 м.

#### 8.4 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток 63х63х6 мм., (згідно з ДСТУ 2251-93 «Кутики сталеві гарячекатані рівнополічні. Сортамент») довжиною  $L=1,6$  м, та горизонтальний електрод – металева полоса з перетином 60х5 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – по контуру прямокутником (рис. 8.1).

Розрахунок проведемо за допустимим опором розтіканню струму заземлювача.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунту – чорнозем, нижнього шару ґрунту – глина (питомий опір  $\rho_2 = 40 \text{ Ом}\cdot\text{м}$ ). Умовна товщина верхнього шару ґрунту:  $H=0,55 \text{ м}$ . Відстань між вертикальними заземлювачами (електродами)  $A=3 \text{ м}$ . Глибина закладення горизонтального контура заземлення  $t=0,6 \text{ м}$ . Опір заземлювача, який нормується:  $R_{3Н} = 4 \text{ Ом}$ . Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

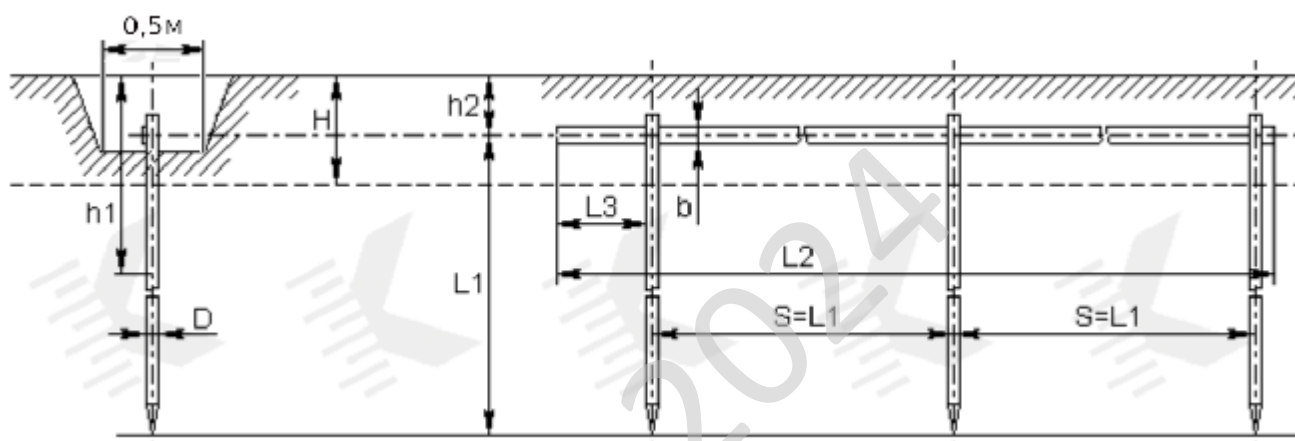


Рисунок 8.1 – Схема штучного заземлення

Виконаємо розрахунок.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t + L/2=0,6 + 1,6/2 = 1,4 \text{ м.}$$

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

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

де  $\psi = 1,36$  – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багат шаровому ґрунті [12];

$\rho_2 = 40 \text{ Ом}\cdot\text{м}$  – табличне значення питомого опору нижнього шару ґрунту (глина) [12].

Еквівалентний діаметр вертикального електрода (кутка) [12]:

$$D_B = 0,95 \cdot K = 0,95 \cdot 63 = 59,85 \text{ мм} = 0,0598 \text{ м},$$

де  $K = 63 \text{ мм}$  – розмір металевого кутка (заданий).

Відношення  $A/L = 3/1,6 = 1,87$

Опір розтіканню електричного струму одного електрода вертикального заземлювача з урахуванням заглиблення заземлювача [12]:

$$R_0 = 0,366 \cdot (\rho/L) \cdot [\lg(2L/D_B) + (1/2) \cdot \lg((4T+L)/(4T-L))] = \\ = 0,366 \cdot (54,5/1,6) \cdot [(\lg(2 \cdot 1,6/0,0598) + (1/2) \cdot \lg((4 \cdot 1,4 + 1,6)/(4 \cdot 1,4 - 1,6)))] = 23 \text{ Ом}.$$

Визначаємо коефіцієнт екранування вертикальних електродів  $K_{\text{ев}} = 0,62$  при орієнтовній кількості вертикальних електродів, яке дорівнює 5 [12].

Визначаємо необхідну кількість вертикальних електродів заземлювача (без врахування горизонтального заземлювача), при  $R_{3Н} = 4 \text{ Ом}$ :

$$N = R_0 / (K_{\text{ев}} R_{3Н}) = 23 / (0,62 \cdot 4) = 9,3 \approx 9 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси:

$$L_{\Pi} = 1,05 \cdot A \cdot N = 1,05 \cdot 3 \cdot 9 = 28,35 \approx 28 \text{ м}.$$

Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору ґрунту  $K_{\Pi}$  [12]:

$$R_{\Pi} = 0,366 \cdot (\rho \cdot K_{\Pi} / L_{\Pi}) \cdot \lg(2 \cdot L_{\Pi}^2 / (B \cdot t)) = \\ = 0,366 \cdot (40 \cdot 5 / 28) \cdot \lg((2 \cdot 28^2) / (0,06 \cdot 0,6)) = 12,2 \text{ Ом}.$$

де  $K_{\Pi} = 5$  – табличне значення кліматичного коефіцієнта питомого опору ґрунту для відповідної кліматичної зони для з'єднуючої полоси [12]:

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

Загальний опір розтіканню електричного струму заземлювача [12]:

$$R = (R_0 \cdot R_{\Pi}) / (R_0 \cdot \eta_{\Pi} + N \cdot R_{\Pi} \cdot K_{\text{ев}}) = \\ = (23 \cdot 12,2) / (23 \cdot 0,6 + 9 \cdot 12,2 \cdot 0,62) = 3,48 \text{ Ом}.$$

де  $\eta_{\Pi} = 0,6$  – табличне значення коефіцієнта екранування з'єднуючої полоси [12].

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

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Оскільки при 9 вертикальних електродах  $R$  суттєво менше  $R_{3H}$ , зменшимо кількість вертикальних електродів  $N$  до 8 і виконаємо перерахунок. У результаті остаточно отримали:  $R=3,91$  Ом при кількості вертикальних електродів  $N=8$ .

### 8.5 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 приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів з питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи реалізації AVB-технології 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 реалізації AVB-технології у мережі.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем реалізації AVB-технології у мережі.
- Досліджена система реалізації AVB-технології у мережі.
- На основі отриманих результатів досліджень створена програмна реалізація системи реалізації AVB-технології у мережі.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання реалізації AVB-технології 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.

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.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

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 Threefish.

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

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

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дудзінський А.М. Дослідження та програмна реалізація системи реалізації AVB-технології у мережі // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB® With Applications to Pattern Recognition. CRC Press. 2021. 406 p.
3. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
4. Peter Shirley, Steve Marschner. Fundamentals of Computer Graphics. 2009
5. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
6. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
7. Інженерна комп'ютерна графіка: підручник / В.В. Проців [та ін.] / М-во освіти і науки України, Нац. гірн. унт-т. – Дніпро: НГУ, 2017. – 247 с.
8. Проців В.В. Прикладна комп'ютерна графіка [Текст]: Навч. посібник / В.В. Проців, К.А. Зіборов, К.М. Бас, Г.К. Ванжа; М-во освіти і наук, Нац. гірн. унт. – Д.: НГУ, 2016. – 187 с.
9. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
10. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
11. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

					ВКРМ-123.24.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78



Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

20. 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.

21. 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.

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

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

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

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

35. Smirnov, O., 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

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

42. 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.

43. 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.

44. 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.

45. 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.

46. 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.

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

48. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» *Комп'ютерні науки та кібербезпека*. № 4. С. 30-37. 2019.

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

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

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

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

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

					<b>ВКРМ-123.24.0010.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Додаток А  
(обов'язковий)

Технічне завдання

Зміст

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

					<b>ВКРМ-123.24.0010.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Дудзінський А.М.				<i>Дослідження та програмна реалізація системи реалізації AVB- технології у мережі</i>	Літ.	Аркуш	Аркушів
Перевірів	Петренко В.І.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М			
Затв.	Смірнов О.А.							

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи реалізації AVB-технології 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. № 19-13 від 07.08.2024 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи реалізації AVB-технології 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;

					ВКРМ-123.24.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи реалізації AVB-технології 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.

					<b>ВКРМ-123.24.0010.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

## 5.7 Вимоги до складу та параметрів технічних засобів

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

## 5.8 Вимоги до інформаційної і програмної сумісності

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

### 5.8.1 Обладнання

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

### 5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

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

## 7 Економічні вимоги

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

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

## 8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута техніка безпеки та протипожежна профілактика.

					ВКРМ-123.24.0010.00.00.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 – 84 аркуші.

## 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 18.12.2024 р.

					<b>ВКРМ-123.24.0010.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

**ЗАТВЕРДЖУЮ**

Керівник випускної кваліфікаційної роботи за  
другим (магістерським) рівнем вищої освіти

\_\_\_\_\_ Петренко В.І.

*Дослідження та програмна реалізація  
системи реалізації AVB-технології у мережі*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 19

Літера: РП

### Основна програма

```

import socket
import time
import threading

class AVBSystem:
    def __init__(self, network_interface):
        self.network_interface = network_interface
        self.srp_module = SRPModule()
        self.traffic_module = TrafficShapingModule()
        self.sync_module = SynchronizationModule()
        self.streams = []

    def configure_network(self):
        self.sync_module.synchronize_clocks()
        self.traffic_module.setup_qos(self.network_interface)
        self.srp_module.setup_streams()

    def add_stream(self, stream_data):
        self.streams.append(stream_data)
        self.srp_module.reserve_resources(stream_data)

    def start_streaming(self):
        for stream in self.streams:
            self.traffic_module.start_transmission(stream)

class SynchronizationModule:
    def __init__(self):
        self.sync_interval = 5 # Інтервал синхронізації у секундах

    def synchronize_clocks(self):
        # Імітація синхронізації годинників у мережі
        threading.Thread(target=self.sync_process).start()

    def sync_process(self):
        while True:
            # Імітація обміну пакетами для синхронізації
            print("Synchronizing clocks...")
            time.sleep(self.sync_interval)

class TrafficShapingModule:
    def __init__(self):
        self.qos_enabled = False

    def setup_qos(self, interface):
        # Налаштовується якість обслуговування (QoS) для інтерфейсу
        self.qos_enabled = True
        print(f"QoS налаштовано на інтерфейсі {interface}")

    def start_transmission(self, stream_data):
        # Імітація передачі даних з урахуванням QoS
        threading.Thread(target=self.transmission_process,
            args=(stream_data,)).start()

    def transmission_process(self, stream_data):
        # Імітація передачі потоку з затримкою та пріоритетом
        print(f"Передача потоку {stream_data['id']} почалася")
        while True:
            # Імітація передачі пакету даних
            time.sleep(1)
            print(f"Пакет відправлено для потоку {stream_data['id']}")

class SRPModule:
    def __init__(self):

```

```

self.stream_registry = []

def setup_streams(self):
    # Ініціалізація системи з потоками
    print("Ініціалізація потоків SRP")

def reserve_resources(self, stream_data):
    # Резервування ресурсів для потоку
    self.stream_registry.append(stream_data)
    print(f"Ресурси зарезервовано для потоку {stream_data['id']}")

class MonitoringModule:
    def __init__(self, avb_system):
        self.avb_system = avb_system

    def monitor_network(self):
        # Моніторинг мережі для виявлення проблем
        threading.Thread(target=self.monitor_process).start()

    def monitor_process(self):
        while True:
            # Імітація моніторингу трафіку
            time.sleep(3)
            print("Моніторинг мережі...")

class LoggerModule:
    def log_event(self, message):
        # Логування подій системи
        print(f"[LOG]: {message}")

class ErrorHandlingModule:
    def handle_error(self, error_message):
        # Обробка помилок
        print(f"[ERROR]: {error_message}")

class AVBStream:
    def __init__(self, stream_id, stream_type, bandwidth, priority):
        self.stream_id = stream_id
        self.stream_type = stream_type
        self.bandwidth = bandwidth
        self.priority = priority

    def get_details(self):
        # Повернення деталей про потік
        return {
            "id": self.stream_id,
            "type": self.stream_type,
            "bandwidth": self.bandwidth,
            "priority": self.priority
        }

# Створення об'єкта системи AVB
avb_system = AVBSystem("eth0")

# Налаштування мережі
avb_system.configure_network()

# Додавання потоків
audio_stream = AVBStream(1, "audio", 512, 1)
video_stream = AVBStream(2, "video", 10240, 2)

avb_system.add_stream(audio_stream.get_details())
avb_system.add_stream(video_stream.get_details())

# Початок передачі потоків
avb_system.start_streaming()

# Моніторинг та логування
monitor = MonitoringModule(avb_system)

```

```
monitor.monitor_network()

logger = LoggerModule()
logger.log_event("Система запущена успішно")

# Модуль обробки помилок
error_handler = ErrorHandlingModule()
error_handler.handle_error("Помилка під час резервування ресурсів")
```

КБПЗ\_2024

## Файл network\_interface\_support.py

```
import psutil

class NetworkInterfaceSupport:
    def __init__(self):
        self.interfaces = self.get_network_interfaces()

    def get_network_interfaces(self):
        # Отримуємо список мережєвих інтерфейсів з системи
        return psutil.net_if_addrs()

    def configure_interface(self, interface_name):
        # Налаштування інтерфейсу
        if interface_name in self.interfaces:
            print(f"Налаштування інтерфейсу: {interface_name}")
            # Тут можна додати конфігурацію QoS, IP-адресу, тощо
        else:
            print(f"Інтерфейс {interface_name} не знайдено")

    def list_interfaces(self):
        # Повернення списку доступних інтерфейсів
        return [iface for iface in self.interfaces]
```

## Файл dynamic\_qos.py

```
import threading
import time

class DynamicQoS:
    def __init__(self, avb_system):
        self.avb_system = avb_system
        self.adjustment_interval = 10
        self.qos_levels = {}

    def start_adjustment(self):
        # Запуск динамічного регулювання QoS
        threading.Thread(target=self.adjust_qos_process).start()

    def adjust_qos_process(self):
        while True:
            for interface in self.avb_system.network_interfaces:
                # Перевіряємо навантаження на кожному інтерфейсі
                load = self.check_interface_load(interface)
                self.adjust_qos_level(interface, load)
                time.sleep(self.adjustment_interval)

    def check_interface_load(self, interface):
        # Імітація перевірки навантаження на інтерфейсі
        print(f"Перевірка навантаження на інтерфейсі {interface}")
        return 50 # Імітоване значення навантаження

    def adjust_qos_level(self, interface, load):
        # Регулювання рівня QoS залежно від навантаження
        if load > 70:
            print(f"Підвищення QoS для інтерфейсу {interface}")
        else:
            print(f"Залишаємо поточний рівень QoS для інтерфейсу {interface}")
```

## Файл security\_module.py

```
from cryptography.fernet import Fernet

class SecurityModule:
    def __init__(self):
        self.key = Fernet.generate_key()
        self.cipher = Fernet(self.key)

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

    def decrypt_data(self, encrypted_data):
        # Дешифрування отриманих даних
        decrypted = self.cipher.decrypt(encrypted_data).decode()
        return decrypted

    def secure_transmission(self, stream_data):
        encrypted = self.encrypt_data(stream_data)
        print(f"Передача зашифрованих даних: {encrypted}")
```

КБПЗ\_2024

## Файл traffic\_monitoring.py

```
import matplotlib.pyplot as plt
import time
import threading

class TrafficMonitoring:
    def __init__(self, avb_system):
        self.avb_system = avb_system
        self.traffic_data = []

    def start_monitoring(self):
        # Запуск моніторингу трафіку
        threading.Thread(target=self.monitor_traffic).start()

    def monitor_traffic(self):
        while True:
            # Імітація збору даних про трафік
            traffic_load = self.get_traffic_load()
            self.traffic_data.append(traffic_load)
            print(f"Поточне навантаження: {traffic_load} Mbps")
            time.sleep(5)

    def get_traffic_load(self):
        # Імітація отримання навантаження
        return 100 # Імітоване значення навантаження

    def visualize_traffic(self):
        # Візуалізація зібраних даних
        plt.plot(self.traffic_data)
        plt.title("Мережеве навантаження")
        plt.xlabel("Час (секунди)")
        plt.ylabel("Навантаження (Mbps)")
        plt.show()
```

## Файл auto\_recovery.py

```
import threading
import time

class AutoRecovery:
    def __init__(self, avb_system):
        self.avb_system = avb_system

    def start_recovery_process(self):
        # Запуск процесу автоматичного відновлення
        threading.Thread(target=self.recovery_check).start()

    def recovery_check(self):
        while True:
            # Імітація перевірки на наявність збоїв
            if self.detect_failure():
                self.recover_system()
                time.sleep(10)

    def detect_failure(self):
        # Імітація виявлення збою
        print("Перевірка на наявність збоїв...")
        return False

    def recover_system(self):
        # Процес відновлення системи
        print("Відновлення системи після збою...")
        self.avb_system.configure_network()
```

## Файл automation\_integration.py

```
class AutomationIntegration:
    def __init__(self):
        self.connected_systems = []

    def connect_to_system(self, system_name):
        # Підключення до іншої системи автоматизації
        print(f"Підключення до системи {system_name}")
        self.connected_systems.append(system_name)

    def send_control_signal(self, system_name, command):
        if system_name in self.connected_systems:
            print(f"Відправка команди '{command}' до {system_name}")
        else:
            print(f"Система {system_name} не підключена")
```

КБПЗ\_2024

**Файл stream\_support.py**

```

class StreamSupport:
    def __init__(self):
        self.supported_streams = ["audio", "video", "text"]

    def add_stream_type(self, stream_type):
        if stream_type not in self.supported_streams:
            self.supported_streams.append(stream_type)
            print(f"Додано новий тип потоку: {stream_type}")

    def list_supported_streams(self):
        return self.supported_streams

```

**Файл priority\_management.py**

```

class PriorityManagement:
    def __init__(self, avb_system):
        self.avb_system = avb_system

    def adjust_priority(self, stream_id, new_priority):
        for stream in self.avb_system.streams:
            if stream['id'] == stream_id:
                stream['priority'] = new_priority
                print(f"Пріоритет для потоку {stream_id} змінено на {new_priority}")

```

**Файл api\_integration.py**

```

from flask import Flask, request, jsonify

app = Flask(__name__)

@app.route('/add_stream', methods=['POST'])
def add_stream():
    data = request.get_json()
    # Додавання потоку
    print(f"Додано потік: {data}")
    return jsonify({"status": "success"})

def start_api():
    app.run(host='0.0.0.0', port=5000)

```

**Файл performance\_analysis.py**

```

class PerformanceAnalysis:
    def __init__(self, avb_system):
        self.avb_system = avb_system

    def analyze_performance(self):
        # Аналіз продуктивності мережі

```

```
print("Аналіз продуктивності мережі...")
```

```
def optimize_transmission(self):
    # Оптимізація передачі даних
    print("Оптимізація передачі даних...")
```

#### Файл `extended_functionality.py`

```
import threading
import time
import random
from datetime import datetime

class BackupModule:
    def __init__(self):
        self.backup_interval = 60 # Інтервал створення резервних копій у секундах
        self.backup_data = {}

    def start_backup_process(self):
        # Запуск процесу резервного копіювання
        threading.Thread(target=self.backup_process).start()

    def backup_process(self):
        while True:
            # Створення резервної копії
            timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            self.backup_data[timestamp] = self.generate_backup()
            print(f"Резервна копія створена: {timestamp}")
            time.sleep(self.backup_interval)

    def generate_backup(self):
        # Імітація створення резервної копії даних потоку
        return {"status": "backup_complete", "data": random.sample(range(1000),
10)}

class DeviceAutoConfigModule:
    def __init__(self):
        self.devices = {}

    def detect_new_device(self, device_id):
        # Виявлення нового пристрою у мережі
        print(f"Виявлено новий пристрій: {device_id}")
        self.configure_device(device_id)

    def configure_device(self, device_id):
        # Імітація автоматичної конфігурації пристрою
        config = {"IP": f"192.168.1.{random.randint(2, 254)}", "QoS": "medium"}
        self.devices[device_id] = config
        print(f"Пристрій {device_id} сконфігуровано з параметрами: {config}")

class LoggingModule:
    def __init__(self):
```

```

self.logs = []

def log_event(self, message):
    # Логування подій
    log_entry = {"timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
"message": message}
    self.logs.append(log_entry)
    print(f"[LOG]: {log_entry}")

def get_logs(self):
    # Повернення списку логів
    return self.logs

class AdvancedMonitoringModule:
    def __init__(self, avb_system):
        self.avb_system = avb_system
        self.monitor_interval = 15
        self.anomaly_count = 0

    def start_monitoring(self):
        # Запуск розширеного моніторингу мережі
        threading.Thread(target=self.monitor_process).start()

    def monitor_process(self):
        while True:
            print("Моніторинг активності мережі...")
            anomalies = self.detect_anomalies()
            if anomalies:
                print(f"Виявлено аномалії: {anomalies}")
                self.anomaly_count += len(anomalies)
                time.sleep(self.monitor_interval)

    def detect_anomalies(self):
        # Імітація виявлення аномалій у мережі
        anomalies = []
        if random.choice([True, False]):
            anomalies.append("Велика затримка пакету")
        if random.choice([True, False]):
            anomalies.append("Втрата пакету")
        return anomalies

class ResourceReportingModule:
    def __init__(self, avb_system):
        self.avb_system = avb_system

    def generate_report(self):
        # Генерація звіту про використання ресурсів мережі
        report = {
            "total_bandwidth": self.calculate_total_bandwidth(),
            "stream_count": len(self.avb_system.streams),
            "device_count": len(self.avb_system.network_interfaces)
        }
        print(f"Звіт про використання ресурсів: {report}")

```

```

return report

def calculate_total_bandwidth(self):
    # Імітація розрахунку загальної пропускної здатності
    return sum([stream["bandwidth"] for stream in self.avb_system.streams])

class DeviceTrackingModule:
    def __init__(self):
        self.tracked_devices = {}

    def register_device(self, device_id, device_info):
        # Реєстрація пристрою у системі
        self.tracked_devices[device_id] = device_info
        print(f"Пристрій {device_id} зареєстровано з інформацією:
{device_info}")

    def track_devices(self):
        # Імітація відстеження пристроїв у мережі
        for device_id, info in self.tracked_devices.items():
            print(f"Відстеження пристрою {device_id}: {info}")

class NetworkDeviceConfigurationModule:
    def __init__(self):
        self.configurations = {}

    def configure_device(self, device_id, configuration):
        # Налаштування мережевого пристрою
        self.configurations[device_id] = configuration
        print(f"Пристрій {device_id} налаштовано: {configuration}")

    def update_configuration(self, device_id, new_config):
        # Оновлення конфігурації пристрою
        if device_id in self.configurations:
            self.configurations[device_id].update(new_config)
            print(f"Конфігурація для {device_id} оновлена:
{self.configurations[device_id]}")

class LoadBalancingModule:
    def __init__(self, avb_system):
        self.avb_system = avb_system

    def balance_load(self):
        # Балансування навантаження між інтерфейсами
        for interface in self.avb_system.network_interfaces:
            load = self.get_interface_load(interface)
            if load > 80:
                print(f"Перерозподіл навантаження для інтерфейсу {interface}")

    def get_interface_load(self, interface):
        # Імітація отримання навантаження на інтерфейс
        return random.randint(50, 100)

class MulticastSupportModule:

```

```
def enable_multicast(self, interface):
    # Вмикання підтримки мультикасту на інтерфейсі
    print(f"Мультикаст підтримка увімкнена на інтерфейсі {interface}")

def configure_multicast_group(self, group_ip):
    # Налаштування мультикаст групи
    print(f"Конфігурація мультикаст групи: {group_ip}")

class AlertingSystem:
    def __init__(self):
        self.alert_threshold = 90 # Поріг навантаження у відсотках

    def check_network_load(self, load):
        # Перевірка навантаження та виведення попередження
        if load > self.alert_threshold:
            self.send_alert(f"Навантаження у мережі перевищує
{self.alert_threshold}%")

    def send_alert(self, message):
        # Надсилання попередження
        print(f"[ALERT]: {message}")

class AutoConfigurationModule:
    def auto_update_config(self, device_id, configuration):
        # Автоматичне оновлення конфігурації пристрою
        print(f"Автоматичне оновлення конфігурації для пристрою {device_id}:
{configuration}")

class LoadTestingModule:
    def run_tests(self):
        # Запуск тестування навантаження у мережі
        print("Запуск тестування навантаження...")
        results = self.perform_load_tests()
        print(f"Результати тестування: {results}")

    def perform_load_tests(self):
        # Імітація тестування навантаження
        return {"status": "completed", "average_load": random.randint(60, 95)}

class AnomalyDetectionModule:
    def detect_and_handle_anomalies(self):
        # Виявлення та обробка аномалій у мережі
        anomalies = ["Втрата пакету", "Перевищення затримки"]
        print(f"Виявлено аномалії: {anomalies}")
        for anomaly in anomalies:
            self.handle_anomaly(anomaly)

    def handle_anomaly(self, anomaly):
        # Обробка аномалії
        print(f"Обробка аномалії: {anomaly}")
```

```
import threading
import time
import random
from datetime import datetime

class AnomalyDetectionModule:
    def __init__(self):
        self.anomaly_log = []

    def detect_anomalies(self):
        # Виявлення аномалій у мережі
        anomaly_detected = random.choice([True, False])
        if anomaly_detected:
            anomaly = self.generate_anomaly()
            self.handle_anomaly(anomaly)
        else:
            print("Аномалії не виявлено")

    def generate_anomaly(self):
        # Генерація імітованої аномалії
        anomalies = ["Втрата пакету", "Висока затримка", "Перевантаження мережі"]
        return random.choice(anomalies)

    def handle_anomaly(self, anomaly):
        # Обробка виявленої аномалії
        timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        self.anomaly_log.append({"timestamp": timestamp, "anomaly": anomaly})
        print(f"[ALERT]: Виявлено аномалію: {anomaly} о {timestamp}")

    def get_anomaly_log(self):
        # Повернення журналу аномалій
        return self.anomaly_log

class ResourceUsageReportingModule:
    def __init__(self):
        self.reports = []

    def generate_resource_report(self):
        # Генерація звіту про використання ресурсів
        report = {
            "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
            "cpu_usage": random.randint(10, 90),
            "memory_usage": random.randint(10, 80),
            "network_bandwidth": random.randint(100, 1000) # Мбіт/с
        }
        self.reports.append(report)
        print(f"Звіт про ресурси створено: {report}")
        return report
```

```
def get_reports(self):
    # Повернення усіх звітів
    return self.reports

class StreamManagementModule:
    def __init__(self):
        self.streams = {}

    def register_stream(self, stream_id, stream_details):
        # Реєстрація нового потоку у системі
        self.streams[stream_id] = stream_details
        print(f"Потік {stream_id} зареєстровано: {stream_details}")

    def update_stream(self, stream_id, new_details):
        # Оновлення параметрів потоку
        if stream_id in self.streams:
            self.streams[stream_id].update(new_details)
            print(f"Параметри потоку {stream_id} оновлено:
{self.streams[stream_id]}")

    def remove_stream(self, stream_id):
        # Видалення потоку з системи
        if stream_id in self.streams:
            del self.streams[stream_id]
            print(f"Потік {stream_id} видалено")

class PerformanceOptimizationModule:
    def optimize_performance(self, avb_system):
        # Оптимізація продуктивності системи
        print("Оптимізація продуктивності...")
        for stream in avb_system.streams:
            self.adjust_bandwidth(stream)

    def adjust_bandwidth(self, stream):
        # Імітація регулювання пропускної здатності
        if stream["priority"] == 1:
            stream["bandwidth"] += 100
            print(f"Пропускна здатність для потоку {stream['id']} збільшена")

class ScenarioAutomationModule:
    def __init__(self):
        self.scenarios = []

    def create_scenario(self, scenario_name, actions):
        # Створення нового сценарію автоматизації
        scenario = {"name": scenario_name, "actions": actions}
        self.scenarios.append(scenario)
        print(f"Сценарій '{scenario_name}' створено")

    def run_scenario(self, scenario_name):
        # Запуск сценарію
        for scenario in self.scenarios:
            if scenario["name"] == scenario_name:
```

```

        print(f"Запуск сценарію '{scenario_name}'")
        for action in scenario["actions"]:
            self.execute_action(action)

    def execute_action(self, action):
        # Імітація виконання дії
        print(f"Виконання дії: {action}")

class RemoteUpdateModule:
    def __init__(self):
        self.update_logs = []

    def perform_remote_update(self, device_id, software_version):
        # Імітація віддаленого оновлення ПЗ
        log_entry = {
            "device_id": device_id,
            "version": software_version,
            "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        }
        self.update_logs.append(log_entry)
        print(f"Пристрій {device_id} оновлено до версії {software_version}")

    def get_update_logs(self):
        # Повернення логів оновлень
        return self.update_logs

class DeviceCompatibilityModule:
    def check_compatibility(self, device_id, device_type):
        # Перевірка сумісності пристрою з системою
        compatible_types = ["AVB-enabled", "Ethernet", "Wi-Fi"]
        if device_type in compatible_types:
            print(f"Пристрій {device_id} сумісний з типом {device_type}")
        else:
            print(f"Пристрій {device_id} не сумісний")

class HistoricalDataModule:
    def __init__(self):
        self.history = {}

    def save_stream_history(self, stream_id, data):
        # Збереження історії потоків
        if stream_id not in self.history:
            self.history[stream_id] = []
        self.history[stream_id].append({
            "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
            "data": data
        })
        print(f"Історія для потоку {stream_id} збережена")

    def replay_stream_history(self, stream_id):
        # Відтворення історії потоку
        if stream_id in self.history:
            print(f"Відтворення історії для потоку {stream_id}:")

```

```
        for entry in self.history[stream_id]:
            print(entry)

class RemoteDiagnosticsModule:
    def perform_diagnostics(self, device_id):
        # Імітація віддаленого діагностування пристрою
        diagnostics = {
            "status": "success",
            "device_id": device_id,
            "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
            "issues": []
        }
        print(f"Діагностика пристрою {device_id}: {diagnostics}")

    def repair_device(self, device_id):
        # Імітація ремонту пристрою
        print(f"Ремонт пристрою {device_id} виконується...")
        time.sleep(2)
        print(f"Пристрій {device_id} успішно відремонтовано")

class QoEModule:
    def __init__(self):
        self.qoe_logs = []

    def measure_qoe(self, stream_id):
        # Вимірювання якості обслуговування (QoE)
        qoe_score = random.uniform(1.0, 5.0) # Рейтинг від 1 до 5
        log_entry = {
            "stream_id": stream_id,
            "qoe_score": qoe_score,
            "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        }
        self.qoe_logs.append(log_entry)
        print(f"QoE для потоку {stream_id}: {qoe_score}")

    def get_qoe_logs(self):
        # Повернення логів QoE
        return self.qoe_logs
```