

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
« ____ » _____ 2022 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
**“Дослідження та програмна реалізація системи інформаційно-
комунікаційної платформи на базі рішень Cisco”**

Виконав здобувач вищої освіти
II курсу, групи КН-21М-1,4
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Гут В.В.
« ____ » _____ 2022 р.

Керівник проекту
доктор технічних наук, доцент
_____ Коваленко О.В.
« ____ » _____ 2022 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 “Інформаційні технології”
Спеціальність 122 “Комп’ютерні науки”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2022 року

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

Гуту Владиславу Вадимовичу

(прізвище, ім'я, по батькові)

- | | | | | | | | | | | | |
|--|--|--|----------------------------|---|---|--|--|--|---------------------|--|--|
| 1. Тема роботи | <i>Дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco</i> | | | | | | | | | | |
| 2. Керівник роботи | <i>Коваленко Олександр Володимирович, докт. техн. наук, доцент</i>
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання) | | | | | | | | | | |
| затверджені наказом вищого навчального закладу № 18-13 від 17.08.2022 року | | | | | | | | | | | |
| 3. Строк подання студентом роботи до захисту | <i>10.12.2022 р.</i> | | | | | | | | | | |
| 4. Мета та завдання випускної кваліфікаційної роботи: | <i>Метою розробки є дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco</i> | | | | | | | | | | |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | <table border="1"><tr><td><i>1. Призначення та область використання.</i></td><td><i>6. Наукова новизна.</i></td></tr><tr><td><i>2. Перегляд аналогічних існуючих систем.</i></td><td><i>7. Економічна ефективність розробленої програми.</i></td></tr><tr><td><i>3. Опис і обґрунтування проектних рішень.</i></td><td><i>8. Заходи з охорони праці та техніки безпеки.</i></td></tr><tr><td><i>4. Етапи програмування системи.</i></td><td><i>9. Висновки.</i></td></tr><tr><td><i>5. Впровадження системи в промислову експлуатацію</i></td><td></td></tr></table> | <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> | <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Економічна ефективність розробленої програми.</i> | <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> | <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> | <i>5. Впровадження системи в промислову експлуатацію</i> | |
| <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> | | | | | | | | | | |
| <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Економічна ефективність розробленої програми.</i> | | | | | | | | | | |
| <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> | | | | | | | | | | |
| <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> | | | | | | | | | | |
| <i>5. Впровадження системи в промислову експлуатацію</i> | | | | | | | | | | | |
| 6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) | | | | | | | | | | | |
| <i>Наукова новизна</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Структурна схема системи</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Функціональна схема системи</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Діаграма процесів</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> | | | | | | | | | | |
| <i>Показники економічної ефективності</i> | <i>1 аркуш</i> | | | | | | | | | | |

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2022	14.11.2022
Охорона праці	Оришака О.В.	06.10.2022	16.11.2022

7. Дата видачі завдання « 6 » вересня 2022 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	10.10.2022 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2022 р.	
3.	Розробка моделі компонента	20.10.2022 р.	
4.	Розробка структур даних	25.10.2022 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2022 р.	
6.	Програмування алгоритмів	10.11.2022 р.	
7.	Розрахунок економічної ефективності	13.11.2022 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2022 р.	
9.	Оформлення ПЗ	17.11.2022 р.	
10.	Попередній захист роботи	10.12.2022 р.	

Дата видачі завдання
« 6 » вересня 2022 р.

Підпис керівника

Коваленко О.В.
(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2022 р.

Підпис здобувача

Гут В.В.
(прізвище та ініціали)

АНОТАЦІЯ

Гут В.В. Дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2022.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інформаційно-комунікаційної платформи на базі рішень Cisco.

Метою розробки є дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco.

Об'єктом дослідження є процес інформаційно-комунікаційної платформи на базі рішень Cisco.

Предметом дослідження є методи інформаційно-комунікаційної платформи на базі рішень Cisco.

Методи дослідження базуються на методах теорії обробки телекомунікаційного трафіку, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco.

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 архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі RAD Studio Delphi 10.4.

Ключові слова: комп'ютерні науки, інформаційно-комунікаційна платформа, Cisco

ABSTRACT

Hut V.V. Research and software implementation of the information and communication platform system based on Cisco solutions. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the information and communication platform system based on Cisco solutions.

The purpose of the development is the research and software implementation of the information and communication platform system based on Cisco solutions.

The object of research is the process of an information and communication platform based on Cisco solutions.

The subject of the study is the methods of the information and communication platform based on Cisco solutions.

Research methods are based on methods of the theory of telecommunication traffic processing, methods of mathematical statistics, software development methods.

The result of the work is the software implementation of the information and communication platform system based on Cisco solutions.

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 PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the RAD Studio Delphi 10.4 environment.

Keywords: computer science, information and communication platform, Cisco

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	5
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	8
1.1 Призначення системи.....	8
1.2 Область застосування.....	11
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	18
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	18
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	35
2.3 Розгорнута постановка завдання	41
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	42
3.1 Опис функціонування системи	42
3.2 Розробка структурної схеми.....	55
3.3 Розробка функціональної схеми	59
3.4 Розробка діаграми процесів.....	62
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	64
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	64
4.2 Захист розробленого програмного забезпечення.....	74
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	76
6 НАУКОВА НОВИЗНА	85

					ВКРМ-122.22.0001.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Гут В.В.</i>					М	1	127
<i>Перев.</i>	<i>Коваленко О.В.</i>							
Н.контр.	<i>Гермак В.С.</i>					ЦНТУ КН-21М-1,4		
Затв.	<i>Смірнов О.А.</i>							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	86
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	86
7.2 Розрахунок трудомісткості розробки програмної продукції.....	88
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	90
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	95
7.5 Визначення собівартості розробки та ціни програмної продукції.....	99
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	101
7.7 Визначення експлуатаційних витрат.....	102
7.8 Визначення економічної ефективності програмної продукції.....	104
7.9 Висновок.....	106
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	107
8.1 Вступ.....	107
8.2 Характеристика умов праці програміста	107
8.3 Розробка заходів з умов поліпшення охорони праці.....	112
8.4 Розрахункова частина	113
8.5 Висновки до розділу.....	115
9 ОСНОВНІ ВИСНОВКИ.....	117
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	119

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

ВСТУП

Актуальність теми. У наш час існує множина різних програм, що дозволяють вести телефонні переговори через Інтернет або локальну мережу. Така можливість уже нікого не дивує, для цього потрібні лише комп'ютер, підключений до мережі, відповідна програма й мікрофон з навушниками. Звичайно, таке рішення явно не підходить для організації телефонії в серйозній фірмі (все-таки подібні засоби носять скоріше розважальний характер), однак ідея передачі голосу через мережу передачі даних дуже приваблива, особливо якщо фірма має множину офісів у різних містах. І в цьому випадку рано або пізно виникає питання про впровадження IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco.

IP-телефонія інформаційно-комунікаційної платформи на базі рішень Cisco, по суті, є способом організації телефонного зв'язку з використанням мережі передачі даних для передачі голосу. Переваги такої організації телефонного зв'язку очевидні, і головне з них – істотне зниження витрат на дзвінки між офісами, розташованими в різних містах. Крім цього, даний підхід дозволяє ввести єдиний номерний план для всієї організації, коли не потрібно пам'ятати телефонні коди міст, у яких перебувають філії компанії. Ну й звичайно, не варто забувати про впровадження додаткових сервісів.

Корпоративна IP-телефонія інформаційно-комунікаційної платформи на базі рішень Cisco дозволяє об'єднати вже існуюче в організації телефонне устаткування (звичайні телефони, підключені до АТМУ) і спеціалізовані IP-телефони в одну систему, що використовує для передачі голосового трафіку мережі передачі даних.

В зв'язку з тим, що багато фірм мають корпоративну мережу передачі даних, побудовану з використанням активного мережного устаткування фірми Cisco Systems, у цьому магістерському проекті, особлива увага приділена

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

рішенням, які пропонує саме ця компанія.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco.

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

– Огляд існуючих систем інформаційно-комунікаційної платформи на базі рішень Cisco.

– Дослідження системи інформаційно-комунікаційної платформи на базі рішень Cisco.

– Програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco.

Об'єктом дослідження є процес інформаційно-комунікаційної платформи на базі рішень Cisco.

Предметом дослідження є методи інформаційно-комунікаційної платформи на базі рішень Cisco.

Методи дослідження базуються на методах теорії обробки телекомунікаційного трафіку, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод інформаційно-комунікаційної платформи на базі рішень Cisco.

– Розроблено вітчизняний продукт інформаційно-комунікаційної платформи на базі рішень Cisco, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інформаційно-комунікаційної платформи на базі рішень Cisco.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

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 LVI Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2022, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №13.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Рішення для побудови мереж IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco засновано на використанні архітектурної моделі Cisco AVVID (Architecture for Voice, Video and Integrated Data) і призначено для рішення наступних основних задач:

- побудова сучасної багатофункціональної системи цифрової телефонії на базі корпоративної IP-мережі;
- підключення системи корпоративної IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco до телефонної мережі загального користування й стикування з існуючими ділянками традиційної телефонної мережі компанії;
- забезпечення широкого кола сучасних сервісів для абонентів корпоративної мережі IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco.

Як додаткова можливість пропонуване рішення дозволяє вирішити задачу створення мережі відеотелефонії, що може бути частиною корпоративної IP-телефонної системи.

За допомогою засобів IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco можливо побудувати мережу, що охоплює від декількох десятків користувачів у мережі малого підприємства або віддаленого офісу компанії до декількох сотень тисяч абонентів у мережі великої корпорації.

Завдяки можливостям архітектури пропонуване рішення дозволяє технологічно й економічно ефективно побудувати географічно розподілену мережу корпоративної телефонії й відеотелефонії.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Рішення IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco складається з наступних основних компонентів:

– Інтелектуальна мережна інфраструктура на базі протоколу IP, що включає в себе маршрутизатори, комутатори, шлюзи й інше мережне встаткування. IP інфраструктура є основою для подальшого впровадження користувальницьких додатків і повинна забезпечувати підтримку таких життєво важливих для мережі сервісів, як безпека, мережне керування й механізми якості обслуговування (QoS). У рамках архітектури Cisco AVVID інтелектуальна мережна інфраструктура використовується поряд з передачею даних для функціонування корпоративної телефонної й відеотелефонної системи.

– Інтелектуальні клієнтські пристрої з підтримкою протоколу IP, у тому числі цифрові IP-телефони Cisco, відеопристрої, персональні комп'ютери зі спеціалізованим програмним забезпеченням для рішення різних бізнес-задач, програмні емулятори телефонів (наприклад, IP Communicator) і так далі.

– Керування корпоративною системою IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco, а також, відеотелефонії Cisco здійснюється спеціалізованим додатком Cisco Call Manager (CCM) або кластером Cisco Call Manager. Крім того, у системі можуть використовуватися додаткові службові пристрої й додатки, такі як, корпоративна служба каталогів, що служить централізованим сховищем інформації про абонентів телефонної і відео системи, а так само службові пристрої для забезпечення аудіо й відео конференцій, H.323 гейткіпери й т.д.

– Сучасні телефонні додатки, що виникли завдяки розвитку інтегрованих систем з підтримкою голосу, відео й даних, наприклад, система уніфікованої обробки повідомлень (Unified Messaging), інтелектуальні Центри Обробки Викликів (Contact Center), мультимедійні системи організації конференцій. Впровадження подібних додатків дозволяє забезпечити додаткові можливості для користувачів/абонентів корпоративної телекомунікаційної мережі, підвищити зручність і ефективність використання системи.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Керуючий сервер Cisco Call Manager забезпечує керування встановленням телефонних з'єднань і відео з'єднань у системі. Call Manager також управляє наданням додаткових функцій як абонентам, що використовують IP-телефони, так і відеопристрої. Він також забезпечує адміністратора мережі засобами для налаштування й керування взаємодією різних компонентів системи IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco.

Спеціалізовані цифрові IP-телефони Cisco підключаються в локальну мережу, що комутується, Ethernet 10/100 і забезпечують як традиційну функціональність цифрових телефонів, так і ряд нових можливостей, властивих IP-телефонам Cisco.

Для стикування з існуючими системами традиційної телефонії, у тому числі із установленими раніше АТМУ, і підключення до телефонної мережі загального користування застосовуються голосові шлюзи. Дана функціональність реалізована на базі цілого ряду мультисервісних маршрутизаторів Cisco. Існують також голосові модулі для деяких моделей комутаторів Cisco Catalyst і самостійні пристрої, що забезпечують функціональність голосових шлюзів.

Доповнюють рішення відео компоненти:

- Абонентські відео пристрої.
- Cisco VT Advantage – включає недорогу Cisco USB камеру й програмне забезпечення для персонального комп'ютера, що підключається до IP-телефону абонента. AVTA дозволяє розширити можливості абонентського IP-телефону за рахунок передачі відео на ПК абонента при встановленні телефонного з'єднання.
- Відеотермінали сторонніх виробників, що підтримують протоколи H.323 і SCCP.
- Додатковим компонентом рішення Cisco відео телефонії є пристрої IP/VC MCU серії 3500 (IP/VC 3540 і 3511), що забезпечують аудіо– і відеоконференції в системі.
- Також обов'язковим компонентом є H.323 гейткіпери, які використовуються для інтеграції з H.323 аудіо й відео мережами.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

– ISDN шлюзи сімейства IP/VC забезпечують інтеграцію з існуючими системами H.320 відеоконференцій.

1.2 Область застосування

Корпоративна IP-телефонія інформаційно-комунікаційної платформи на базі рішень Cisco – це інтеграція системи телефонії в корпоративну мережу передачі даних. У найпростішому випадку система IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco буде складатися з наступних компонентів:

- IP-АТМУ – сервер обробки викликів;
- голосовий шлюз – пристрій для підключення до телефонної мережі загального користування (ТфОП) і/або АТМУ;
- IP-телефони – кінцеві пристрої з інтерфейсом підключення до IP-мережі;
- LAN/WAN – мережа передачі даних, що складає з комутаторів і маршрутизаторів з підтримкою функцій забезпечення якості обслуговування для передачі чутливого до затримок голосового трафіку.

Перелічимо які реальні переваги надає система корпоративної телефонії на базі протоколу IP.

По-перше, загальна кабельна інфраструктура для комп'ютерної мережі й телефонної мережі. Більшість моделей IP-телефонів оснащено вбудованим Ethernet-комутатором, що дозволяє підключити до одного порту комутатора й IP-телефон, і комп'ютер користувача. Таким чином, витрати на побудову кабельної інфраструктури будинки знижуються практично у два рази.

По-друге, відсутність потреби встановлювати в кожний з віддалених офісів власну АТМУ або виноси АТМУ центрального офісу.

Для роботи IP-телефону необхідно тільки IP-з'єднання до сервера обробки викликів (через WAN-канал зв'язку). При відмові WAN-каналу зв'язку функції

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

сервера обробки виклику тимчасово бере на себе маршрутизатор віддаленого офісу, і в цьому випадку зв'язок із центральним і віддаленим офісами здійснюється через ТфОП.

По-третє, скорочення експлуатаційних витрат. Немає необхідності містити роздільний персонал для адміністрування телефонної й комп'ютерної мереж. Крім того, завдяки концентрації всіх інтелектуальних ресурсів у центральному офісі, що обслуговує персонал у віддалених офісах не потрібно взагалі.

По-четверте, скорочення витрат на міжміські телефонні розмови при об'єднанні офісів у різних регіонах у єдину корпоративну телефонну мережу. Причому це стосується не тільки IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco, але й просто Voice over IP корпоративних мереж. Це пов'язане з тим, що пакетна мережа більше ефективна з погляду використання каналів зв'язку, чим мережа з комутацією каналів, і, отже, дешевше.

В-п'ятих, наявність додаткових сервісів, надаваних IP-телефонією. Розглянемо самі потрібні функції для користувача (переклад виклику, переадресація виклику, організація конференц-зв'язку, пошук номера абонента по ім'ю й/або прізвища й т.д.). Всі ці функції надаються й у традиційних телефонних станціях, однак частина з них доступна тільки на дорогих системних телефонних апаратах. Що стосується аналогових абонентів, яких, як правило, не менш 90% загального числа абонентів, вони теж мають можливість використовувати ряд цих функцій, але для цього необхідно запам'ятовувати ряд двузначних і більше кодів. В остаточному підсумку, тільки 10% користувачів зможуть організувати конференцзв'язок або здійснити переклад виклику на іншого абонента. Тому виникає доречне питання: навіщо вкладати гроші в «розумну» телефонну станцію й при цьому використовувати її можливості на 10%? Повернемося до системних телефонних апаратів. Так, вони відкривають перед нами доступ до цих функцій, але за вартістю вони порівнянні з IP-телефонами. У цілому рішення на базі IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco в порівнянні з рішенням традиційної телефонії, що включає 100% системних

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

телефонів, буде дешевше.

В-шостих, можливість тісної інтеграції IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco з додатками, які допомагають оптимізувати робочий процес. Оскільки IP-телефонія інформаційно-комунікаційної платформи на базі рішень Cisco споконвічно є додатком, що працює на сервері в корпоративній IP-мережі, то провести інтеграцію з іншими додатками набагато простіше й дешевше, ніж інтегрувати традиційну телефонну станцію з тими ж додатками.

Інтеграція традиційних телефонних станцій з додатками виробляється, як правило, з використанням серверів із платами Dialogic, які по інтерфейсах FXO, ISDN PRI, ISDN BRI приймають виклики від телефонної станції й обробляють їхнього різного роду додатками. Такий варіант взаємодії має наступні недоліки:

- кількість одночасно оброблюваних викликів обмежено інтерфейсами підключення сервера до АТМУ. У контексті IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco сервер додатків, підключений до мережі одним інтерфейсом FastEthernet, буде еквівалентний серверу додатків у контексті традиційної телефонії з 1000 FXO портами або 33 ISDN PRI потоками;
- відсутня можливість керування телефонною станцією й абонентськими пристроями з боку сервера додатків.

У контексті IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco наявність відкритих інтерфейсів взаємодії, таких як STI, TAPI, JTAPI, забезпечує простоту створення й налагодження, що дозволяє реалізувати ефективні додатки з мінімальними витратами. Як приклад широко використовуваних додатків:

- IP-теледодаток, що зустрічає вітанням зовнішній виклик і дозволяє з використанням донабору з'єднатися із внутрішнім абонентом або перемкнутися на секретаря;
- консоль секретаря – додаток, що дозволяє оптимізувати роботу секретаря: за допомогою зручного інтерфейсу пошуку секретар може швидко

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

відшукати необхідного абонента, визначити стан його телефонного апарата й перевести виклик, просто «перетягнувши» мишкою зовнішній виклик на даного абонента. Додатково можна реалізувати розподіл навантаження між декількома секретарями;

– набір номера з поштового клієнта або web-інтерфейсу;

– IP-контакт-центр – додаток, що дозволяє зареєструвати кожний обіг клієнта або партнера, направивши його тому співробітникові, що зможе найбільше ефективно його обробити, потім передати обіг по ланцюжку обслуговування, не втрачаючи інформації про нього. Контакт-центр на основі IP дозволяє створювати агентське робоче місце на як завгодно великому видаленні від головного офісу, максимально використовуючи ресурси філій.

Необхідно також згадати про такий параметр, як доступність. Одна з його складових – надійність, обумовлена середнім наробітком на відмову MTBF (Mean Time Between Failures). Таким чином, доступність системи телефонії означає ймовірність, з якої система буде безвідмовно працювати, а для кінцевого користувача – мінімальний час несправності системи в рік, протягом якого неможливі нові виклики або відбувається скидання встановлених викликів. Система високого класу повинна забезпечувати параметр доступності рівний 99,999%, часто називаний «п'ять дев'яток», тобто простої даної системи становить менш 6 хвилин у рік. У системі IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco, як і в традиційних телефонних станціях, рівень доступності підвищується з використанням резервування основних компонентів. Якщо в традиційній телефонії це резервування TDM-Шини, модуля керування, блоків живлення, то в IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco – не тільки сервера обробки викликів, але й компонентів, що забезпечують взаємодію IP-телефонів і серверів обробки викликів – комутаторів і маршрутизаторів.

Таким чином, щоб забезпечити доступність «п'ять дев'яток» у системі IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco,

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

необхідно дотримуватися всіх рекомендацій (керівництва по дизайну) виробника встановлення як при створенні систем IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco, так і при побудові комп'ютерних мереж високої доступності.

Нові можливості

Відзначимо деякі останні нововведення в IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco, які істотно розширюють обрії її застосування.

Video IP-телефонія інформаційно-комунікаційної платформи на базі рішень Cisco

Один з нових додатків – Video IP-телефонія інформаційно-комунікаційної платформи на базі рішень Cisco. Сьогодні цей додаток найбільше яскраво демонструє переваги IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco перед традиційною телефонією, у якій нічого подібного реалізувати неможливо.

IP-АТМУ Cisco CallManager починаючи з версії 4.0(1) дозволяє забезпечити тісну інтеграцію між двома раніше роздільними засобами комунікацій – телефонією й відеотелефонією. Для формування зображення на забезпечення Cisco VT Advantage 1.0 і спеціальні USB-відеокамери. Яких-небудь додаткових витрат на встановлення, програмне забезпечення, інсталяцію й навчання технічного персоналу не потрібно. Необхідно лише підключити USB-камеру Cisco VT Advantage до комп'ютера користувача, що використовує IP-телефон моделей 7940/60/70G і в інтерфейсі адміністрування Cisco CallManager указати, що цей телефон забезпечує відеозв'язок. Абонент відеотелефонії може використовувати зручний інтерфейс і усе раніше доступні тільки в стандартній телефонії функції (утримання/повернення виклику, переклад виклику, переадресація виклику й т.д.).

Установлення відеотелефонного спілкування між абонентами реалізується шляхом звичайного набору номера з IP-телефону. Вартість одного робочого

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

місця, оснащеного Cisco VT Advantage і портативною камерою, становить 190 дол. (без обліку вартості комп'ютера й IP-телефону).

Рішення має підвищену продуктивність, забезпечує гарну якість зображення (до 30 кадрів у секунду) і звуку. Система здатна динамічно управляти шириною смуги пропускання залежно від завантаженості телекомунікаційних ліній. Максимально можлива швидкість передачі відеоінформації становить 4,5 Мбіт/с.

Для організації відеоконференц-зв'язку в архітектурі Cisco AVVID додається пристрій IP/VC серії 3500, що надає ресурс відеоконференц-сервера й інтегрується із кластером CallManager по протоколах SCCP або H.323. Пристрій IP/VC 3540 має модульну архітектуру й забезпечує до 140 портів в 48 конференціях (при швидкості 384 кбіт/с).

Програмне забезпечення Cisco VT Advantage протестовано на сумісність із устаткуванням відеоконференц-зв'язку компанії Tandberg. Припустима також інтеграція на рівні базових функцій із продуктами інших виробників, що використовують архітектуру Cisco AVVID, наприклад, із системами компанії Polycom. Розширені можливості взаємодії з рішеннями третіх фірм дають можливість працювати з різними відеотерміналами, що підтримують розповсюджений стандарт H.323.

Безпека в системі IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco

Нові засоби безпеки, включені в Cisco CallManager 4.0, розширюють можливості захисту інтелектуальних мереж. В Cisco CallManager 4.0 використовуються нові стандартні цифрові сертифікати, що підтверджують вірогідність мережних пристроїв і захищають мережу від вторгнення хакерів, а також новітні засоби шифрування, що забезпечують конфіденційність голосового зв'язку.

Для підвищення захищеності використовується також CSA (Cisco Security Agent – агент безпеки Cisco) – рішення, що дозволяє забезпечити безпеку на

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

декількох рівнях, включаючи боротьбу із проникненням хакерів у хост-системи, захист від комп'ютерних вірусів, цілісність операційної системи й ін.

Чому замовник як і раніше зупиняє свій вибір на системі традиційної телефонії? Адже за ціною порт в IP-АТМУ на порядок нижче граничної вартості IP-телефонів у порівнянні з аналоговими телефонами. На це в першу чергу й звертають увагу замовники. Часто доводиться зіштовхуватися з аргументом, що недоцільно оснащувати робоче місце слюсаря на заводі багатофункціональним IP-телефоном. Однак не варто забувати про те, що існують голосові шлюзи з різною щільністю аналогових портів, до яких можна підключати звичайні аналогові телефони. Крім того, якщо аналоговий телефон реалізує функції Caller ID, те деякі голосові шлюзи забезпечують передачу на цей телефон номера й ім'я абонента, що дзвонить.

Таким чином, загальна вартість володіння системою IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco протягом декількох років виявляється якщо не менша, то, принаймні, порівнянна із загальною вартістю володіння системою традиційного телефонного зв'язку.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		17

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

Короткий огляд протоколів VoIP

У технології VoIP використовуються наступні протоколи:

– H.323. Протокол ITU для інтерактивної конференції. Був споконвічно призначений для мультимедійної взаємодії в мережах без установа з'єднання, таких як ЛОМ.

– MGCP (Media Gateway Control Protocol). Призначений для керування VoIP шлюзів, підключених до зовнішніх пристроїв керування викликами. MGCP надає сервіс сигналіngu для недорогих кінцевих пристроїв, таких як шлюзи, які не підтримують у повному об'ємі стек сигналіngu, наприклад H.323.

– SIP (Session Initiation Protocol). Протокол, що визначає команди й відповіді для встановлення й завершення телефонних викликів. Також деталізує такі моменти як безпека, проксі й транспортні сервіси.

– RTP (Real-Time Transport Protocol). RTP доставляє голос через мережу. Забезпечує черговість і маркування часу для правильної послідовної обробки пакетів.

– RTCP (RTP Control Protocol). Використовується для передачі керуючої інформації для протоколу RTP. Будь-яке RTP-з'єднання має відповідне RTCP-з'єднання. RTCP використовується для надання інформації про якість сервісу.

На рисунку 2.1 наведено відповідність протоколів VoIP рівням моделі OSI.

Як видно з рисунку на каналному рівні застосовуються протоколи: Frame Relay, ATM, Ethernet, MLPPP, PPP, HDLC. На мережному проткол IP. На транспортному – RTP/UDP (голос), TCP/UDP (керування). На сеансовому

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

H.323/SIP/MGCP. На представницькому рівні використовуються кодеки, а на прикладному софтвері й додатки Call Manager.

Прикладний (додатків)	Софтвері й додатки Call Manager
Представницький (рівень подання)	Кодеки
Сеансовий рівень	H.323/SIP/MGCP
Транспортний рівень	RTP/UDP (голос), TCP/UDP (керування)
Мережний рівень	IP
Канальний рівень	Frame Relay, ATM, Ethernet, MLPPP, PPP, HDLC ...
Фізичний рівень	Фізичне середовище передачі

Рисунок 2.1 – Відповідність протоколів VoIP рівням моделі OSI

Принципи встановлення з'єднання

Абонентські пристрої (Dial Peers)

Абонентський пристрій (Dial Peer) – це адресуєма точка дозвону. Такі точки встановлюють логічні з'єднання, називані етапами дозвону (Call Legs), для завершення встановлення дзвінка. Маршрутизатори Cisco, що підтримують голосові функції, підтримують два типи абонентських пристроїв: POTS Dial Peer і VoIP Dial Peer.

POTS (Plane old telephone service) Dial Peer підключаються до традиційних телефонних мереж або традиційних телефонних апаратів. Такі пристрої виконують функції по наданню адреси (телефонного номера або діапазону телефонних номерів) для кінцевого пристрою (мережі) і також указують на конкретний голосовий порт, до якого кінцевий пристрій (мережа) підключено.

VoIP Dial Peer підключаються через мережу передачі даних і надають адресу призначення (телефонний номер або діапазон номерів) для кінцевого

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

пристрою в мережі й асоціюють адресу призначення з наступним маршрутизатором, на який дзвінок повинен передатися.

Коли відбувається виклик, пристрій генерує цифри номера дозвону як спосіб вказівки пристрою, на якому дзвінок повинен завершитися, тобто пристрою, на яке відбувається дзвінок. Коли ці цифри попадають на голосовий порт маршрутизатора, маршрутизатор повинен мати спосіб вирішити, куди виклик повинен бути маршрутизований. Маршрутизатор знаходить це рішення, переглядаючи список абонентських пристроїв.

Адресу абонентського пристрою, називана шаблоном призначення (destination pattern), зконфігуровано на кожному абонентському пристрої. Шаблон призначення може відповідати як одному телефонному номеру, так і діапазону телефонних номерів. Маршрутизатор використовує абонентські пристрої для встановлення логічних з'єднань (Call Legs) як у вихідному, так і у вхідних напрямках.

Коли до маршрутизатора Cisco Systems з голосовими функціями підключається традиційний телефонний пристрій (варіант POTS Dial Peer), у конфігурації маршрутизатора вказується телефонний номер цього пристрою й порт, до якого він підключено. Таким чином, маршрутизатор "знає", куди направляти вхідний дзвінок на цей номер.

У випадку VoIP Dial Peer конфігурація маршрутизатора включає телефонний номер призначення (діапазон номерів) і мережна адресу наступного маршрутизатора.

Етапи з'єднання

Етапи встановлення з'єднання (Call Legs) – це логічні з'єднання між будь-якими двома телефонними пристроями, такими як шлюзи, маршрутизатори, додатки Cisco Call Manager або кінцеві телефонні пристрої.

Коли надходить вхідний виклик, він обробляється окремо, поки не буде визначений пункт призначення. Після цього встановлюється вихідне з'єднання, і вхідний виклик комутирується з вихідним портом.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Наскрізний дзвінок складається із чотирьох етапів з'єднання: два з погляду маршрутизатора, на якому дзвінок виникає, і два з погляду маршрутизатора, на якому телефонне з'єднання завершується.

Вхідний етап з'єднання виникає, коли виклик входить у маршрутизатор або шлюз, що виходить – коли виклик відбувається маршрутизатором або шлюзом.

Процес встановлення телефонного з'єднання можна описати наступними кроками (рисунк 2.2):

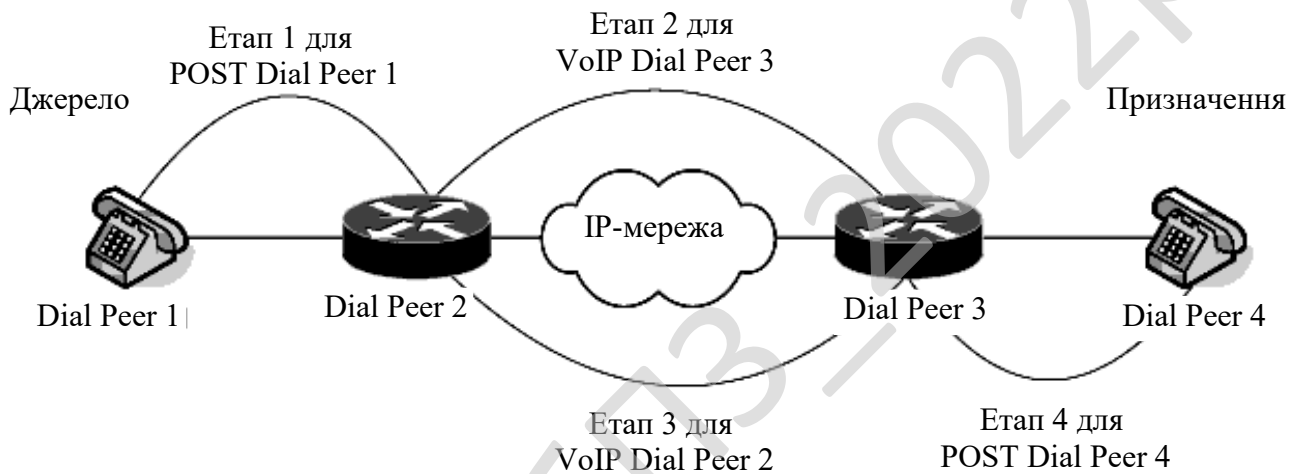


Рисунок 2.2 – Етапи з'єднання

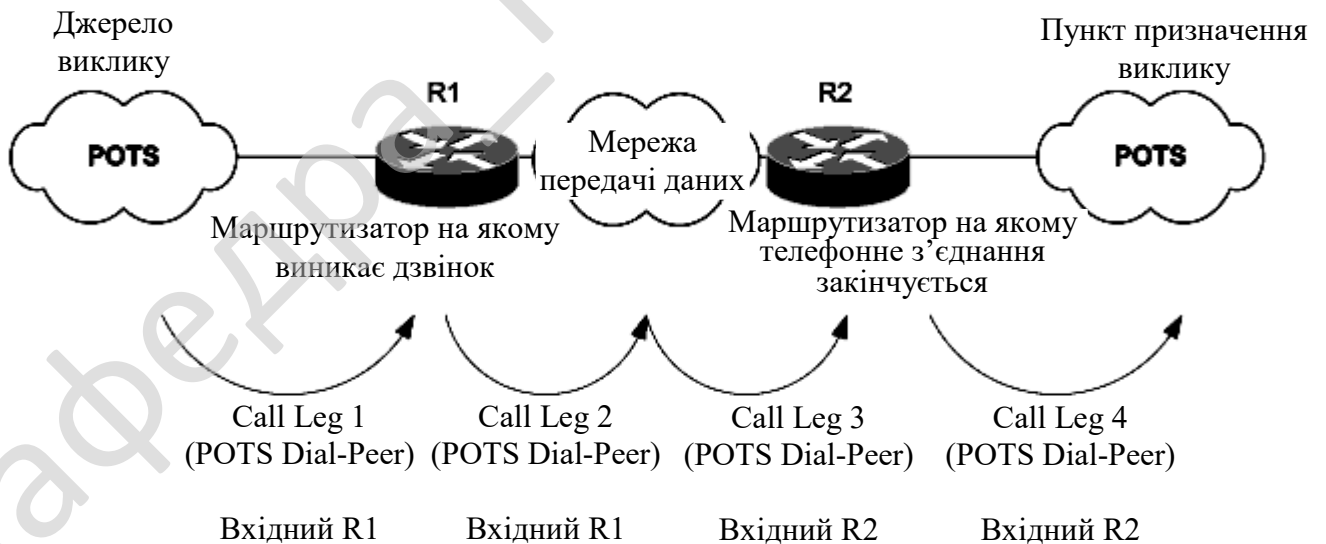


Рисунок 2.3 – Етапи з'єднання з погляду маршрутизаторів

1. Дзвінок із традиційного телефону приходить на R1 і абонентський пристрій, що ініціював виклик, ідентифіковано.

2. Після асоціювання вхідного виклику з абонентським пристроєм R1 створює вхідний етап з'єднання й призначає йому ідентифікатор Call ID (Call Leg 1).

3. R1 використовує рядок набору з метою визначення абонентського пристрою для здійснення вихідного кроку з'єднання.

4. Після визначення абонентського пристрою, з яким буде встановлюватися з'єднання, R1 створює вихідний крок з'єднання й призначає йому ідентифікатор (Call Leg 2).

5. Мережний запит надходить на маршрутизатор 2 (R2), на якому відбувається ідентифікація викликаючого мережного абонентського пристрою.

6. Після визначення мережного абонентського пристрою, з якого надійшов запит, R2 створює вхідне з'єднання й призначає йому ідентифікатор (Call Leg 3). Тут R1 і R2 погоджують параметри при необхідності.

7. R2 використовує рядок набору з метою визначення абонентського пристрою для здійснення вихідного кроку з'єднання.

8. Після визначення абонентського пристрою R2 створює вихідний виклик із призначенням йому ідентифікатора й завершує процес з'єднання (Call Leg 4).

Протоколи VoIP

Стек протоколів H.323 є одним з найпоширеніших на сьогодні. Це найстарший і найбільш стабільний із всіх використовуваних зараз протоколів, тому він вимагає особливої уваги.

Протокол ініціації сесій (SIP) – це відносно новий протокол, що одержує широке поширення. Він є значно більше молодим відносно H.323 і тому поки не одержав такого ж масштабного поширення.

Протокол MGCP – це керуючий VoIP-протокол, що найбільше часто використовується для керування шлюзами в VoIP-мережі.

Відносно новий протокол MGCP одержав широке поширення як частина архітектури Cisco AVVID.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

стеці H.323. MCU забезпечують можливість конференцій.

Стек протоколів H.323

IP, TCP, UDP

Протоколи IP, TCP і UDP безсумнівно є протоколами стека TCP/IP, але вони тут розглядаються тому, що надають транспортний сервіс для стека протоколів H.323.

Кожний термінал, шлюз, гейткіпер (контролер зони H.323) і MCU повинен мати свою унікальну IP-адресу. Це також відноситься й до ПК із додатками, які використовують H.323. IP надає кожній точці H.323-адресу й забезпечує механізм маршрутизації H.323-пакетів у мережі. TCP використовується для встановлення початкового з'єднання між терміналами H.323 і шлюзами/гейткіперами. Протокол UDP використовується для передачі безпосередньо голосу через мережу.

H.225

H.225 забезпечує встановлення й контроль викликів з усією необхідною сигналізацією для здійснення з'єднання між двома кінцевими точками.

H.245

Керуюча сигналізація H.245 застосовується для узгодження використання каналу й можливостей. Керуючі повідомлення несуть інформацію, що відноситься до наступних моментів:

- обмін інформацією про доступні можливості;
- відкривання й закривання логічного каналу, використовуваного для медійного потоку;
- повідомлення керування потоком;
- загальні команди.

Після встановлення виклику всі процеси передачі інформації проходять по логічних каналах.

RAS

RAS – це протокол, що використовується між кінцевими точками (терміналами й шлюзами) і гейткіперами. Він застосовується для здійснення

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

реєстрації, контролю доступу, статусу й змін доступної смуги пропускання, а також для відключення кінцевих точок від гейткіперу. RAS використовує порт UDP 1719.

RTP

RTP надає наскрізний мережний транспорт для додатків, що передають дані реального часу. Він використовує для передачі даних протокол UDP. Передача даних супроводжується керуючим протоколом (RTCP) для моніторингу доставки даних.

Кодеки

Кодеки використовуються не тільки протоколом H.323, а всіма протоколами VoIP для визначення алгоритмів компресії й декомпресії, застосовуваних для передачі аудіо/відео по мережі. H.323 підтримує більшість стандартів кодування аудіо й відео, включаючи G.7XX для аудіо й H.26X для відео.

Рисунок 2.5 ілюструє взаємодію протоколів стека H.323.

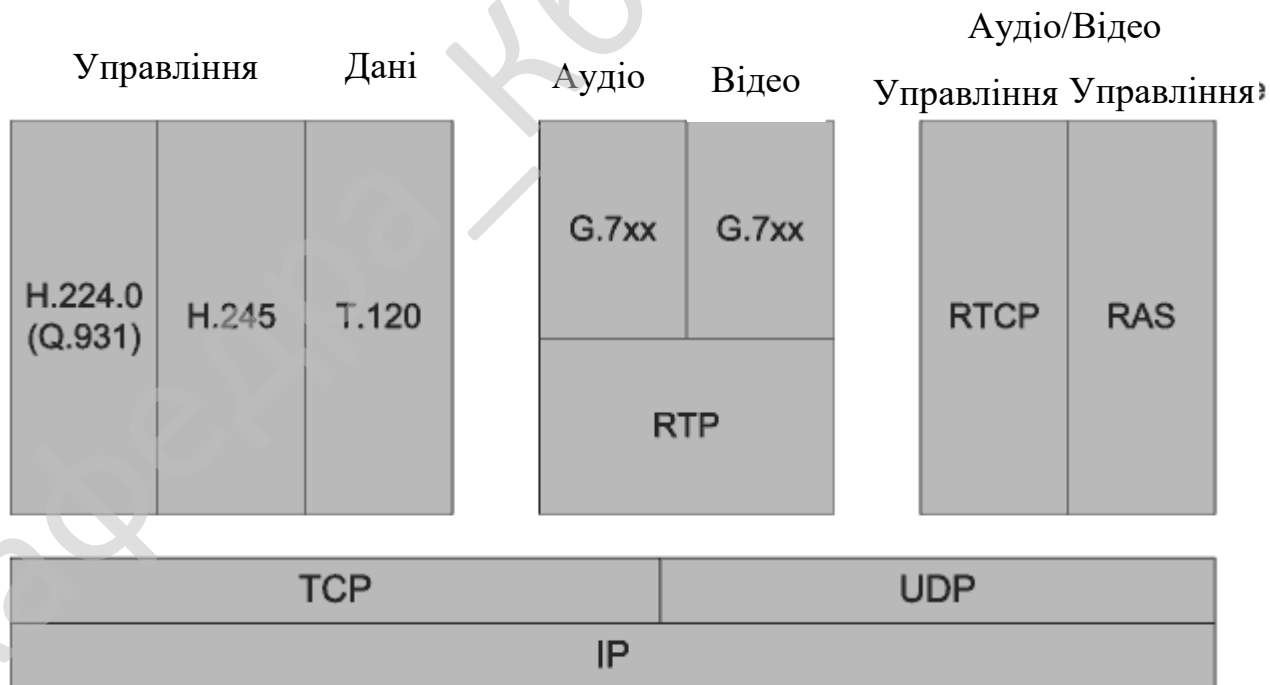


Рисунок 2.5 – Стек протоколів H.323

Етапи з'єднання

Процедури з'єднання при формуванні дзвінка по протоколі H.323 можуть бути згруповані в п'ять етапів:

1. Виявлення й реєстрація.
2. Установлення виклику.
3. Сигнальний потік.
4. Медійний потік і потік керування.
5. Завершення виклику.

Виявлення й реєстрація пристроїв

На стадії виявлення й реєстрації гейткіпер (контролер зони H.323) ініціює процес "розвідування" для визначення гейткіперу, з яким кінцева точка повинна взаємодіяти. Це може бути зконфігуровано статично, або процес виявлення може відбуватися за допомогою багатоадресних повідомлень. Після виявлення термінал або шлюз реєструється на заданому гейткіпері.

Реєстрація використовується кінцевою точкою для визначення зони, з якої вона може бути асоційована. Зона – це набір компонентів, керованих одним гейткіпером. Після визначення зони гейткіпер (контролер зони H.323) інформується про адресу зони (рисунок 2.6).

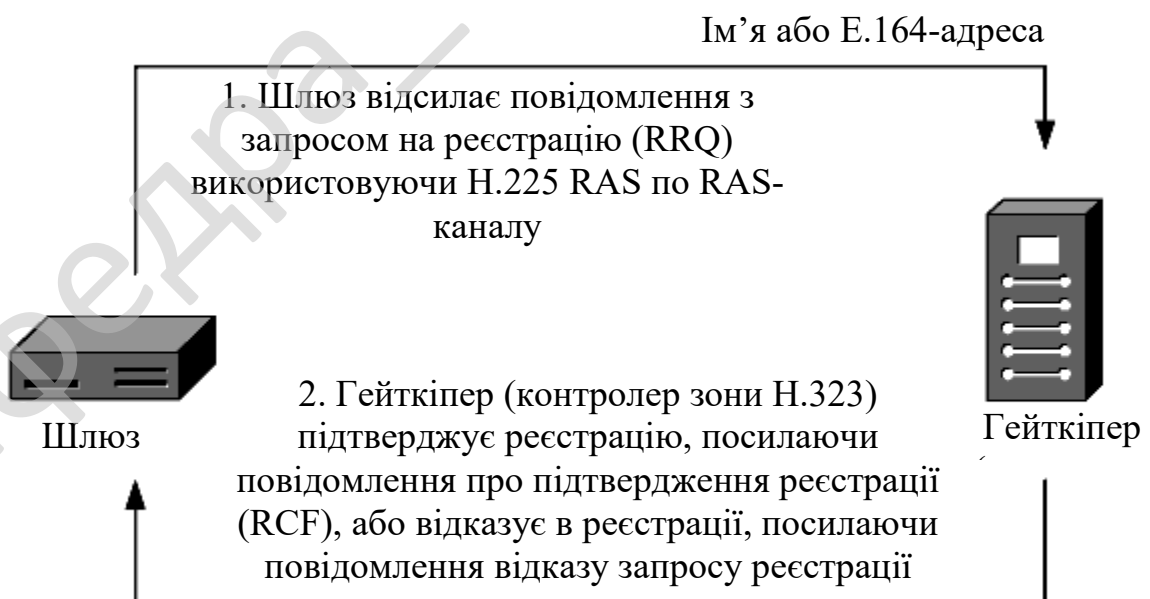


Рисунок 2.6 – Процес реєстрації

Внутризонові виклики

Допустимо, шлюзи (термінали) уже зареєстровані, і шлюз X хоче зробити виклик на термінал, підключений до шлюзу Y. Шлюз X посилає ARQ (Admission Request) повідомлення гейткіперу, запитуючи дозвіл на встановлення виклику на телефон, що обслуговується шлюзом Y. Гейткіпер (контролер зони H.323) дозволяє виклик з вимогою сигналізації (дозвону) прямо, посилаючи повідомлення ACF (Admission Confirmation) (рисунок 2.7).



Рисунок 2.7 – H.323. Установлення внутризонового виклику

Міжзонові виклики

Допустимо, гейткіпер (контролер зони H.323) А контролює зону А, а гейткіпер (контролер зони H.323) В – зону В, шлюз X зареєстрований на гейткіпері А, а шлюз Y – на гейткіпері В, і шлюз А хоче встановити з'єднання з терміналом, підключеним до шлюзу Y.

Процес установлення виклику містить наступні етапи (рисунок 2.8).

1. Шлюз X запитує з'єднання зі шлюзом Y у свого локального гейткіперу.
2. Запит місця розташування (LRQ – Location request). Гейткіпер (контролер зони H.323) шлюзу X не знає IP-адресу шлюзу Y і запитує адресу в

гейткіперу шлюзу Y.

3. Місце розташування підтвержене (LCF – Location confirm). Гейткіпер (контролер зони Н.323) шлюзу Y відповідає IP-адресою шлюзу Y.

4. Гейткіпер (контролер зони Н.323) шлюзу X підтверджує його запит і надає йому IP-адресу шлюзу Y.

5. Установлення з'єднання між шлюзами.

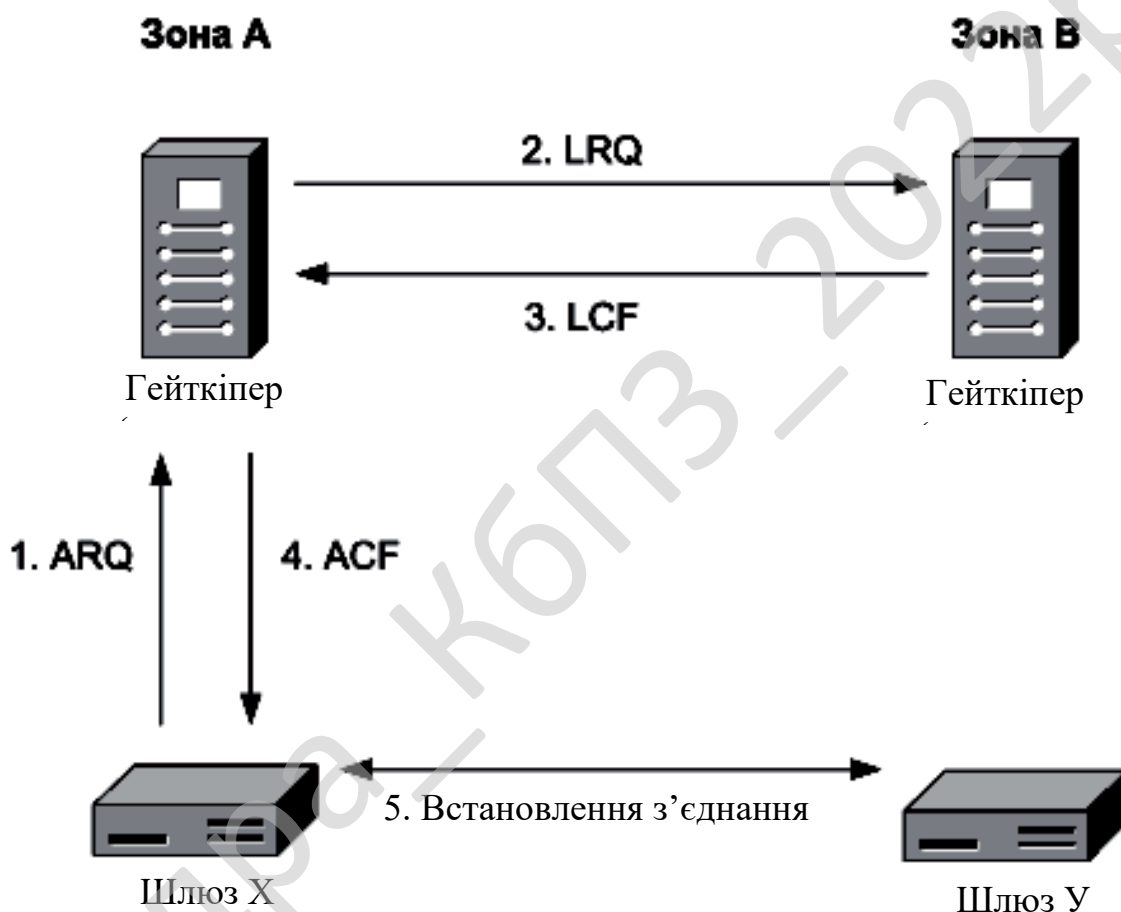


Рисунок 2.8 – Н.323. Установлення міжзонового виклику

Установлення з'єднання

На етапі встановлення з'єднання шлюзи взаємодіють прямо. Альтернативний метод установлення з'єднання – це гейткіпер (контролер зони Н.323)-маршрутизуєма сигналізація, при якій всі повідомлення по установці з'єднання проходять через гейткіпер (контролер зони Н.323).

Установлення з'єднання засноване на протоколі ITU-Q.931 (H.225 є підмножиною Q.931), що визначає метод установлення, обробки й завершення мережного з'єднання по цифровій мережі ISDN. Процес складається із шести фаз (рисунок 2.9):

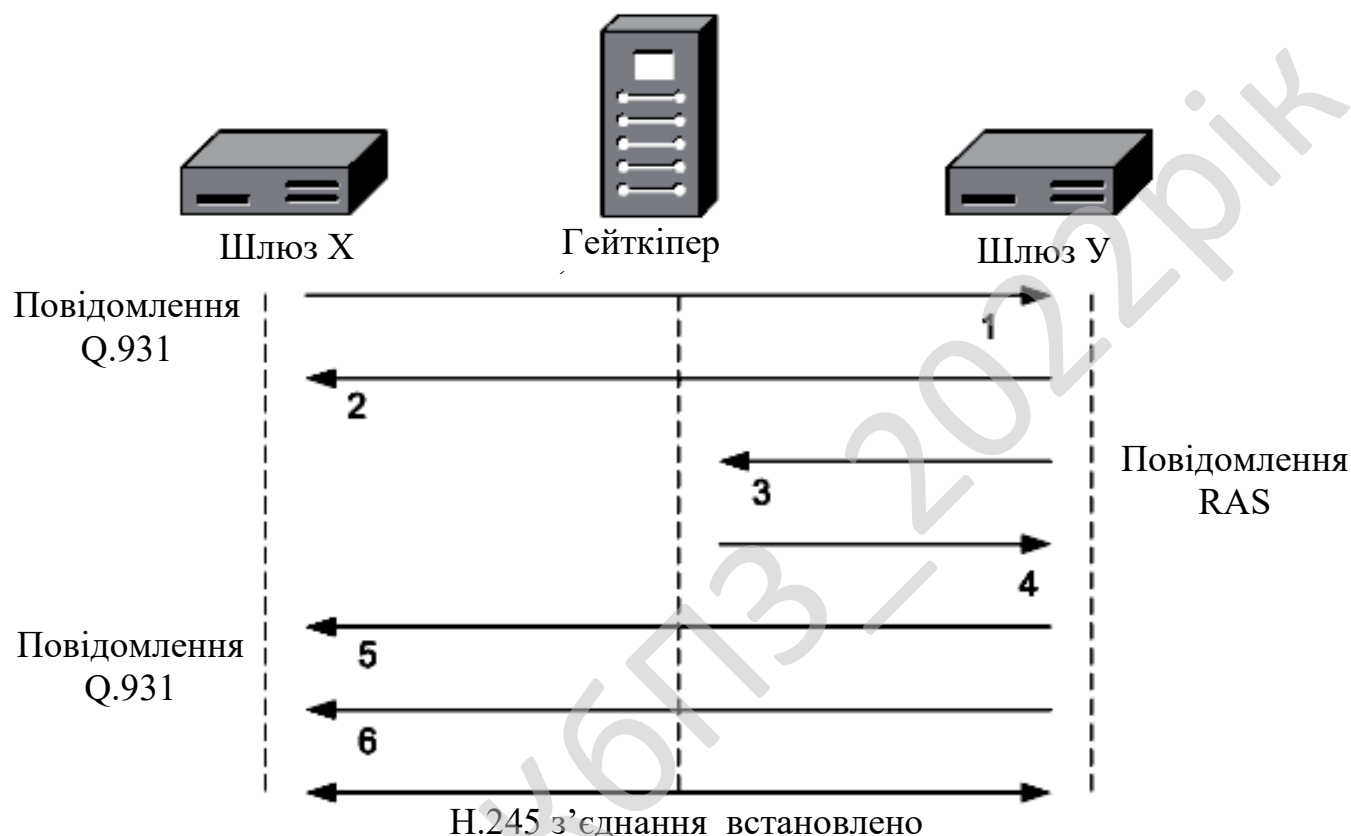


Рисунок 2.9 – Установлення з'єднання

1. Шлюз X посилає H.225 повідомлення встановлення дозвону для запиту з'єднання.

2. Шлюз Y посилає обернено H.225 повідомлення, заявляючи про можливість продовження процесу.

3. Шлюз Y запитує в гейткіперу правомірність дзвінка, посилаючи йому RAS-повідомлення (ARQ) по каналі RAS.

4. Гейткіпер (контролер зони H.323) підтверджує, що дзвінок правомірний, посилаючи шлюзу Y ACF-повідомлення.

5. Шлюз Y посилає H.225-повідомлення шлюзу X, сповіщаючи його, що з'єднання встановлене.

6. Шлюз Y посилає H.225-повідомлення шлюзу X, сповіщаючи його, що виклик установлений.

Установлення логічних каналів

Після того як з'єднання встановлене, взаємодія відбувається по логічних каналах. H.245 використовується для визначення процесу керування цими каналами. На один виклик може доводитися кілька каналів для різних типів трафіку (відео, аудіо, дані). H.245 LCSE (Local Channel Signaling Entity) відкриває логічний канал для кожного потоку. Канали можуть бути як односпрямованими, так і двонаправленими.

Установлення логічних каналів відбувається в такий спосіб:

1. Шлюз X повідомляє шлюз Y, які можливості він підтримує, посылаючи H.245 Terminal Capability Set повідомлення.

2. Шлюз Y підтверджує запит, посылаючи H.245 Terminal Capability Set Acknowledge повідомлення.

3. Аналогічний п.1, але тільки у зворотному напрямку.

4. Аналогічний п.2, але тільки у зворотному напрямку.

5. Шлюз X відкриває медіаканал зі шлюзом Y, посылаючи H.245 повідомлення Open Logical Channel, включаючи адресу RTCP каналу.

6. Шлюз Y підтверджує встановлення логічного каналу зі шлюзом X, посылаючи H.245-повідомлення Open Logical Channel Acknowledge, включаючи RTP-адресу, виділена шлюзом Y, і RTCP-адресу, отримана від шлюзу X.

7. Аналогічний п.5, але тільки у зворотному напрямку.

8. Аналогічний п.6, але тільки у зворотному напрямку.

Медійний потік і потік керування

Медійний потік управляється RTCP. RTCP використовує виділений логічний канал для кожного RTP-потіку. Кінцеві точки можуть спробувати змінити виділену смугу пропускання, що вони споконвічно запросили. Для збільшення виділеної смуги пропускання кінцеві точки повинні запросити на цей дозвіл у гейткіперу.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Завершення виклику зупиняє медиапоток і закриває логічні канали. Воно може бути запитане як кінцевими точками, так і гейткіпером. Завершення виклику також завершує H.245-сесію, звільняє H.225/Q.931 з'єднання й надає гейткіперу підтвердження про роз'єднання по RAS.

Сигналізація між кінцевими точками без посередника в H.323

Якщо шлюзи знають IP-адреси один одного, то можливо їхня взаємодія без гейткіперу. Цей процес можна описати наступними кроками:

1. Шлюз ініціює H.225.0-сесію зі шлюзом призначення.
2. Процедура встановлення виклику, що базується на Q.931, створює сигнальний канал між кінцевими точками.
3. Кінцеві точки відкривають канал для функцій керування H.245. Відбувається обмін можливостями й дескрипторами логічних каналів.
4. Відкривається RTP-сесія.

MGCP

Протокол MGCP являє собою приклад моделі із централізованим керуванням викликами. Він визначає керування телефонними шлюзами із центрального керуючого компонента, названого телефонним агентом (Call Agent). Шлюзи взаємодіють із агентами, які здійснюють сигналізацію й обробку викликів.

Компоненти MGCP

В MGCP-оточенні використовуються наступні компоненти:

- кінцеві точки;
- шлюзи;
- телефонний агент (назвемо для стислості агентом).

Кінцеві точки – це точки з'єднання пакетної мережі й традиційний телефоній мережі. Вони можуть бути фізичними й логічними. Шлюзи – це вузли об'єднання кінцевих точок.

Телефонний агент MGC (Media Gateway Controller) являє собою центральний керуючий елемент в MGCP-оточенні. MGC здійснює керування

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

діяльністю шлюзів у припущенні, що шлюзи фіксують події й доповідають про їх. Агент, ґрунтуючись на подіях, інструктує шлюзи про дії, які необхідно вживати. Він також ініціює всі VoIP-етапи з'єднання.

Поняття MGCP

Базові поняття MGCP:

- Виклики й з'єднання. Дозволяють установлювати наскрізні з'єднання двох і більше кінцевих точок.
- Події й сигнали. Дозволяють телефонним агентам інструктувати шлюзи.
- Цифрові карти й пакети. Дозволяють шлюзам визначати пункт призначення викликів.

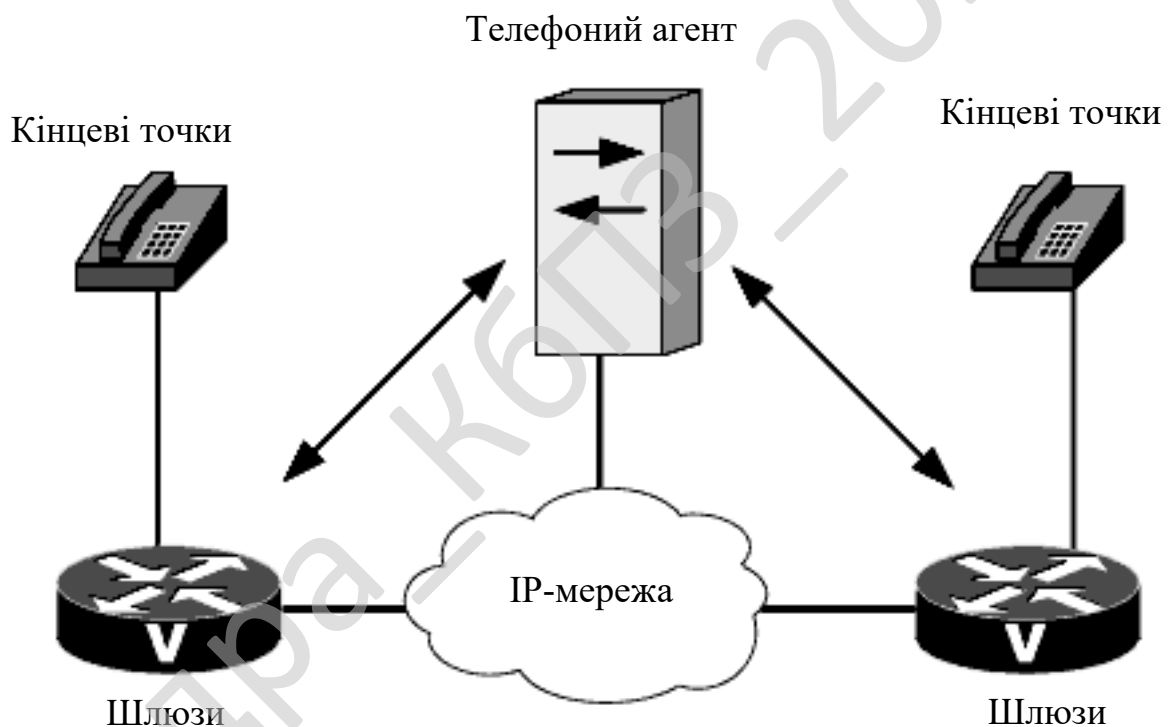


Рисунок 2.10 – Компонента MGCP

Взаємодія агентів і шлюзів

Процес взаємодії телефонного агента зі шлюзами для забезпечення телефонного виклику можна описати наступною послідовністю дій (рисунок 2.11).

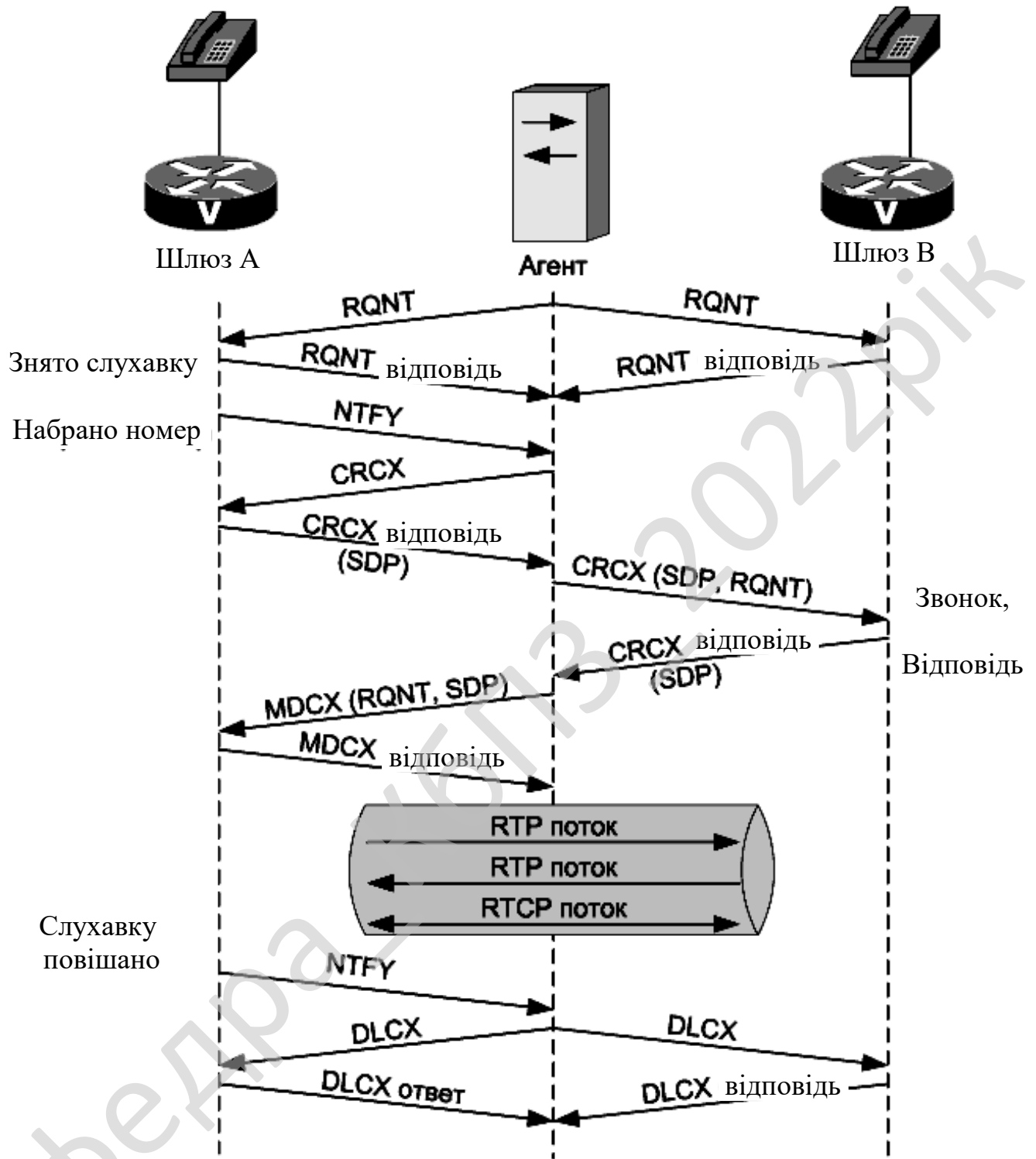


Рисунок 2.11 – Взаємодія шлюзів з агентом

Процес взаємодії телефонного агента зі шлюзами відбувається наступним чином:

1. Агент направляє повідомлення RQNT (Request Notification) кожному

шлюзу. Цей запит дає інструкцію шлюзам чекати події off-hook (коли знімається слухавка) і дати гудок, коли така подія відбудеться. Агент також повідомляє про необхідність моніторингу інших подій. Надаючи цифрову карту в запиті, агент дозволяє шлюзам зібрати цифри перед тим як інформувати про подію агента (інакше шлюз не буде "знати", коли набір номера завершується, буде змушений посилати агентові всі цифри набору по однієї).

2. Шлюз відповідає на запит. Із цього моменту агент і шлюзи чекають подій.

3. Користувач на шлюзі А підняв трубку. Випливаючи інструкції, шлюз дає телефонний гудок. Тому що в шлюзу є карта номерів, він починає збирати набираються цифри, що, поки не буде отримана відповідність (або поки набрані цифри не покажуть, що відповідність неможливо).

4. Шлюз А посилає оповіщення (NTFY) агентові, повідомляючи його, що необхідна подія відбулася. Оповіщення містить у собі кінцеву точку, подію й набрані цифри.

5. Після підтвердження можливості дзвінка агент інструктує шлюз А створити з'єднання (CRCX) з його кінцевою точкою.

6. Шлюз відповідає дескриптором сесії. Дескриптор визначає, як мінімум, IP-адресу й UDP-порт для наступної RTP-сесії. Шлюз не має дескриптора сесії віддаленої сторони, і з'єднання переходить у режим очікування.

7. Агент відправляє запит на з'єднання шлюзу В. У запиті агент надає дескриптор сесії, що він одержав від шлюзу А. Агент також посилає інструкції про те, які в цей момент події важливі і які сигнали шлюзу генерувати. У цьому випадку такою подією є off-hook, сигналом – дзвінок.

8. Шлюз У відповідає на запит і повідомляє свій дескриптор сесії.

9. Агент передає дескриптор сесії шлюзу А в запиті MDCX (Modify Connection). Тепер шлюзи можуть установити RTP-сесії для передачі голосу.

10. Наприкінці виклику одна з кінцевих точок розпізнає перехід у стан on-hook (трубка повішена). Допустимо, це трапилося на шлюзі А. Тому що агент

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Snake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.
- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабується під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентів на одній

формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємі FMX компонент TМето на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

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 інформаційно-комунікаційної платформи на базі рішень Cisco.

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Cisco AVVID

Рішення Cisco для побудови мереж IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco засновано на використанні архітектурної моделі Cisco AVVID (Architecture for Voice, Video and Integrated Data) і призначено для рішення наступних основних задач:

- побудова сучасної багатофункціональної системи цифрової телефонії на базі корпоративної IP-мережі;
- підключення системи корпоративної IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco до телефонної мережі загального користування й стикування з існуючими ділянками традиційної телефонної мережі компанії;
- забезпечення широкого кола сучасних сервісів для абонентів корпоративної мережі IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco.

Крім того, дане рішення дозволяє створити мережу відеотелефонії, що може бути частиною корпоративної IP телефонної системи.

За допомогою засобів IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco можна побудувати й невелику мережу в кілька десятків користувачів малого підприємства або віддаленого офісу компанії, і мережа великої корпорації в кілька сотень тисяч абонентів.

Архітектура пропонованого рішення дозволяє технологічно й економічно ефективно створити географічно розподілену мережу корпоративної телефонії.

Рішення IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco складається з наступних основних компонентів:

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

– Інтелектуальна мережна інфраструктура на базі протоколу IP, що включає маршрутизатори, комутатори, шлюзи й інше мережне встаткування. IP-інфраструктура є основою для подальшого впровадження користувальницьких додатків і повинна забезпечувати підтримку таких життєво важливих для мережі сервісів, як безпека, мережне керування й механізми якості обслуговування (QoS). У рамках архітектури Cisco AVVID інтелектуальна мережна інфраструктура використовується поряд з передачею даних для функціонування корпоративної телефонної й відеотелефонної системи.

– Інтелектуальні клієнтські пристрої з підтримкою протоколу IP, у тому числі цифрові IP-телефони Cisco, відеопристрої, персональні комп'ютери зі спеціалізованим програмним забезпеченням для рішення різних бізнес-задач, програмні емулятори телефонів (наприклад, IP Communicator) і так далі.

– Керування корпоративною системою IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco, а також відеотелефонії Cisco здійснюється спеціалізованим додатком Cisco Call Manager або кластером Cisco Call Manager. Крім того, у системі можуть використовуватися додаткові службові пристрої й додатки, такі як корпоративна служба каталогів, що служить централізованим сховищем інформації про абонентів у телефонній і відеосистемі, а також службові пристрою для забезпечення аудіо– і відеоконференцій, H.323-гейткіпери й т.д.

– Сучасні телефонні додатки, що виникли завдяки розвитку інтегрованих систем з підтримкою голосу, відео– і даних, наприклад, система уніфікованої обробки повідомлень (Unified Messaging), інтелектуальні центри обробки викликів (Contact Center), мультимедійні системи організації конференцій. Впровадження подібних додатків створює додаткові можливості для користувачів/абонентів корпоративної телекомунікаційної мережі, підвищує зручність і ефективність використання системи.

Керуючий сервер Cisco Call Manager забезпечує керування встановленням телефонних з'єднань і відеоз'єднань у системі. Call Manager також управляє

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

наданням додаткових функцій абонентам, що використовують як IP-телефони, так і відеопристрої. Він також забезпечує адміністратора мережі засобами для налаштування й керування взаємодією різних компонентів системи IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco.

Спеціалізовані цифрові IP-телефони Cisco підключаються в локальну мережу, що комутується, Ethernet 10/100 і забезпечують як традиційну функціональність цифрових телефонів, так і ряд нових можливостей.

Для стикування із системами традиційної телефонії, у тому числі із установленими раніше АТМУ, і підключення до телефонної мережі загального користування застосовуються голосові шлюзи. Дана можливість реалізована на базі цілого ряду мультисервісних маршрутизаторів Cisco. Існують також голосові модулі для деяких моделей комутаторів Cisco Catalyst і самостійні пристрої, що забезпечують функціональність голосових шлюзів.

Переваги застосування Cisco AVVID:

- швидкість впровадження нових сервісів;
- надійність;
- можливість взаємодії різних мереж;
- зниження матеріальних витрат.

Архітектура AVVID складається із чотирьох рівнів:

1. Інфраструктурний рівень – це фундамент мережі.
2. Рівень обробки викликів, що виконує функції комутації викликів. Його функції схожі з функціями АТМУ при використанні традиційних технологій телефонії.
3. Рівень додатків, що забезпечують додаткову функціональність.
4. Клієнтський рівень, на якому розташовуються пристрої й додатки, з якими користувач безпосередньо взаємодіє.

Моделі розгортання

Cisco підтримує наступні моделі розгортання IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco:

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

- однооб'єктна;
- с централізованою обробкою викликів;
- с розподіленою однокластерною обробкою викликів;
- с розподіленою багатокластерною обробкою викликів.



Рисунок 3.1 – Рівні архітектури AVVID

Однооб'єктна модель

В однооб'єктній моделі розгортання (рисунок 3.2) всі додатки CCM і DSP-ресурси фізично розташовані в одному місці.

Відмітні риси моделі:

- CCM, DSP і додатки перебувають в одному місці;

- підтримка приблизно 36 тис. IP-телефонів на кластер;
- кілька кластерів можуть бути з'єднані за допомогою транків;
- для зовнішніх дзвінків використовується телефонна мережа загального призначення (PSTN).

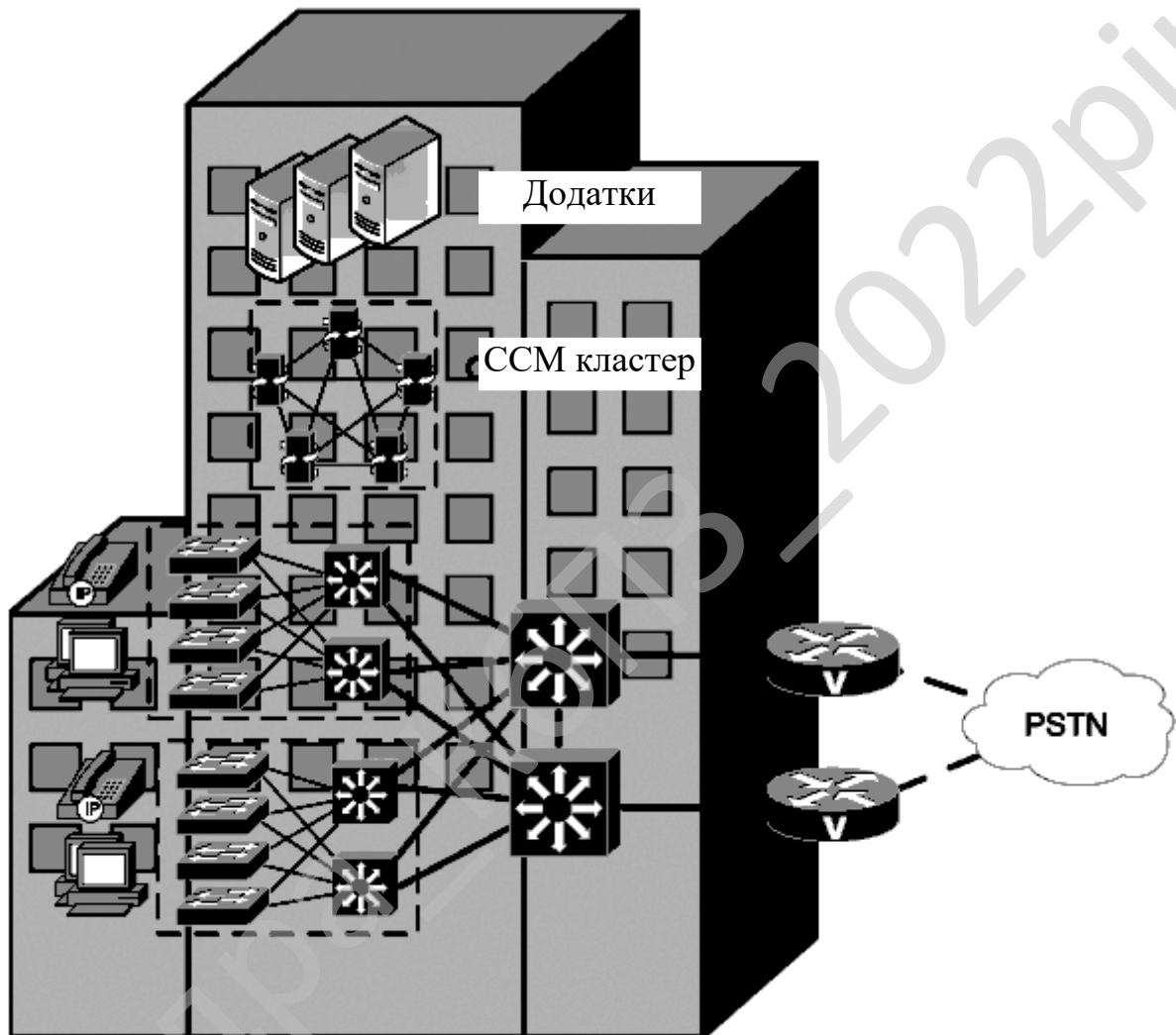


Рисунок 3.2 – Однооб'єктна модель AVVID

Модель із централізованою обробкою викликів

Дана модель (рисунок 3.3) має на увазі CCM-кластер на центральному вузлі й з'єднання до віддалених вузлів через мережу передачі даних (IP-мережа з дотриманням QoS). Віддалені вузли звертаються до центрального CCM-кластеру

виклики через телефонну мережу загального користування, коли мережа передачі даних перевантажена, для запобігання погіршення якості встановлених викликів.

Телефонна мережа загального призначення використовується як резервна. Можна також застосовувати технологію ISDN як резервний канал передачі даних, але ISDN не годиться для передачі голосу, тому що не підтримує вимоги QoS. Навіть якщо віддалений офіс втратить зв'язок із кластером ССМ, обробка викликів може бути здійснена за допомогою технології безвідмовної телефонії віддаленого вузла (SRST – Survivable Remote Site Telephony), доступної при використанні маршрутизаторів Cisco IOS. На час втрати зв'язку з ССМ ця технологія буде забезпечувати внутрішню комутацію викликів у віддалених точках.

Моделі з розподіленою обробкою викликів

ССМ-кластери можуть бути присутнім у всіх вузлах, у цьому випадку локальна комутація викликів буде вироблятися без участі центрального вузла. Також є можливість рознести компоненти одного ССМ-кластера по різних точках (рисунок 3.4 і рисунок 3.5).

Процес реєстрації IP-телефонів

Щораз, коли IP-телефон завантажується, відбувається наступний процес:

1. Якщо використовується комутатор Cisco з підтримкою живлення, комутатор посилає спеціальний FLP (Fast Link Pulse) сигнал. Комутатор використовує цей сигнал, щоб визначити, чи підключений до нього IP-телефон, що вимагає живлення. У знеструмленому стані IP-телефон Cisco повертає сигнал, запитуючи тим самим комутатор подати напругу 48 вольтів постійного струму.
2. Після того як IP-телефон одержав живлення й завантажився, комутатор посилає йому CDP (Cisco Discovery Protocol) пакет з інформацією віртуальної голосової локальної мережі (якщо зконфігуровано).
3. IP-телефон надсилає широкомовний запит DHCP-серверу. DHCP-сервер повертає телефону, як мінімум, IP-адресу, маску підмережі й IP-адреса Cisco TFTP-сервера.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

4. Телефон з'єднується з TFTP-сервером і одержує з його свою конфігураційну інформацію, що містить перелік CCM (до трьох CCM).

5. Телефон намагається зареєструватися з першим CCM з переліку, отриманого з TFTP-сервера.

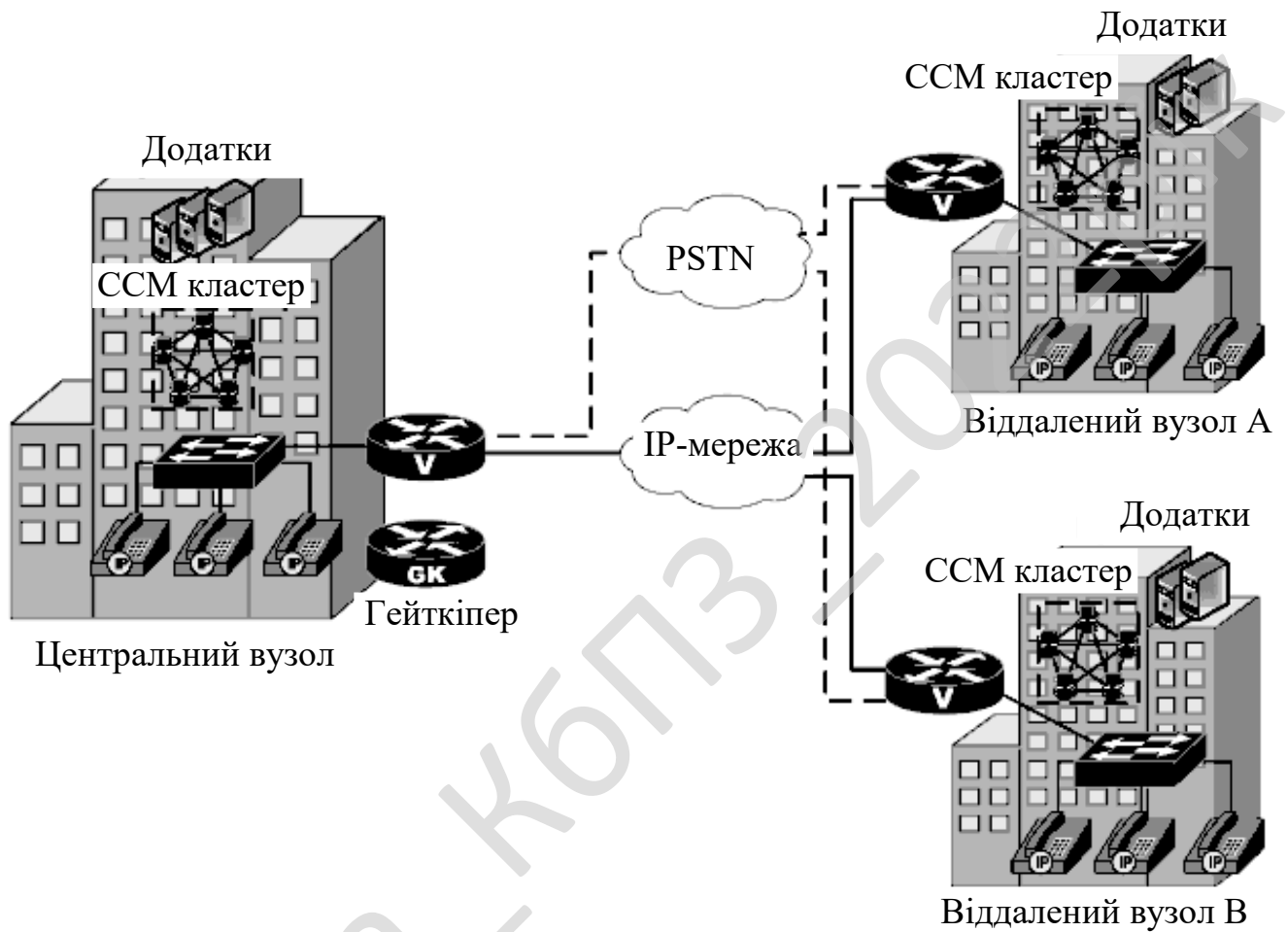


Рисунок 3.4 – Багатокластерна модель із розподіленою обробкою викликів

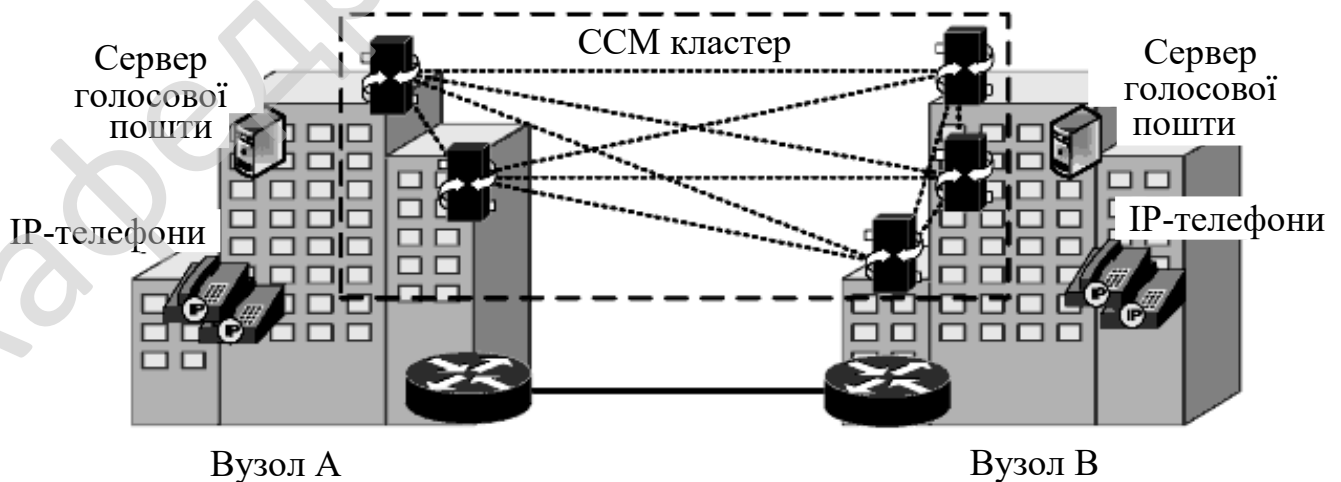


Рисунок 3.5 – Однокластерна модель із розподіленою обробкою викликів

Комутація на Cisco Call Manager

ССМ маршрутизує два типи викликів:

1. Внутрішні (on-cluster).
2. Зовнішні (off-cluster).

Комутація внутрішніх викликів

Коли надходить виклик з IP-телефону, ССМ аналізує набраний номер. Якщо він відповідає DN (Directory Number), зареєстрованому на тому же ССМ-кластері, ССМ направляє виклик на IP-телефон призначення, асоційований з відповідної DN. Це внутрішній (on-cluster) виклик. ССМ дозволяє обробляти такі виклики без напрямку його на зовнішній шлюз.

Не тільки IP-телефони можуть виступати в ролі пристроїв, здатних ініціювати й приймати внутрішні виклики, це може бути будь-який пристрій із зареєстрованим на ССМ DN. Наприклад, такими пристроями можуть бути Cisco софтверні й аналогові телефони, підключені до MGCP-шлюзів або шлюзів, що працюють по протоколі Skinny.

Комутація зовнішніх викликів

Якщо на IP-телефоні набирається номер, для якого не нашлося відповідного DN, виходить, має місце зовнішній (off-cluster) виклик. ССМ у цьому випадку переглядає свою таблицю зовнішніх маршрутів, щоб визначити, куди направити виклик. ССМ використовує концепцію таблиць маршрутів і шаблонів трансляції для визначення, куди і як направляти зовнішній виклик.

Можна створювати плани маршрутизації для зовнішніх викликів, використовуючи триярусну архітектуру, що надає кілька рівнів маршрутизації й маніпуляцій із цифрами. Шаблон маршруту (Route Pattern) визначає по номеру дзвінка список маршрутів (Route List), що вибере доступний шлях для вихідного дзвінка на основі пріоритетів. Ці шляхи Cisco визначає як "групи маршрутів" (Route Group).

Рівні вибору маршруту показані на рисунку 3.6.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

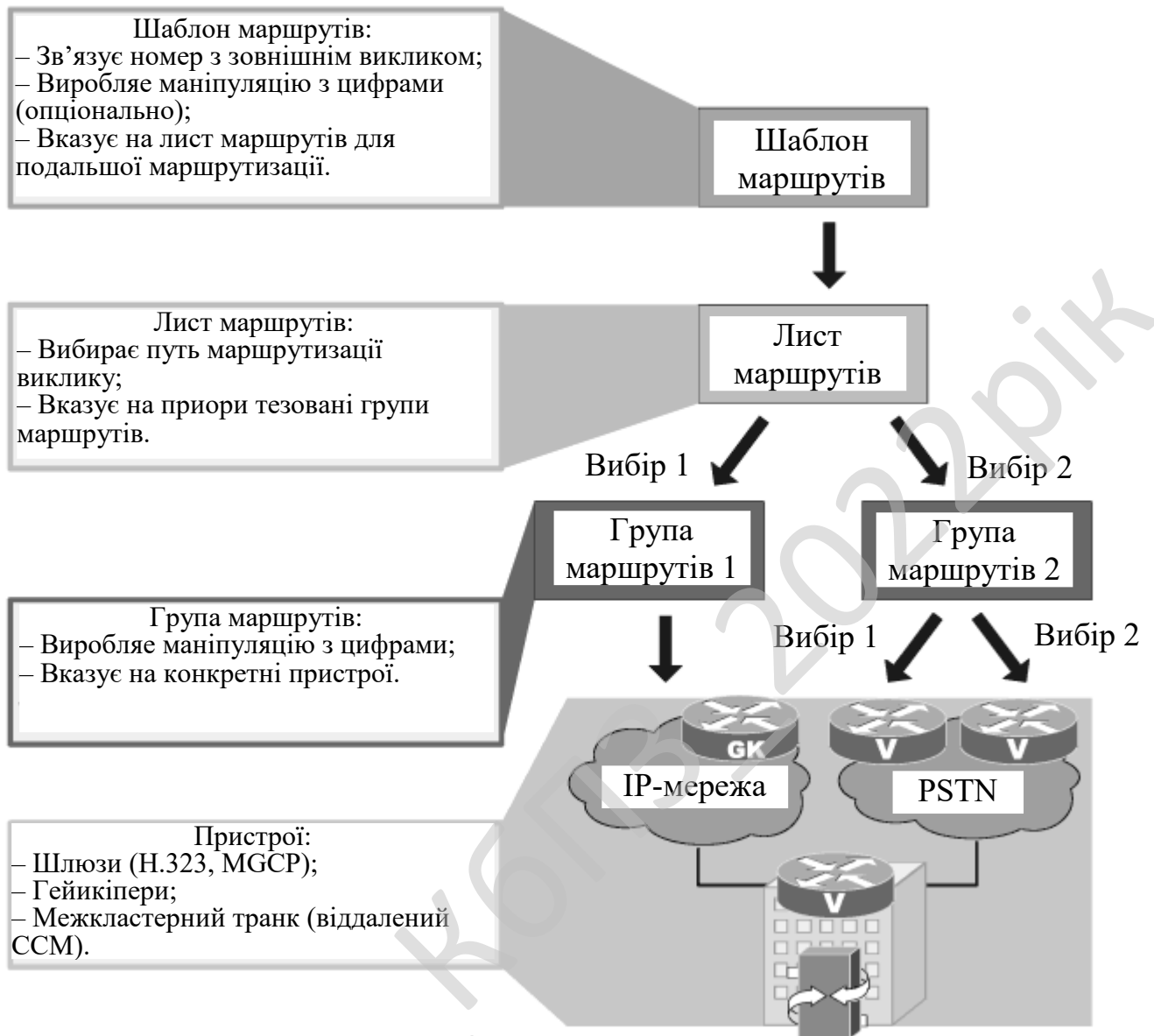


Рисунок 3.6 – Елементи маршрутизації зовнішніх викликів в CCM

Процес конфігурування маршрутів для зовнішніх викликів містить наступні етапи:

- додавання шлюзів;
- створення груп маршрутів з доступних пристроїв;
- створення списків маршрутів з доступних груп маршрутів;
- створення шаблону маршруту й асоціювання його з доступним списком маршрутів або шлюзом.

Шаблон маршруту є ключовим елементом у плані маршрутизації. Він

визначає набраний номер і направляє виклик на підходящий шлюз. Коли набраний номер відповідає шаблону маршруту, ССМ направляє виклик на відповідний список маршрутів або шлюз.

Шлюзи

Шлюзи – це пристрої, що дозволяють ССМ взаємодіяти з не-IP-мережами, такими як телефонна мережа загального користування (PSTN). Cisco розділяє свої шлюзи на дві головні категорії – аналогові й цифрові. Аналогові шлюзи можуть бути шлюзами станцій або транковими шлюзами.

Шлюзи станцій використовують порти FXS для підключення кінцевих пристроїв, таких як аналогові телефони й факси. Транкові шлюзи використовують порти FXO і підключаються до телефонної мережі загального користування або АТМУ для забезпечення зв'язку з іншими телефонними системами по аналогових лініях.

Цифрові шлюзи забезпечують те ж підключення до телефонної мережі загального користування або АТМУ, однак вони використовують цифрові технології підключення, такі як PRI CCS і транки T1 CAS.

ССМ підтримує три типи шлюзів:

– MGCP-шлюзи. Використовує модель клієнт-сервер, у якій ССМ управляє шлюзом. MGCP-шлюзи підтримують всі додаткові сервіси ССМ, надмірність ССМ і безперебійність викликів. Додатковою перевагою таких шлюзів є їхнє нескладне конфігурування.

– Non-IOS MGCP-шлюзи. Аналогічні MGCP-шлюзам, але не підтримують безперебійність викликів.

– H.323-шлюзи. Використовують однорангову модель. Більша частина конфігурації виробляється безпосередньо на шлюзі. При одноранговій моделі ССМ не має контролю над шлюзом, що приводить до зменшення кількості доступних сервісів ССМ при використанні таких шлюзів. Зате H.323-шлюзи підтримують додаткову функціональність Cisco IOS – CAC і SRST.

Крім перерахованих тут, існує ще один тип шлюзів – міжкластерний

						ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			52

транк. Це логічний шлюз, що використовується для комунікації між кластерами ССМ.

Конфігурація абонентських пристроїв

Для того щоб сконфігурувати POTS Dial Peer (традиційний телефонний пристрій), потрібно:

1. Зконфігурувати абонентський пристрій типу POTS.
2. Зконфігурувати телефонний номер.
3. Указати, до якого порту пристрій підключений.

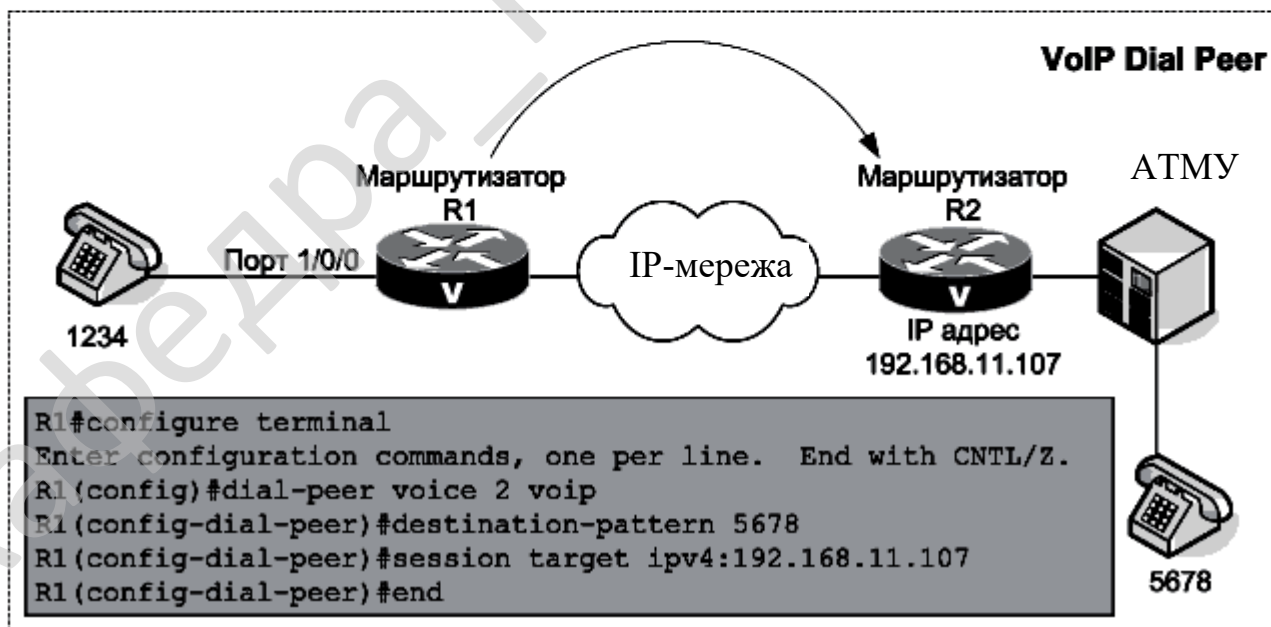
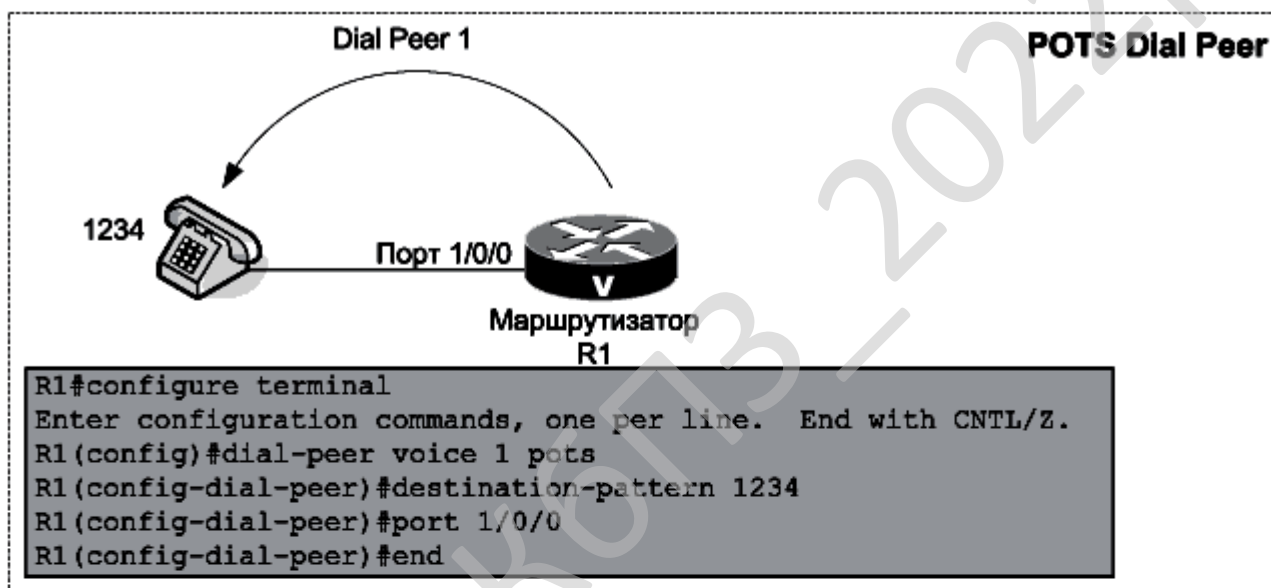


Рисунок 3.7 – Приклади конфігурації

Telecommunication Standardization Sector) для інтернаціонального плану нумерації. Знак плюс відноситься поперед рядка, що визначає шаблон призначення, і означає, що рядок повинен відповідати рекомендаціям E.164.

– Рядок: набір цифр, що визначають телефонний номер. У рядку крім цифр можуть використовуватися спеціальні символи:

– Зірочка (*) і ґрати (#). Ці символи завжди є на стандартних клавіатурах кнопочних телефонів. Вони можуть використовуватися в автоматичних телефонних системах, таких як автовідповідач.

– Кома (,) вставляє односекундну паузу між цифрами, що набираються. Кома може бути використана, наприклад, коли набирається 9 для виходу в телефонну мережу загального призначення через АТМУ – пауза дає АТМУ час для комутації з телефонною мережею загального призначення.

– Точка (.) – відповідає будь-якій цифрі. Використовується для завдання діапазону телефонних номерів.

– Квадратні дужки ([]) – позначають діапазон. Наприклад "20[0-4]." відповідає діапазону номерів від 2000 до 2004.

– Т. Опціональний контрольний символ, що позначає, що значення шаблону є рядком змінної довжини. Маршрутизатор накопичує набираються цифри, що, доти, поки інтервал між ними не перевищить зконфігурованого значення (яке за замовчуванням становить 10 секунд). Після закінчення набору, щоб не чекати, поки мине таймаут, можна набрати ґрати й тоді маршрутизатор почнуть обробляти запит негайно.

3.2 Розробка структурної схеми

Систему IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco розглянемо на прикладі бездротової мережі передачі даних і голосу, побудованої з використанням мостів радіо-Ethernet Cisco AIR-BRI342 і системи IP Telephony AVVID.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

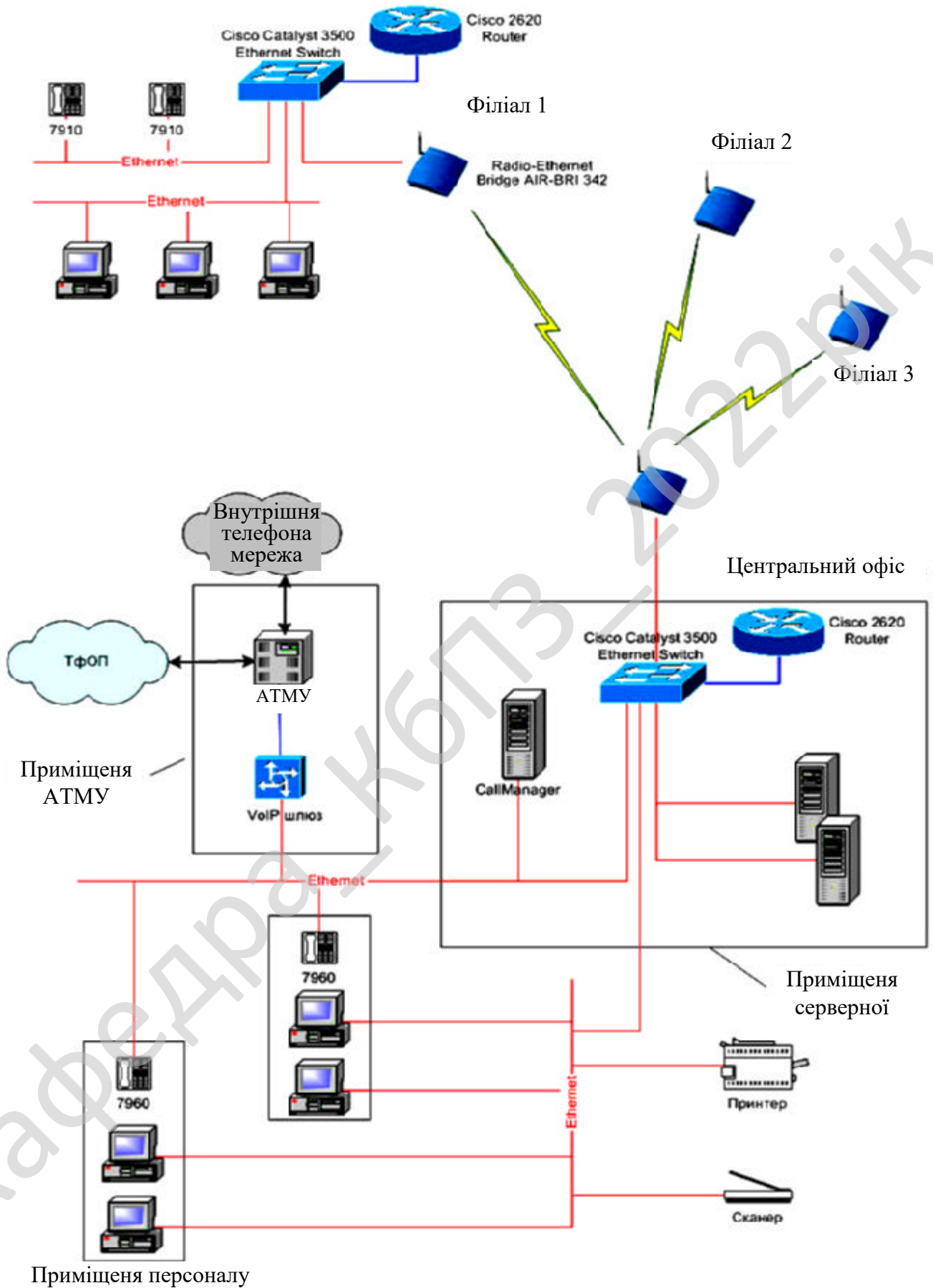


Рисунок 3.8 – Структурна схема системи

На рисунку 3.8 зображена структурна схема розроблювальної системи.

Дана радімережа поєднує центральний офіс компанії з її філіями. Для виключення конфліктів голосового трафіку з локальним трафіком даних у ЛОМ необхідно розділити пристрою IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco й інші мережні пристрої по різним колізійним сегментах шляхом організації VLAN за допомогою Ethernet-Комутаторів. Ця міра особливо актуальна, якщо у ЛОМ працюють напівдуплексні пристрої Ethernet. Що стосується властиво радімережі, те всі радіомосты по визначенню перебувають в одному колізійном сегменті.

Відомо що, при передачі голосу по VoIP у радімережах множинного доступу з використанням колізійних протоколів MAC-рівня типу CSMA/CA, основна проблема, що приводить до тимчасових затримок, ця відсутність гарантованої пропускної здатності каналу для даного з'єднання. Протокол CSMA/CA припускає випадкову установку з'єднання й, як наслідок, асинхронний механізм передачі радіокадрів, що зовсім неприйнятно для передачі голосового трафіку, критичного до тимчасових затримок. Випадковий характер установки з'єднань приводить до також випадкового значення пропускної здатності каналу для даного з'єднання. З ростом числа абонентів мережі ймовірність появи колізій зростає й, відповідно, зростають коливання швидкості передачі даних щодо середньостатистичного значення. Ці коливання можуть становити десятки відсотків, у результаті чого виникають непередбачені затримки трафіку. Як показали численні експерименти по передачі VoIP у середовищі радіо-Ethernet, затримка трафіку тут значно перевищує суму інших складових загальної затримки, характерних для технології VoIP: затримки кодека, затримки в чергах, затримки в буфері, що згладжує. Коли загальна затримка в каналі, по якому встановлений сеанс голосового зв'язку, перевищить 150 мс – суб'єктивну межу слухового сприйняття затримки, відбувається катастрофічна втрата якості мови.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Типовий метод рішення проблеми, наприклад, засобами математичного забезпечення Cisco Systems містить у собі:

- усунення перевантажень у мережі шляхом профілювання трафіку за допомогою GTS (Generic Traffic Shaping);
- пріоритезація трафіку на основі інструментів класифікації PBR, BGP, CAR;
- забезпечення необхідного рівня сервісу в мережі за допомогою сигналізації QoS, що використовує протокол маркування кадрів Ethernet відповідно до рекомендації 802.1p/Q або протокол RSVP.

Для реалізації перерахованих функцій у кожній точці мережі необхідно встановити прикордонний маршрутизатор, як мінімум, серії 2600. Однак, слід зазначити принаймні два недоліки наведеного методу. Перший полягає в тому, що механізм GTS обмежує швидкість вихідного потоку даних, тобто, виконує свою функцію тільки на ділянці маршрутизатор – радіомост у даній точці мережі. Але, оскільки в радіосегмент дані однаково надходять несинхронно, імовірність нерівномірного завантаження мережі не вдається звести до нуля.

Другий недолік полягає в тім, що математичне забезпечення, що дозволяє реалізувати маркування кадрів Ethernet, не входить у базову комплектацію маршрутизатора. Для цього необхідно доповнити його розширеною версією програмного забезпечення IP Plus , а також збільшити об'єм flash-пам'яті до 16 Мбайт і RAM – до 48 Мбайт на загальну суму 2400 USD. Якщо взяти до уваги, що вартість маршрутизатора Cisco 2610 у базовій комплектації становить 1995 USD, а радіомоста Cisco AIR-BRI342 – 1949 USD, те одне тільки забезпечення QoS у мережі радіо-Ethernet підвищує витрати на 60%.

Таким чином, у бездротовій мережі передачі даних і голосу на базі радіо-Ethernet основна робота із забезпечення якості мови виконується маршрутизатором. Нагадаємо, що все вищесказане відноситься до переважної більшості мостів радіо-Ethernet, що використовує протоколи множинного

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

доступу типу CSMA/CA. Однак, у цей час з'являється встаткування цього типу, що використовують неколізійні каналні протоколи.

3.3 Розробка функціональної схеми

В процесі практичної реалізації теоретичних принципів розробки системи, додатково розглянутих вище, була розроблена структурна схема системи, яка зображена на рисунку 3.9.

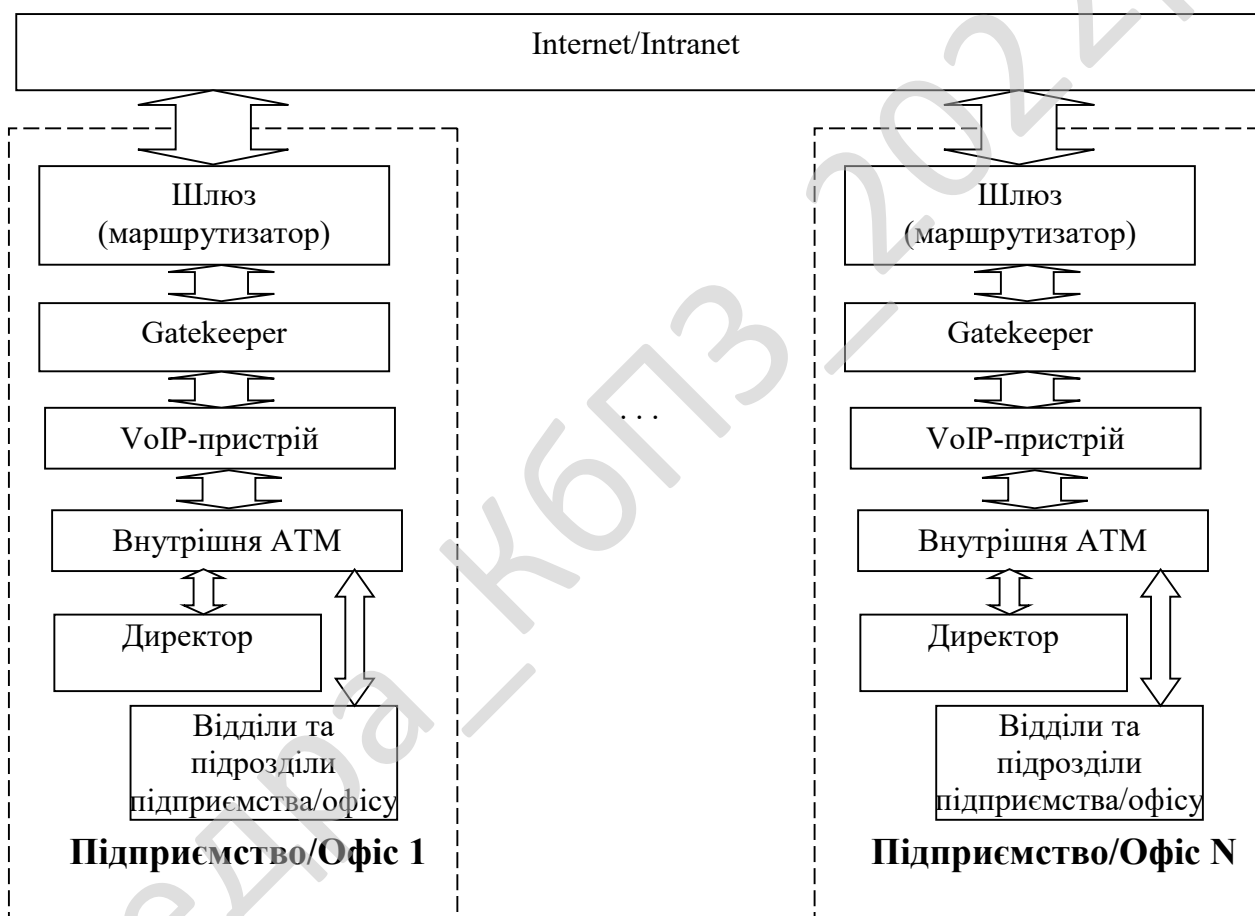


Рисунок 3.9 – Функціональна схема системи

Завдяки структурній схемі можна чітко побачити основні структурні блоки системи та взаємозв'язки між ними. При розробці структурної схеми основний упор робився на існуючі розробки ПЗ і їх модулі допомоги. Аналіз рисунка 3.9 дозволяє чітко прослідити як працює програма. Розглянемо схему

зверху вниз, в напрямку від пристрою до кінцевої програми – за допомогою внутрішнього пула доступу що забезпечує закриті канали зв'язку абоненти генеральний директор, відділи бухгалтерії, логістики, складу, збуту, економічної безпеки – можуть взаємодіяти з дочірніми виробниками з використанням внутрішньої АТМ. і чітко налагодженої технічної системи взаємодії. Технічна частина взаємодії забезпечується VoIP та гейткіпером з використанням маршрутизатора.

Розглянемо системний взаємозв'язок програмних модулів і інформаційних файлів. Технологія внутрішньої організації задається послідовністю реалізованих процедур – схем взаємозв'язку програмних модулів і інформаційних масивів. Така схема являє собою декомпозицію загального процесу рішення задачі на окремі процедури перетворення масивів, іменованими модулями (це – виклик IP-адресату, завершення сеансу зв'язку, редагування легенди IP-номерів, редагування бази даних IP-адрес, моніторинг бази даних IP-адрес та легенд або організація селективного зв'язку і т.і.).

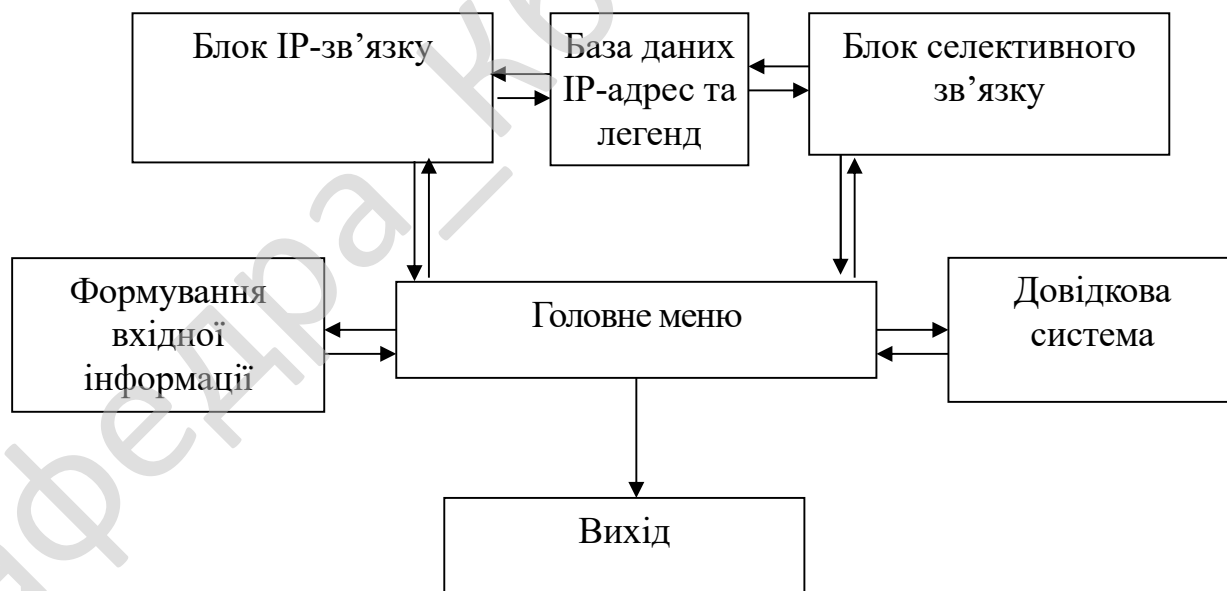


Рисунок 3.10 – Функціональна схема основних модулів програми

Основне призначення створюваної корпоративної мережі зв'язку на основі IP-технології – це автоматизація процесу керування. Отже, структуру програм можна описати наступними основними блоками.

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

Модуль “Головне меню” призначений для запуску основних процедур програми і завершення роботи з програмою.

Модуль роботи з довідниками містить у собі два довідники:

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

Модуль роботи з базою даних абонентів корпоративної мережі включає в себе наступні підмодулі:

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

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

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

Модуль «Формування вхідної інформації» призначений для введення первинних даних і перегляду раніше занесених. Даний модуль реалізує задачі

обліку телефонних номерів, забезпечуючи введення номерів та їх легенд, ранжування за важливістю та їх знищенню.

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

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

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

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

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

3.8 Діаграма процесів системи

Одним з важливих критеріїв при розробці будь-якого програмного забезпечення це грамотна розробка структури роботи системи потоків і процесів (рисунок 3.11). Розглянемо діаграму процесів. На діаграмі процесів можна точно зрозуміти як працює і взаємодіє ПЗ в цілому.

Починається і закінчується програма в першому блоці є основною точкою відрахунку діаграми– «Основний блок ПЗ (початок/кінець)». При переміщенні по стрілках можна побачити загальну схему взаємодії блоків і їх входження один в одного. Чітко простежується взаємодія основного блоку програми з первинними функціями програми без яких надійне функціонування ПЗ неможливе. Перед тим як зв'язатися з кінцевим абонентом ПЗ перевіряє ступень захищеності та ступень працездатності каналу зв'язку, проходить модуль захисту ПЗ та Викликає кінцевого адресу чи групу адресів (селективний зв'язок). При виникненні помилки ПЗ запускає обробник помилок і локалізує помилку.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

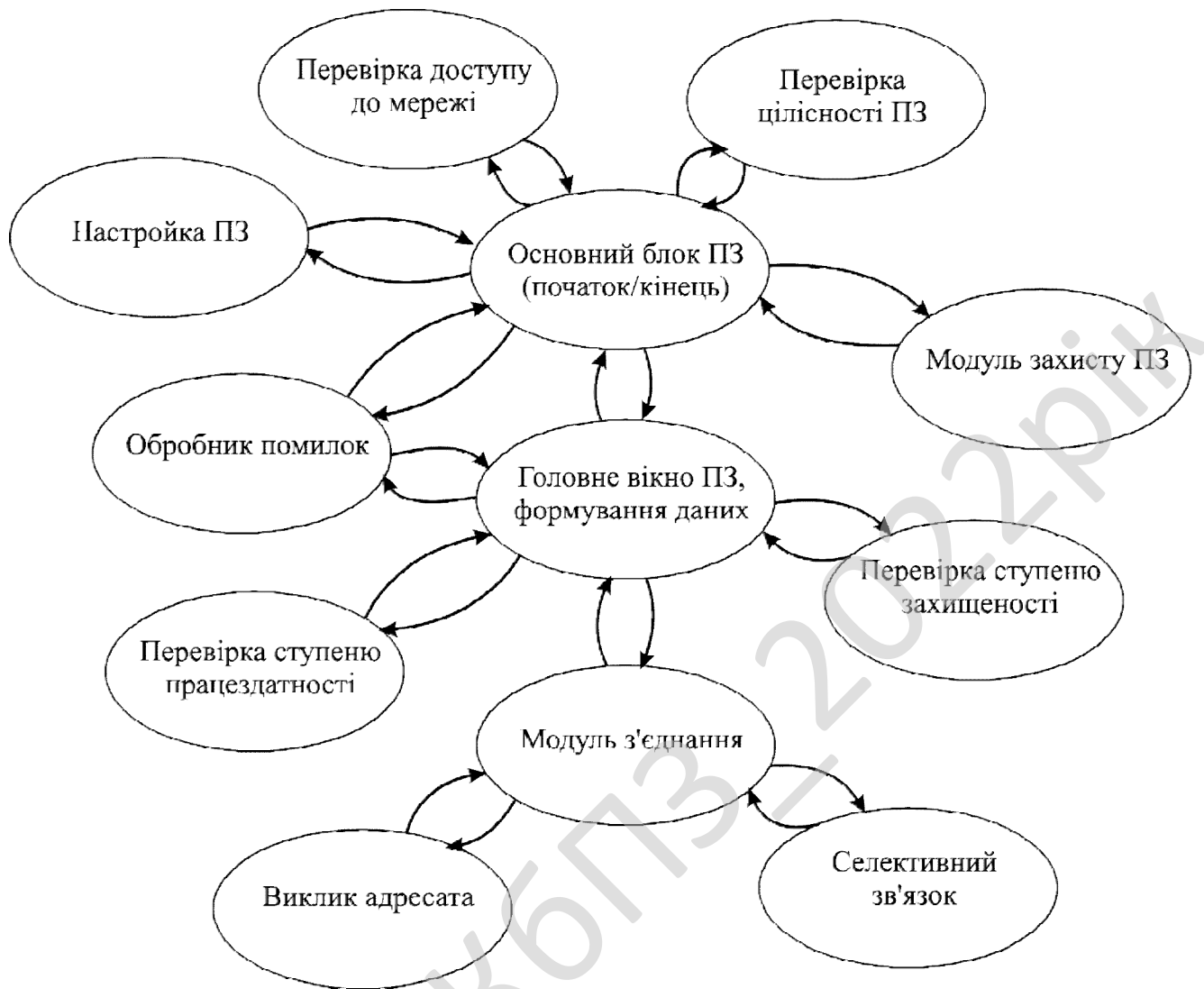


Рисунок 3.11 – Схема взаємодії процесів

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

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. 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 я враховував, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єкно-модульна, проект, що розробляється, вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулям захисту каналу зв'язку з використанням IP технології, модулю обробки помилок програми і аудіо модулю. Створення повноцінного коду програмного продукту з розробленою ціною політикою і системою знижок покупцям програмної продукції. На рисунку 4.1 зображена основна блок-схема програми. При детальному розгляді програма розбита на декілька основних важливих блоків чи групи блоків, таких як:

- Блок ініціалізації початкових значень типів даних, констант, змінних, класів, масивів і графічних даних.
- Блок підключення додаткових файлів.
- Блок пошуку та читання файлів програмного забезпечення, якщо файли ПЗ не знайдені відбуваються дії для завершення ПЗ з виведенням повідомлення. Локальне повідомлення реалізоване WIN API32 функцією MessageDlg.
- Блоків налагодження на робочий режим.
- Блок підпрограми зв'язку.
- Блоків завершення роботи програми.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

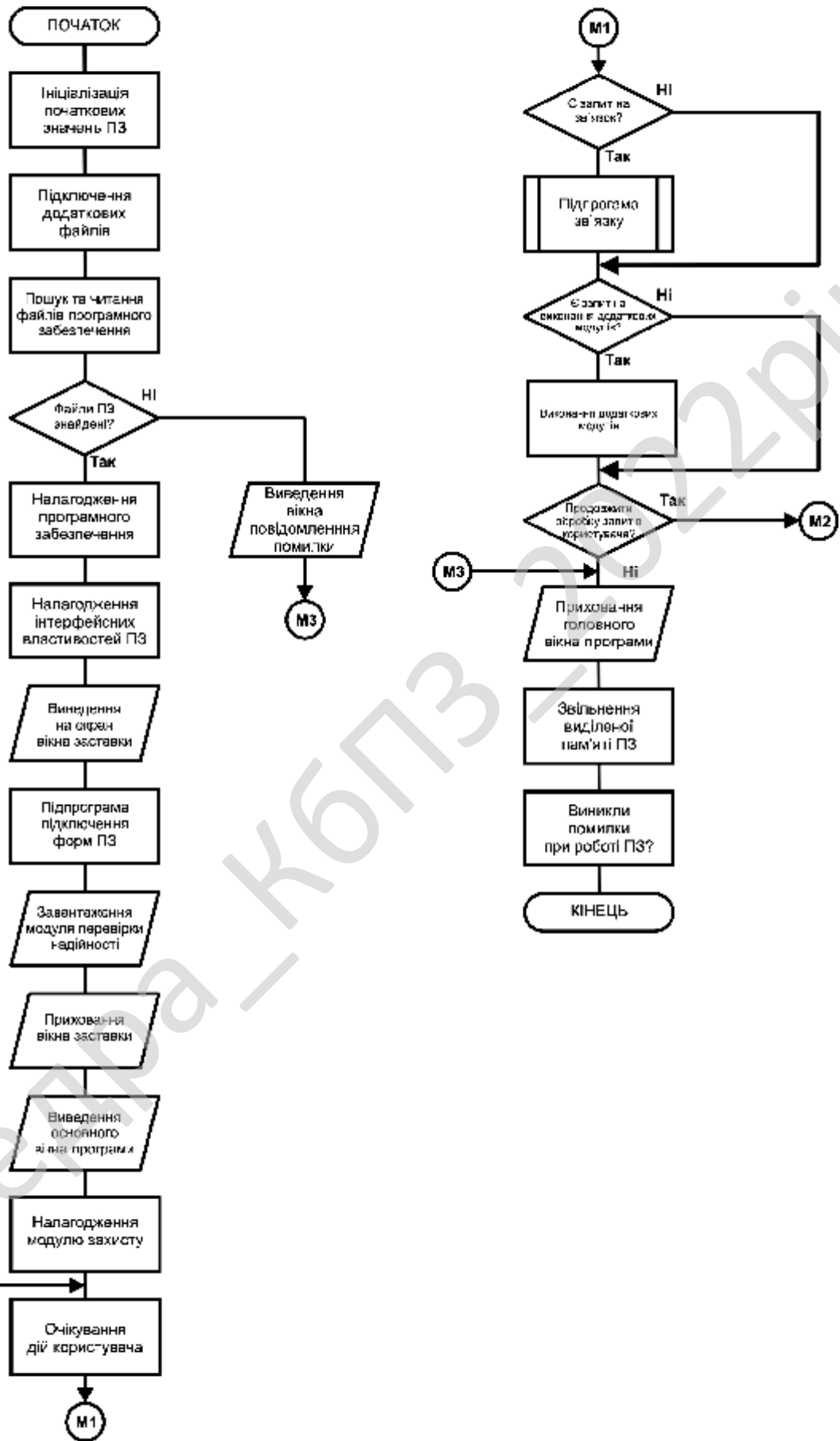


Рисунок 4.1 – Блок-схема програми

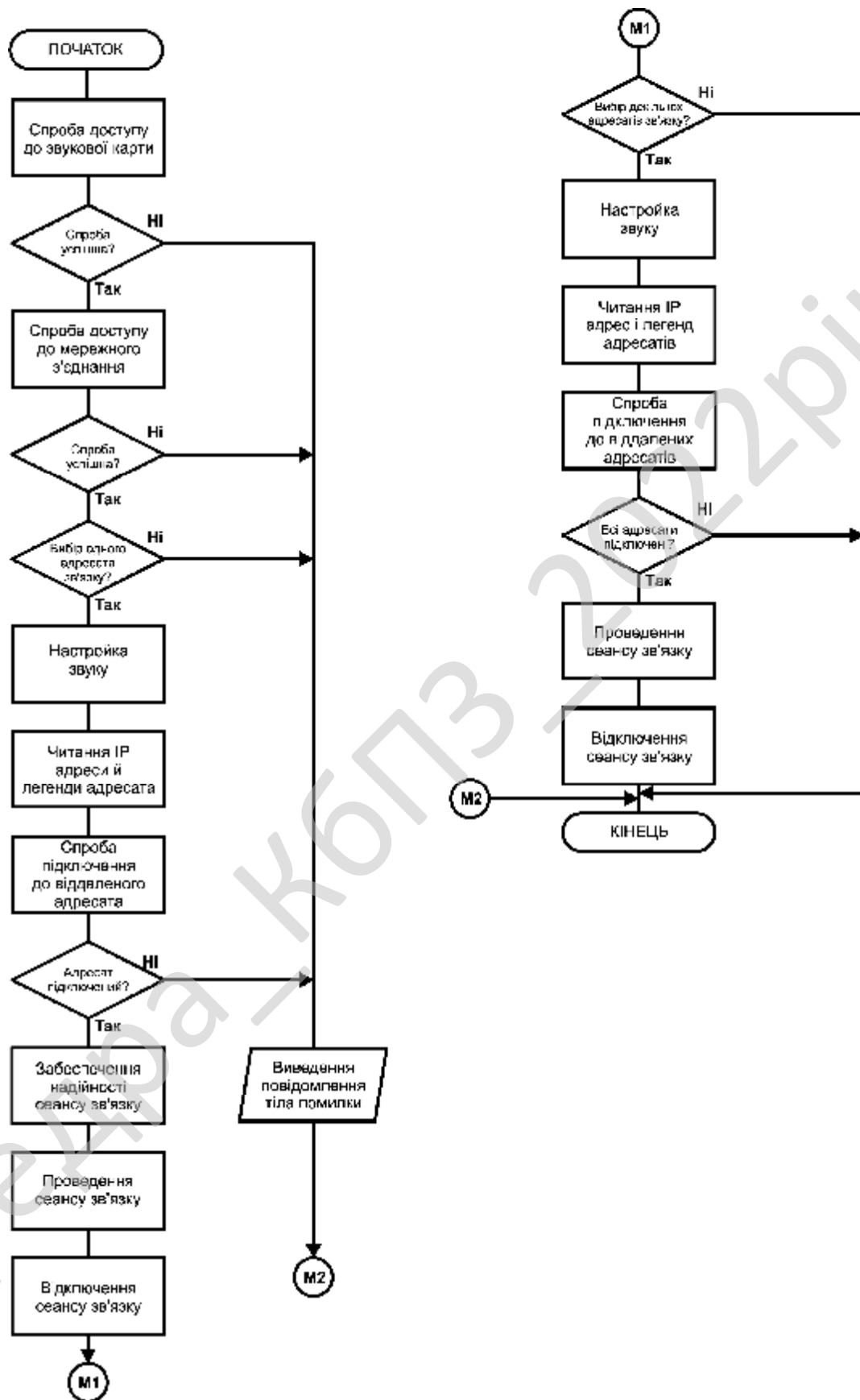


Рисунок 4.2 – Блок-схема підпрограми зв'язку

Блок підпрограми зв'язку відображений на рисунку 4.2 і описує основний алгоритм аналізу та роботи каналу зв'язку.

Саме в цьому блоці в залежності від вибору адресу проводиться спроба доступу до мережного з'єднання (internet/intranet), вибір адресу чи адресів зв'язку, налаштування та проведення сеансу зв'язку.

Опис технологічного процесу обробки даних

Технологічний процес складається з двох основних етапів – збір і облік даних по IP-адресам й формування відповідної легенди та власне сама робота програми у режимі IP-телефонії та селективного зв'язку. Вони можуть виконуватися в будь-який календарний момент часу і включають операції введення, з'єднання і ін. Операції мають програмне виконання, підлегле єдиній алгоритмічній схемі. Програма реалізована в середовищі Delphi. Робота з програмою починається з виведення інформаційного вікна й активізації системи меню й здійснюється в діалоговому і по-дійному режиму. При цьому під діалогом розуміється надання користувачу декількох альтернатив і обробка його вибору. У діалогову систему входять головне меню з відповідними спливаючими підменю а також діалогові вікна. Під подіями розуміються процеси, що активізуються користувачем (наприклад – натискання функціональних клавіш), а також програмні події – з'єднання по IP-технології.

Програма складається з наступних основних модулів.

Основна процедура – конфігурація середовища оточення, формування основного екрана програми, створення системи головного меню і відповідних підменю, активізація меню.

Процедура обробки головного меню – запуск відповідної процедури. Процедура введення даних – забезпечення введення інформації у бази даних IP-адрес та легенд до них, контроль за допустимістю значень, забезпечення введення даних шляхом вибору зі списку.

Допоміжні процедури і функції – реалізація запитів, повідомлень, формування списків вибору IP-адрес та легенд, а також контроль за даними, що

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

вводяться.

Усі модулі в програмі зв'язані між собою за даними, що аналізуються на вході і виробляються на виході. Дані в модулі надходять через діалог з користувачем, параметри і документи інформаційної бази. Передача даних від одного модуля до іншого здійснюється тільки через збережені документи.

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

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

Обґрунтування інформаційного забезпечення

Перелік первісних даних

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

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

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

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

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

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

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

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

Сьогодні застосовуються два основні різновиди ІКМ: з кодуванням по m -закону й по A -закону. У результаті стиску сигнал з амплітудою, що кодується 12-13 бітами, описується всього вісьма бітами. Розрізняються ці різновиди ІКМ деталями процесу стиску (m -закон кодування переважніше використовувати при малій амплітуді сигналу й при малому відношенні сигнал/шум). У Північній Америці використовується кодування по m -закону, а в Європі – по A -закону. Тому при міжнародному зв'язку в багатьох випадках потрібне перетворення m -закону в A -закон, відповідальність за яке несе країна, у якій використовується m -закон кодування. В обох випадках кожний відлік кодується 8 бітами, або одним байтом, який можна вважати звуковим фрагментом. Для передачі послідовності таких фрагментів необхідна пропускна здатність каналу, рівна 64 Кбіт/с. Оскільки ІКМ була першою стандартною технологією, що одержала широке застосування в цифрових системах передачі, пропускна здатність каналу, рівна 64

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

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

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

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

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

Алгоритмом, побудованим на описані вище принципах, є алгоритм адаптивної диференціальної імпульсно-кової модуляції (АДІКМ) (G.726). Алгоритм передбачає формування сигналу помилки проорокування і його наступне адаптивне квантування. При досить гарних характеристиках алгоритму, АДІКМ практично не застосовується для передачі мови по мережах з комутацією пакетів, тому що цей алгоритм дуже чутливий до втрат цілих блоків відліку, що відбуваються при втратах пакетів у мережі. У таких випадках порушується синхронізація кодера й декодера, що приводить до катастрофічного погіршення якості відтворення мови навіть при малій імовірності втрат [11-27].

Перелік вихідних даних

У ході розробки корпоративної системи зв'язку з використанням ІР-технологій визначено, що вихідною інформацією є мова або відеозображення на пункті отримання інформації. Як може здатися на перший погляд, вузькополосне кодування мови, що вимагає обчислювальної потужності, є самим складним завданням, виконуваної устаткуванням ІР-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco. Однак це не так: алгоритми кодування мови стандартизовані й відмінно документовані, більше того, на ринку

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

доступні досить ефективні їхні реалізації для всіх популярних DSP-платформ. З іншого боку, в устаткуванні IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco повинні бути реалізовані багато інших функцій, спосіб реалізації яких не є об'єктом стандартизації.

На передавальній стороні устаткування IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco працює за принципом «закодував, передав і забув». На прийомній стороні все набагато складніше. Пакети приходять із мережі із затримкою, що міняється за випадковим законом. Більше того, пакети можуть прийти не в тій послідовності, у якій були передані, а деякі пакети можуть взагалі бути загублені. Приймач повинен справлятися з усіма цими труднощами, забезпечуючи на виході нормальний звуковий потік з тактовою синхронізацією, або генерованим на основі прийнятого потоку даних, або одержуваним із ТфОП по каналах Е1. Прив'язка мовних потоків до місцевого тактового синхросигналу здійснюється шляхом непомітної на слух деформації періодів мовчання у відтвореному сигналі. До цього залишається додати необхідність передачі факсимільної інформації в реальному часі з автоматичним розпізнаванням сигналів факсимільних апаратів і передачу DTMF-сигналів з коректним їхнім відновленням у приймачі.

Сигнали багаточастотного набору номера (DTMF) – просто звукові сигнали, передані по телефонному каналі. При передачі їх по цифровій телефонній мережі не виникає ніяких проблем, тому що кодування за допомогою алгоритму G.711 не накладає ніяких обмежень на вид звукових сигналів – це може бути мова, сигнали модему, або тональні сигнали – всі будуть успішно відтворені на приймачі [17-25].

Вузькополосні кодеки, щоб досягти низьких швидкостей передачі, використовують той факт, що сигнал, який вони кодують, представляє саме мову. Сигнали DTMF при проходженні через такі кодеки спотворюються й не можуть бути успішно розпізнані приймачем на прийомній стороні [27].

Коли користувачеві ТфОП потрібно ввести якусь додаткову інформацію у

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

віддалену систему при вже встановленому з'єднанні, необхідно забезпечити можливість надійної передачі DTMF-сигналів через мережу IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco. У випадках, коли система, взаємодіючи з користувачем, просто відносить запитання й чекає введення, тривалість і момент передачі сигналу не важливі. В інших випадках система видає користувачеві список і просить його нажати, наприклад, кнопку «#», як тільки він почує потрібну інформацію; тут ситуація більше складна, і необхідна більше точна прив'язка вчасно.

Існуючі методи передачі сигналів DTMF по мережах IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco [24,26].

– **Обов'язковий метод.** Спеціальне повідомлення протоколу H.245 може містити символи цифр і «*», «#». У цьому випадку використовується надійне TCP-з'єднання, так що інформація не може бути загублена. Однак через особливості TCP можуть мати місце значні затримки [24];

– **Нестандартний метод.** Він може бути застосований у терміналах H.323v2 при використанні процедури fastStart і відсутності каналу H.245. Для передачі сигналів DTMF відкривається спеціальна RTP-сесія, у якій передаються кодовані значення прийнятих цифр, а також дані про амплітуду й тривалість сигналів. Може бути використана та ж сесія, що й для мови, але зі спеціальним типом корисного навантаження. Використання RTP дозволяє прив'язати DTMF-сигнали до реального часу, що є важливою перевагою даного методу [26].

У принципі, перший метод може бути більше кращим, однак у випадку міжнародних викликів і при використанні віддалених систем, що вимагають твердої прив'язки введення користувача до часу, може виявитися необхідним застосувати другий метод. Шлюзи IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco повинні обов'язково придушувати переключені сигнали DTMF, що пройшли через основний мовний канал. У протилежному випадку, при відновленні сигналів, про які була прийнята інформація, можуть виникнути неприємні ефекти накладення й розмноження сигналів. На основі

даного огляду функцій устаткування IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco можна зробити вивід, про те що, незважаючи на існування стандартних алгоритмів кодування мови, у розроблювачів є величезний простір для діяльності, спрямованої на подальше вдосконалювання технології IP-телефонії інформаційно-комунікаційної платформи на базі рішень Cisco.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм RC5, який являє собою блоковий шифр із безліччю параметрів: розміром блоку, розміром ключа й числом раундів. В алгоритмі RC5 передбачені три операції: XOR, додавання й циклічні зрушення. На більшості процесорів операції циклічного зрушення виконуються за постійний час, змінні циклічні зрушення являють собою нелінійну функцію. Циклічні зрушення залежать як від ключа, так і від даних.

В RC5 використовується блок змінної довжини, але в приводиться прикладі, що буде розглянутий, 64-бітовий блок даних. Шифрування використовує $2r+2$ залежних від ключа 32-бітових слів – $S_0, S_1, S_2, \dots, S_{2r+1}$ – де r – число раундів. Для шифрування спочатку потрібно розділити блок відкритого тексту на два 32-бітових слова: A и B . (При впакуванні байтів у слова в алгоритмі RC5 дотримується угода про прямий порядок (little-endian) байтів: перший байт займає молодші біти регістра A й т. ін.) Потім:

$$A = A + S_0$$

$$B = B + S_0$$

Для i від 1 до r :

$$A = ((A \oplus B) \lll B) + S_{2i}$$

$$B = ((B \oplus A) \lll A) + S_{2i+1}$$

Вихід перебуває в регістрах A и B .

Розшифрування теж нескладно. Потрібно розбити блок відкритого тексту на два слова, A й B , а потім:

Для i від r до 1 із кроком -1:

$$B = ((B - S_{2i+1}) \gg\gg A) \oplus A$$

$$A = ((A - S_{2i}) \gg\gg B) \oplus B$$

$$B = B - S_i$$

$$A = A - S_0$$

Символом « $\gg\gg\gg$ » позначене циклічне зрушення вправо. Звичайно ж, всі додавання й вирахування виконуються по модулю 2^{32} .

Створення масиву ключів складніше, але теж прямолінійно. Спочатку байти ключа копіюються в масив L із 32-бітових слів, доповнюючи при необхідності заключне слово нулями. Потім масив S ініціалізується за допомогою лінійного конгруентного генератора по модулю 2^{32} :

$$S_0 = P$$

Для i від 1 до $2(r+1) - 1$:

$$S_i = (S_{i-1} + Q) \bmod 2^{32}$$

де $P = 0xb7e15163$ і $Q = 0x9e3779b9$.

Нарешті, потрібно підставити L в S :

$$i = j = 0$$

$$A = B = 0$$

Виконати $3n$ раз (де n – максимум від $2(r+1)$ і c):

$$A = S_i = (S_i + A + B) \ll\ll 3$$

$$B = L_i = (L_i + A + B) \ll\ll (A + B)$$

$$i = (i + 1) \bmod 2(r+1)$$

$$j = (j + 1) \bmod c$$

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Встановлення програмного додатку на автоматизовані робочі міста

Встановлення програмного додатку на ПК здійснюється простим копіюванням програмної папки IP_TEL_NET на жорсткий диск ПК. Нижче наведено вміст вказаної папки:

- IpTel.exe – виконавчий файл програми;
- abc.ttf – файл зі шрифтом;

Налаштування зв'язку між кінцевими абонентами

1. Після запуску програми на екран виводиться вікно завантаження програмного забезпечення та інформації про автора, яке спливає (рисунок 5.1).

2. Після цього необхідно пересвідчитися у наявності доступу до мережі. Для цього необхідно перевірити наявність мережної карти у персональному комп'ютері та підключеного шнура до мережі. Перевірити системні параметри мережної карти. Зайти в меню **Пуск**→**Налаштування**→**Мережні підключення**→**Налаштування мережних підключень**: вибрати: **IP-адрес, шлюз, маску під мережі, використовуваний протокол (TCP/IP)**.

На мережній карті повинна горіти зелена лампочка. Це каже про те, що фізично мережна карта працездатна та готова до роботи з мережею.

3. Потім необхідно перевірити параметри установки файрволу:

- системи дозволу IP-портів;
- привілеї користувача;
- поточний статус файрволу та перевірити наявність на одночасну роботу декількох файрволів на персональному комп'ютері.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76



Рисунок 5.1 – Вікно завантаження програмного забезпечення



Рисунок 5.2 – Вікно інформації про автора

4. Перевірити на наявність вірусів антивірусом та файрволом, або включити максимальний рівень захисту від вірусів. Антивіруси забезпечують найвищий рівень захисту. Виявляються й видаляються всі типи вірусів і троянських програм, ворожі об'єкти Java/Active, блокується доступ до небезпечних WEB-ресурсів. Захист від вірусів охоплює всі операційні системи й групові додатки, які використовуються в сучасних корпоративних мережах:

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

клієнтські ОС Windows 95/98 / ME / NT Workstation / 2000 Professional / XP Home & Professional/Vista, DOS, Macintosh; серверні ОС Windows NT Server, Windows 2000 Server / Advanced Server / Datacenter, Novell Netware 4.11 / 4.2 / 5.0 / 5.1 / 6.0, FreeBSD, Linux, HP-UX, AIX, SCO, Solaris; групові додатки MS Exchange 5.5/2000 і Lotus Notes/Domino; інтернет-шлюзи Windows NT/2000 і Sun Solaris; мережні пристрої зберігання даних NetApp; ОС мікрокомп'ютерів (PDA) Palm OS, Windows CE / Pocket PC, EPOC (Psion).

Антивірус призначений для захисту від вірусів файл-серверів. Файлові сервери – одне із самих уразливих місць корпоративної мережі: якщо поразка вірусом робочого місця приведе до відмови одного комп'ютера, то зараження файл-сервера може порушити роботу всієї мережі. Крім того, заражений сервер може зберігати працездатність, будучи джерелом вірусів для користувачів мережі. Антивірус встановлюється на сервер і у фоновому режимі перевіряє всі файли, до яких відбувається звертання від клієнтів. Він забезпечується останніми версіями антивірусних баз, автоматично роблячи відновлення через Інтернет або по локальній мережі. Крім того, антивірус здатний самостійно здійснювати регулярні перевірки дисків за розкладом, не вимагаючи участі адміністратора

Файрвол контролює весь трафік на робочій станції, дозволяючи тільки відомим додаткам підключатися до комп'ютера, блокуючи небезпечний трафік, що генерований хакерами, шкідливим кодом, агентами розподілених атак типу "відмова в обслуговуванні", уразливими або неавторизованими додатками й іншими типами атак, що піддають погрозі збереження даних і працездатність мережі. Адміністратор мережі може настроїти роботу файрвола прозоро для користувачів, так що користувач навіть не буде знати про присутність на його машині брандмауера.

Антивірусний монітор, повинен працювати в режимі реального часу. Всі операції читання або запису файлів відбуваються після перевірки їх на предмет вірусів. Така організація дозволяє автоматично захищати не тільки сервер від клієнтів, але й клієнтів від сервера, у випадку якщо він заражений вірусами.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Заражені файли автоматично виліковують, видаляються або переміщуються в карантинну зону, після чого адміністраторові відсилається повідомлення.

Сканер вірусів може бути використаний для ручної примусової перевірки дисків на предмет вірусів. За допомогою автоматичного розкладу можна робити регламентні перевірки всіх дисків сервера. Планувати такі операції можна на неробочий час із метою більше ефективного використання обчислювальних ресурсів процесора. Виявлені заражені файли автоматично виліковують, віддаляються або переміщуються в карантинну область, з наступним повідомленням адміністратора.

Антивірусний захист у фоновому режимі захищає робоче місце від можливого зараження вірусами в реальному часі. Він сканує файли, електронну пошту й WEB-ресурси безпосередньо при звертанні до них користувачів. У будь-який момент часу користувач або адміністратор мережі може перевірити локальні або мережні диски на предмет вірусів. Перевірка може бути запущена автоматично за розкладом.

Централізоване керування політикою захисту дозволяє адміністраторові мережі задавати стандартний набір правил (політику) роботи програм і впроваджувати їх по всій мережі підприємства або окремих його частин. Це значно заощаджує час адміністратора, рятуючи його від необхідності відвідувати кожне робоче місце, а також повністю виключає ймовірність зараження вірусами внаслідок некоректного налаштування антивірусних програм користувачами.

5. Якщо є IP-телефон, то підключити його згідно інструкції користувача, яка до нього додається.

6. Запустити розроблене у ході виконання магістерської роботи програмне забезпечення на кінцевих персональних комп'ютерах.

7. Визначити IP-адресу кінцевих машин. Для цього необхідно виконати наступні дії:

- Отримати інформаційні дані від адміністратора локальної мережі.
- У разі неможливості виконання попереднього пункту необхідно

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

виконати наступні дії: зайти на закладку: **Пуск**→**Налаштування**→**Мережні підключення** вибрати ярлик мережного підключення й натиснути на ньому правою кнопкою миші. У контекстному меню, яке з'явиться вибрати: **Властивість**→**Вибір пункту TCP/IP**→в закладці **IP-адрес**. Там буде відображено IP-адресу персонального комп'ютера у локальній мережі.

8. Заповнити базу даних IP-адрес та легенду до кожної з неї.

9. Після виконання усіх операцій, перерахованих вище, працюємо з розробленим програмним забезпеченням, згідно інструкції користувача.

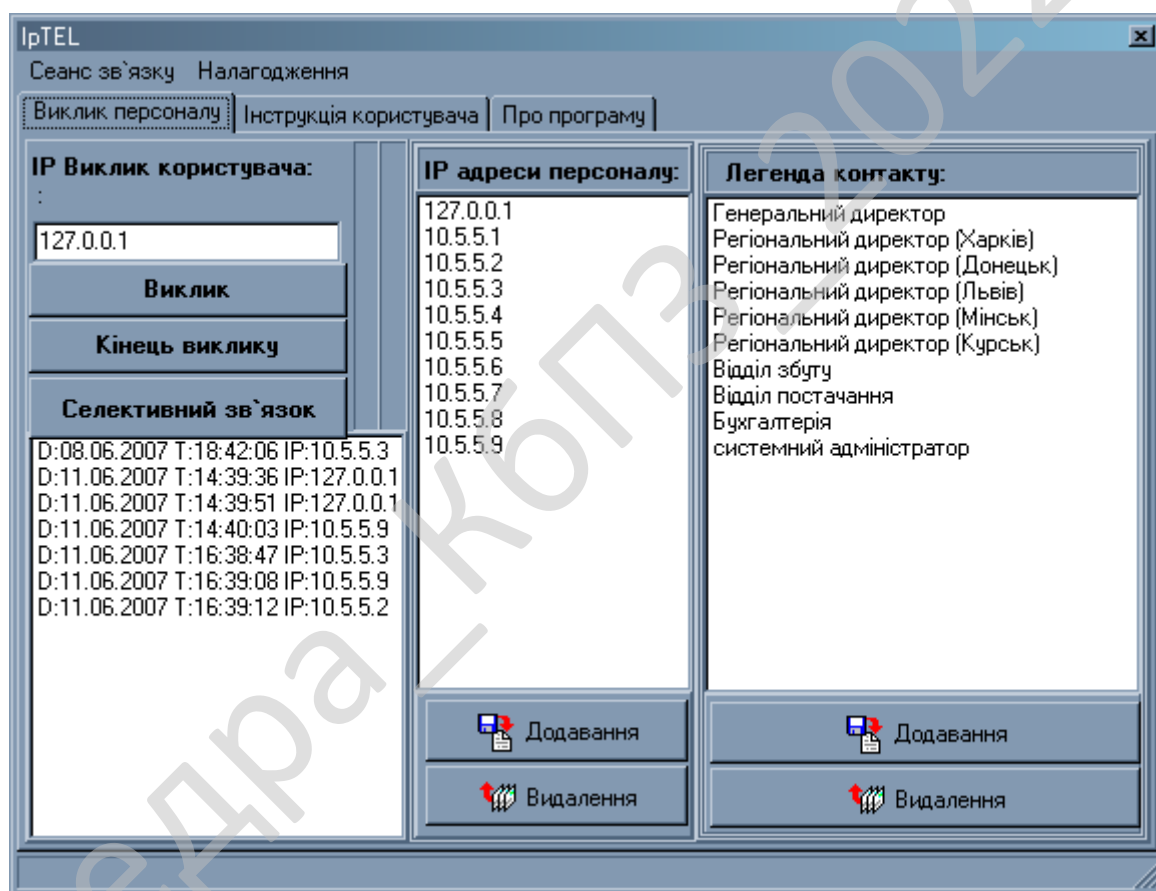


Рисунок 5.3 – Вікно з'єднання з абонентами (головна форма програми)

Вхід в систему

Система захищена програмним чином і її активізація здійснюється через введення пароля у відповідне поле та натисненням кнопки. По замовчуванню встановлено пароль «123». За умови коректного введення пароля (рисунок 5.5)

система надає користувачеві відповідний доступ для входу в систему, в іншому випадку програма закінчує свою роботу.

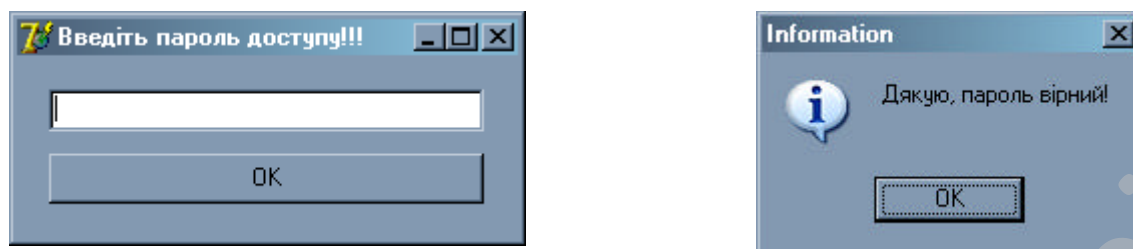


Рисунок 5.4 – Вікно правильного введення паролю

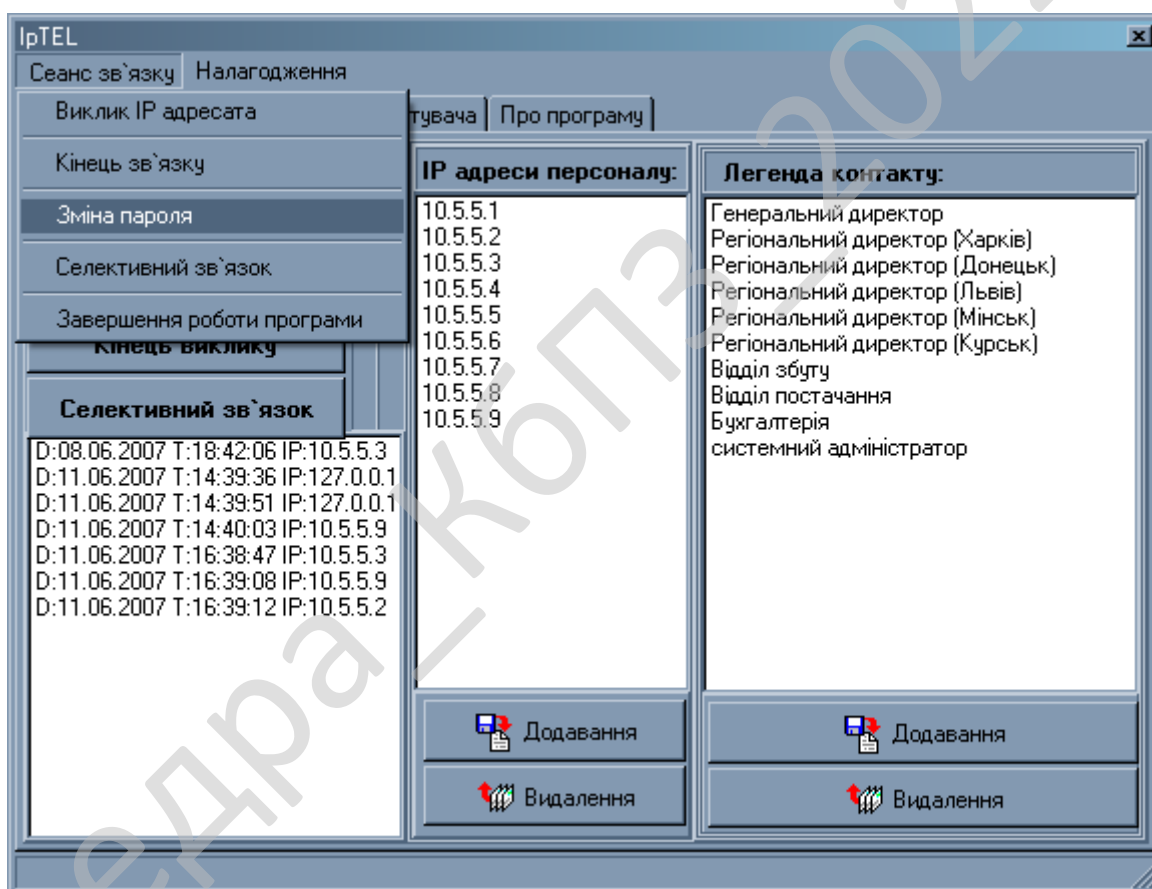


Рисунок 5.5 – Вхідження у режим зміни пароля

Існує можливість зміни паролю. Для цього у вікні зміни паролю необхідно ввести старий пароль, ввести новий пароль та продублювати новий пароль. Фіксація паролю здійснюється кнопкою “Ок”.

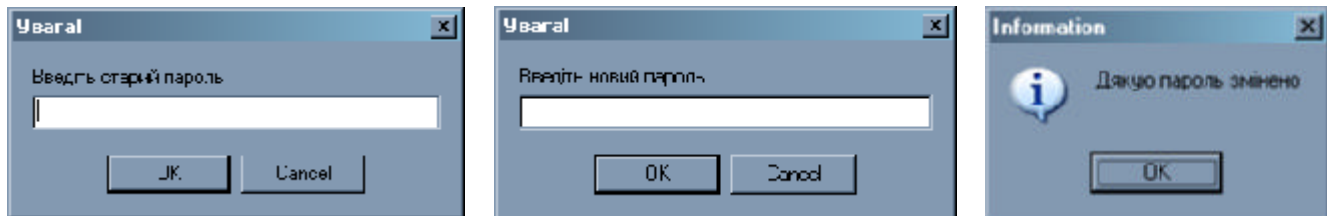


Рисунок 5.6 – Вікно зміни паролю

Організація бази клієнтів

В додатку є можливість редагування IP-адрес та легенди до них.

Зміни вносяться безпосередньо в сітку відповідних таблиць, їх фіксація здійснюється за допомогою кнопки **OK**.

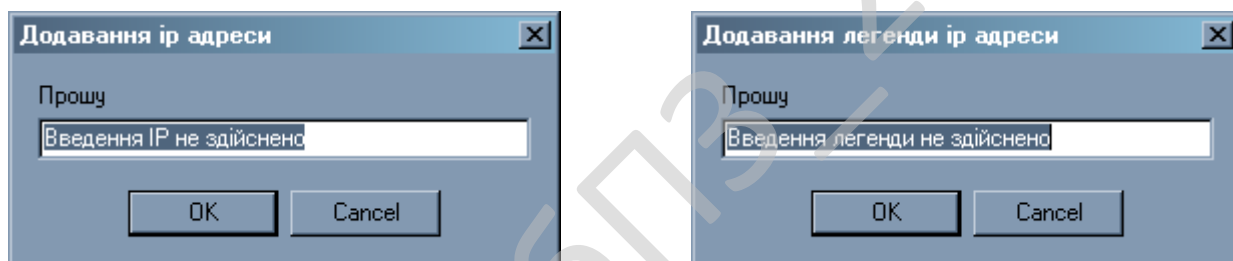


Рисунок 5.7 – Форма редагування адрес та легенд до них

Організація селективного зв'язку

Для організації селективного зв'язку необхідно за нажати клавішу Ctrl та за допомогою лівої кнопки миші виділити ті номери з якими потрібно організувати зв'язок (рисунок 5.8).

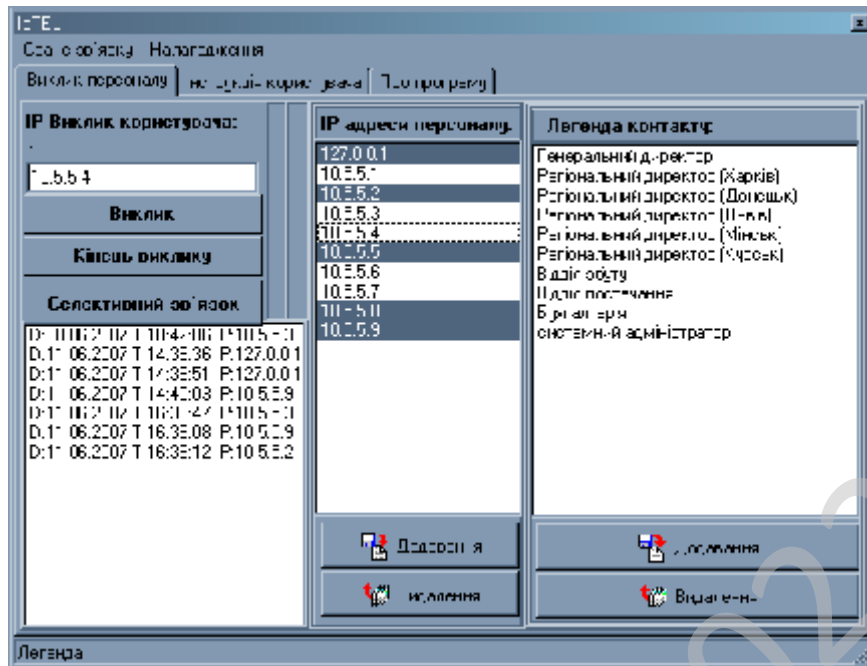
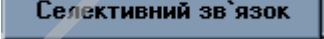


Рисунок 5.8 – Форма вибору номерів для селективного зв'язку

Після цього, необхідно натиснути клавішу . З'явиться наступне вікно:

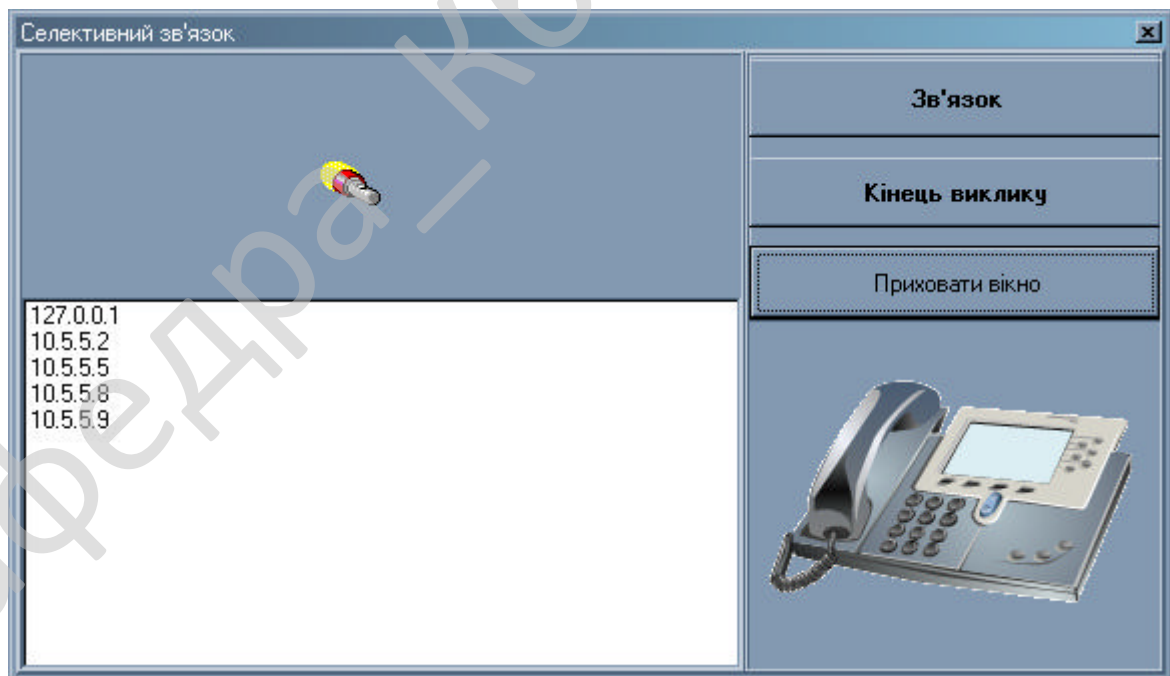


Рисунок 5.9 – Вікно селективного зв'язку

Можливі проблеми та їх усунення

Причинами неможливості доступу мережі можуть бути:

- проблеми з мережею;
- проблеми з мережною картою;
- неактивність сервера БД або некоректність його встановлення.

Перевірку працездатності мережної карти приведена у пункті 5.3.

Перевірка запуску сервера БД здійснюється через запуск менеджера сервера БД із панелі управління Windows

При цьому на ПК не повинно бути файрволів або проксі, які не дають йому приймати з'єднання на порту 3050 по протоколу tcp/ip (або файрвол/проксі потрібно налаштувати, щоб не заважали СУБД Interbase).

Після цього потрібно переконатись, що якщо у вас в мережі немає серверів Netware, то не один клієнт або сервер не має встановленої підтримки протоколів IPX/SPX (NWLink і т.п). Цей протокол не потрібний для роботи.

Взагалі, якщо ОС встановлена коректно, і мережа правильно налаштована, то InterBase або Firebird повинні також функціонувати нормально. З приводу налаштування мережі слід звертатись до її адміністратора або програмної документації ОС.

Причинами помилки можуть бути наступні:

- клієнтська частина (gds32.dll, fbclient.dll) не відповідає версії сервера.
- клієнтська частина не підтримує локальний протокол взагалі (наприклад, в Firebird 1.5.1 for Windows, Classic). Також локальний протокол не працює, якщо на Win 2003 або WinXP активізовані служби Terminal Services.
- особливості конкретного логіна або версії операційної системи.

У всіх випадках проблем з локальним протоколом рекомендується перевірити всі вище перераховані варіанти, і якщо їх не вдалось усунути – використовувати локальний мережевий протокол для з'єднання з БД.

									<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	ВКРМ-122.22.0001.00.00.ПЗ				84

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інформаційно-комунікаційної платформи на базі рішень Cisco.

Метою розробки є дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco.

Об'єктом дослідження є процес інформаційно-комунікаційної платформи на базі рішень Cisco.

Предметом дослідження є методи інформаційно-комунікаційної платформи на базі рішень Cisco.

Методи дослідження базуються на методах теорії обробки телекомунікаційного трафіку, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод інформаційно-комунікаційної платформи на базі рішень Cisco.

– Розроблено вітчизняний продукт інформаційно-комунікаційної платформи на базі рішень Cisco, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Проаналізувавши ринок програм аналогів була визначена ймовірна кількість програмних рішень. Також був підрахований необхідний час для розробки та впровадження програми і визначена орієнтовна вартість програмного забезпечення розробника. Час на розробку склав 60 днів (три місяці). В магістерській роботі було проведене дослідження та виконана програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме: невеликий розмір, висока швидкодія, невеликі системні потреби, незалежність від встановлених на комп'ютері баз даних, зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	100
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	5
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	100000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	19,5
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	50
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де: $\prod V_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33+0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 2,86 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 100 = 168 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	168	Ф 7.1-7.4
Впровадження	13	Д13
Всього	209	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{нз} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

$T_{нз}$ – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{209 \cdot 1}{60 - 5} = 3,8 \approx 3,5 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	12	1080	18
Монітор	60	12	720	12
Клавіатура	30	12	360	6
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	60	1	60	1
Принтер лазерний	120	2	240	4
Принтер струминний	60	1	60	1
Сканер	20	2	40	0,67
Концентратор-маршрутизатор	30	2	60	1
Кабельні господарства ЛОМ на 1 м. п.	2,5	100	250	4,17
Копіювальний апарат	140	2	280	4,67
Усього за рік:			3 _ч	58,51

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{др}^c = \frac{3_{ч} \cdot n_{міс}}{1,2}, \quad (7.6)$$

$$\Phi_{др}^c = \frac{59 \cdot 3}{1,2} = 147,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 147,5 / (60 \cdot 8) = 0,3 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2016, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	0,3	0,1
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,1	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,1	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	0,3	
Всього		0,8	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	0,4	0,1
	Підтримка постійних клієнтів	0,1	
	Оформлення договорів, ведення тендерів	0,1	
	Контроль взаєморозрахунків з постачальниками	0,2	
Всього		0,8	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	0,1	0,1
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,1	
	Розміщення графіки і контенту на Інтернет сторінках	0,1	
Всього		0,8	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	0,2	0,1
	Верстка друкованих видань	0,2	
	Додрукова підготовка макетів	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		0,8	

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Продакт-менеджер	0,1	13000	3900
Інженер-програміст	3,5	11000	115500
Інженер-електронщик	0,3	12000	10800
Інженер-системотехнік	0,1	12000	3600
Адміністратор мережі	0,1	12000	3600
Дизайнер WEB	0,1	14000	4200
Інженер-верстальник	0,1	12000	3600
Бухгалтер-економіст	0,1	13250	3975
Всього за період розробки	$R_{cn} = 4,4$	-	$\Phi_{роб} = 149175$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{сд} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{сд} = \frac{149175}{4,4 \cdot 60} = 565 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

$$B_{y\delta} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

S_y – питома площа на одне робоче місце, m^2 ;

$C_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 780...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 29000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 29000 = 1858000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 185800 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{нв} = R_{сн}^1 \cdot C_m, \quad (7.10)$$

де: C_m – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом Інтернет магазину Компбест за 31.10.22 – джерело <https://compbest.com.ua>.

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
1	2	3
Персональний комп'ютер		10947

Продовження таблиці 7.6

1	2	3
Системний блок		Fujitsu P720 Tower
Процесор	Intel Core i3-4130 (2 (4) ядра по 3.40 GHz); Cache Memory 3	-
Системна плата	Fujitsu D3221-A1 Intel Haswell с TDP до 95 Вт	-
Відеокарта	AMD Radeon RX 550 4GB GDDR5 Re Dragon PowerColor (AXRX 550 4GBD5 DH)	-
Жорсткий диск	HDD Seagate Barracuda 750 Gb 7200 32Mb SATAII ST3750528AS (ST3750528AS)	-
Оперативна пам'ять	DIMM 4096Mb DDR3 PC3-10600 CL9 Transcend JetRam, non-Reg., no-ECC , CL 9 (2 модулі)	-
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	-
Корпус	GRESSO GE-7525, 500W (120mm big fan), 2xIDE, full-ATX,БЖ 2xSATA, 1xFDD, Air Duct, 2xUSB 2.0, Mic+Audio, silver/black	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4 B, int. 3.5", 1*USB2.0+AUDIO+1394 multi: All Type Cards, black	220
інше	Клавіатура, мишка	Подарунок

Продовження таблиці 7.6

1	2	3
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000 1, 170/160, D-SUB, Wide)	3600
Принтер лазерний	HP LaserJet Pro 400 M401DN	6700
Принтер струминний	МФУ Epson M200	4800
Копіювальний апарат	МФУ HP LaserJet Enterprise M527dnm	16594

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	15	10947	16420,5	180625,5
Принтер лазерний	2	6700	1340	14740
Принтер струменевий	1	4800	480	5280
Копіювальний апарат	1	16594	1659,4	18253,4
Всього	–	–	–	218898,9

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1858000	-	-
2. Передавальні пристрої	185800	-	-
Всього по групі	2043800	5	102190
Група 4			
3. Обчислювальна техніка	218899	-	-
Всього по групі	218899	50	109449,5
4. Нематеріальні активи	100000	10	10000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	243000	20	48600
7. Господарський інвентар	28000	25	7000
Всього по групі	280031	-	57857,75
Разом	$K_p = 2642730$		$A_p = 279497,25$

Примітка: вартість автомобіля Sens (Standard+) взята по даним з автосалону, джерело [http://kirovograd-avto.ukravto.ua /catalog/tm-9/model-80/description](http://kirovograd-avto.ukravto.ua/catalog/tm-9/model-80/description), складає 243000 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 565 \cdot 209 / 100 = 1181 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 1181 \cdot 10 \cdot 0,01 = 118 \text{ грн.}$$

Відрахування на соціальні потреби та військовий збір за нормативом $H_c = 19,5\%$ від суми основної та додаткової зарплати:

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 19,5 (1181 + 118) = 253 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 1181 \cdot 15 \cdot 0,01 = 177 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.; Z_{M2} – вартість запам'ятовуючих пристроїв, грн.; Z_{M3} – вартість фарби, картриджей, тонеру, грн.; N_e – кількість екземплярів програм, шт.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

Кількість паперу визначаємо за потребою. В розрахунку приймаємо одну пачку паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 200$ грн., визначаємо вартість паперу за період розробки $N_m = 3$ міс:

$$Z_{M1} = C_n \cdot N_m, \quad (7.16)$$

$$Z_{M1} = 200 \cdot 3 = 600 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості, що дорівнює кількості екземплярів програм та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 20 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 35 грн./шт.

$$Z_{M2} = 20 \cdot 100 + 35 = 2035 \text{ грн.}$$

Згідно методики розрахунку, одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де: C_z – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для HP LaserJet Pro 400 M401DN – 600 грн.; чорнила для Epson M200 – 669 грн.; картриджу для HP LaserJet Enterprise M527, M527dn, M527f, M527c BASF 87X Black BASF-KT-CF287X – 2719 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (600 + 669 + 2719) / 100 = 40 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 1181 \cdot 15 \cdot 0,01 = 177 \text{ грн.}$$

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 100$ прим.):

$$A_m = \frac{A_p \cdot N_{\text{міс}}}{N_e \cdot 12}, \quad (7.20)$$

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 279497 \cdot 3 / (100 \cdot 12) = 699 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1. Основна зарплата виконавців	Z_o	1181
2. Додаткова зарплата виконавців	Z_o	118
3. Відрахування на соціальні потреби	C_{oc}	253
4. Загальногосподарські витрати	G_{ocn}	177
5. Витрати на матеріали	Z_M	40
6. Освоєння нових операційних систем, мов програмування	O_n	177
7. Амортизація основних фондів	A_m	699
8. Повна собівартість програмного забезпечення	C_n	2645
9. Плановий прибуток	P_p	1323
10. Ціна підприємства $C_n = C_n + P_p$	C_n	3968
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	793,6
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	4761,6

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 1181 + 118 + 253 + 177 + 40 + 177 + 699 = 2645 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 2645 = 1323 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.9.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	4762
Всього капітальних витрат	–	4762

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на підтримку	Z_p	153720	107604
2. Витрати на електроенергію	$Z_{ел}$	0	0
3. Витрати на амортизацію	$Z_{ам}$	0	1191
Всього витрат за рік	I	153720	108795

Витрати на підтримку модуля:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування кожного комп'ютера за рік, год.;

Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість годин обслуговування зменшилося з 2500 годин на рік до 1750 годин на рік і витрати на підтримку сервісу складуть:

$$Z_{p \text{ баз}} = 2500 \cdot 42 \cdot 1,2 \cdot 1,22 = 153720 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 1750 \cdot 42 \cdot 1,2 \cdot 1,22 = 107604 \text{ грн}.$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

Витрати на електроенергію не обраховується.

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	4762	–	1190,5
Всього відрахувань	-	–	4762	–	1190,5

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (C_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.; E_p – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (3968 - 2645) \cdot 100 - (0,05 \cdot 2043800 + 0,5 \cdot 218899 + 0,25 \cdot 37031 + 0,1 \cdot 100000 + 0,2 \cdot 243000) \cdot 3/12 = 62426 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p^*}{(C_n - C_n) \cdot N_e}, \quad (7.26)$$

де: K_p^* – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_6 = \frac{598930}{(3968-2645) \cdot 100 \cdot 12 / 3} = 1,13 \text{ років} .$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	100
2. Повна собівартість розробленої програми	Грн.	2645
3. Ціна розробленої програми	Грн.	3968
4. Плановий прибуток від реалізації розробленої програми	Грн.	1323
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2642730
7. Загальний прибуток від реалізації програмної продукції	Грн.	132300
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	62426
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,13
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	4762
11. Величина економічного ефекту у користувача програмної продукції	Грн.	43735
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,1

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n (K_n - K_{\bar{o}}), \quad (7.27)$$

де: $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (153720 - 108795) - 0,25 \cdot 4762 = 43735 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{4762}{153720 - 108795} = 0,1 \text{ року.}$$

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона здоров'я працівників, забезпечення безпеки умов праці, ліквідація професійних захворювань і виробничого травматизму повинна складати одну з головних завдань роботодавця.

Основою охорони праці є науковий аналіз умов праці, технологічних процесів, виробничого обладнання, робочих місць, трудових операцій, організації виробництва з метою виявлення шкідливих і небезпечних виробничих факторів, їх властивостей, особливостей впливу на організм людини. На підставі такого аналізу розробляються заходи та засоби, спрямовані на мінімізацію несприятливого впливу виробничих факторів, створення безпечних та нешкідливих умов праці.

Для того, щоб об'єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам, необхідно здійснити санітарно-гігієнічну характеристику умов праці відділу, в якому працює програміст, над розробкою даного програмного продукту.

В зв'язку з цим необхідно сконцентрувати увагу на небезпечних і шкідливих чинниках пов'язаних з постійною роботою за комп'ютером.

Електробезпека є одним із критичних питань для співробітників, що працюють із технікою, яка одержує живлення з електричної мережі. При невиконанні норм електробезпеки можлива поразка електричним струмом.

8.2 Характеристика умов праці програміста

Приміщення в якому проводиться розробка і дослідження програмного продукту за завданням наведені в таблиці 8.1

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

Таблиця 8.1 – Характеристика умов праці

Шкідливі та небезпечні фактори на робочому місці	Джерела утворення небезпек	Примітка (дані наведені для приміщення)
Електрична напруга вище 127В; Шум; Електромагнітні випромінювання; Статична електрика; Іонізація повітря; Пожежна безпека у приміщенні; Не якісне освітлення;	Кондиціонер 6 ПЕОП Принтер Папір Світильники	Розміри приміщення (м) Довжина – 4,4 Ширина – 3 Висота – 3 Кількість працюючих – 2

Згідно Державними санітарними правилами і нормами ДСанПіН 3.3.2.007-98 [2] може працювати 2 працівників. Мінімальна площа приміщення на 1 людину повинна складати не менше 6 м². Висновок – за умовами завдання це виконується.

В приміщенні відсутні умови, які можуть створювати підвищену або особливо підвищену небезпеку, тому воно відноситься до класу звичайних приміщень згідно ПУЕ. Джерелом живлення є трифазна мережа напруги 380/220 В з глухо заземленою нейтралі, з частотою 50 Гц згідно За пожежо-вибухонебезпекою приміщення відноситься до класу В. В таблиці 8.2 наведена загальна характеристика приміщення щодо вибухопожеженобезпеки та важкістю робіт.

Таблиця 8.2 – Загальна характеристика приміщення щодо вибухопожежонебезпеки та важкістю робіт

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

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

У таблиці 8.3 приведені оптимальні значення параметрів мікроклімату для категорії ваги робіт І-а, а також фактичні значення цих параметрів у розглянутому приміщенні. У приміщеннях з використанням обчислювальної техніки рекомендується застосування тільки оптимальних значень показників мікроклімату, тобто таких, при яких людина відчуває себе комфортно.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Вологість, %	Швидкість повітря, м/с	Температура, °С	Вологість, %	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,11
Тепла	23-25	50-70	0,1	24-25	50-65	0,1

Таким чином, показники мікроклімату в приміщенні, загалом, відповідають установленим нормам. В холодний період року використовується індивідуальне опалення, завдяки якому дотримується температурний режим в приміщенні в залежності від температури повітря навколишнього середовища.

Для підтримки температури в літню пору встановлений кондиціонер, який має достатню потужність по холоду, а для підтримки вологості є зволожувач повітря.

Джерелами запиленості повітря в приміщенні є одяг людей і пил, що проникає з вулиці. З метою боротьби з пилом робляться регулярні вологі прибирання і провітрювання.

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

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

Особливістю роботи за дисплеєм ЕОМ є постійна й значна напруга функцій зорового аналізатора, обумовленого необхідністю розходження самосвітних об'єктів (символів, знаків тощо) при наявності відблисків на екрані, рядковій структурі екрана, мерехтіння зображення, недостатньою чіткістю

Таблиця 8.5 – Рівні звукового тиску від різних джерел.

Джерело шуму	Рівень шуму, дБА
Жорсткий диск	45
Вентилятор	45
Принтер	55
Сканер	50

8.3 Розробка заходів з умов поліпшення охорони праці

У комплексі заходів, що сприяють удосконаленню організації праці програміста, збереженню здоров'я і підвищенню працездатності, його велике значення має організація робочих місць.

Проводячи аналіз умов праці в розглянутому приміщенні, ми одержали наступні результати:

- обсяг приміщення, що приходить на одному працюючого, відповідає нормативному значенню;
- показники мікроклімату відповідають нормативному значенню;
- акустичні умови роботи в нормі.

Важлива роль в ефективному забезпеченні праці належить моральному мікроклімату. Відносини працівників повинні ґрунтуватися на об'єктивності, доброзичливості, взаємодопомозі, глибокій повазі до кожного члена колективу, турботі про молодих співробітників.

Для забезпечення найбільш ефективного виконання обов'язків, плануючи розпорядок, слід дотримуватися таких принципів:

- для зняття втоми через кожні 1,5-2 год. робити перерви для відпочинку на 5-10 хв.;
- для усунення монотонності виконуваних робіт чергувати характер праці.

За умови неправильної організації праці та відпочинку, втома може нагромаджуватися щоденно й переходити в перевтому або захворювання. У зв'язку з цим режим праці та відпочинку користувачів ПК необхідно будувати з урахуванням працездатності, яка змінюється протягом доби.

Виходячи із наступного можна зробити висновок, що основною причиною втомлюваності та зниження працездатності працівника, який постійно працює за комп'ютером є психофізіологічний фактор, тому основною пропозицією правильна організація робочого місця з урахуванням ергономічних вимог, а також дотримання регламентованого режиму праці та відпочинку.

8.4 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: прокат сталевий круглого перерізу діаметром 40 мм., довжиною $L=3$ м., та горизонтальний електрод – металева полоса з перетином $40 \cdot 4$ мм. Напряга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – у ряд.

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта – чорнозем, нижнього шару ґрунта – глина (питомий опір $\rho_2 = 40$ Ом·м). Умовна товщина верхнього шару ґрунта: $H=0,7$ м. Відстань між вертикальними заземлювачами (електродами) $A=3$ м. Глибина закладення горизонтального контура заземлення $t=0,8$ м. Опір заземлювача, який нормується: $R_{3H} = 4$ Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Розрахунок.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,8+3/2=2,3 \text{ м.}$$

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

зайняті роботою з комп'ютерною технікою. Наведено приклад розрахунку ризику травмування людей, зайнятих певним видом діяльності. Можна зробити наступний висновок, що шкідливі та небезпечні виробничі фактори існують практично на будь якому робочому місці. Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів о 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.

Кафедра _ КБПЗ _ 2022 рік

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		116

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи інформаційно-комунікаційної платформи на базі рішень Cisco.

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 інформаційно-комунікаційної платформи на базі рішень Cisco.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем інформаційно-комунікаційної платформи на базі рішень Cisco.

– Досліджена система інформаційно-комунікаційної платформи на базі рішень Cisco.

– На основі отриманих результатів досліджень створена програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інформаційно-комунікаційної платформи на базі рішень Cisco.

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

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		117

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

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

Програма реалізована на мові високого рівня RAD Studio Delphi 10.4. 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 RC5.

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.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 43735 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,1 роки.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		118

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гут В.В. Дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco // Збірник праць молодих науковців ЦНТУ. – Вип. 13. – Кропивницький: ЦНТУ, 2022. Конахович Г.Ф. Сети передачи пакетных данных / Г.Ф. Конахович, В.М.Чуприн. – К.:МК-Пресс, 2006. – 272 с.

2. Королев А.В. Адаптивная маршрутизация в корпоративных сетях / А.В. Королев, Г.А. Кучук, А.А. Пашнев. – Х.: ХВУ, 2003. – 224 с.

3. Кучерявый Е.А. Управление трафиком и качество обслуживания в сети Интернет / Евгений Андреевич Кучерявый. – СПб.: Наука и техника, 2004. – 336 с.

4. Кучук Г.А. Управление ресурсами инфотелекоммуникаций / Г.А. Кучук, Р.П. Гахов, А.А. Пашнев. – М.: Физматлит, 2006. – 220 с.

5. Лагутин В.С., Степанов С.Н. Телетрафик мультисервисных сетей связи / В.С. Лагутин, С.Н. Степанов. – М.: Радио и связь, 2000. – 320 с.

6. Майника Э. Алгоритмы оптимизации на сетях и графах: пер. с англ. / Э. Майника; под ред. Е.К. Масловского. – М.: Мир, 1981. – 321 с.

7. Смирнов А.А. Разработка математической GERT-модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / А.А.Смирнов, Мохамад Гани Абу Таам // Информационные системы в управлении, образовании, промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2014. – 498 с.

8. Смирнов А.А. Метод управления доступом в интеллектуальных узлах коммутации / Мохамад Гани Абу Таам, А.А.Смирнов // Информационные технологии и защита информации в информационно-коммуникационных

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		119

системах: монографія / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2015. – 486 с.

9. Смирнов А.А. Математическая GERT-модель технологии передачи метаданных в облачные антивирусные системы / В.В.Босько, А.А.Смирнов, И.А.Березюк, Мохамад Гани Абу Таам // Збірник наукових праць "Системи обробки інформації". – Випуск 1(117). – Х.: ХУПС – 2014. – С. 137-141.

10. Смирнов А.А. Структурно-логическая GERT-модель технологии распространения компьютерных вирусов / А.А.Смирнов, И.А.Березюк, Мохамад Гани Абу Таам // Системи управління, навігації та зв'язку. – Випуск 1(29). – П.: ПНТУ. – 2014. – С. 120-125.

11. Смирнов А.А. Сравнительные исследования математических моделей технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Гани Абу Таам, А.А. Смирнов, А.В. Коваленко, С.А. Смирнов // Збірник наукових праць "Системи обробки інформації". – Випуск 9(125). – Х.: ХУПС – 2014. – С. 105-110.

12. Смирнов А.А. Математическая модель интеллектуального узла коммутации с обслуживанием информационных пакетов различного приоритета / Мохамад Гани Абу Таам, А.А. Смирнов, Н.С. Якименко, С.А. Смирнов // Збірник наукових праць Харківського університету Повітряних Сил. Випуск 4 (41). – Харків: ХУПС. – 2014. – С. 48-52.

13. Смирнов А.А. Исследование показателей качества функционирования интеллектуальных узлов коммутации в телекоммуникационных системах и сетях / Мохамад Гани Абу Таам, А.А. Смирнов, Н.С. Якименко, С.А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 4(17). – Харків: ХУПС. – 2014. – С.90-95.

14. Смирнов А.А. Усовершенствованный алгоритм управления доступом к «облачным» телекоммуникационным ресурсам / Мохамад Гани Абу Таам, А.А. Смирнов, Н.С. Якименко, С.А. Смирнов // Збірник наукових праць "Системи обробки інформації". – Випуск 1(126). – Х.: ХУПС – 2015. – С. 150-153.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		120

15. Смирнов А.А. Анализ и исследование методов управления сетевыми ресурсами для обеспечения антивирусной защиты данных / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Системи озброєння і військова техніка. – Випуск 3(43) – Х.: ХУПС – 2015. – С. 100-107.

16. Смирнов А.А. Исследование эффективности метода управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 3(19). – Х.: ХУПС. – 2015. – С. 134-141.

17. Mohamad Abou Taam Method of controlling access to intellectual switching nodes of telecommunication networks and systems / A.A. Smirnov, Mohamad Abou Taam, S.A. Smirnov // International Journal of Computational Engineering Research (IJCER). – Volume 5, Issue 5. – India. Delhi. – 2015. – P. 1-7.

18. Смирнов А.А. GERT-модель технологии передачи данных в облачные антивирусные системы / А.А. Смирнов, В.В. Босько, Мохамад Гани Абу Таам // Збірник тез доповідей науково-практичної конференції «Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку». м. Харків. 12-13 березня 2014 р. – Харків. АВВ МВС. – 2014. – С. 18-19.

19. Смирнов А.А. Математическое моделирование технологии передачи сигнатур в облачные антивирусные системы / Мохамад Гани Абу Таам, А.А. Смирнов // Збірник тез VI міжнародної науково-практичної конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 17-18 квітня 2014 р. – Харків: ХНЕУ. – 2014. – С. 260.

20. Смирнов А.А. Анализ требований к качеству обслуживания в информационно-телекоммуникационных системах / А.А. Смирнов, Мохамад Гани Абу Таам // Збірник тез XVI міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 11-12 квітня 2014 р. – Кіровоград: КНТУ. – 2014. – С. 124-126.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		121

21. Смирнов А.А. Дослідження та реалізація GERT-моделі технології розповсюдження комп'ютерних вірусів для захисту телекомунікаційних систем / Мохамад Гані Абу Таам, С.А. Смирнов // Збірник тез науково-практичної конференції «Інформаційні технології та комп'ютерна інженерія». м. Кіровоград. 4 грудня 2014 р. – Кіровоград: КНТУ. – 2014. – С. 168.

22. Смирнов А.А. Исследование математических моделей технологии распространения компьютерных вирусов / А.А. Смирнов, Мохамад Гани Абу Таам, С.А. Смирнов // Збірник наукових праць міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 25-28 лютого 2015 р. – Київ: Європейський університет. – 2015. – С. 90-91.

23. Смирнов А.А. Метод управления доступом к «облачным» ресурсам для защиты телекоммуникационных систем / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Збірник тез всеукраїнської науково-практичної конференції «Інформаційна безпека держави, суспільства та особистості». м. Кіровоград. 16 квітня 2015. – Кіровоград: КНТУ. – 2015. – С. 50-52.

24. Смирнов А.А. Разработка метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Гани Абу Таам, С.А. Смирнов // Збірник тез VII міжнародної науково-практичної конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 17-18 квітня 2015 р. – Харків: ХНЕУ. – 2015. – С. 14.

25. Смирнов А.А. Реализация метода управления доступом в интеллектуальных узлах коммутации / А.А. Смирнов, Мохамад Гани Абу Таам // Збірник тез XVII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 17-18 квітня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 91-92.

26. Смирнов А.А. Реализация математической модели интеллектуального узла коммутации для обеспечения защищенности телекоммуникационной сети / Мохамад Гани Абу Таам, А.А. Смирнов,

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		122

С.А. Смирнов // Збірник тез II Міжнародної науково-практичної Інтернет-конференції «Інформаційна та економічна безпека» (INFECO-2015)». м. Харків. 21-22 травня 2015 р. – Харків: ХІБС УБС НБУ. – 2015. – С. 20-24.

27. Смирнов А.А. Разработка математической модели технологии распространения компьютерных вирусов в информационно-телекоммуникационных сетях / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Сборник тезисов XI международной конференции "Стратегия качества в промышленности и образовании". г. Варна. Болгария. 01 – 06 июня 2015 г – Варна. ТУВ. – 2015. – С. 488-491

28. Смирнов А.А. Метод управления доступом к облачным телекоммуникационным ресурсам для обеспечения защиты данных / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Збірник тез Міжнародної науково-практичної конференції «Комп'ютерні технології та інформаційна безпека». м. Кіровоград. 2-3 липня 2015 р. – Кіровоград: КНТУ. – 2015. – С. 4-5.

29. Смирнов А.А. Имитационная модель системы управления доступом к облачным антивирусным телекоммуникационным ресурсам / Мохамад Гани Абу Таам, А.А. Смирнов, С.А. Смирнов // Збірник тез першої всеукраїнської науково-практичної конференції «Перспективні напрями захисту інформації». м. Затока. 7-9 вересня 2015 р. – Одеса: ОНАЗ. – 2015. – С. 90-94.

30. МСЭ-Т Рекомендация G.101. Международные телефонные соединения и цепи – Общие определения //11/2003. [Электронный ресурс]. – Режим доступа до ресурсу: [http://www. telecom61.ru/SharedFiles/Download.aspx? ...pageid=106](http://www.telecom61.ru/SharedFiles/Download.aspx?...pageid=106)

31. Одом Ш. Коммутаторы CISCO / Ш. Одом, Х. Ноттингем – М.: "Кудиц-Образ", 2003. – 528 с.

32. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов / В.Г. Олифер, Н.А. Олифер. – 2-е изд. – СПб.: Питер, 2007. – 958 с.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		123

33. Руководство по технологиям объединенных сетей. 4-е изд. / пер.с англ. и ред. А.Н. Крикуна – М.: Изд. дом «Вильямс», 2005. – 1040 с.

34. Свами М.Н., Тхуласираман К. Графы, сети и алгоритмы: пер. с англ. / М.Н. Свами, К. Тхуласираман; под ред. В.А. Горбатова. – М.: Мир, 1984. – 454 с.

35. Семенов С.Г. Анализ методов прогнозирования в телекоммуникационных сетях автоматизированных систем управления / С.Г.Семенов // Збірник наукових праць «Системи управління, навігації та зв'язку», – К.:ЦНДІ навігації і управління, – 2008.-Вип. 2(6) .- С.134-137

36. Семенов С.Г. Математическая модель процесса доставки информационных пакетов в компьютерной сети системы критического применения / С.Г.Семенов, И.В.Ильина // Науково-технічний журнал «Радіоелектронні і комп'ютерні системи» Х.:ХАІ, – 2008.-Вип. 1(28) – С.162-165

37. Семенов С.Г. Оптимизация трафика на основе сбалансированной загрузки информационно-телекоммуникационной сети // Системи обробки інформації. – Х.: ХВУ, 2004. – № 8(36). – С.206-210

38. Семенов С.Г. Математическая модель мультисервисного канала связи на основе экспоненциальной GERT-сети / С.Г. Семенов, Є.В. Мелешко, Я.В. Ілюшко // Системи озброєння і військова техніка. – Х.:ХУ ПС. – 2011. –Вип. 3(27). – С. 64-67.

39. Семенов С.Г. Математична модель системи криптографічного захисту електронних повідомлень на основі GERT-мережі / С.Г. Семенов, О.О. Сур // Системи управління, навігації та зв'язку. – К.:ЦНДІ навігації і управління. – 2012. – Том 1. Вип. 1(21). – С. 131-137

40. Семенов С.Г. Исследования вероятностно-временных характеристик мультисервисного канала связи с использованием математического аппарата GERT-сети / С.Г. Семенов, В.В. Босько, І.А.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		124

Березюк // Системи обробки інформації. – Х.: ХУ ПС. – 2012. – Том 1. Вип. 3(101). – С. 139-142.

41. Семенов С.Г. Моделирование защищенного канала связи с использованием экспоненциальной GERT-сети / С.Г. Семенов, А.А. Можаяев // Информатика, математическое моделирование, экономика. – Смоленськ.: Смоленский филиал АНО ВПО ЦС РФ "Российский университет кооперации". – 2012. – Том.1. – С. 152-160.

42. Семенов С.Г. Методика математического моделирования защищенной ИТС на основе многослойной GERT-сети / С.Г. Семенов // Вісник Національного технічного університету «Харківський політехнічний інститут». – Х.:НТУ «ХПІ». – 2012. –№62 (968). – С 173-181.

43. Семенов С.Г. Защита данных в компьютеризированных управляющих системах / С.Г. Семенов, В.В. Давыдов, С.Ю. Гавриленко. – LAP Lambert Academic Publishing GmbH & Co. KG (Саарбрюккен, Германия), 2014. – 236 с.

44. Смирнов А.А. Анализ и сравнительное исследование перспективных направлений развития цифровых телекоммуникационных систем и сетей / А.А.Смирнов, В.В.Босько, Е.В.Мелешко // Системи обробки інформації. – Х.: ХУ ПС, 2008. – Вип.7(74). – С.120-123.

45. Смирнов А.А. Усовершенствование метода управления очередями в многопротокольных узлах телекоммуникационной сети / А.А.Смирнов, Е.В.Мелешко // Збірник тез та доповідей другої всеукраїнської науково-практичної конференції «Системний аналіз. Інформатика. Управління». Запоріжжя. Тези доповідей. Запоріжжя: КПУ, 2011.

46. Современные телекоммуникации. Технологии и экономика / [В.Л. Банкет, О.В. Бондаренко, П.П. Воробьенко и др.]; под ред. С.А. Довгого. – М.: Эко-Трендз, 2003. – 320 с.

47. Столлингс В. Современные компьютерные сети / Вильям Столлингс.– СПб.: Питер, 2003. – 778 с.

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		125

48. Таненбаум Э. Компьютерные сети / Эндрю Таненбаум; пер. с англ. А. Леонтьев. – СПб.: Питер, 2002. – 848 с.
49. Телекоммуникационные системы и сети: учебное пособие. В 3 томах / [В.В. Величко, Е.А. Субботин, В.П. Шувалов, А.Ф. Ярославцев]; под ред. В.П. Шувалова. – М.: Горячая линия-Телеком, 2005, т. 3 – 592 с.
50. Уолрэнд Дж. Телекоммуникационные и компьютерные сети / Дж. Уолрэнд. – М.: Постмаркет, 2001. – 480 с.
51. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 1103 с.
52. Шелухин О.И. Фрактальные процессы в телекоммуникациях: моногр. / О.И. Шелухин, А.М. Тенякшев, А.В. Осин – М.: Радиотехника, 2003. – 480 с.
53. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ>
54. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>
55. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>
56. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.
57. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>
58. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ ІВМ сумісного типу / Кіровоград. ін-т с.-г. машинобуд.; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград:

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		126

КІСМ, 1997. – 20 с. Режим доступу до ресурсу:

<http://dspace.kntu.kr.ua/jspui/handle/123456789/4358>

59. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу:

<https://zakon.rada.gov.ua/rada/show/va042282-99>

60. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : “Колос”, 1973. – 238 с.

61. Центр післядипломної освіти та підвищення кваліфікації. – Режим доступу до ресурсу: <https://cpo.stu.cn.ua>

62. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення 19.09.22).

					ВКРМ-122.22.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		127

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-122.22.0001.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Гут В.В.				<i>Дослідження та програмна реалізація системи інформаційно- комунікаційної платформи на базі рішень Cisco</i>	Літ.	Аркуш	Аркушів
Перевірів	Коваленко О.В.					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КН-21М-1,4			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи інформаційно-комунікаційної платформи на базі рішень Cisco.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 18-13 від 17.08.2022 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи інформаційно-комунікаційної платформи на базі рішень Cisco.

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-122.22.0001.00.00.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 інформаційно-комунікаційної платформи на базі рішень Cisco;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-122.22.0001.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, 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 RAD Studio Delphi 10.4.

					ВКРМ-122.22.0001.00.00.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 розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2022 року.

8 Вимоги щодо охорони праці

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

					ВКРМ-122.22.0001.00.00.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 – 127 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

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 10.12.2022 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 20.12.2022 р.

					ВКРМ-122.22.0001.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

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

_____ Коваленко О.В.

*Дослідження та програмна реалізація
системи інформаційно-комунікаційної платформи на базі рішень Cisco*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 48

Літера: РП

Кропивницький – 2022 року

Основна програма – Програма організації зв'язку

```

program iptel;

uses
  Forms,
  Dialogs,
  Windows,
  Sysutils,
  Messages,
  usimple in 'usimple.pas' {Form1},
  Unit2 in 'Unit2.pas' {Form2},
  Unit3 in 'Unit3.pas' {Form3},
  Unit1 in 'Unit1.pas' {Form0},
  Unit4 in 'Unit4.pas' {Form4};
{$R simple.RES}
var
  i:integer;
{
  function Crypt(Text,Key: String; Encode: boolean): String;
  var
    i, KeyLength: integer;
    Sign: ShortInt;
  begin
    KeyLength:=Length(Key);
    if Encode then Sign :=-1 else Sign:=1;
    for i:=1 to Length(Text) do
      Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
    Result:=Text;
  end;}
begin
Application.Initialize;
{
if FileExists('main.dat')=false then
begin
  MessageDlg('Файл      main.dat      не      знайдено!      завершення
програми',mtInformation,[mbOK],0);
  Application.Terminate;
end else
begin
  AssignFile(F, 'main.dat');
  Reset(F);
  Readln(F, S);
  CloseFile(F);
  if Crypt(InputBox('Увага!', 'Введіть пароль', Y),KEY,false)=S then
begin
  MessageDlg('Дякую, пароль вірний!', mtInformation,[mbOK],0);}

```

```
Try
    Form2:=TForm2.Create(Application);
    Form2.Show;
    Form2.Update;
    for i:=1 to 10 do
    begin
        sleep(200);
        Form2.ProgressBar1.Position:=i*10;
        Form2.Update;
    end;
    Application.Title := 'IpTel';
Application.CreateForm(TForm1, Form1);
Application.CreateForm(TForm3, Form3);
Application.CreateForm(TForm0, Form0);
Application.CreateForm(TForm4, Form4);
Finally
    Form2.Free;
End;
Application.Run;
{
    end
    else
    begin
        MessageDlg('Невірний пароль!', mtInformation, [mbOK], 0);
        Application.Terminate;
    end;
end;}
end.
```

Підпрограма завантаження загрузочного вікна

```
unit U_SPLASH;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, jpeg, ExtCtrls, StdCtrls, Gauges;

type
    TForm_SPLASH = class(TForm)
        Image1: TImage;
        Label1: TLabel;
        Gauge1: TGauge;
        Timer1: TTimer;
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form_SPLASH: TForm_SPLASH;

implementation

{$R *.dfm}

end.
```

Головне вікно програми

```
unit usimple;

interface

uses

    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    telefoon, StdCtrls, ComCtrls, ExtCtrls, jpeg, Menus, Buttons;

type

    TForm1 = class(TForm)
        Telefoon1: TTelefoon;
        StatusBar1: TStatusBar;
        PageControl1: TPageControl;
        TabSheet1: TTabSheet;
        Panel1: TPanel;
        Panel3: TPanel;
        Panel4: TPanel;
        Panel2: TPanel;
        Label1: TLabel;
        Label4: TLabel;
        Edit1: TEdit;
        Button2: TButton;
        Button1: TButton;
        Panel5: TPanel;
        TabSheet2: TTabSheet;
        Panel6: TPanel;
        Label2: TLabel;
        Panel7: TPanel;
        MainMenu1: TMainMenu;
        Panel8: TPanel;
        ListBox1: TListBox;
        BitBtn1: TBitBtn;
        BitBtn2: TBitBtn;
        ListBox2: TListBox;
        Panel9: TPanel;
        Label3: TLabel;
        Panel10: TPanel;
        BitBtn3: TBitBtn;
        BitBtn4: TBitBtn;
        N1: TMenuItem;
        IP1: TMenuItem;
        N5: TMenuItem;
        N6: TMenuItem;
        N7: TMenuItem;
    end;
end;
```

```
N8: TMenuItem;
N9: TMenuItem;
N10: TMenuItem;
BitBtn5: TBitBtn;
Memol: TMemo;
TabSheet3: TTabSheet;
Panell1: TPanel;
Image3: TImage;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
N11: TMenuItem;
N12: TMenuItem;
ListBox3: TListBox;
Label9: TLabel;
Label10: TLabel;
N2: TMenuItem;
PB1: TProgressBar;
PB2: TProgressBar;
T1: TTimer;
Imagel: TImage;
Timer1: TTimer;

procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure ListBox1Click(Sender: TObject);
procedure ListBox2Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure N3Click(Sender: TObject);
procedure N6Click(Sender: TObject);
procedure N2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure T1Timer(Sender: TObject);
procedure Timer1Timer(Sender: TObject);

private
    { Private declarations }
public
    { Public declarations }
```

```

end;

var
    Form1: TForm1;

implementation

uses Unit3, Unit2, Unit4, Unit1;

var
    KEY:string='2#$T%&(*qwrda@@@#45';
{$R *.DFM}
procedure TForm1.Button1Click(Sender: TObject);
begin
    Telefoon1.placecall(Edit1.text);
    ListBox3.items.add('D:'+DateToStr(now)+'
T:'+TimeToStr(now)+'
IP:'+Edit1.text);
    T1.Enabled:=true;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
    Telefoon1.calling:=false;
    T1.Enabled:=false;
    PB1.Position:=0;
    PB2.Position:=0;
end;
procedure TForm1.ListBox1Click(Sender: TObject);
begin
    Edit1.text:=ListBox1.Items.Strings[ListBox1.ItemIndex];
end;
procedure TForm1.ListBox2Click(Sender: TObject);
begin
    Label4.Caption:=ListBox2.Items.Strings[ListBox2.ItemIndex];
end;
procedure TForm1.BitBtn1Click(Sender: TObject);
var
    InputString: string;
begin
    InputString:= InputBox('Додавання ip адреси', 'Прошу', 'Введення IP не
здійснено');
    if (InputString<>'Введення IP не здійснено') then
    begin
        Listbox1.Items.add(InputString);
    end;
end;
procedure TForm1.BitBtn3Click(Sender: TObject);
var

```

```

    InputString:= string;
begin
    InputString:= InputBox('Додавання легенди ip адреси', 'Прошу', 'Введення
легенди не здійснено');
    if (InputString<>'Введення легенди не здійснено') then
    begin
        Listbox2.Items.add(InputString);
    end;
end;
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
    ListBox1.Items.Strings[ListBox1.ItemIndex]:= '';
    ListBox1.Items.Delete(ListBox1.ItemIndex);
end;

procedure TForm1.BitBtn4Click(Sender: TObject);
begin
    ListBox2.Items.Strings[ListBox1.ItemIndex]:= '';
    ListBox2.Items.Delete(ListBox1.ItemIndex);
end;
procedure TForm1.N10Click(Sender: TObject);
begin
    Application.Terminate;
end;
procedure TForm1.BitBtn5Click(Sender: TObject);
var
    i:integer;
    G:boolean;
begin
    Form3.ListBox1.Clear;
    g:=false;
    for i:=0 to (ListBox1.Items.Count - 1) do
    begin
        if ListBox1.Selected[i] then
        begin
            Form3.ListBox1.Items.add(ListBox1.Items.Strings[i]);
            g:=true;
        end;
        if g then Form3.show;
    end;
end;
procedure TForm1.N12Click(Sender: TObject);
var
    F: TextFile;
    H,S:string;
    OLD:string;

```

```

function Crypt(Text,Key: String; Encode: boolean): String;
var
  i, KeyLength: integer;
  Sign: ShortInt;
begin
  KeyLength:=Length(Key);
  if Encode then Sign :=-1 else Sign:=1;
  for i:=1 to Length(Text) do
    Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
    Result:=Text;
  end;
begin
if FileExists('main.dat') then
begin
  AssignFile(F, 'main.dat');
  Reset(F);
  Readln(F, S);
  CloseFile(F);
if Crypt(TextBox('Увага!', 'Введіть старий пароль', ''),KEY,false)=S then
begin
  H:=Crypt(TextBox('Увага!', 'Введіть новий пароль', ''),KEY,false);
  if H<>' ' then
  begin
    MessageDlg('Дякую пароль змінено',mtInformation,[mbOK],0);
    AssignFile(F, 'main.dat');
    Rewrite(F);
    Writeln(F,H);
    CloseFile(F);
  end else MessageDlg('Введіть значення!',mtInformation,[mbOK],0);
end
else
begin
  MessageDlg('Файл main.dat не знайдено чи пароль невірний! завершення програми',mtInformation,[mbOK],0);
  Application.Terminate;
end;
end;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
  randomize;
  PB1.Position:=0;
  PB2.Position:=0;
  T1.Enabled:=false;
  Timer1.Enabled:=true;
  if FileExists('MainLegend.dat')=false then

```

```

begin
    MessageDlg('Файл MainLegend.dat не найдено!',mtInformation,[mbOK],0);
end else
begin
    ListBox2.Items.LoadFromFile('MainLegend.dat');
end;
if FileExists('MainIP.dat')=false then
begin
    MessageDlg('Файл MainIP.dat не найдено!',mtInformation,[mbOK],0);
end else
begin
    ListBox1.Items.LoadFromFile('MainIP.dat');
end;
if FileExists('MainHISTORY.dat')=false then MessageDlg('Файл MainHISTORY.dat
не найдено!',mtInformation,[mbOK],0)
else ListBox3.Items.LoadFromFile('MainHISTORY.dat');
if FileExists('mainHelp.dat')=false then MessageDlg('Файл mainHelp.dat не
найденно!',mtInformation,[mbOK],0)
else Memo1.Lines.LoadFromFile('mainHelp.dat');
end;
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
if FileExists('MainLegend.dat')=false then MessageDlg('Файл MainLegend.dat не
найденно!',mtInformation,[mbOK],0)
else ListBox2.Items.SaveToFile('MainLegend.dat');
if FileExists('MainIP.dat')=false then MessageDlg('Файл MainIP.dat не
найденно!',mtInformation,[mbOK],0)
else ListBox1.Items.SaveToFile('MainIP.dat');
if FileExists('MainHISTORY.dat')=false then MessageDlg('Файл MainHISTORY.dat
не найдено!',mtInformation,[mbOK],0)
else ListBox3.Items.SaveToFile('MainHISTORY.dat');
end;
procedure TForm1.N3Click(Sender: TObject);
begin
TabSheet2.Visible:=true;
end;
procedure TForm1.N6Click(Sender: TObject);
begin
    Telefoon1.calling:=false
end;
procedure TForm1.N2Click(Sender: TObject);
begin
Form0.show;
end;
procedure TForm1.FormShow(Sender: TObject);
begin

```

```
PB1.Position:=0;
PB2.Position:=0;
end;
procedure TForm1.T1Timer(Sender: TObject);
var
i1,i2:integer;
begin
PB1.Position:=random(100);
PB2.Position:=PB1.Position;
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
Form4.show;
Form1.hide;
Timer1.Enabled:=false;
end;
end.
```

Кафедра _ КБПЗ _ 2022 рік

Підпрограма налагодження звукової карти AudioSettings

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ComCtrls, StdCtrls, AMixer, MMSystem;
type
  TForm0 = class(TForm)
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    TrackBar: TTrackBar;
    CheckBox: TCheckBox;
    Label1: TLabel;
    Label2: TLabel;
    Mixer: TAudioMixer;
    Label3: TLabel;
    Label4: TLabel;
    ComboBox3: TComboBox;
    LabelStereo: TLabel;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure ComboBox2Change(Sender: TObject);
    procedure MixerControlChange(Sender: TObject; MixerH, ID: Integer);
    procedure TrackBarChange(Sender: TObject);
    procedure CheckBoxClick(Sender: TObject);
    procedure ComboBox3Change(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
    Setting: Boolean;
  public
    { Public declarations }
  end;
var
  Form0: TForm0;
implementation
uses usimple;
{$R *.DFM}
procedure TForm0.FormCreate(Sender: TObject);
var A: Integer;
begin
  For A := 0 to Mixer.MixerCount - 1 do
    ComboBox3.Items.Add ('Mixer '+IntToStr(A));

```

```

If (ComboBox3.Items.Count > 0) then
    ComboBox3.ItemIndex := 0;
ComboBox3Change (Sender);
end;
procedure TForm0.ComboBox1Change(Sender: TObject);
var A:Integer;
begin
    ComboBox2.Items.Clear;
    ComboBox2.Items.Add
(Mixer.Destinations[ComboBox1.ItemIndex].Data.szName);
    For A:=0 to Mixer.Destinations[ComboBox1.ItemIndex].Connections.Count-1 do
ComboBox2.Items.Add(Mixer.Destinations[ComboBox1.ItemIndex].Connections[A].Dat
a.szName);
    If ComboBox2.Items.Count>0 then
    begin
        ComboBox2.ItemIndex:=0;    ComboBox2Change (Self);
    end;
end;
procedure TForm0.ComboBox2Change(Sender: TObject);
var L,R,M:Integer;
    VD,MD:Boolean;
    Stereo:Boolean;
    IsSelect:Boolean;
begin
    Mixer.GetVolume (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,L,R,M,Stereo,VD,MD,IsSelect);
    Setting:=True;
    TrackBar.Visible:=not VD;
    Label1.Visible:=not VD;
    Label3.Visible:=VD;
    If TrackBar.Visible then
        TrackBar.Position:=L;
    CheckBox.Visible:=not MD;
    Label2.Visible:=not MD;
    Label4.Visible:=MD;
    If CheckBox.Visible then
        CheckBox.Checked:=M<>0;
    If (Stereo) then
        LabelStereo.Caption := '- stereo -'
    else
        LabelStereo.Caption := '- mono -';
    Setting:=False;
end;
procedure TForm0.MixerControlChange(Sender: TObject; MixerH, ID: Integer);
begin
    ComboBox2Change (Self);
end;

```

```

procedure TForm0.TrackBarChange(Sender: TObject);
begin
  If (not Setting) then
  begin
    Setting:=True;
    Mixer.SetVolume          (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer (CheckBox.Checked));
    Setting:=False;
  end;
end;
procedure TForm0.CheckBoxClick(Sender: TObject);
begin
  If not Setting then
  begin
    Setting:=True;
    Mixer.SetVolume          (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer (CheckBox.Checked));
    Setting:=False;
  end;
end;
procedure TForm0.ComboBox3Change(Sender: TObject);
var A:Integer;
begin
  If (ComboBox3.ItemIndex >= 0) AND (ComboBox3.ItemIndex < Mixer.MixerCount)
then
  Mixer.MixerId := ComboBox3.ItemIndex;
  ComboBox1.Items.Clear;
  If Mixer.MixerCount>0 then
  begin
    For A:=0 to Mixer.Destinations.Count-1 do
      ComboBox1.Items.Add (Mixer.Destinations[A].Data.szName);
    If ComboBox1.Items.Count>0 then
    begin
      ComboBox1.ItemIndex:=0;    ComboBox1Change (Self);
    end;
  end
else
  begin
    ComboBox1.OnChange:=nil;    ComboBox2.OnChange:=nil;
    TrackBar.OnChange:=nil;    CheckBox.OnClick:=nil;
    MessageDlg ('No mixer present in the system !',mtError,[mbOK],0);
  end;
  Setting:=False;
end;
procedure TForm0.Button1Click(Sender: TObject);
begin

```

```
Form0.hide; Form1.show;  
end;  
procedure TForm0.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
Form0.hide; Form1.show;  
end;  
end.
```

Кафедра _ КБПЗ _ 2022 рік

Підпрограма виклику вікна налагодження звукової карти AudioSettings

```
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  jpeg, ExtCtrls, StdCtrls, ComCtrls;

type
  TForm2 = class(TForm)
    Image1: TImage;
    RichEdit1: TRichEdit;
    ProgressBar1: TProgressBar;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.DFM}

end.
```

Підпрограма селективного зв'язку

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls, jpeg, ComCtrls;

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    ListBox1: TListBox;
    Animat1: TAnimate;
    Panel4: TPanel;
    BitBtn2: TBitBtn;
    BitBtn1: TBitBtn;
    BitBtn3: TBitBtn;
    Image1: TImage;
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

uses usimple;

{$R *.DFM}

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
  form3.hide;
end;

procedure TForm3.BitBtn1Click(Sender: TObject);
```

```
begin
Animatel.Active:=true;
form1.telefoon1.placecall(form1.edit1.text);
end;

procedure TForm3.BitBtn3Click(Sender: TObject);
begin
Form1.Telefoon1.calling:=false;
Animatel.Active:=false;
end;

end.
```

Кафедра _ КБПЗ _ 2022 рік

Підпрограма парольного захисту

```

unit Unit4;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, Mask, ExtCtrls;
type
  TForm4 = class(TForm)
    Panel1: TPanel;
    M: TMaskEdit;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form4: TForm4;
implementation
uses usimple;
VAR
  KEY:string='2#%$T%&(*qwrda$@@@#45';
  DAT:string='VUW';
{$R *.DFM}

  function Crypt(Text,Key: String; Encode: boolean): String;
  var
    i, KeyLength: integer;
    Sign: ShortInt;
  begin
    KeyLength:=Length(Key);
    if Encode then Sign :=-1 else Sign:=1;
    for i:=1 to Length(Text) do
      Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
    Result:=Text;
  end;
procedure TForm4.BitBtn1Click(Sender: TObject);
var
  F: TextFile;
  i:integer;
  s:string;
  Y:string;
  UUU:string;
begin
  if FileExists('main.dat')=TRUE then

```

```
begin
  AssignFile(F, 'main.dat');
  Reset(F);
  Readln(F, S);
  CloseFile(F);
UUU:=Crypt(M.text,KEY,false);
if UUU=S then
  begin
    Form4.hide;
    Form1.show;
    MessageDlg('Дякую, пароль вірний!',mtInformation,[mbOK],0);
  end
  else MessageDlg('Введіть пароль!',mtInformation,[mbOK],0);
end
else begin
  MessageDlg('Файл main.dat не знайдено! завершення
програми',mtInformation,[mbOK],0);
  Application.Terminate;
end;
end;
end.
```

Кафедра _ КБПЗ _ 2022 рік

Розроблений компонент налагодження звука

```

unit AMixer;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  MMSystem;

type
  TAudioMixer=class;

  TPListFreeItemNotify=procedure (Pntr:Pointer) of object;
  TMixerChange=procedure (Sender:TObject;MixerH:HMixer;ID:Integer) of object;
    {MixerH is handle of mixer, which sent this message.
     ID is ID of changed item (line or control).}

  TPointerList=class(TObject)
  private
    FOnFreeItem:TPListFreeItemNotify;
    Items:Tlist;
  protected
    function GetPointer (Ind:Integer):Pointer;
    function GetCount :integer;
  public
    constructor Create;
    destructor Destroy; override;
    procedure Clear;
    procedure Add (Pntr:Pointer);
    property Count:Integer read GetCount;
    property Pointer[Ind:Integer]:Pointer read GetPointer; default;
    property OnFreeItem:TPListFreeItemNotify read FOnFreeItem write
FOnFreeItem;
  end;

  TMixerControls=class(TObject)
  private
    heap:pointer;
    FControls:TPointerList;
  protected
    function GetControl (Ind:Integer):PMixerControl;
    function GetCount:Integer;
  public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;

```

```

    property Control[Ind:Integer]:PMixerControl read GetControl; default;
    property Count:Integer read GetCount;
end;

```

```

TMixerConnection=class(TObject)

```

```

private

```

```

    XMixer:TAudioMixer;
    FData:TMixerLine;
    FControls:TMixerControls;

```

```

public

```

```

    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Controls:TMixerControls read FControls;
    property Data:TMixerLine read FData;

```

```

end;

```

```

TMixerConnections=class(TObject)

```

```

private

```

```

    XMixer:TAudioMixer;
    FConnections:TPointerList;

```

```

protected

```

```

    procedure DoFreeItem (Pntr:Pointer);
    function GetConnection (Ind:Integer):TMixerConnection;
    function GetCount:Integer;

```

```

public

```

```

    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Connection[Ind:Integer]:TMixerConnection read GetConnection;

```

```

default;

```

```

    property Count:Integer read GetCount;

```

```

end;

```

```

TMixerDestination=class(TObject)

```

```

private

```

```

    XMixer:TAudioMixer;
    FData:TMixerLine;
    FControls:TMixerControls;
    FConnections:TMixerConnections;

```

```

public

```

```

    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Connections:TMixerConnections read FConnections;
    property Controls:TMixerControls read FControls;
    property Data:TMixerLine read FData;

```

```

end;

```

```

TMixerDestinations=class(TObject)
private
    FDestinations:TPointerList;
protected
    function GetDestination (Ind:Integer):TMixerDestination;
    procedure DoFreeItem (Pntr:Pointer);
    function GetCount:Integer;
public
    constructor Create (AMixer:TAudioMixer);
    destructor Destroy; override;
    property Count:Integer read GetCount;
    property Destination[Ind:Integer]:TMixerDestination read GetDestination;
default;
end;

TAudioMixer = class(TComponent)
private
    XWndHandle:HWND;

    FDestinations:TMixerDestinations;
    FMixersCount:Integer;
    FMixerHandle:HMixer;
    FMixerId:Integer;
    FMixerCaps:TMixerCaps;
    FDriverVersion: MMVERSION;
    FManufacturer: String;
    FProductId: Word;
    FNumberOfLine: Integer;
    FProductName: String;
    FOnLineChange:TMixerChange;
    FOnControlChange:TMixerChange;
protected
    procedure SetMixerId (Value:Integer);
    procedure MixerCallBack (var Msg:TMessage);
    procedure CloseMixer;
published
    constructor Create (AOwner:TComponent); override;
    destructor Destroy; override;
    property DriverVersion: MMVERSION read FDriverVersion;
    property ProductId: WORD read FProductId;
    property NumberOfLine: Integer read FNumberOfLine;
    property Manufacturer: string read FManufacturer;
    property ProductName: string read FProductName;
    property MixerId:Integer read FMixerId write SetMixerId;
    {Opened mixer - value must be in range 0..MixersCount-1
     If no mixer is opened this value is -1}

```

```

property OnLineChange:TMixerChange read FOnLineChange write FOnLineChange;
property OnControlChange:TMixerChange read FOnControlChange write
FOnControlChange;
public
function GetVolume (ADestination, AConnection:Integer; var LeftVol,
RightVol, Mute:Integer; var Stereo, VolDisabled, MuteDisabled,
MuteIsSelect:Boolean):Boolean;
    {This function return volume of selected Destination and Connection.
    ADestination must be from range 0..Destinations.Count-1
    AConnection must be in range
0..Destinations[ADestination].Connections.Count-1
    If you want to read master volume of some Destination, you have to
    set AConnection to -1.
    If LeftVol, RightVol or Mute is not supported by queried connection,
    it's return value will be -1.

    LeftVol and RightVol are in range 0..65536

    If Mute is non-zero then the connection is silent (or vice-versa - see
MuteIsSelect parameter)
    If specified line is recording source then Mute specifies if programs
will
    record from this connection (it is copy of "Select" Checkbox in
    standard Windows Volume Control program)
    Stereo is true, then this control is stereo.
    VolDisabled or MuteDisabled is True when you cannot apply settings to
this
    control (but can read it).
    MuteIsSelect returns True is "mute" work here as select - opposite of
mute.

    Return value of the function is True if no error has occurred,
    otherwise it returns False.}
function SetVolume (ADestination, AConnection:Integer; LeftVol, RightVol,
Mute:Integer):Boolean;
    {This function sets volume.
    If you set RightVol to -1 and connection is stereo then LeftVol will be
    copied to RightVol.
    If LeftVol or Mute is -1 then this value will not be set.
    Note that "Mute" can be "select" (which is reversed mute) - see
function
    GetVolume, parameter MuteIsSelect.

    Return value is True if ADestination and AConnection are correct,
    otherwise False.}

```

```

function GetPeak(ADestination, AConnection:Integer; var LeftPeak,
RightPeak:Integer):Boolean;

function GetMute (ADestination, AConnection:Integer; var
Mute:Boolean):Boolean;

function SetMute (ADestination, AConnection:Integer; Mute:Boolean):Boolean;

property Destinations:TMixerDestinations read FDestinations;
  {Ind must be in range 0..DestinationsCount-1}
property MixerCaps:TMixerCaps read FMixerCaps;
property MixerCount:Integer read FMixersCount;
  {Number of mixers present in system; mostly 1}
property MixerHandle:HMixer read FMixerHandle;
  {Handle of opened mixer}
end;

procedure Register;

implementation

{-----}
{TPointerList}
{-----}

constructor TPointerList.Create;
begin
  Items := TList.Create;
end;

destructor TPointerList.Destroy;
begin
  Clear;
  Items.Free;
end;

procedure TPointerList.Add (Ptr:Pointer);
begin
  Items.Add (Ptr);
end;

function TPointerList.GetPointer (Ind:Integer):Pointer;
begin
  Result := nil;
  If (Ind < Count) then
    Result := Items[Ind];
end;

```

```

procedure TPointerList.Clear;
var I:Integer;
begin
  for I := 0 to Items.Count-1 do begin
    If Assigned (FOnFreeItem) then
      FOnFreeItem (Items[I])
    end;
    Items.Clear;
  end;

function TPointerList.GetCount:Integer;
begin
  Result := Items.Count;
end;

{-----}
{TMixerControls}
{-----}
constructor TMixerControls.Create (AMixer:TAudioMixer; AData:TMixerLine);
var MLC:TMixerLineControls;
    A,B:Integer;
    P:PMixerControl;
begin
  FControls := TPointerList.Create;
  GetMem (P, SizeOf(TMixerControl)*AData.cControls);
  heap := P;
  MLC.cbStruct := SizeOf(MLC);
  MLC.dwLineID := AData.dwLineID;
  MLC.cbmxcctrl := SizeOf(TMixerControl);
  MLC.cControls := AData.cControls;
  MLC.pamxcctrl := P;
  A := MixerGetLineControls (AMixer.MixerHandle, @MLC,
MIXER_GETLINECONTROLSF_ALL);
  If A = MMSYSERR_NOERROR then
  begin
    For B := 0 to AData.cControls-1 do
    begin
      FControls.Add (P);
      P := PMixerControl (DWORD(P) + sizeof (TMixerControl));
    end;
  end;
end;

destructor TMixerControls.Destroy;
begin
  FControls.free;

```

```

    freemem(heap);
    inherited;
end;

function TMixerControls.GetControl (Ind:Integer):PMixerControl;
begin
    Result := FControls.Pointer[Ind];
end;

function TMixerControls.GetCount:Integer;
begin
    Result := FControls.Count;
end;

{-----}
{TMixerConnection}
{-----}

constructor TMixerConnection.Create (AMixer:TAudioMixer; AData:TMixerLine);
begin
    FData := AData;
    XMixer := AMixer;
    FControls := TMixerControls.Create (AMixer, AData);
end;

destructor TMixerConnection.Destroy;
begin
    FControls.Free;
    inherited;
end;

{-----}
{TMixerConnections}
{-----}

constructor TMixerConnections.Create (AMixer:TAudioMixer; AData:TMixerLine);
var A,B:Integer;
    ML:TMixerLine;
begin
    XMixer := AMixer;
    FConnections := TPointerList.Create;
    FConnections.OnFreeItem := Dofreeitem;
    ML.cbStruct := SizeOf(TMixerLine);
    ML.dwDestination := AData.dwDestination;
    For A := 0 to AData.cConnections-1 do
        begin

```

```

    ML.dwSource := A;
    B := MixerGetLineInfo (AMixer.MixerHandle, @ML,
MIXER_GETLINEINFOF_SOURCE);
    If B = MMSYSERR_NOERROR then
        FConnections.Add (Pointer(TMixerConnection.Create (XMixer, ML)));
    end;
end;

destructor TMixerConnections.Destroy;
begin
    FConnections.Free;
    inherited;
end;

procedure TMixerConnections.DoFreeItem (Pntr:Pointer);
begin
    TMixerConnection(Pntr).Free;
end;

function TMixerConnections.GetConnection (Ind:Integer):TMixerConnection;
begin
    Result := FConnections.Pointer[Ind];
end;

function TMixerConnections.GetCount:Integer;
begin
    Result := FConnections.Count;
end;

{-----}
{TMixerDestination}
{-----}

constructor TMixerDestination.Create (AMixer:TAudioMixer; AData:TMixerLine);
begin
    FData := AData;
    XMixer := AMixer;
    FConnections := TMixerConnections.Create (XMixer, FData);
    FControls := TMixerControls.Create (XMixer, AData);
end;

destructor TMixerDestination.Destroy;
begin
    Fcontrols.Free;
    FConnections.Free;
    inherited;
end;

```

```

end;

{-----}
{TMixerDestinations}
{-----}

constructor TMixerDestinations.Create (AMixer:TAudioMixer);
var A,B:Integer;
    ML:TMixerLine;
begin
    FDestinations := TPointerList.Create;
    FDestinations.OnFreeItem := DoFreeItem;
    if (AMixer = nil) then
        Exit;
    For A := 0 to AMixer.MixerCaps.cDestinations-1 do
    begin
        ML.cbStruct := SizeOf(TMixerLine);
        ML.dwDestination := A;
        B := MixerGetLineInfo (AMixer.MixerHandle, @ML,
MIXER_GETLINEINFOF_DESTINATION);
        If B = MMSYSERR_NOERROR then
            FDestinations.Add (Pointer(TMixerDestination.Create (AMixer, ML)));
        end;
    end;
end;

procedure TMixerDestinations.DoFreeItem (Pntr:Pointer);
begin
    TMixerDestination(Pntr).Free;
end;

destructor TMixerDestinations.Destroy;
begin
    FDestinations.Free;
    inherited;
end;

function TMixerDestinations.GetDestination (Ind:Integer):TMixerDestination;
begin
    Result := nil;
    If (Assigned (FDestinations)) then
        Result := FDestinations.Pointer[Ind];
    end;
end;

function TMixerDestinations.GetCount:Integer;
begin
    Result := FDestinations.Count;
end;

```

```

end;

{-----}
{TAudioMixer}
{-----}

constructor TAudioMixer.Create (AOwner:TComponent);
begin
  inherited Create (AOwner);
  XWndHandle := AllocateHwnd (MixerCallBack);
  FMixersCount := mixerGetNumDevs;
  FMixerId := -1;
  if (FMixersCount = 0) then
    FDestinations := TMixerDestinations.Create (nil)
  else
    begin
      FDestinations := nil;
      SetMixerId (0);
    end;
end;

destructor TAudioMixer.Destroy;
begin
  CloseMixer;
  if XWndHandle <> 0 then
    DeAllocateHwnd (XWndHandle);
  inherited;
end;

procedure TAudioMixer.CloseMixer;
begin
  If FMixerId >= 0 then
    begin
      mixerClose (FMixerHandle);
      FMixerId := -1;
    end;
  FDestinations.Free;
  FDestinations := nil;
end;

procedure TAudioMixer.SetMixerId (Value:Integer);
label AllOK;
begin
  If (Value < 0) OR (Value >= FMixersCount) then
    Exit;
  CloseMixer;

```

```

If mixerOpen (@FMixerHandle, Value, XWndHandle, 0, CALLBACK_WINDOW OR
MIXER_OBJECTF_MIXER) = MMSYSERR_NOERROR then
    goto Allok;

// we will go here very rarely, but sometimes it could help
If mixerOpen (@FMixerHandle, Value, XWndHandle, 0, CALLBACK_WINDOW) =
MMSYSERR_NOERROR then
    goto Allok;
If mixerOpen (@FMixerHandle, Value, 0, 0, 0) = MMSYSERR_NOERROR then
    goto Allok;

// an error has occurred
FMixerId := -1;
FDestinations := TMixerDestinations.Create (nil);

Exit;

Allok:
    FMixerId := Value;
    mixerGetDevCaps (MixerId, @FMixerCaps, SizeOf (TMixerCaps));

    if FMixerCaps.wMid = MM_MICROSOFT then
        FManufacturer := 'Microsoft'
    else
        FManufacturer := IntToStr (FMixerCaps.wMid) + ' = Unknown';
    FDriverVersion := FMixerCaps.vDriverVersion;
    FProductId := FMixerCaps.wPid;
    FProductName := StrPas (FMixerCaps.szPName);
    FNumberOfLine := FMixerCaps.cDestinations;

    FDestinations := TMixerDestinations.Create (Self);
end;

procedure TAudioMixer.MixerCallBack (var Msg:TMessage);
begin
    case Msg.Msg of
        MM_MIXM_LINE_CHANGE:
            If Assigned (OnLineChange) then
                OnLineChange (Self, Msg.wParam, Msg.lParam);
        MM_MIXM_CONTROL_CHANGE:
            If Assigned (OnControlChange) then
                OnControlChange (Self, Msg.wParam, Msg.lParam);
    else
        Msg.Result := DefWindowProc (XWndHandle, Msg.Msg, Msg.WParam,
Msg.LParam);

```

```

    end;
end;

const MIXER_LONG_NAME_CHARS = 64;

type MIXERCONTROLDETAILS_LISTTEXT = record
    dwParam1:DWORD;
    dwParam2:DWORD;
    szName:Array [0..MIXER_LONG_NAME_CHARS-1] of Char;
end;

type ListTextArray = array [0..1000] of MIXERCONTROLDETAILS_LISTTEXT;

function TAudioMixer.GetVolume (ADestination,AConnection:Integer;var LeftVol,
RightVol,      Mute:Integer;var      Stereo,      VolDisabled,      MuteDisabled,
MuteIsSelect:Boolean):Boolean;
var MD:TMixerDestination;
    MC:TMixerConnection;
    Cntrl:TMixerControls;
    MCD:TMixerControlDetails;
    Cntrl:PMixerControl;
    A,B:Integer;
    ML:TMixerLine;
    details:array [0..100] of Integer;
    ltext:^ListTextArray;
    MCDText:TMixerControlDetails;

begin
    Result := False;
    If (not Assigned (FDestinations)) then
        Exit;
    Stereo := False;
    MuteDisabled := True;
    VolDisabled := True;
    LeftVol := -1;
    RightVol := -1;
    Mute := -1;
    MuteIsSelect := False;
    MD := Destinations[ADestination];
    MC := nil;
    If AConnection <> -1 then
        MC := MD.Connections[AConnection];

    If MD <> nil then
        begin

```

```

Result := True;

//      If Mute = -1 then
begin
  If AConnection <> -1 then
  begin
    Cntrls := MD.Controls;
    ML := MD.Data;
    If (MC <> nil) AND (Cntrls <> nil) then
    begin
      A := 0;
      while (Mute = -1) AND (A < Cntrls.Count) do
      begin
        Cntrl := Cntrls[A];
//          If (Cntrl.dwControlType AND MIXERCONTROL_CONTROLTYPE_MIXER =
MIXERCONTROL_CONTROLTYPE_MIXER) OR
//          (Cntrl.dwControlType AND MIXERCONTROL_CONTROLTYPE_MUX =
MIXERCONTROL_CONTROLTYPE_MUX) then

//          If (Cntrl.dwControlType AND MIXERCONTROL_CT_CLASS_MASK =
MIXERCONTROL_CT_CLASS_LIST) then

          If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
            // Mux is similar to mixer, but only one line can be selected
at a time
          begin
            MCD.cbStruct := SizeOf(TMixerControlDetails);
            MCD.dwControlID := Cntrl.dwControlID;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
              MCD.cChannels := 1
            else
              MCD.cChannels := ML.cChannels;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
              MCD.cMultipleItems := Cntrl.cMultipleItems
            else
              MCD.cMultipleItems := 0;
            MCD.cbDetails := 4;
            MCD.paDetails := @Details;
            B := mixerGetControlDetails
(FMixerHandle,@MCD,MIXER_GETCONTROLDETAILSF_VALUE);
            If B <> MMSYSERR_NOERROR then
            begin
              Inc (A);
              continue;
            end;

```

```

MCDText.cbStruct := sizeof (MCDText);
MCDText.dwControlID := Cntrl.dwControlID;
If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
    MCDText.cChannels := 1
else
    MCDText.cChannels := ML.cChannels;
If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
    MCDText.cMultipleItems := Cntrl.cMultipleItems
else
    MCDText.cMultipleItems := 0;
GetMem (ltext, MCDText.cChannels * MCDText.cMultipleItems *
sizeof (MIXERCONTROLDETAILS_LISTTEXT));
MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
MCDText.paDetails := ltext;
B := mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
If B <> MMSYSERR_NOERROR then
begin
    FreeMem (ltext);
    Inc (A);
    continue;
end;
B := MCD.cChannels - 1;
while (B < integer(MCD.cMultipleItems)) do
begin
    if (ltext[B].dwParam1 = MC.Data.dwLineID) then
        break;
    Inc (B, MCD.cChannels);
end;
FreeMem (ltext);
If (B < integer (MCD.cMultipleItems)) then
begin
    Mute := Details[B];
    MuteDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
    MuteIsSelect := True;
    break;
end;
end;
Inc (A);
end;
end;
end;
end;
end;

```

```

end;

If AConnection = -1 then
begin
  Cntrls := MD.Controls;
  ML := MD.Data;
end
else
begin
  If MC <> nil then
  begin
    Cntrls := MC.Controls;
    ML := MC.Data;
  end
  else
    Cntrls := nil;
end;
If Cntrls <> nil then
begin
  A := 0;
  while ((LeftVol = -1) OR (Mute = -1)) AND (A < Cntrls.Count) do
  begin
    Cntrl := Cntrls[A];
    If Cntrl <> nil then
    begin
      If ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) OR
          ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) AND (Mute
= -1))) AND
          (Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE <>
MIXERCONTROL_CONTROLF_MULTIPLE)
      then
        begin
          if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) then
            MCD.cbStruct := SizeOf(TMixerControlDetails)
          else
            MCD.cbStruct := SizeOf(TMixerControlDetails);
          MCD.dwControlID := Cntrl.dwControlID;
          If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
            MCD.cChannels := 1
          else
            MCD.cChannels := ML.cChannels;
          MCD.cMultipleItems := 0;
          MCD.cbDetails := SizeOf(Integer);
          MCD.paDetails := @details;
          B := mixerGetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);

```

```

    If B = MMSYSERR_NOERROR then
    begin
        If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) AND
(LeftVol = -1) then
        begin
            VolDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
            LeftVol := details[0];
            If MCD.cChannels > 1 then
            begin
                RightVol := Details[1];
                Stereo := True;
            end
            else
                RightVol := LeftVol;
            end
            else If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE)
AND (Mute = -1) then
            begin
                MuteDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
                If Details[0] <> 0 then
                    Mute := 1
                else
                    Mute := 0;
                end
            // NEW ->
            (*
                else If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF)
AND (Mute = -1) then
                begin
                    MuteDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
                    If Details[0] <> 0 then
                        Mute := 1
                    else
                        Mute := 0;
                    MuteIsSelect := True;
                end;*)
            // <- NEW
            end;
        end;
        end;
        end;
        Inc (A);
    end;

    {
        If LeftVol = -1 then

```

```

        VolDisabled := True;
    If Mute = -1 then
        MuteDisabled := True;}
    end;
end;
end;

```

```

function TAudioMixer.SetVolume (ADestination, AConnection:Integer; LeftVol,
RightVol, Mute:Integer):Boolean;
var MD:TMixerDestination;
    MC:TMixerConnection;
    Cntrls:TMixerControls;
    MCD:TMixerControlDetails;
    Cntrl:PMixerControl;
    A,B:Integer;
    ML:TMixerLine;
    details:array [0..100] of Integer;
    VolSet,MuteSet:Boolean;
    ltext:^ListTextArray;
    MCDText:TMixerControlDetails;
begin
    Result := False;
    If (not Assigned (FDestinations)) then
        Exit;
    MC := nil;
    MD := Destinations[ADestination];
    If MD <> nil then
        begin
            If AConnection <> -1 then
                MC := MD.Connections[AConnection];

                VolSet := LeftVol = -1;
                MuteSet := Mute = -1;
                Result := True;

            If not MuteSet then
                begin
                    If AConnection <> -1 then
                        begin
                            Cntrls := MD.Controls;
                            ML := MD.Data;
                            If (MC <> nil) AND (Cntrls <> nil) then
                                begin
                                    A := 0;
                                    while not MuteSet AND (A < Cntrls.Count) do
                                        begin

```

```

Cntrl := Cntrls[A];
If (*(Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR*)
    (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
begin
    MCD.cbStruct := SizeOf(TMixerControlDetails);
    MCD.dwControlID := Cntrl.dwControlID;
    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
        MCD.cChannels := 1
    else
        MCD.cChannels := ML.cChannels;
    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
        MCD.cMultipleItems := Cntrl.cMultipleItems
    else
        MCD.cMultipleItems := 0;
    MCD.cbDetails := 4;
    MCD.paDetails := @Details;
    MuteSet := True;
    mixerGetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
    if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
        For B := 0 to Cntrl.cMultipleItems-1 do
            Details[B] := 0;

            GetMem (ltext, MCD.cChannels * MCD.cMultipleItems * sizeof
(MIXERCONTROLDETAILS_LISTTEXT));
            MCDText.cbStruct := sizeof (MCDText);
            MCDText.dwControlID := Cntrl.dwControlID;
            MCDText.cChannels := MCD.cChannels;
            MCDText.cMultipleItems := MCD.cMultipleItems;
            MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
            MCDText.paDetails := ltext;
            mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
            B := MCD.cChannels - 1;
            while (B < integer (MCD.cMultipleItems)) do
            begin
                if (ltext[B].dwParam1 = MC.Data.dwLineID) then
                    break;
                Inc (B, MCD.cChannels);
            end;
            FreeMem (ltext);

            If (B < integer (MCD.cMultipleItems)) then
            begin
                Details[B] := Mute;

```

```

        mixerSetControlDetails          (FMixerHandle,
MIXER_GETCONTROLDETAILSF_VALUE);
        break;
    end;
end;
    Inc (A);
end;
end;
end;
end;
end;

If AConnection = -1 then
begin
    Cntrl := MD.Controls;
    ML := MD.Data;
end
else
begin
    If MC <> nil then
    begin
        Cntrl := MC.Controls;
        ML := MC.Data;
    end
    else
    begin
        Cntrl := nil;
    end;
end;
If Cntrl <> nil then
begin

    A := 0;
    while (not VolSet OR not MuteSet) AND (A < Cntrl.Count) do
    begin
        Cntrl := Cntrl[A];
        If Cntrl <> nil then
        begin
            If (((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) AND not
VolSet) OR
                ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) AND not
MuteSet) (* NEW -> *) (*OR
                ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF) AND not
MuteSet)*) (* <- NEW *) AND
                (Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE <>
MIXERCONTROL_CONTROLF_MULTIPLE)
            then

```

```

begin
  MCD.cbStruct := SizeOf(TMixerControlDetails);
  MCD.dwControlID := Cntrl.dwControlID;
  If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
    MCD.cChannels := 1
  else
    MCD.cChannels := ML.cChannels;
  MCD.cMultipleItems := 0;
  MCD.cbDetails := SizeOf(Integer);
  MCD.paDetails := @Details;
  If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) then
    begin
      Details[0] := LeftVol;
      If RightVol = -1 then
        Details[1] := LeftVol
      else
        Details[1] := RightVol;
      VolSet := True;
    end
  else if ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) (*
NEW -> *) (* OR
(Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF) *)
(* <- NEW *) then
    begin
      For B := 0 to MCD.cChannels - 1 do
        Details[B] := Mute;
        MuteSet := True;
      end;
      mixerSetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
    end;
  end;
  Inc (A);
end;

end;
end;
end;

```

```

function TAudioMixer.GetMute (ADestination, AConnection: Integer; var Mute:
Boolean):Boolean;

```

```

var

```

```

  MD : TMixerDestination;

```

```

  MC : TMixerConnection;

```

```

  mlcMixerLineControlsMute : TMIXERLINECONTROLS;

```

```

  mcdMixerDataMute : TMIXERCONTROLDETAILS;

```

```

pmcMixerControlMute : PMIXERCONTROL;
pmcdsMixerDataUnsignedMute : PMIXERCONTROLDETAILSBOOLEAN;
mlMixerLine : TMixerLine;
Cntrl:PMixerControl;
Cntrls:TMixerControls;
ML:TMixerLine;
A,B:Integer;
details:array [0..100] of Integer;
ltext:^ListTextArray;
MCDText:TMixerControlDetails;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  MC := nil;
  Mute := False;
  MD := Destinations[ADestination];
  if MD <> nil then
    begin
      if AConnection = -1 then
        mlMixerLine := MD.Data
      else
        begin
          MC := MD.Connections[AConnection];
          if MC <> nil then
            mlMixerLine := MC.Data
          else
            Exit;
          end;
        end;

      GetMem(pmcMixerControlMute, SizeOf(TMIXERCONTROL));
      GetMem(pmcdsMixerDataUnsignedMute, SizeOf(TMIXERCONTROLDETAILSBOOLEAN));

      with mlcMixerLineControlsMute do
        begin
          cbStruct := SizeOf(TMIXERLINECONTROLS);
          dwLineID := mlMixerLine.dwLineID;
          dwControlType := MIXERCONTROL_CONTROLTYPE_MUTE;
          cControls := 1;
          cbmxcntrl := SizeOf(TMIXERCONTROL);
          pamxcntrl := pmcMixerControlMute;
        end;

        if (mixerGetLineControls(FMixerHandle, @mlcMixerLineControlsMute,
MIXER_GETLINECONTROLSF_ONEBYTYPE) = MMSYSERR_NOERROR) then
          begin

```

```

with mcdMixerDataMute do
begin
    cbStruct := SizeOf(TMIXERCONTROLDETAILS);
    dwControlID := pmcMixerControlMute^.dwControlID;
    cChannels := 1;
    cMultipleItems := 0;
    cbDetails := SizeOf(TMIXERCONTROLDETAILSBOOLEAN);
    paDetails := pmcMixerDataUnsignedMute;
end;

if mixerGetControlDetails(FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) = MMSYSERR_NOERROR then
begin
    Mute := pmcMixerDataUnsignedMute^.fValue = 1;
    Result := True;
end;
end
else
begin
    If (AConnection <> -1) then
    begin
        Cntrls := MD.Controls;
        ML := MD.Data;
        If (MC <> nil) AND (Cntrls <> nil) then
        begin
            A := 0;
            while (Result = False) AND (A < Cntrls.Count) do
            begin
                Cntrl := Cntrls[A];
                If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR
                    (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
                begin
                    mcdMixerDataMute.cbStruct := SizeOf(TMixerControlDetails);
                    mcdMixerDataMute.dwControlID := Cntrl.dwControlID;
                    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
                        mcdMixerDataMute.cChannels := 1
                    else
                        mcdMixerDataMute.cChannels := ML.cChannels;
                    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
                        mcdMixerDataMute.cMultipleItems := Cntrl.cMultipleItems
                    else
                        mcdMixerDataMute.cMultipleItems := 0;
                    mcdMixerDataMute.cbDetails := 4;
                    mcdMixerDataMute.paDetails := @Details;

```

```

        mixerGetControlDetails      (FMixerHandle,      @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE);

        GetMem      (ltext,      mcdMixerDataMute.cChannels      *
mcdMixerDataMute.cMultipleItems * sizeof (MIXERCONTROLDETAILS_LISTTEXT));
        MCDText.cbStruct := sizeof (MCDText);
        MCDText.dwControlID := Cntrl.dwControlID;
        MCDText.cChannels := mcdMixerDataMute.cChannels;
        MCDText.cMultipleItems := mcdMixerDataMute.cMultipleItems;
        MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
        MCDText.paDetails := ltext;
        mixerGetControlDetails      (FMixerHandle,      @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);

        B := mcdMixerDataMute.cChannels - 1;
        while (B < integer (mcdMixerDataMute.cMultipleItems)) do
        begin
            if (ltext[B].dwParam1 = MC.Data.dwLineID) then
                break;
            Inc (B, mcdMixerDataMute.cChannels);
        end;
        FreeMem (ltext);

        If (B < integer (mcdMixerDataMute.cMultipleItems)) then
        begin
            Result := True;
            Mute := Details[B] <> 0;
            break;
        end;
        end;
        Inc (A);
    end;
end;
end;
end;
end;

FreeMem (pmcMixerDataUnsignedMute);
FreeMem (pmcMixerControlMute);
end;
end;

function TAudioMixer.SetMute (ADestination, AConnection: Integer; Mute:
Boolean): Boolean;
var
    MD : TMixerDestination;
    MC : TMixerConnection;
    mlcMixerLineControlsMute : TMIXERLINECONTROLS;

```

```

mcdMixerDataMute : TMIXERCONTROLDETAILS;
pmcMixerControlMute : PMIXERCONTROL;
pmcMixerDataUnsignedMute : PMIXERCONTROLDETAILSBOOLEAN;
mlMixerLine : TMixerLine;
Cntrl:PMixerControl;
Cntrls:TMixerControls;
ML:TMixerLine;
A,B:Integer;
details:array [0..100] of Integer;
ltext:^ListTextArray;
MCDText:TMixerControlDetails;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  MC := nil;
  MD := Destinations[ADestination];
  if MD <> nil then
    begin
      if AConnection = -1 then
        mlMixerLine := MD.Data
      else
        begin
          MC := MD.Connections[AConnection];
          if MC <> nil then
            mlMixerLine := MC.Data
          else
            Exit;
          end;
        end;

      GetMem(pmcMixerControlMute, SizeOf(TMIXERCONTROL));
      GetMem(pmcMixerDataUnsignedMute, SizeOf(TMIXERCONTROLDETAILSBOOLEAN));

      with mlcMixerLineControlsMute do
        begin
          cbStruct := SizeOf(TMIXERLINECONTROLS);
          dwLineID := mlMixerLine.dwLineID;
          dwControlType := MIXERCONTROL_CONTROLTYPE_MUTE;
          cControls := 0;
          cbmxctrl := SizeOf(TMIXERCONTROL);
          pamxctrl := pmcMixerControlMute;
        end;

        if (mixerGetLineControls(FMixerHandle, @mlcMixerLineControlsMute,
MIXER_GETLINECONTROLSF_ONEBYTYPE) = MMSYSERR_NOERROR) then
          begin

```

```

with mcdMixerDataMute do
begin
    cbStruct := SizeOf(TMixerControlDetails);
    dwControlID := pmcMixerControlMute^.dwControlID;
    cChannels := 1;
    cMultipleItems := 0;
    cbDetails := SizeOf(TMIXERCONTROLDETAILSBOOLEAN);
    paDetails := pmcdsMixerDataUnsignedMute;
end;

if Mute then
    pmcdsMixerDataUnsignedMute^.fValue := 1
else
    pmcdsMixerDataUnsignedMute^.fValue := 0;

if
(mixerSetControlDetails(FMixerHandle,@mcdMixerDataMute,MIXER_SETCONTROLDETAILS
F_VALUE) = MMSYSERR_NOERROR) then
    Result := True;
end
else
begin
    If (AConnection <> -1) then
    begin
        Cntrls := MD.Controls;
        ML := MD.Data;
        If (MC <> nil) AND (Cntrls <> nil) then
        begin
            A := 0;
            while (Result = False) AND (A < Cntrls.Count) do
            begin
                Cntrl := Cntrls[A];
                If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR
                    (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
                begin
                    mcdMixerDataMute.cbStruct := SizeOf(TMixerControlDetails);
                    mcdMixerDataMute.dwControlID := Cntrl.dwControlID;
                    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
                        mcdMixerDataMute.cChannels := 1
                    else
                        mcdMixerDataMute.cChannels := ML.cChannels;
                    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
                        mcdMixerDataMute.cMultipleItems := Cntrl.cMultipleItems
                    else
                        mcdMixerDataMute.cMultipleItems := 0;

```

```

mcdMixerDataMute.cbDetails := 4;
mcdMixerDataMute.paDetails := @Details;
if (mixerGetControlDetails (FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) <> MMSYSERR_NOERROR) then
begin
  Inc (A);
  continue;
end;
if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
  For B := 0 to Cntrl.cMultipleItems-1 do
    Details[B] := 0;
If Mute then
begin
  GetMem (ltext, mcdMixerDataMute.cChannels *
mcdMixerDataMute.cMultipleItems * sizeof (MIXERCONTROLDETAILS_LISTTEXT));
  MCDText.cbStruct := sizeof (MCDText);
  MCDText.dwControlID := Cntrl.dwControlID;
  MCDText.cChannels := mcdMixerDataMute.cChannels;
  MCDText.cMultipleItems := mcdMixerDataMute.cMultipleItems;
  MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
  MCDText.paDetails := ltext;
  mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
  B := mcdMixerDataMute.cChannels - 1;
  while (B < integer (mcdMixerDataMute.cMultipleItems)) do
  begin
    if (ltext[B].dwParam1 = MC.Data.dwLineID) then
      break;
    Inc (B, mcdMixerDataMute.cChannels);
  end;
  FreeMem (ltext);
  If (B < integer (mcdMixerDataMute.cMultipleItems)) then
    Details[B] := 1;
  end;
  if (mixerSetControlDetails (FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) = MMSYSERR_NOERROR) then
  begin
    Result := True;
    break;
  end;
end;
Inc (A);
end;
end;
end;

```

```

end;

FreeMem(pmcMixerDataUnsignedMute);
FreeMem(pmcMixerControlMute);
end;
end;

function TAudioMixer.GetPeak (ADestination, AConnection:Integer; var LeftPeak,
RightPeak:Integer): Boolean;
var
  MD : TMixerDestination;
  MC : TMixerConnection;
  mcdMixerDataPeak : TMIXERCONTROLDETAILS;
  pmcMixerControlPeak : PMIXERCONTROL;
{ pmcMixerDataSignedPeak : PMIXERCONTROLDETAILSSIGNED;}
  mlMixerLine : TMixerLine;
  A:Integer;
  Cntrl:TMixerControls;
  Details:Array [1..100] of Integer;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  LeftPeak := 0;
  RightPeak := 0;
  MD := Destinations[ADestination];
  if MD <> nil then
  begin
    if AConnection = -1 then
    begin
      mlMixerLine := MD.Data;
      Cntrl := MD.Controls;
    end
    else
    begin
      MC := MD.Connections[AConnection];
      if MC <> nil then
      begin
        mlMixerLine := MC.Data;
        Cntrl := MC.Controls;
      end
      else
        Exit;
      end;
    end;
  end;
  GetMem(pmcMixerControlPeak, SizeOf(TMIXERCONTROL));

```

```

A := 0;
while (A < Cntrls.Count) do
begin
    If (Cntrls[A].dwControlType AND MIXERCONTROL_CT_CLASS_MASK) =
MIXERCONTROL_CT_CLASS_METER then
        break;
    Inc (A);
end;
If A = Cntrls.Count then
begin
    FreeMem(pmcMixerControlPeak);
    Exit;
end;

with mcdMixerDataPeak do
begin
    cbStruct := SizeOf(TMIXERCONTROLDETAILS);
    dwControlID := Cntrls[A].dwControlID;
    cChannels := mlMixerLine.cChannels;
    If (Cntrls[A].fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE) =
MIXERCONTROL_CONTROLF_MULTIPLE then
        cMultipleItems:=Cntrls[A].cMultipleItems
    else
        cMultipleItems:=0;
    cbDetails := SizeOf(TMIXERCONTROLDETAILSSIGNED);
    paDetails := @Details;
end;
if
(mixerGetControlDetails(FMixerHandle,@mcdMixerDataPeak,MIXER_GETCONTROLDETAILS
F_VALUE) = MMSYSERR_NOERROR) then
begin
    LeftPeak := Details[1];
    if mlMixerLine.cChannels = 2 then
        RightPeak := Details[2]
    else
        RightPeak := LeftPeak;
    Result := True;
end;
FreeMem(pmcMixerControlPeak);
end;
end;
procedure Register;
begin
    RegisterComponents('Samples', [TAudioMixer]);
end;
end.

```