Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM”

КБПЗ_2024

Виконав здобувач вищої освіти
ІІ курсу, групи КІ-23М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Найдьонов І.В.
«____» _____ 2024 р.

Керівник проекту
доктор технічних наук, професор
_____ Коваленко О.В.
«____» _____ 2024 р.
Рецензент _____
_____

м. Кропивницький

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *магістр*
Галузь знань _12 _____*"Інформаційні технології"*_____
Спеціальність 123 *"Комп'ютерна інженерія"*
Освітньо-професійна (освітньо-наукова) програма _*"Комп'ютерна інженерія"*_

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

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

Найдьонову Ігорю Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM*

2. Керівник роботи *Коваленко Олександр Володимирович, докт. техн. наук, професор*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

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

3. Строк подання студентом роботи до захисту *2.12.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою розробки є дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM*

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

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

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

Наукова новизна	*1 аркуш*
Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*
Показники економічної ефективності	*1 аркуш*

КБПЗ-2024

6. Консультанти розділів роботи

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

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

КАЛЕНДАРНИЙ ПЛАН

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

КсПЗ_2024

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

_____ _____
(прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2024 р.

Підпис здобувача

_____ _____
(прізвище та ініціали)

АНОТАЦІЯ

Найдьонов І.В. Дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

Метою розробки є дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

Об'єктом дослідження є процес інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

Предметом дослідження є методи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

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

Результат роботи – програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, IP-ATM

КБПЗ_2024

ABSTRACT

Naidyonov I.V. Research and software implementation of an integrated telecommunications solution system based on IP-ATM technology. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for an integrated telecommunications solution system based on IP-ATM technology.

The purpose of the development is the research and software implementation of an integrated telecommunications solution system based on IP-ATM technology.

The object of the research is the process of an integrated telecommunications solution based on IP-ATM technology.

The subject of the research is the methods of an integrated telecommunications solution based on IP-ATM technology.

The research methods are based on methods of coding theory and computer network theory, methods of mathematical statistics, methods of software development.

The result of the work is a software implementation of an integrated telecommunications solution based on IP-ATM technology.

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 the software are provided.

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

The program was developed in the Python environment.

Keywords: computer engineering, IP-ATM

КбПЗ_2024

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 3
ВСТУП 4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 8
1.1 Призначення системи 8
1.2 Область застосування 9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти 11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування 18
2.3 Розгорнута постановка завдання 23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 25
3.1 Опис функціонування системи 25
3.2 Розробка структурної схеми 28
3.3 Розробка функціональної схеми 36
3.4 Розробка діаграми процесів 55
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ 58
4.1 Розробка блок-схем та опис алгоритмів функціонування системи 58
4.2 Захист розробленого програмного забезпечення 69
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ 74
6 НАУКОВА НОВИЗНА 79

КБПЗ_2024

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ 80
7.1 Визначення цільової аудиторії кінцевого готового продукту 80
7.2 Оцінка привабливості шляхом застосування методів експертних оцінок ... 82
7.3 Вибір методу оцінки вартості ПЗ 84
7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості 85
7.5 Пропозиція алгоритму просування проєкту розробки ПЗ 87
7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ 89
7.7 Визначення ключових факторів успіху конкретного проєкту 90
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ 91
8.1 Вступ 91
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером 92
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ... 93
8.4 Розробка заходів з умов поліпшення охорони праці 96
8.5 Розрахункова частина 97
9 ОСНОВНІ ВИСНОВКИ 100
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 102

К6ПЗ_2024

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АТМУ – автоматизована телефона мережа установи
БД – база даних
КМЗ – корпоративна мережа зв’язку
ЛОМ – локальна обчислювальна мережа
ОЗП – оперативно-запам’ятовувальний пристрій
ПАТМ – пристрій автоматизованої телефонної мережі
ПЗ – програмне забезпечення
ПП – програмний продукт
СПД – система передачі даних
СУБД – система управління БД
ТфОП – телефонний оператор
AVVID – архітектурна модель фирмы Cisco Systems
CNG – Comfort Noise Generator. Генератор комфортного шуму
DSP – Digital Signal Processor. Процесор цифрової обробки сигналів
DTMF – Dual Tone Multi-Freqency. Багаточастотна система кодування цифр номера
GK – Gatekeeper. Гейткіпер . Виконує функції керування зоною мережі H.323
GW – Gateway. Шлюз. Апаратно-програмний комплекс, що забезпечує обмін даними між мережами різних типів
H.323 – Рекомендація ITU-T, що визначає системи мультимедійного зв'язку в мережах з пакетною комутацією
H.248 – Протокол керування транспортним шлюзом
IP – Internet Protocol. Протокол міжмережної взаємодії
LPC – Linear Prediction Coding. Кодування з лінійним пророкуванням
MCU – Multipoint Control Unit. Пристрій керування конференцією
MG – Media Gateway. Транспортний шлюз

КБПЗ_2024

MGCP – Media Gateway Control Protocol. Протокол керування шлюзами

MP – Multipoint processor. Процесор для обробки інформації користувачів при централізованих конференціях

OSI – Open System Interconnection. Взаємодія відкритих систем

PPP – Point-to-Point Protocol. Протокол двостороннього зв'язку

RADIUS – Remote Authentication Dial-In User Service. Протокол автентифікації й авторизації абонентів, а також обліку обсягу наданих їм послуг

RAS – Registration Admission and Status. Протокол взаємодії термінального встаткування з gatekeeper. Входить у сімейство протоколів H.323

RSVP – Resource Reservation Protocol. Протокол резервування ресурсів

RTCP – Real-time Transport Control Protocol. Протокол контролю транспортування інформації в реальному часі

SIP – Session Initiation Protocol. Протокол ініціювання сеансів зв'язку

TAPI – Telephony Applications Programming Interface. Інтерфейс для програмування телефонних додатків

TCP – Transmission Control Protocol. Протокол керування передачею (даних) Основний транспортний протокол у стеці протоколів TCP/IP.

TCP/IP – Transmission Control Protocol/Internet Protocol. Стек протоколів, що забезпечують організацію зв'язку між комп'ютерами в мережі Інтернет

UDP – User Datagram Protocol. Протокол передачі дейтаграмм користувача. 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

КБПЗ_2024

ВСТУП

Актуальність теми. Сьогодні бізнес-телефонія – це інтегровані телекомунікаційні рішення на основі технологій VoIP, віртуальні й фізичні IP-ATM, різноманітні SIP-пристрої, віртуальні номери й багато чого іншого.

Ми живемо у швидко мінливому світі. Швидкість передачі інформації й прийняття рішень стрімко збільшується рік у рік. Років двісті назад проходили тижні й місяці, поки ямщик довозив лист до пункту призначення. Двадцять років тому лист доставлявся поштою за кілька днів, а іноді, як і раніше, тижнів, і лише у випадках екстреної важливості відправляли телеграму, щоб адресат одержав її дуже швидко – протягом доби. Усього десять років тому на пересилання повідомлення електронної пошти було потрібно кілька хвилин і здавалося, що це вже межа технологічних можливостей.

Сьогодні рахунок йде на секунди: вибравши одержувача в списку контактів у програмному клієнті Jabber на ПК або мобільному телефоні, ви можете не просто відправити йому миттєве повідомлення, але й подзвонити на робочий або мобільний телефон, а при необхідності, підключити когось із колег до розмови, продемонструвати документ, про яке мова йде, включити камеру або перевести розмова в режим відео-дзвінка.

В IP-ATM реалізований цілий ряд корисних функцій: автосекретаря, підключення віддалених співробітників, використання смартфона як внутрішнього абонента станції, запуску додатків CTI і т.д. Причому в складі корпоративної мережі можуть перебувати різні апарати: провідні IP-, SIP-телефони, DECT-телефони, софтфони й інші

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

КБПЗ_2024

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

– Огляд існуючих систем інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

– Дослідження системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

– Програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

Об'єктом дослідження є процес інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

Предметом дослідження є методи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

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

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

– Удосконалено метод інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

– Розроблено вітчизняний продукт інтегрованого телекомунікаційного рішення на основі технології IP-ATM, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

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.

КБПЗ_2024

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

К6ПЗ_2024

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Впровадження рішень, що надають можливості IP-ATM, допомагає співробітникам організацій істотно підвищити продуктивність своєї роботи й поліпшити спільну діяльність. Збільшення ж продуктивності веде до зростання прибутку й скороченню основних статей витрат:

– на співробітників (фонд оплати праці) – до 20%;
– на відрядження – до 15%;
– на офіс – до 5%;
– на навчання – до 15%.

Як правило, завдяки використанню сучасних IP-технологій підвищується ефективність і знижуються витрати практично в будь-якій компанії. У сегменті малого й середнього бізнесу такі технології теж стають вагомою підмогою в оптимізації роботи й збільшенні прибутку, однак далеко не всі керівники розуміють важливість впровадження якісних комунікаційних рішень, віддаючи пріоритет іншим напрямкам: вони вважають, що, якщо телефон і так дзвонить, цього цілком достатньо. Стратегічне завдання всіх учасників телекомунікаційного ринку складається в пропаганді й популяризації технологій у цілому, а не тільки в просуванні окремих брендів.

Успішний бізнес неможливо представити без корпоративної телефонної мережі. Грамотна організація системи зв'язку в компанії дозволяє не тільки оптимізувати витрати на організацію комунікацій, але й автоматизувати повсякденні операції офісних співробітників. Лідери ринку здатні запропонувати різні варіанти рішень. В Україні в умовах економічної кризи серйозний спад торкнувся саме підприємства малого й середнього бізнесу (SMB), які із цієї причини усе більш ретельно вибирають рішення.

КБПЗ_2024

Основний акцент в останні роки робиться саме на просуванні IP-рішень, у тому числі й для SMB, причому головними вимогами замовників є оптимальний набір функцій, мобільність співробітників, масштабованість і надійність за розумною ціною.

Устаткування допомагає створювати комплексні інтегровані системи, що поєднують всі комунікаційні процеси компанії. Такі проекти будуються за принципом All-over-SIP, коли використовується різне встаткування, що підтримує протокол SIP. У якості бюджетних IP-телефонів для даних рішень можна використовувати Yealink SIP-T21P E2. Телефони Yealink сумісні з безліччю популярних віртуальних АТМ і мають привабливе співвідношення ціна/якість

1.2 Область застосування

У малому й середньому бізнесі роль телекомунікацій традиційно висока. Все більшу популярність у цьому сегменті стали набирати хмарні рішення, що не вимагають серйозних первісних вкладень при організації системи зв'язку. Для телефонізації офісу існує чимало хмарних рішень – віртуальних АТМ, у пакеті з якими часто надається й реальний міський номер. Очевидні плюси такого вибору – швидке розгортання й широкі можливості масштабування. Компанія обтяжує себе лише порівняно невеликою орендною платою, і їй не потрібно витрачатися на покупку «залізної АТМ».

Важливим фактором досягнення успіху вважається здатність компаній реагувати на зміни. Курси валют, ціни на сировину, товари й послуги останнім часом швидко міняються. Як наслідок, організації постійно переїжджають, відкривають і закривають офіси, створюють нові напрямки й відмовляються від старих і найчастіше в таких умовах не можуть собі дозволити довгострокових дорогих інвестицій. Тому хмарні сервіси здобувають все більшу популярність.

КБПЗ_2024

Донедавна в силу ряду причин до хмарних сервісів ставилися скептично: не всі розуміли, навіщо потрібно переходити на такі рішення при нестабільній якості сервісу в більшості провайдерів, та й цінова вигода була неочевидна.

Ситуація помінялася з появою якісних і надійних хмарних віртуальних рішень, які з погляду функціональності й надійності ні в чому не уступають офісному встаткуванню, але на ділі виявляються дешевше, навіть із урахуванням сукупних витрат за 4-5 років.

Один із прикладів – рішення «Бізнес-комунікації за запитом» (BCaaS). Оскільки в кризу в багатьох компаній немає засобів на розвиток телефонної інфраструктури, хмарна телефонія й хмарний відео-конференц-зв'язок можуть виявитися оптимальним вибором, адже при повній відсутності первісних інвестицій і низьких щомісячних платежів можна одержати доступ відразу до всіх функціональних можливостей.

Рішення дозволяє вибрати спосіб надання сервісів для бізнес-комунікацій: «усе із хмари», коли замовникові не потрібно створювати власну інфраструктуру, або гібридне рішення, що забезпечує використання як хмарних, так і локально реалізованих сервісів, які вигідніше розгорнути на території компанії або на власному встаткуванні. Останній варіант застосовується для доповнення або розширення функціональності вже існуючих рішень, коли ризики або інвестиції в розвиток власних систем не виправдані

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ_2024

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

Для того щоб почати користуватися IP-телефонією, необхідно визначитися з вибором програмного клієнта – це самий доступний варіант дозвону. Гарний SIP-клієнт забезпечить зручність здійснення викликів, прийняття дзвінків і зберігання контактів. Якими якостями повинні володіти програмні телефони? Як вибрати найкращий клієнт для IP-телефонії? Відповісти на ці питання допоможе наш докладний огляд.

Кращі SIP-клієнти для Windows

Найпоширенішою операційною системою є Windows. Тому SIP-клієнтів під цю операційну систему досить багато. У список самих популярних клієнтів ми внесемо наступні програми:

- 3CX VOIP Phone;
- IP-Phone;
- X-Lite;
- Zoiper;
- QIP Infium.

Розглянемо їх більш докладно.

3CX VOIP Phone

Програмний телефон 3CX VOIP Phone для Windows є одним із самих популярних SIP-клієнтів. Перевагою програми є невелика вага, що забезпечує швидке завантаження на будь-якому каналі. Після запуску на екрані з'явиться приємний інтерфейс, що нагадує самий звичайний мобільний телефон. При першому старті програма запропонує додати аккаунт.

КБПЗ_2024

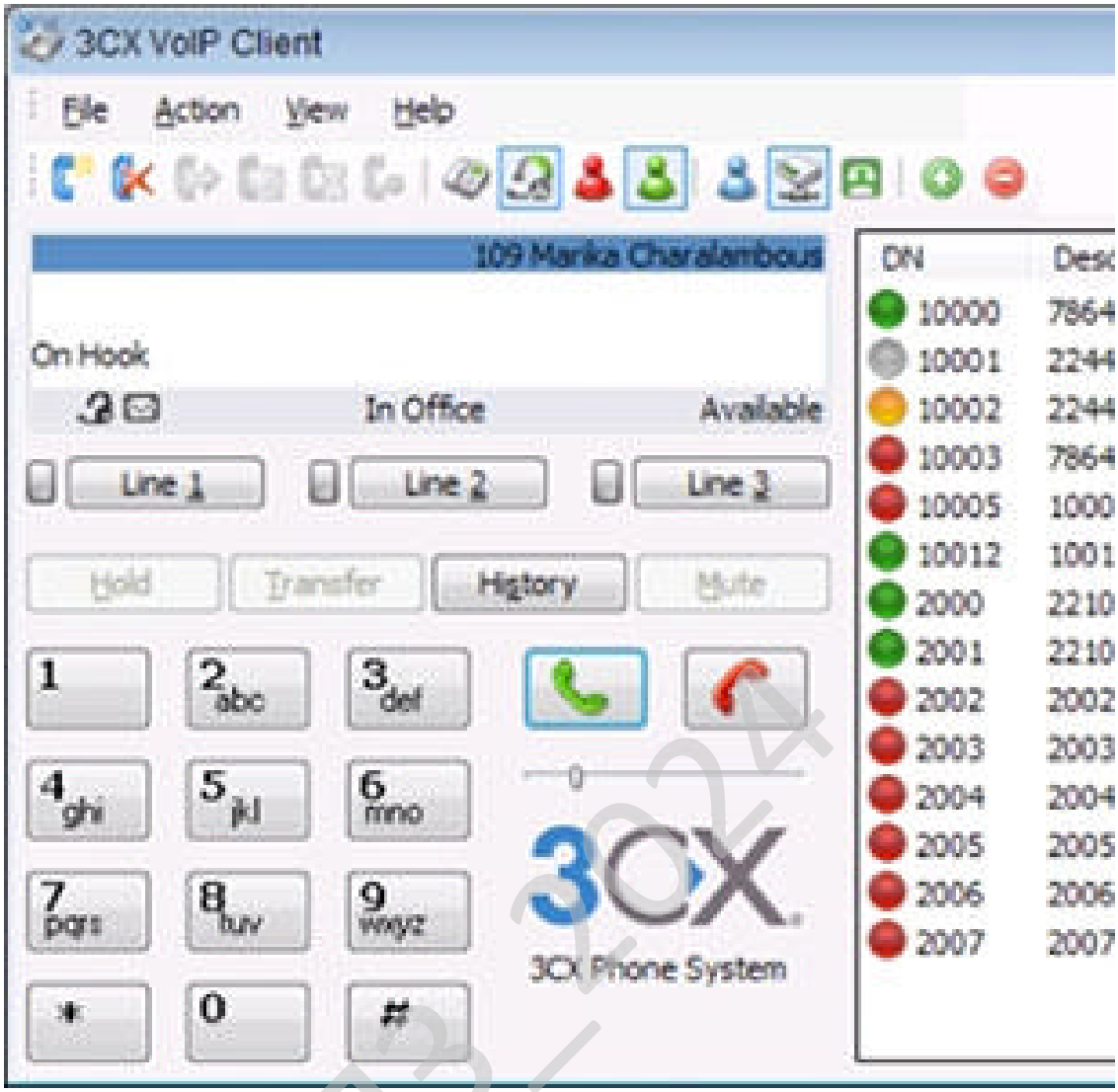

Рисунок 2.1 – Інтерфейс користувача 3CX VOIP Phone

КБПЗ_2024

3CX VOIP Phone підтримує відразу декілька аккаунтів, що придасться тим, хто користується послугами відразу декількох провайдерів.

Якщо ви скачали 3CX VOIP Phone з офіційного сайту, варто додати українську мову – це робиться в налаштуваннях, які викликаються за допомогою середньої кнопки (як на смартфонах з Android). Який функціонал цієї програми?

– Можливість здійснення відеовикликів.

– Установка статусу – доступний, немає на місці, автовідповідь, переадресація на голосову пошту.

– Можливість здійснення дзвінків відразу по декількох лініях.

– Перегляд історії викликів.

– Запис розмов у файл.

– Підтримка швидкого набору.

Програма досить проста, але українізація в неї страждає – деякі пункти як і раніше залишаються на англійській. Зайшовши в налаштування ми виявимо, що їх дуже мало. Користувачі можуть вибрати звукові пристрої й камеру, визначити поводження програми при знаходженні комп'ютера в різних режимах, задати придушення шумів. Тут же задається зовнішній вигляд програми – є підтримка різних скинів.

IP-Phone

Програма IP-Phone є найпростішим і зрозумілим програмним IP-телефоном. Можна сказати, що це найпростіший SIP-клієнт для Windows. Сама програма україномовна, хоча деякі пункти на англійському. Безсумнівним достоїнством є те, що цей телефон уміє запитувати й відображати баланс на використовуваному аккаунті. Також підтримується відображення номерів доступу. Варто сказати, що програма орієнтована на SIP-провайдеру Comtube і працює тільки з ним. Підтримується тільки один аккаунт.

При налаштуванні даної програми вказуються логін і пароль від аккаунта на сайті Comtube. Авторизація по SIP ID не підтримується.

X-Lite

Програмний телефон X-Lite має англомовний інтерфейс і досить приємний зовнішній вигляд. Прямо на головному екрані розташовуються великі цифрові кнопки – промахнутися повз них не вийде. Нижче вибираються контакти й вибрані номери, відображається історія викликів. Ледве вище виставляється статус, там же викликається віконце для обміну текстовими повідомленнями. Тут же розташовуються регулятор гучності й рядок пошуку контактів.

КБПЗ-2024

Рисунок 2.2 – Інтерфейс користувача X-Lite

У налаштуваннях програми практично нічого немає – отут редагуються звукові оповіщення, вибираються звукові пристрій, задаються кодеки для роботи з тими або іншими провайдерами. Тут же вибирається мова, якому потрібно скачати й установити окремо. Налаштування аккаунта (причому одного) розташовуються в окремій вкладці, тут можна виконати тонке налаштування.

Zoiper

SIP-клієнт Zoiper випускається в трьох видах:

– Безкоштовна версія.

– Бізнес-Версія.

– Індивідуальна версія.

Безкоштовна версія клієнта доступна для всіх користувачів, але вона трохи обмежена в можливостях. У бізнес-версії присутні підтримка додаткових

алгоритмів стиску звуку, поліпшена й розширена робота з контактами, є інтеграція в браузер, функція пересилання файлів і конференцзв’язок. Вартість програми становить 33 євро. Індивідуальна версія платна, тут виконується брендування під замовника, можливе замовлення додаткових опцій і індивідуального дизайну, відображається баланс. Вартість цієї версії підраховується індивідуально.

Безкоштовна версія Zoiper підтримує протоколи SIP, IAX і XMPP, уміє працювати з декількома аккаунтами. Контакти зберігаються локально або на серверах Zoiper, у глобальній адресній книзі з пошуком. Українська мова вже включена в стартовий дистрибутив. Налаштування дуже й дуже добре, програму можна налаштувати по своєму смаку.

Функціонал програми:

– Підтримка відеовикликів.
– Генерація комфортного шуму.
– Шумозаглушення.
– Підтримка скинів.

Переадресації в безкоштовній версії немає, як немає й розширеної роботи з контактами. Інтерфейс не дуже зручний і не дуже гарний. Також занадто нав'язливо пропонується придбати бізнес-версію – позначка про це зустрічається буквально на кожному кроці.

Модуль в QIP Infium

У популярному мультипротокольному мессенджері QIP Infium є присутнім модуль для роботи з SIP-телефонією. Налаштування зводиться до прописування даних аккаунта й вибору необхідних кодеків. В інтерфейсі присутня простенька панелька з дозвоном, є можливість підключення USB-телефону. Також реалізована можливість швидкого доступу до сервісів провайдеру, але працює вона не завжди – імовірно, все залежить від налаштувань.

КБПЗ – 2024

Ця «дзвонилка» підійде тим, хто не хоче захаращувати оперативну пам'ять додатковими додатками. Але потрібно пам'ятати, що ніякого додаткового функціонала (крім придушення луни) тут немає.

Рисунок 2.3 – Інтерфейс користувача QIP Infium

Клієнти для інших ОС

Користуватися IP-телефонією можна не тільки на комп'ютері, але й на смартфоні. Оптимальний SIP-клієнт для Android – це 3CXPhone – VoIP/SIP phone, його можна скачати через Google Play. Програма дуже функціональна, відрізняється швидкістю роботи й приємним інтерфейсом. Недоліком є відсутність української мови.

Якщо ви користуєтеся сервісом Comtube, ми рекомендуємо вам скачати з Google Play додаток IP-Phone для Android. Програма відрізняється легкістю в налаштуваннях, уміє показувати вартість дзвінків, може заміняти собою стандартну дзвонилку. Це ідеальний клієнт для Comtube, що не завантажує смартфон і не приводить до швидкої витрати батареї. Інтерфейс

КОПЗ_2024

україномовний, передбачений показ сервісної інформації – баланс, тариф, дані користувача.

Таблиця 2.1 – Порівняльна таблиця SIP-клієнтів

	Мультиаккаунт	Гнучкість налаштувань	Наявність версії для смартфонів	Протоколи	Відеозв'язок	Підтримка скинів	Запис розмов	Текстовий чат	Особливості
3CX VOIP Phone	Є	Немає	Є	SIP	Є	Є	Є	Немає	Зручний інтерфейс
IP-Phone	Немає	Немає	Є	SIP	Немає	Немає	Є	Немає	Відображення балансу
X-Lite	Немає	Немає	Немає	SIP	Немає	Немає	Немає	Є	
Zoiper	Є	Є	Є	SIP, IAX, XMPP	Немає	Є	Немає	Немає	Факс
QIP Infium	Є	Немає	Є	SIP, XIMSS (CGP)	Є	Є	Немає	Немає	Простий інтерфейс

Самий популярний SIP-клієнт для Iphone – це програма 3CXPhone. Вона вміє працювати з декількома обліковими записами й має велику гнучкість у налаштуваннях, як і її побратим для Windows. Не дивно, що цей SIP-клієнт став самим популярним. Попитом користується й клієнт ForFone – програма має зрозумілий інтерфейс, дозволяє спілкуватися голосом і в режимі чата. У

КбПЗ-2024

число недоліків входить те, що вона орієнтована на роботу з однойменним SIP-провайдером.

Абоненти Comtube зможуть скористатися додатками від свого провайдеру – їх можна знайти в AppStore. Для здійснення дзвінків необхідна програма IP-Phone, аналогічна програмам для вищезгаданих платформ.

Самим зручним і функціональним SIP-клієнтом для Windows є 3CX VOIP Phone. Тут ми бачимо гнучкість налаштувань і приємний інтерфейс, а також зручну книгу контактів із призначенням швидкого доступу. Для роботи з Comtube ми рекомендуємо використовувати IP-Phone.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

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

КБПЗ_2024

Швидкість виконання коду Python

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

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

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

Використання Python

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

Застосування мови програмування Python:

1. BitTorrent – протокол для обміну даними.

2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.

3. Blender – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.

4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.

5. World of Tanks.

6. Вільна енциклопедія Вікіпедія.

7. Пошукова система Google.

КБПЗ_2024

8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.

9. YouTube – популярне відеосховище.

Версії Python

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

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

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

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

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

Коли процесор виконує інструкції програми, він бере участь у процесі, який є відомим як цикл fetch – decode – execute (отримати – декодувати –

КБПЗ_2024

виконати). Цей цикл виконується для кожної інструкції у програмі і складається з трьох кроків:

Отримати

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

Декодувати

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

Виконати

Останній крок циклу – виконати операцію.

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

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

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

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

КБПЗ_2024

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

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

1. Компілятор.
2. Інтерпретатор.

Завантаження Python

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

Середовище програмування для Python

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

Текстові редактори та інтегровані середовища програмування для Python:

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

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

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

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

КБПЗ_2024

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

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

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

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

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

КБПЗ_2024

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

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) сформувати висновки про виконаний обсяг робіт та одержані результати.

К6ПЗ_2024

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Послуги операторів дозволяють компаніям із сегмента SMB бути максимально мобільними при роботі із клієнтами й не пропускати жодного дзвінка. При формуванні пакетів послуг він орієнтується саме на їхній потребі: забезпечення зв'язком віддалених співробітників, швидке й малозатратне масштабування рішення, низька вартість інсталяції й експлуатації. Крім того, хмарна АТМ дозволяє оптимізувати витрати на вихідні виклики в інші міста й країни завдяки опції «пакет хвилин», вигідної компаніям, співробітники яких часто дзвонять клієнтам і партнерам, що перебувають у віддалених регіонах.

Хмарна АТМ дозволяє успішно проводити такі переговори й оцінювати якість обслуговування клієнтів. Завдяки таким можливостям, як інтелектуальне керування вхідними викликами, переадресація, IVR, гнучке налаштування внутрішньої маршрутизації, віртуальний факс, запис розмов, підключення номерів інших операторів з різних регіонів Україні й інтелектуальне керування чергою телефонних викликів, продуктивність роботи істотно підвищується. При цьому співробітники, можуть користуватися програмними телефонами, установленими на їх ПК або мобільних пристроях.

Сучасні рішення пропонує бізнесу три хмарних продукти: віртуальну АТМ, центр обробки викликів і систему CRM. Кожний з них надає широкі можливості для розширення продажів, поліпшення якості обслуговування, підвищення продуктивності співробітників і зміцнення лояльності клієнтів. Наприклад, віртуальна АТМ не тільки дозволяє швидко й недорого розгорнути в організації систему телефонії, але й оптимізувати бізнес-процеси, задіяні для спілкування із клієнтами.

КБПЗ_2024

У лінійці гібридних IP-ATM представлені моделі як для невеликих офісів, так і для компаній зі штатом до 500 чоловік. Система здатна обслуговувати до 500 абонентів і підтримувати 80 дзвінків одночасно

Один з найважливіших способів удосконалювання бізнесу – застосування інформаційно-аналітичних звітів. За допомогою детальної статистики, що враховує всі дзвінки, можна правильно оцінити ефективність роботи й рівень зайнятості кожного співробітника; зрозуміти, у який час активність клієнтів найбільш висока, чи довго їм доводиться чекати відповіді, скільки дзвінків пропускається й з якої причини; визначити, які рекламні канали найбільш ефективні. Це дозволяє одержати повне подання про роботу компанії, приймати зважені управлінські рішення, а також впровадити систему KPI для підвищення ефективності підрозділів, зайнятих продажами або клієнтським обслуговуванням.

Різні схеми розподілу викликів у ВАТМ дозволяють налаштувати її так, щоб на дзвінки покупців завжди відповідав найбільш кваліфікований продавець або, скажемо, клієнтові не доводилося довго чекати відповіді на лінії. Ця функція допоможе одночасно збільшити продажі й підвищити рівень обслуговування.

За рахунок запису розмов удасться поліпшити контроль за роботою співробітників, а також зрозуміти, чи дотримуються вони прийнятого в компанії сценарію продажів. Прослухавши запису, легко довідатися, що цікавить покупців найчастіше, які товари й послуги, запитувані ними, відсутні в асортименті, чим залучають їхні пропозиції конкурентів. У процесі навчання персоналу як приклад можна використовувати грамотно проведені переговори й аналізувати типові помилки. Нарешті, що зберігається інформація завжди допоможе добратися до істини при виникненні спірних ситуацій. Таким чином, завдяки запису розмов можна підвищити лояльність клієнтів і компетентність співробітників, а також знайти нові резерви для росту бізнесу.

Компанії із сегмента SMB, створюючи власний ЦОВ, одержують можливість користуватися інструментами, які застосовують у своїй роботі великі професійні центри обробки викликів. На його основі легко організувати контроль

КБПЗ-2024

за обслуговуванням клієнтів і розподіл вступників дзвінків між вільними в конкретний момент співробітниками. Крім того, ЦОВ дозволяє здійснювати збір і систематизацію безлічі даних, що мають відношення до роботи із клієнтами, і формувати інформаційно-аналітичні звіти, за допомогою яких можна підвищити ефективність діяльності підприємства.

При наявності центра обробки викликів під контролем виявляються всі найважливіші показники якості обслуговування клієнтів: кількість прийнятих і пропущених дзвінків, довжина черги, ступінь завантаженості персоналу, число не зайнятих переговорами операторів, частка – у відсотковому відношенні – успішних з'єднань і негайних відповідей, продуктивність роботи операторів, середня тривалість розмови.

На основі інформації, що надходить у ЦОВ, можна становити звіти, розділяючи їх по темах і різних параметрах. Дані про те, що найбільше цікавить замовників, допоможуть вчасно скорегувати роботу співробітників і написати сценарії відповідей на питання, що задаються часто. Така оптимізація працезатрат буде сприяти підвищенню якості обслуговування.

Крім того, ЦОВ дозволяє відслідковувати й інші параметри, що прямо впливають на лояльність клієнтів і рівень продажів. Мова йде про звіти «Рівень обслуговування» і «Середній час очікування в черзі». Як відомо, сприятливе відношення до компанії в значній мірі залежить від того, скільки часу людина проводить чекаючи відповіді. Скасування виклику означає втрату потенційного клієнта, а це вкрай небажаний результат для тих підприємств, які здійснюють продажу по телефоні. Керівник може самостійно встановити максимально припустимий показник, по досягненні якого йому прийде автоматичне оповіщення, що сигналізує про необхідність вжити невідкладних заходів.

Оскільки ЦОВ має всі перераховані функції, можна в реальному часі спостерігати, як співробітники компанії спілкуються із клієнтами, і завчасно виявляти конфліктні ситуації, щоб не доводити їх до серйозних розбіжностей.

КБПЗ-2024

У системі CRM можливості телефонії об'єднані з функціями електронної пошти. Для цього передбачений єдиний програмний інтерфейс, що дозволяє приймати й робити дзвінки, вести клієнтську базу, зберігати дані про всі виконані дзвінки (дата, тривалість розмови, номера телефонів), здійснювати запису переговорів і прослуховувати їх, відправляти повідомлення електронної пошти й переглядати їх і т.д. Все це полегшує рішення різних бізнес-завдань і є гарною підмогою для навчання персоналу й урегулювання спірних питань.

3.2 Розробка структурної схеми

Хмарне рішення має значно більшу гнучкість, чим традиційне. Якщо буде потрібно розширити штат або скоротити кількість співробітників, відкрити або закрити офіс, не прийде позбуватися від непотрібного встаткування або, навпаки, витрачатися на нове «залізо». Досить просто підписатися на необхідний сервіс або відмовитися від непотрібного – це і є так звана гнучкість масштабування. Причому завжди можна використовувати /стаціонарний, мобільний або SIP-телефон або безкоштовно підключити програмний.

Наприклад, провідні й бездротові SIP-телефони Panasonic, представлені кілька років назад на українському ринку, користуються чималим попитом у вітчизняних підприємців завдяки гарній якості звуку й надійності. У результаті, відповідно до звіту аналітичного агентства J'son & Partners Consulting, в 2016 році компанія посіла друге місце по продажах цього встаткування на українському ринку.

Наприкінці зими були представлені нові моделі SIP-телефонів Panasonic: провідні телефони KX-HDV і SIP-DECT телефон KX-TGP600. Всі вони розроблені з урахуванням вимог найбільших IP-операторів країни. Однієї з відмінних рис апаратів є конкурентоспроможна ціна. Подальший розвиток напрямку SIP-терміналів Panasonic має намір здійснювати в тісному співробітництві з відомими українськими операторами IP-телефонії.

КБПЗ_2024

Хмарне рішення – це не тільки мінімум вкладень, але й максимальна гнучкість, що звільняє від необхідності модернізувати власну інфраструктуру, а також можливість розширення функцій: якщо в компанії з'явилися вільні засоби і їсти бажання спробувати нові можливості, можна перейти на преміальний тариф, а при необхідності заощадити – вибрати більше простий тарифний план. Безсумнівними перевагами виявляються швидке впровадження й простота експлуатації, оскільки провайдер бере на себе всі завдання: від запуску рішення до підтримки відновлень.

Оцінюючи переваги хмарних рішень, неважко зрозуміти, чому, так само як і на світових ринках, вони стають усе більше й більше популярними в Україні. Причина збільшення попиту на хмарні послуги в період кризи й економічної нестабільності очевидна: організаціям зараз невигідні капітальні витрати, і перевага віддається операційним. По даним IDC, в Україні ріст ринку хмарних послуг в 2023 році склав 38%. Хмарні технології можуть стати важливим каталізатором розвитку малого й середнього бізнесу, що, у свою чергу, повинне сприяти відновленню темпів економічного росту.

Клієнти з успіхом використовують хмарну АТМ для виконання різних завдань. Так, наприклад, торговому дому, у штаті якого значиться близько 600 співробітників, знадобилося рішення, здатне забезпечити оперативне розгортання телекомунікаційної інфраструктури для знову, що відкриваються відділів, і підрозділів. Іншому підприємству – мережі салонів «Оптика» – потрібно було впровадити рішення для розвитку клієнтського сервісу й об'єднання семи крапок продажів. А пакет сервісів для компанії «Мобільний Актив», що надає послуги мобільного маркетингу, обійшовся у вісім разів дешевше, ніж передбачалося споконвічно.

Коли хмарні АТМ доповнюються пакетом спеціалізованих послуг, замовник одержує у своє розпорядження «віртуальний офіс». Однак користувачам віртуальних рішень варто пам'ятати про можливі ризики.

КБПЗ_2024

Представники різних сфер бізнесу виділяють цілий спектр переваг, завдяки яким IP-телефонія виявляється більше вигідної для підприємства, чим традиційна. Так, у ході проведеного опитування керівники підтвердили, що IP-телефонія дозволяє оптимізувати витрати на зв'язок з віддаленими представництвами й партнерами, знизити витрати на експлуатацію мережі, централізувати керування телекомунікаційною інфраструктурою й підвищити ефективність роботи співробітників. У цьому контексті хмарна АТМ, яку можна назвати інтелектуальною надбудовою до IP-телефонії, є необхідним інструментом для створення й подальшого керування телефонізацією компанії.

Основні замовники цього сервісу – середній і малий бізнес, а також великої мережної компанії із широкою географією присутності. Таким організаціям необхідні гнучкі й масштабовані VoIP-рішення, які легко інтегруються з існуючою інфраструктурою й дозволяють створювати проекти, що максимально враховують специфіку бізнес-процесів. Хмарна АТМ найбільшою мірою затребувана підприємствами, що працюють у сфері послуг, роздрібній торгівлі, готельному й ресторанному бізнесі, де якість обслуговування клієнтів відіграє визначальну роль, а бізнес найчастіше має розгалужену структуру.

Хмарна АТМ розміщається в операторській хмарі – на захищеному сервері, надаваному оператором для потреб замовника, – і поставляється разом з устаткуванням, що значно знижує загальну вартість впровадження, а також можливі ризики по сумісності сервісу з існуючою телекомунікаційною інфраструктурою. У результаті клієнт одержує функціональну систему інтелектуальної телефонії, що обслуговує оператор, і звільняється від додаткових витрат на установку, експлуатацію й технічну підтримку в режимі 24/7.

Офісна IP-АТМ

У віртуальних АТМ є очевидні мінуси: можливі проблеми з відказостійкістю (надійність і стабільність роботи системи в значній мірі залежать від якості каналів зв'язку), не гарантується конфіденційність інформації (за безпеку відповідає провайдер). Саме тому багато компаній воліють

КБПЗ_2024

установлювати АТМ в офісі. При цьому власна IP-АТМ здатна працювати з різними видами каналів, тоді як віртуальна звичайно не має таких ресурсів.

Для малих і середніх компаній дуже важливе оперативне розгортання системи й низька вартість володіння. Хмарні АТМ залучають можливістю щодо швидкого розгортання, але за вартістю вони не завжди виявляються більше дешевими рішеннями, чим «звичайні» IP-АТМ. Віртуальні системи більше підходять тим компаніям, які створюються для реалізації коротких проектів, а іншим найчастіше вигідніше й безпечніше користуватися власними IP-Системами.

Установлюючи сучасні системи зв'язку, компанія одержує насамперед надійні і якісні внутрикорпоративні комунікації й скорочення витрат, тому що завдяки використанню різних каналів зв'язку виклики можуть відбуватися по найбільш вигідних тарифах.

Системи зв'язку забезпечують мобільність персоналу за рахунок організації мікростільникових систем DECT і оснащені функціями мобільної інтеграції й підтримки програмних клієнтів. Розширювати й модернізувати мережа можна без заміни всього встаткування, оскільки моделі IP-АТМ різних серій сумісні між собою, а нові покоління продуктів розробляються з урахуванням принципу наступності.

Системи дозволяють підключити конференц-зв'язок, а також скористатися функціями центра обробки викликів і бізнес-додатків – всі ці можливості допомагають співробітникам компаній поліпшити взаємодію й підвищити якість своєї роботи. Устаткування повинне встановлюватися авторизованим фахівцем, у цьому випадку замовник може бути впевнений у тому, що воно буде функціонувати коректно.

У процесі створення платформи особлива увага приділялася захисту інвестицій замовника, для чого забезпечується проста інтеграція у вже наявну інфраструктуру, не потребуючої прокладки додаткових кабелів і заміни телефонних терміналів.

КБПЗ-2024

В убудованій системі уніфікованих комунікацій передбачена можливість використання найбільш затребуваних функцій IP-телефонії, а також голосової пошти й додатків для створення невеликого або середнього контакт-центра, у чому зацікавлені не тільки великі, але й невеликі компанії.

Оскільки до єдиного короткого внутрішнього номера можна підключити три телефони, у тому числі мобільні, SIP- і стаціонарні пристрої, абонент зможе відповісти по кожному з них незалежно від свого місцезнаходження.

IP-ATM дозволяє зв'язати різні офіси територіально розподіленої компанії, у тому числі глобальної. При відкритті нових філій можна підключати абонентів до телефонної мережі, побудованої на базі IP-телефонії, за допомогою ресурсів центральної IP-ATM. Всі ці засоби покликані підтримати ефективну роботу компаній з будь-якої сфери бізнесу.

IP-телефонія як інструмент бізнесу

Завдяки IP-телефонії бізнес одержав можливість набагато ефективніше використовувати телефон як інструмент взаємодії із клієнтами. Всі процеси, так чи інакше пов'язані зі спілкуванням по телефоні, тепер можуть бути детально відстежені й глибоко оптимізовані. Сучасні цифрові ATM оснащені безліччю корисних функцій. Правда, висока вартість, складність налаштування й необхідність обслуговування роблять їх малопривабливими для невеликих компаній.

Але є й ще одна важлива деталь – перехід до VoIP відкрив перед малим і середнім бізнесом можливості, які раніше були доступні тільки великим організаціям з потужною телекомунікаційною інфраструктурою. Використовуючи хмарні комунікаційні сервіси, наприклад віртуальну ATM, будь-який стартап сьогодні може застосовувати у своїй роботі всі існуючі інструменти для розширення вирви продажів або підвищення лояльності клієнтів. Втім, інтерес не обмежується сегментом СМБ – великі компанії теж нерідко вирішують свої завдання за допомогою хмарної телефонії.

КБПЗ-2024

Як показують дослідження, бізнес добре розуміє, що може дати йому IP-телефонія. Так, тільки половина користувачів ВАТМ задовольняються лише звичайними функціями начебто розподілу й переадресації викликів, голосового меню або автовідповідача. Іншим потрібні саме можливості вдосконалювання бізнес-процесів, які надає хмарна АТМ. А найбільшою популярністю в них користуються інструменти, що дозволяють знайти нові крапки росту, – насамперед інформаційно-аналітичні звіти.

Комунікаційні сервіси по хмарній моделі придбали особливу популярність із початком кризи: вони допомагають не тільки заощадити на впровадженні й обслуговуванні, але й збільшити дохід, оптимізуючи роботу компанії різними способами, а також гнучко міняти організаційну структуру без шкоди для діяльності підприємства. Постійний ріст ринку ВАТМ протягом уже декількох років – краще тому підтвердження.

Як і у випадку з іншими технологіями, успішно застосовуваними в бізнесі, затребуваність подібних рішень неминуче веде до появи нових функцій, що враховують різні потреби й бізнес-моделі, і навіть нових класів продуктів, в основу яких покладена функціональність хмарних АТМ, наприклад, для створення операторських центрів. Розширюються й можливості інтеграції IP-телефонії з іншими корпоративними системами, такими як CRM і ERP.

У цілому ж, я думаю, при грамотному застосуванні інструментів IP-телефонії вигода від них може бути набагато більше, ніж просто економія на послугах зв'язку й інфраструктурі.

Гібридні IP-АТМ

На відміну від традиційних офісних АТМ, гібридні IP-АТМ часом мають широкий набір інтелектуальних функцій: вони дають малим і середнім підприємствам безліч конкурентних переваг, про які керівники раніше й мріяти не могли.

У даній роботі реалізована побудова мережі All-over-SIP малого підприємства. Такі рішення легко модернізуються, масштабуються, інтегруються

КБПЗ-2024

зі сторонніми сервісами й додатками: системами обліку, CRM або контакт-центрами. Цей підхід дозволяє краще організувати роботу співробітників.

Рисунок 3.1 – Структурна схема системи

КбПЗ_2024

От лише один приклад: інтелектуальна маршрутизація за допомогою нової «розумної» аналогової станції Yeastar N824. АТМ запам'ятовує маршрут для вхідних і вихідних викликів. Так, якщо абонентові дзвонили з організації, у якій установлена Yeastar N824, а він не зміг відповісти, то при зворотному дзвінку виклик буде спрямований на той додатковий номер, з якого йому дзвонили. Цей принцип працює як при пропущеному дзвінку, так і при зробленому виклику.

Аналогічно, при повторному дзвінку клієнт прямо зв'язується з тим менеджером або оператором, з яким уже вів переговори. Безліч подібних «фішок» (будь те запис всіх розмов, можливість зручного перехоплення дзвінка й

трансфер натисканням однієї кнопки) заощаджують час кожного співробітника й ресурси всієї компанії. При цьому використання таких технологій приводить до загального зниження витрат на зв'язок, що вкрай важливо для невеликих організацій.

Сучасні рішення допомагають підвищити лояльність клієнтів і ефективно обробити кожний вхідний дзвінок, що сприяє збільшенню прибутку. При цьому в станціях Yeastar вся функціональність (крім деяких специфічних модулів для готелів або тарифікатора) надається за замовчуванням і не вимагає додаткового ліцензування, що спрощує розрахунок повної вартості рішення.

До станцій Yeastar можна додавати IP-камери Milesight і IP-домофони ITS. Це ще два кубики в конструктор рішень All-over-SIP. Ці пристрої підключаються по протоколі SIP і стають повноцінними учасниками комунікаційного простору офісу. При виклику з домофона здійснюється належна маршрутизація викликів і відкрити двері може будь-який співробітник, тому для забезпечення доступу в офіс не потрібно будувати окрему мережу.

Крім свого прямого призначення (відеоспостереження й записи), IP-камери Milesight можуть здійснювати відеозв'язок по протоколі SIP, що, наприклад, затребувано при організації віддаленого відео- і аудіоконтроля важливих об'єктів для керівника підприємства. Це можна робити з настільного офісного відеотелефона (наприклад, Yealink VP530) або за допомогою клієнта, установленого на мобільному телефоні. Такий підхід дозволяє створювати проекти «Розумний будинок» і різні комплексні рішення для великих підприємств, а також вирішувати завдання, актуальні для невеликих компаній.

IP-телефонія стала не тільки засобом комунікацій між співробітниками, засобом підтримки колективної роботи й взаємодії з партнерами й замовниками, але й потужним інструментом для продажів і просування продуктів і послуг. Багато організацій уже використовують у своїй діяльності функції ЦОВ, CRM і інші додатки, провайдери пропонують відповідні хмарні сервіси, а постачальники встаткування IP-телефонії створюють нові рішення для корпоративного ринку.

КБПЗ_2024

Сучасні комунікаційні системи, продукти й послуги можуть допомогти малим і середнім підприємствам вести бізнес, збільшити обсяг продажів, просувати послуги, підвищити ефективність бізнесу.

3.3 Розробка функціональної схеми

В процесі практичної реалізації теоретичних принципів розробки системи, додатково розглянутих вище, була розроблена функціональна схема системи, яка зображена на рисунку 3.2. Корпоративна IP-телефонія – це інтеграція системи телефонії в корпоративну мережу передачі даних. У найпростішому випадку система IP-телефонії 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.

Перелічемо 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

КБПЗ_2024

викликів (через WAN-канал зв'язку). При відмові WAN-каналу зв'язку функції 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, 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 в порівнянні з рішенням традиційної телефонії, що включає 100%

КБПЗ_2024

системних телефонів, буде дешевше.

В-шостих, можливість тісної інтеграції IP-телефонії з додатками, які допомагають оптимізувати робочий процес. Оскільки IP-телефонія 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 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 наявність відкритих інтерфейсів взаємодії, таких як CTI, 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 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

КБПЗ_2024

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 простій даної системи становить менш 6 хвилин у рік. У системі IP-телефонії, 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 – 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, необхідно додержуватися всіх рекомендацій (керівництва по дизайну) виробника встаткування як при створенні систем IP-телефонії, так і при побудові комп'ютерних мереж високої доступності.

Нові можливості

Відзначимо деякі останні нововведення в IP-телефонії, які істотно

КБПЗ_2024

розширюють обрії її застосування.

Video IP-телефонія

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

IP-АТМУ Cisco CallManager починаючи з версії(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 встаткування, програмне забезпечення, інсталяцію й навчання технічного персоналу не потрібно. Необхідно лише підключити USB-камеру Cisco VT Advantage до комп'ютера користувача, що використовує Cisco 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 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 пропущення залежно від завантаженості телекомунікаційних ліній. Максимально можлива швидкість передачі відеоінформації становить 4,5 Мбіт/с.

Для організації відеоконференц-зв'язку в архітектуру Cisco додається пристрій Cisco IP/VC серії 3500, що надає ресурс відеоконференц-сервера й інтегрується із кластером CallManager по протоколах SCCP або H.323. Пристрій

КБПЗ_2024

Cisco 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, 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

Нові засоби безпеки, включені в Cisco CallManager, розширюють можливості захисту інтелектуальних мереж. В Cisco CallManager 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 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.

КБПЗ-2024

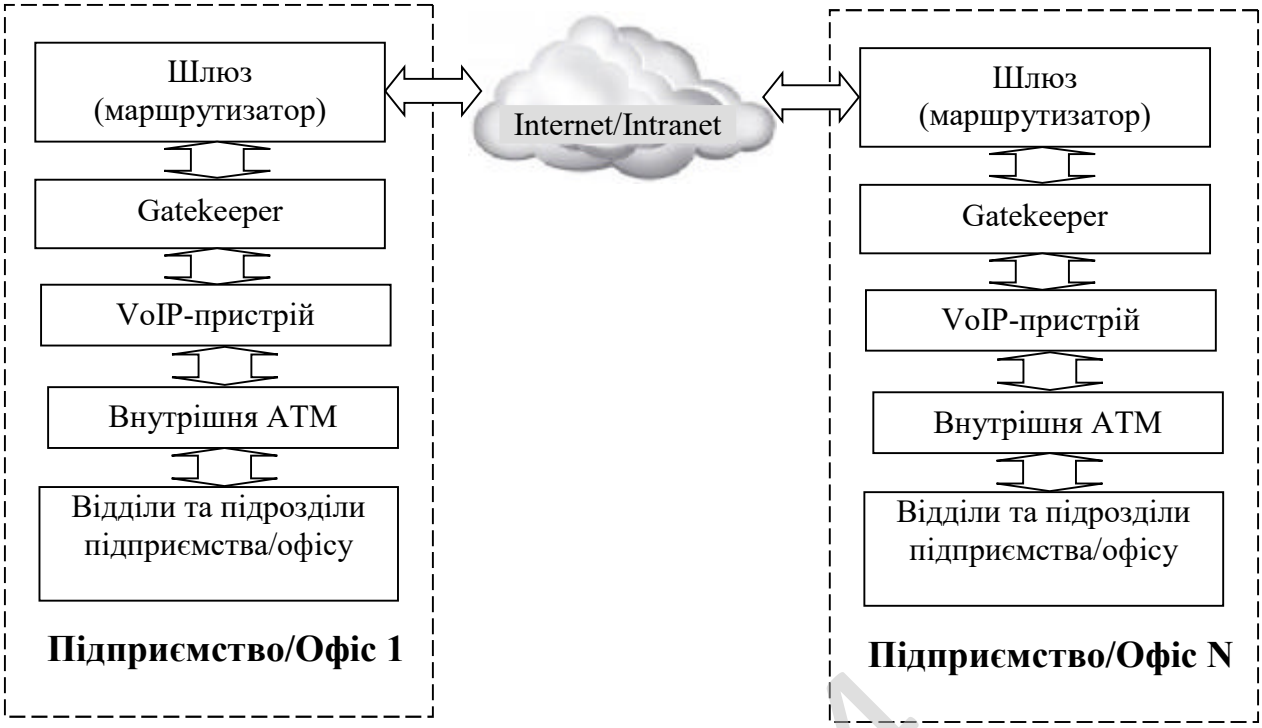

Рисунок 3.2 – Функціональна схема системи

КБПЗ_2024

Завдяки функціональній схемі можна чітко побачити основні функціональні блоки системи та взаємозв’язки між ними. При розробці функціональної схеми основний упор робився на існуючи розробки ПЗ і їх модулі допомоги. Аналіз рисунка 3.2 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 внутримашиної 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.).

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

КБПЗ_2024

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.

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 установлення з'єднання, таких як ЛОМ.

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

– SIP (Session Initiation Protocol). Протокол, що визначає команди й відповіді для встановлення й завершення телефонних викликів. Також деталізує

КБПЗ_2024

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 канальному рівні застосовуються протоколи: Frame Relay, ATM, Ethernet, MLPPP, PPP, HDLC. На мережному проткол IP. На транспортному – RTP/UDP (голос), TCP/UDP (керування). На сеансовому H.323/SIP/MGCP. На представницькому рівні використовуються кодеки, а на прикладному софтфони й додатки Call Manager.

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

Коли відбувається виклик, пристрій генерує цифри номера дозвону як

КбПЗ_2024

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.

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.

КБПЗ_2024

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:

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

КБПЗ_2024

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.

H.323

Як уже говорилося, H.323 є набором протоколів. Всі пристрої, використовувані H.323, можна поділити на чотири категорії: термінали, шлюзи, гейткіпери (Gatekeeper – гейткіпер) і точки багатопунктового контролю (Multipoint Control Unit – MTU).

Термінали, також називані кінцевими точками (endpoints), надають користувальницький інтерфейс до протоколу H.323 і забезпечують двосторонній мультимедійний зв'язок реального часу. Шлюзи виконують роль "перекладачів" для забезпечення взаємодії між H.323 і не-H.323 сутностями. Шлюзи, так само як і термінали, розглядаються як кінцеві точки. Гейткіпери виконують функції контролю викликів, такі як трансляція адрес і керування займаною смугою пропущення. Гейткіпери можна вважати найбільш важливим компонентом у 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 і MCU повинен мати свій унікальний IP-адресу. Це також ставиться 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/гейткіперуми. Протокол UDP використовується для передачі безпосередньо голосу через мережу.

КБПЗ_2024

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 ставиться 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 гейткіперуми. Він застосовується для здійснення реєстрації, контролю доступу, статусу й змін доступної смуги пропущення, 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

КБПЗ_2024

стандартів кодування аудіо й відео, включаючи G.7XX для аудіо й H.26X для відео.

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 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 інформується про адресу зони.

Внутрізонові 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 дозволяє виклик з вимогою сигналізації (дозвону) прямо, посилаючи повідомлення ACF (Admission Confirmation).

Міжзонові виклики

Допустимо, гейткіпер A контролює зону A, а гейткіпер B – зону B, шлюз X зареєстрований на гейткіпері A, а шлюз Y – на гейткіпері B, і шлюз A хоче

КБПЗ_2024

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 шлюзу X не знає IP-адресу шлюзу Y і запитує адресу в гейткіперу шлюзу Y.

3. Місце розташування підтверджене (LCF – Location confirm). Гейткіпер шлюзу Y відповідає IP-адресою шлюзу Y.

4. Гейткіпер шлюзу X підтверджує його запит і надає йому IP-адресу шлюзу Y.

5. Установлення з'єднання між шлюзами.

Установлення з'єднання

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

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:

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 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

КБПЗ_2024

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 двунаправленими.

Установлення логічних каналів відбувається в такий спосіб:

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

2. Шлюз Y підтверджує запит, посилаючи 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 медиаканал зі шлюзом Y, посилаючи H.245 повідомлення Open Logical Channel, включаючи адресу RTCP каналу.

6. Шлюз Y підтверджує встановлення логічного каналу зі шлюзом X, посилаючи 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-потоку. Кінцеві точки можуть спробувати змінити виділену смугу пропущення, що вони споконвічно запросили. Для збільшення виділеної смуги пропущення кінцеві точки повинні запросити на цей дозвіл у гейткіперу.

Завершення виклику зупиняє медиапоток і закриває логічні канали. Воно

КБПЗ-2024

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, називаного 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 діяльністю шлюзів у припущенні, що шлюзи фіксують події й доповідають про

КБПЗ_2024

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.

Взаємодія агентів і шлюзів

Процес взаємодії телефонного агента зі шлюзами для забезпечення телефонного виклику можна описати наступною послідовністю дій.

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 "знать", 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 цифры, 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

КБПЗ_2024

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 В 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 проінструктував повідомити про таку подію, шлюз А посилає агентові повідомлення.

11. Агент розсилає повідомлення DLCX (Delete Connection) кожному шлюзу.

12. Шлюзи видаляють з'єднання й відповідають.

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

КБПЗ_2024

3.4 Розробка діаграми процесів

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

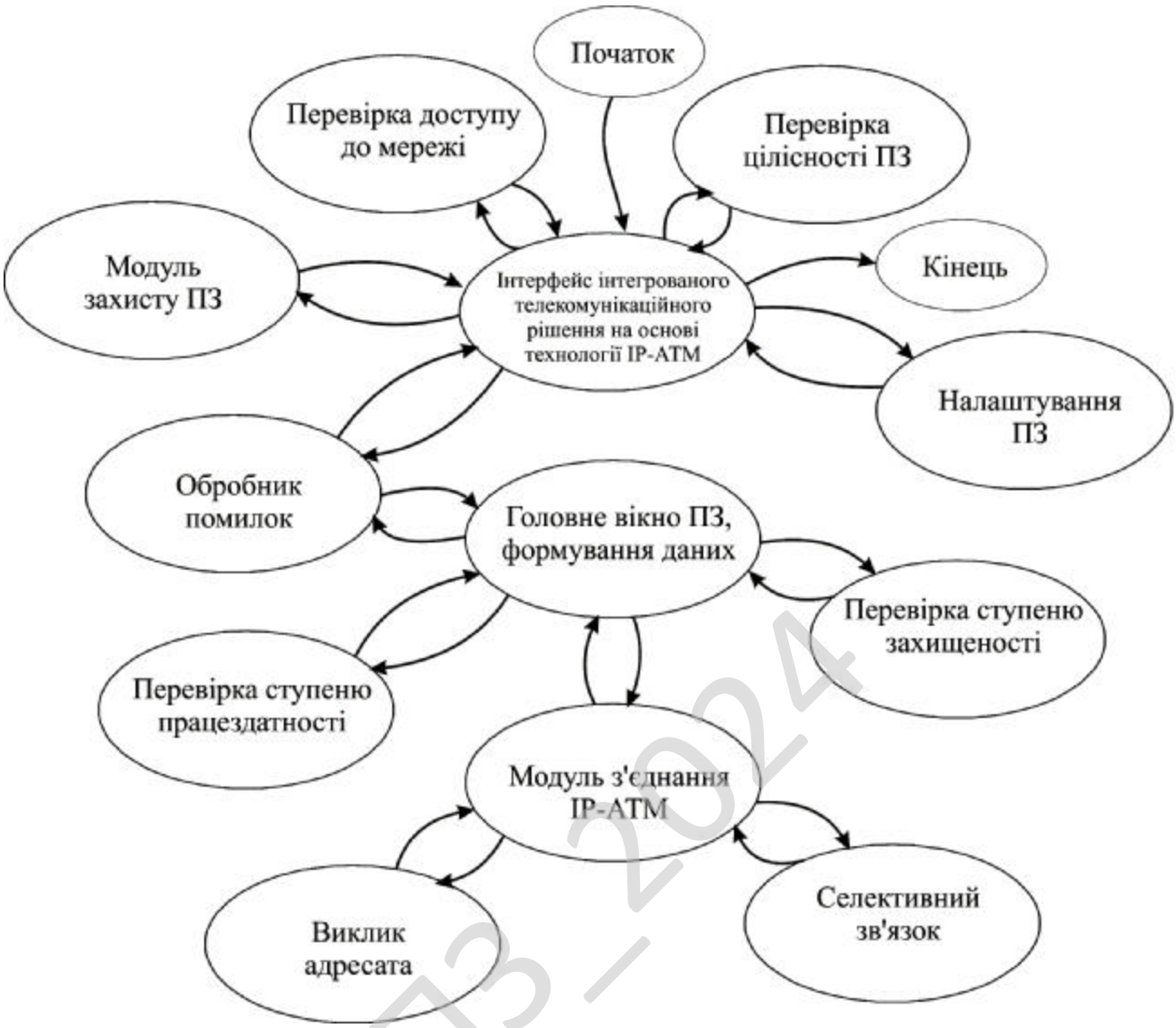

Рисунок 3.3 – Діаграма взаємодії процесів

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється. Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходе взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

КбПЗ_2024

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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

КБПЗ_2024

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

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

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

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ.

КБПЗ_2024

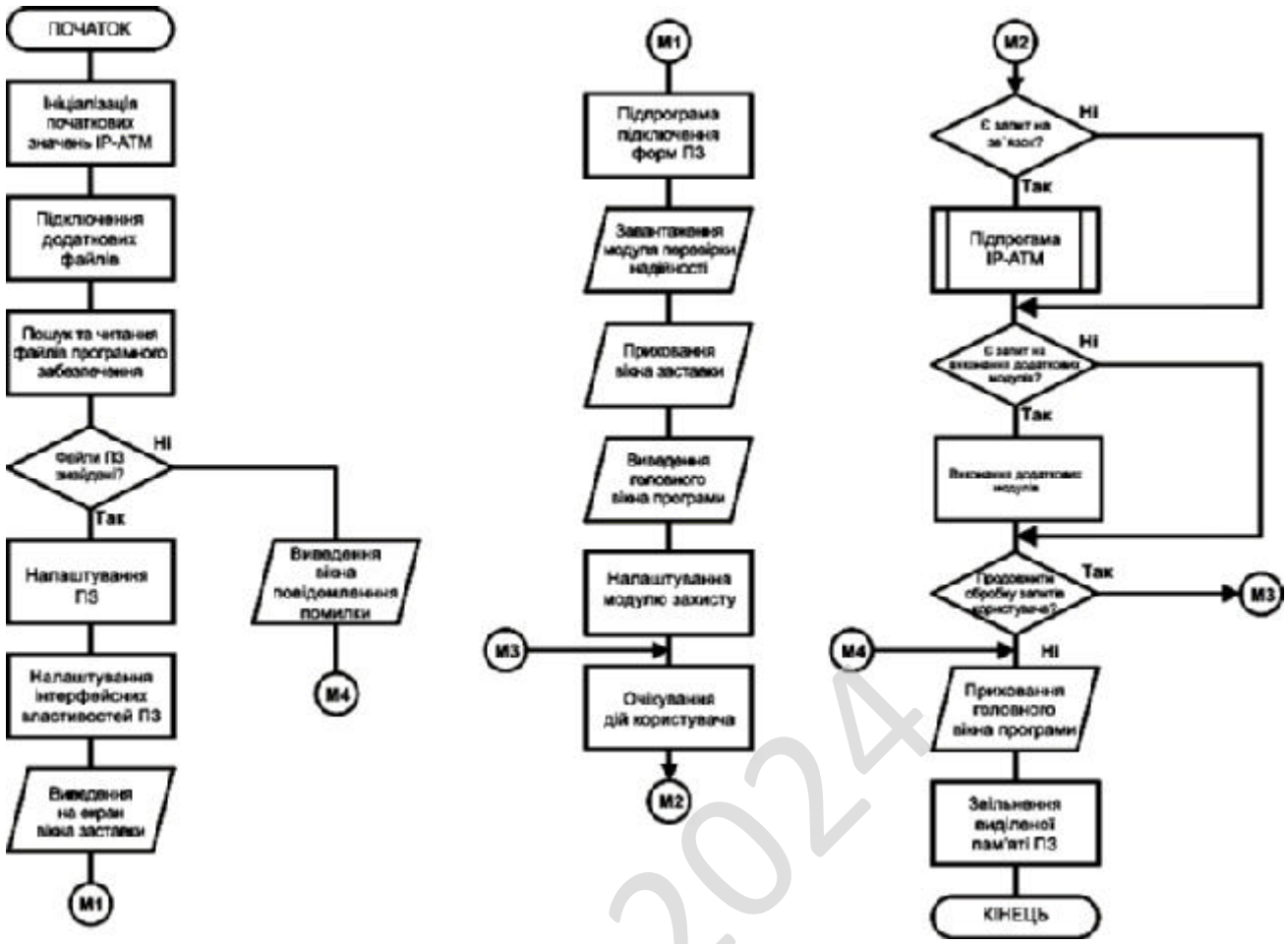

Рисунок 4.1 – Блок-схема основної програми

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

КБПЗ_2024

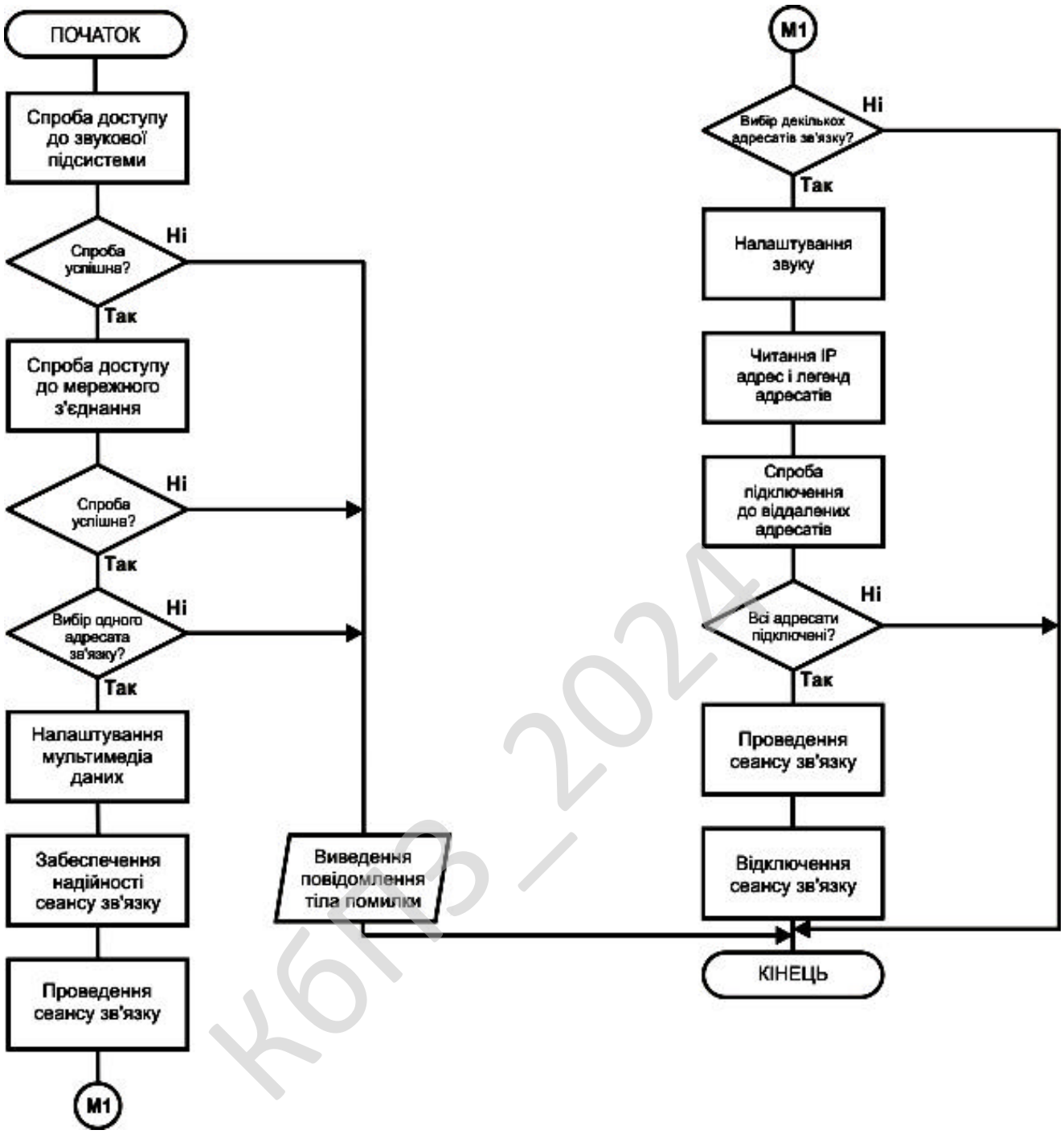

Рисунок 4.2 – Блок-схема роботи підпрограми

Розглянемо використані технології та їх основні компоненти що підтверджують правильність використаних проектних рішень.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених

КБПЗ_2024

мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

– набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;

– набір клієнтів, які використовують сервіси, що надаються серверами;

– мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

– рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;

КБПЗ_2024

– прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;

– рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

– модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

– модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузерами.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є

КБПЗ_2024

важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення – звичайний браузер. Це позбавляє його необхідності завантажувати та інсталювати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої

КБПЗ_2024

обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання. Redmine написано на мові Ruby і є ПЗ розробленим з використанням відомого веб-фреймворку Ruby on Rails, що означає легкість в розгортанні системи та її адаптації під конкретні вимоги. Для кожного проекту можна вести свої вікі та форуми.

Функціональні можливості:

– Ведення декількох проектів.

– Гнучка система доступу з використанням ролей.

– Система відстеження помилок.

КБПЗ-2024

– Діаграми Ганта та календар.

– Ведення новин проекту, документів та управління файлами.

– Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.

– Власна Wiki для кожного проекту.

– Форуми для кожного проекту.

– Облік часових витрат.

– Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.

– Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Bazaar и Darcs).

– Створення записів про помилки на основі отриманих листів

– Підтримка LDAP автентифікації.

– Можливість самореєстрації нових користувачів.

– Багатомовний інтерфейс (у тому числі українська мова).

– Підтримка СКБД: MySQL, PostgreSQL, SQLite.

Діаграма Ганта (Gantt chart, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка робіт за будь-яким проектом. Є одним з методів планування та управління проектами.

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання; показується вертикальна лінія, що відповідає моменту «сьогодні».

КБПЗ_2024

Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу.

Система відстеження помилок Багтрекер – прикладна програма для допомоги розробникам програмного забезпечення (програмістам, тестувальникам тощо) враховувати і контролювати помилки, знайдені у програмах, питання щодо функціональності, рішення та оновлення, побажання користувачів, а також стежити за процесом їх виконання.

Кожному, хто розробляв програмні продукти, добре знайоме співвідношення «20/80» – останні 20 % роботи тривають 80 % часу.

Як це не парадоксально, але нічого дивного в цій пропорції немає, адже саме на завершальній стадії починається тестування проекту, коли виявляються помилки, і що більший проект, то більше буде знайдено помилок.

Водночас досить часто виявляється, що більшість цих помилок були відомі та могли бути виправлені з меншими витратами на попередніх стадіях роботи, але не були вчасно описані, а потім загубилися серед інших важливих завдань.

Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

Сукупність інформації про дефект. Головний компонент такої системи – база даних, що містить відомості про виявлені дефекти. Ці відомості можуть включати в себе:

– номер (ідентифікатор) дефекту;

– хто повідомив про дефект;

– дата і час виявлення дефекту;

– версія продукту, в якій виявлено дефект;

– серйозність (критичність) дефекту та пріоритет рішення;

– опис кроків для відтворення дефекту (неправильної поведінки програми);

КПЗ_2024

– відповідальний за усунення дефекту;

– обговорення можливих рішень та їх наслідків;

– поточний стан виправлення дефекту;

– версії продукту, в якій дефект виправлений.

Крім того, розвинені системи надають можливість прикріплювати файли, які допомагають описати проблему, наприклад, дамп пам'яті або скріншот.

Використання. Основна перевага систем відстеження помилок полягає в забезпеченні чітких централізованих оглядів, запитів на розробку (включаючи помилки і виправлення) та їх стан. У корпоративному середовищі, системи відстеження помилок можуть бути використані для генерації звітів по продуктивності програмістів виправлення помилок. Однак, це може іноді приводити до неточних результатів, тому що різні помилки можуть мати різні ступені пріоритету та серйозності, що пов'язано з складністю їх фіксації.

Життєвий цикл дефекту. Як правило, система відстеження помилок використовує той чи інший варіант «життєвого циклу» помилки, стадія якого визначається поточним станом помилки.

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.

2. Призначений – призначений відповідальний за виправлення дефекту.

3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

– Виправлено (виправлення включені у версію таку-то).

– Дубль (повторює дефект, що вже знаходиться в роботі).

– Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).

– «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

КБПЗ_2024

4. Далі тестувальник проводить перевірку виправлення, залежно від чого дефект або знову переходить у стан «Призначений» (якщо він описаний як виправлений, але не виправлений), або у стан «Закрито».

5. Відкрито повторно – дефект знайдено знову в іншій версії.

Система може надавати адміністраторові можливість налаштування користувачі, які можуть переглядати і редагувати помилки залежно від їх стану, переводити їх в інший стан або видаляти.

У корпоративному середовищі, система відстеження помилок може використовуватися для отримання звітів, що показують продуктивність програмістів при виправленні помилок. Однак, часто такий підхід не дає достатньо точних результатів через те, що різні помилки мають різну ступінь серйозності та складності. При цьому серйозність проблеми прямо не стосується складності її усунення.

Була використана водоспадна (каскадна) модель життєвого циклу ПЗ (waterfall model) – послідовний метод розробки програмного забезпечення, названий так через діаграму схожу на водоспад.

Ця модель розробки запозичена з системної інженерії у виробництві та будівництві – областях, в яких зміни на пізніх етапах дуже дорогі, або неможливі. Наприклад, для створення складних інженерних конструкцій (споруд, літаків, мостів і т.п.). Зміни в проекті фундаменту будинку після того, як покладений дах коштують дуже дорого, тому перфекціонізм на початкових етапах проектування просто необхідний. Інженери, які починали займатись розробкою програмного забезпечення перейшовши з інших галузей, просто адаптували звичну модель, тому що на ранніх етапах розвитку комп’ютерної техніки не було методологій створених саме для програмування. Проте, схожі методології застосовуються для програмного забезпечення й далі, у випадках коли вимоги фіксовані, і вимагається висока якість та надійність, наприклад в системах для військових чи медичних потреб.

КБПЗ_2024

Перший формальний опис водоспадної моделі, після якої вона стала популярною був здійснений В. В. Ройсом у 1970. Попри те, що стаття містить переважно критику методу, на неї часто посилаються.

Переваги методу:

– Ніяких переробок.

– Гарна специфікація перетікає в гарну документацію.

– Зрозуміла модель.

– Розробники можуть мати низьку кваліфікацію.

Недоліки:

– Необхідний перфекціонізм на кожному етапі.

– Важко вносити зміни (якщо взагалі можливо).

– Надлишкове проектування.

– Поділ розробників на "perfect" та "code monkeys".

Модифікації. Через те що цей метод погано підходить для розробки саме ПЗ, частіше використовують його модифікації.

Найвідоміша модифікація – Sashimi. Названа так через японську страву сашимі (суші нарізане і сервіроване так, що складені рядочком шматочки накладаються один на одного). В моделі розробки Сашимі фази життєвого циклу йдуть одна за одною, але при цьому перекриваються одна з одною в часі.

КБПЗ_2024

4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 7624:2014 («Калина»). При розробці національного стандарту ДСТУ 7624:2014 (блоковий шифр «Калина» і режими його роботи) було ухвалене рішення забезпечити прозорість проектування й використовувати консервативний підхід із застосуванням добре досліджених конструкцій, що забезпечують запас стійкості для безпечного застосування алгоритму в умовах істотного прогресу криптоаналітичних технік і засобів обробки даних.

Національний стандарт підтримує розмір блоку й довжину ключа шифрування 128, 256 і 512 біт (довжина ключа дорівнює розміру блоку або у два рази перевищує його), забезпечуючи нормальний, високий і надвисокий рівень стійкості (зараз це єдиний у світі стандарт блокового шифрування, що підтримує 512-бітові симетричні ключі). Різні варіанти забезпечують гнучкість вибору параметрів для розроблювачів систем криптографічного захисту, що дозволяє одержати як найвищий рівень швидкодії, так і найбільший запас стійкості перетворення.

Високорівнева конструкція використовує добре досліджену Square-подібну SPN-структуру, застосовувану в алгоритмах AES/Rijndael, Whirlpool, «Стрибог», «Коник» і багатьох інших. Циклове перетворення побудоване на базі таблиць підстановки (S-блоків) і множення на МДР-матрицю над кінцевим полем, забезпечуючи необхідні криптографічні властивості. Застосування саме такої конструкції дозволяє забезпечити доказову стійкість до диференціального, лінійному й ін. видам криптоаналізу, одночасно забезпечуючи ефективну реалізацію на широкому спектрі програмних і програмно-апаратних платформ. При виборі розміру МДР-матриці був прийнятий в увагу розмір кешу L1 сучасних і перспективних процесорів, що дозволило оптимізувати швидкодія програмної реалізації шифру.

Альтернативний варіант, який розглядався при розробці циклової функції, – ARX перетворення (Addition-rotation-xor). Цей підхід реалізований у шифрі SPECK (розроблений Агентством національної безпеки США й переданий у відкритий доступ), у блокових алгоритмах, на основі яких побудовано сімейство геш-функцій SHA-0,1,2 і ін. Перевагою походу є компактність і швидкодія перетворення. Але, у той же час, є й істотний недолік, пов'язаний з відсутністю методів, що дозволяють виконати строге аналітичне обґрунтування криптографічної стійкості таких розв'язків. Навіть із наймогутнішими у світі можливостями для аналізу, США кілька раз були змушено модифікувати свої стандарти гешування через знайдені уразливості: з 1993 по 1995 рр. діяв SHA-0, з

КБПЗ_2024

1995 по 2001 рр. застосовувався SHA-1, з тих пор використовується SHA-2. Через питання, що піднімаються, до стійкості й цієї версії, у США з 2008 по 2012 рр. був проведений міжнародний конкурс SHA-3. До теперішнього часу розроблений проект стандарту FIPS 202, що описує нову криптографічну геш-функцію.

Таким чином, консервативний і прозорий підхід до проектування нового національного стандарту України обумовив вибір добре перевіреної конструкції на базі S-блоків і лінійного перетворення, для якої можливо забезпечити доказову стійкість до різних видів криптоаналізу.

При порівнянні характеристик S-блоків і інших симетричних перетворень, можна відзначити, що саме національний стандарт України забезпечує найбільшу нелінійність булевих функцій, що дає додатковий запас стійкості до лінійного криптоаналізу. Ще більше значення нелінійності для взаємо- однозначної підстановки можна забезпечити, застосовуючи, наприклад, аффінно-тквівалентні статечні функції в кінцевому полі, але такі перетворення, використані в AES, Camellia і ін. алгоритмах, ставлять шифр під погрозу реалізації алгебраїчної атаки (цей метод криптоаналізу був успішно застосований проти шифру Keeloq, використовуваного в системах автомобільної безпеки).

У якості схеми розгортання ключів була запропонована нова конструкція з наступними властивостями:

– забезпечення криптографічної стійкості до відомих методів аналізу, відсутність «слабких» ключів, які можуть погіршити властивості перетворення;

– зручність програмної й програмно-апаратної реалізації (для формування циклових ключів застосовуються тільки операції, використовувані при шифруванні);

– висока обчислювальна складність відновлення ключа шифрування по одному або декільком цикловим ключам.

Остання властивість забезпечує додатковий захист до атак на реалізацію, коли зловмисник намагається атакувати інженерні розв'язки (вимірюючи споживаний пристроєм струм, навмисне викликаючи збої в роботі через

КБПЗ_2024

навмисний перегрів шифратора та ін.). Ця особливість є істотною перевагою для ряду додатків, зокрема, при реалізації шифрування на смарт-картах, USB-токенах та ін., коли ключ прошитий у пристрої й повинен бути захищений від зовнішнього доступу (наприклад, у модулях доступу до платних цифрових ТБ-каналівм і ін.)

Кількість циклів шифрування залежить від довжини ключа: 10 циклів для 128-бітового, 14 циклів для 256-бітового й 18 циклів для 512-бітового ключа шифрування.

У порівнянні з іншими алгоритмами на основі Square-подібної SPN-структури, блоковий шифр «Калина» має наступні істотні конструктивні відмінності:

– початкове й кінцеве забілювання з використанням модульного додавання (264) для підвищення складності криптоаналітичних атак;

– застосування чотирьох різні S-блоків (замість одного) із властивостями для захисту від алгебраїчних атак, і при порівнянні характеристик з іншими блоковими й потоковими шифрами забезпечують найбільшу нелінійність булевих функцій (104), що дає додатковий запас стійкості перетворення;

– збільшений розмір МДР-перетворення, що поліпшує криптографічні властивості й дозволяє оптимізувати швидкодія на сучасних 64-бітових платформах;

– нову односпрямовану схему формування циклових ключів, що забезпечує захист від атак, ефективність програмної й програмно-апаратної реалізації, разом з додатковою стійкістю до методів аналізу спеціального виду.

Оцінка криптографічної стійкості до диференціального, лінійному, алгебраїчному, інтегральному й іншим методам аналізу (практичний критерій) показала, що шифр є стійким при 6 циклах для 128-бітового блоку, 7 циклах для 256-бітового й 9 циклах для 512-бітового (кожний додатковий цикл забезпечує експонентний ріст складності криптоаналізу). Таким чином, шифр, що містить 10,

КБПЗ-2024

14 і 18 циклів для блоку 128, 256 і 512 біт відповідно, забезпечує захист від розглянутих видів аналізу й має істотний запас стійкості.

Крім блокового шифру, ДСТУ 7624:2014 визначає режими роботи, що відповідають як ISO 10116:2006, так і додаткові, призначені для сучасних систем криптографічного захисту IP-трафіка, прозорого шифрування носіїв інформації й ін. У стандарті визначені обсяги повідомлень, після обробки яких потрібна обов'язкова зміна ключа. Крім того, приводяться рекомендації розроблювачам, що обертають увагу на необхідність запобігання атак з використанням особливостей реалізації засобів шифрування. Зокрема, враховані особливості, що дозволяли організацію атак BEAST і CRIME/BREACH у протоколах SSL/TLS, повторне приймання повідомлення, відновлення конфіденційних параметрів на основі залежності часу шифрування від оброблюваних даних (через промахи кешу при табличній реалізації) і ін.

КБПЗ_2024

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ інтегрованого телекомунікаційного рішення на основі технології IP-ATM яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

– Навігаційне меню: Файл; Дані; Фільтри; Сеанс зв’язку; Налагодження.

– Розділів: IP-ATM; Інструкції; Налаштування; Фільтрів.

– Розділу виведення результату роботи системи.

– Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.

– Функціональних кнопок ПЗ.

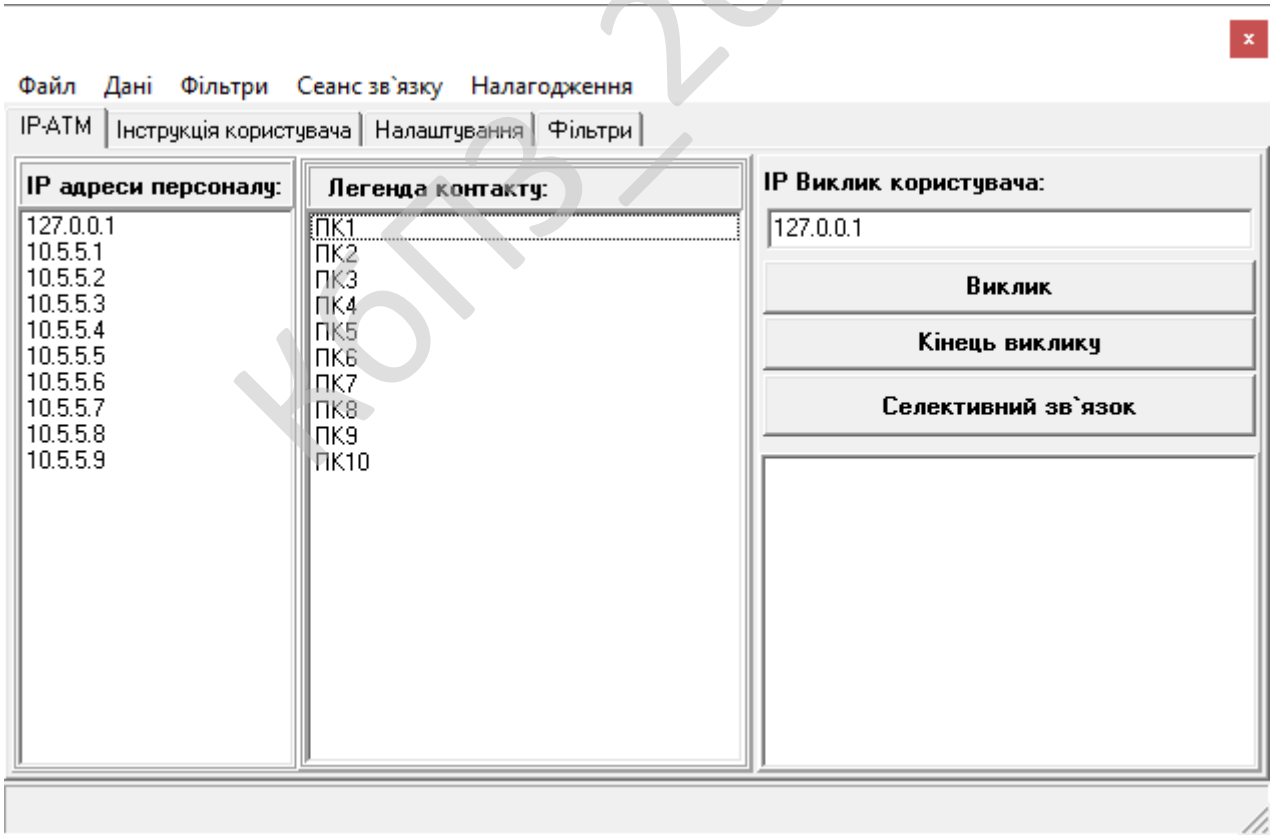

Рисунок 5.1 – Головне вікно ПЗ

КоПЗ – 2024

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

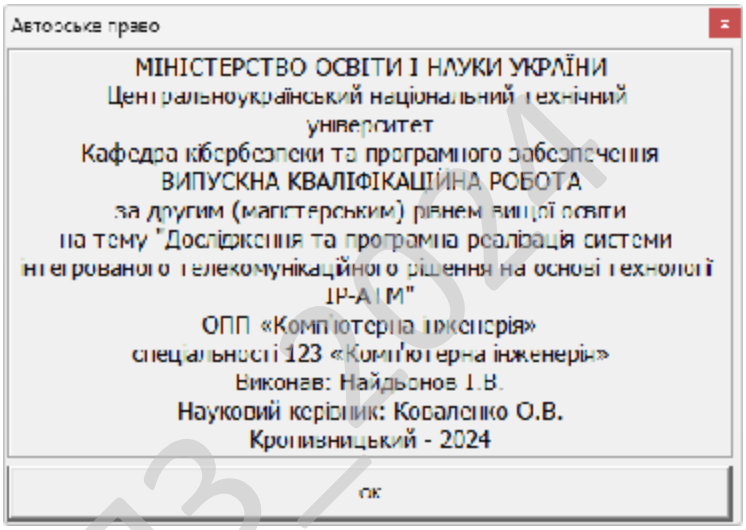

Рисунок 5.2 – Авторське право

КПІ_2024

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача.

Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

– Випуск.

– Встановлення та активація.

– Деактивація.

– Адаптація.

– Обновлення.

– Вмонтування.

– Відстежування версій.

– Видалення.

– Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати IT рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене IT рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

КБПЗ_2024

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

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

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

Проводилась оцінка:

– відповідності поставленим вимогам;

– правильна відповідь для усіх можливих вхідних даних;

– виконання функцій за прийнятний час;

– практичність;

– сумісність з ОС та стороннім ПЗ.

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

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

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

Ці тести демонструють:

– Як виконуються функції програми.

– Як приймаються вихідні дані.

– Як виробляються результати.

КБПЗ_2024

– Як зберігається цілісність зовнішньої інформації.

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

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

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

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

– Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ITc).

– Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його OT).

Будь-який спосіб тестування «чорної скриньки» повинен:

– Виявити такі вхідні дані, які з високою ймовірністю належать набору ITc;

– Сформулювати такі очікувані результати, які з високою імовірністю є елементами набору OT.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

– Некоректних чи відсутніх функцій;

– Помилок інтерфейсу;

– Помилок у зовнішніх структурах даних або в доступі до зовнішньої бази даних;

КБПЗ_2024

– Помилок характеристик (необхідна ємність пам'яті і т.д.);

– Помилок ініціалізації та завершення.

Обрано умови розповсюдження – Shareware.

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

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

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

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

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

КБПЗ_2024

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

Метою розробки є дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

Об'єктом дослідження є процес інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

Предметом дослідження є методи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

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

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

– Удосконалено метод інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

– Розроблено вітчизняний продукт інтегрованого телекомунікаційного рішення на основі технології IP-ATM, який має більш широкі можливості, на відміну від існуючих аналогів.

КБПЗ_2024

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM можуть зацікавити такі категорії осіб і організацій, що представлен на рисунку 7.1.

Провайдери телекомунікаційних послуг – великі оператори зв'язку, які прагнуть покращити якість обслуговування, скоротити затримки в передачі даних і оптимізувати мережеву інфраструктуру.

Інтернет-провайдери (ISP) – для впровадження гнучких і масштабованих мереж, що забезпечують підвищену пропускну здатність і підтримку багатопотокових даних (відео, голос, дані).

Корпорації та великі компанії – організації, які мають великі внутрішні мережі і зацікавлені в інтеграції швидких і безпечних рішень для внутрішнього зв'язку.

Центри обробки даних (Data Centers) – для впровадження високопродуктивних мережевих рішень, які зменшують час передачі між серверами та клієнтами.

Фінансові установи –банки та інші фінансові організації, яким необхідна висока швидкість передачі даних для банкоматів, платіжних терміналів та онлайн-операцій.

Розробники програмного забезпечення для мереж – компанії, що займаються створенням програмного забезпечення для мережевих рішень, можуть інтегрувати розробки у свої продукти.

Державні установи та оборонні організації – для впровадження безпечних, високонадійних телекомунікаційних рішень у критично важливих державних і військових мережах.

Дослідницькі інститути та університети – установи, що займаються вивченням новітніх технологій передачі даних, можуть використовувати результати як базу для подальших досліджень.

Постачальники обладнання для телекомунікацій – для впровадження нових функціональних можливостей у свої пристрої (маршрутизатори, комутатори, термінали).

ІТ-консалтингові компанії – для впровадження цих рішень у своїх клієнтів з метою модернізації інфраструктури.

Рисунок 7.1 – Цільова аудиторія

КПІ_2024

Потенційний інтерес також можуть виявити ті, хто прагне знизити витрати на зв'язок та оптимізувати інфраструктуру для мультимедійних даних, де потрібні висока швидкість і мінімальні затримки.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM за допомогою методу експертних оцінок може включати такі етапи:

1. Формулювання цілей оцінки – оцінити привабливість проекту для потенційних клієнтів та інвесторів.

2. Визначення критеріїв оцінки. Критерії мають бути чіткими, релевантними й охоплювати всі ключові аспекти проекту (рис.7.2).

Рисунок 7.2 – Критерії експертної оцінки

КСПЗ-2024

3. Формування групи експертів з представників ключових галузей: фахівців з телекомунікацій, IT-інженерів та архітекторів мереж, бізнес-аналітиків, представників потенційних користувачів (корпоративний сектор).

4. Проведення оцінювання, при якому кожен експерт оцінює проект за визначеними критеріями за шкалою (наприклад, від 1 до 10). Результати зводимо в таблицю 7.1.

Таблиця 7.1. – Зведені дані в процесі експертної оцінки

Критерій	Експерт 1	Експерт 2	Експерт 3	Середній бал
Технологічні переваги	9	8	10	9.0
Економічна ефективність	8	7	8	7.7
Попит на ринку	7	9	8	8.0
Конкурентоспроможність	6	8	7	7.0
Інтегративність	9	10	8	9.0
Рентабельність	8	7	9	8.0

5. Зведення результатів після зважування критеріїв. Результати зважуються відповідно до важливості критеріїв (коефіцієнти ваги).

Таблиця 7.2. – Зважені результати оцінки

Критерій	Середній бал	Коефіцієнт ваги	Підсумковий бал
Технологічні переваги	9.0	0.3	2.7
Економічна ефективність	7.7	0.2	1.54
Попит на ринку	8.0	0.2	1.6
Конкурентоспроможність	7.0	0.1	0.7
Інтегративність	9.0	0.15	1.35
Рентабельність	8.0	0.05	0.4

Загальний підсумковий бал: 2.7+1.54+1.6+0.7+1.35+0.4=8.292.7 + 1.54 + 1.6 + 0.7 + 1.35 + 0.4 = 8.292.7+1.54+1.6+0.7+1.35+0.4=8.29

КБПЗ-2024

Якщо загальний бал > 8, проект є привабливим і має високу ймовірність успішної реалізації на ринку. У даному прикладі загальний бал 8.29 свідчить про високу привабливість системи для впровадження.

7.3 Вибір методу оцінки вартості ПЗ

Вибір методу оцінки вартості програмної реалізації системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM залежить від цілей проекту, доступності даних, а також від специфіки продукту.

Для програмної реалізації системи на основі технології IP-ATM оптимальним варіантом є комбінування методів:

Cost-Based Pricing для визначення базової вартості. Цей метод враховує всі витрати, пов'язані з розробкою, тестуванням, впровадженням та підтримкою системи, з додаванням бажаного рівня прибутку. Застосовується у разі точного розуміння витрат на розробку та впровадження та для внутрішніх проектів, де потрібно обґрунтувати бюджет.

Value-Based Pricing для врахування вигод клієнтів і встановлення фінальної ціни. Вартість визначається з огляду на цінність, яку система приносить користувачам (економія часу, зниження витрат, підвищення продуктивності тощо). Застосовується якщо продукт має унікальні функції, що значно перевищують пропозиції конкурентів та для розрахунку вартості для корпоративних клієнтів або великих замовників.

Benchmarking для порівняння з конкурентами і коригування ціни відповідно до ринкових умов. Оцінка вартості базується на порівнянні з аналогічними рішеннями на ринку. Краще використовувати якщо існують подібні продукти або рішення на ринку та для швидкого визначення орієнтовної ціни. Проходить в три етапи: збір інформації про вартість конкурентних продуктів; порівняння функціоналу, переваг і недоліків вашої системи; коригування ціни відповідно до переваг вашого продукту.

КБПЗ_2024

Дотримання рекомендацій дозволить створити обґрунтовану вартість, яка буде конкурентною і водночас відображатиме цінність продукту.

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

Для оцінки економічної ефективності впровадження системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM можна застосувати кілька підходів, таких як аналіз витрат і вигод, підвищення продуктивності та зниження витрат на комунікації.

Вхідні та розрахункові дані зводимо в таблицю 7.3.

Таблиця 7.3 – Основні показники впровадження проєкту

1. Визначення початкових витрат (капітальних витрат)
Початкові витрати можуть включати такі елементи:
Вартість обладнання: маршрутизатори, комутатори, сервери, термінали.
Ліцензії на програмне забезпечення.
Інсталяція та налаштування системи: витрати на розробку, тестування, впровадження.
Навчання персоналу: тренінги для співробітників компанії.
Вартість обладнання: 500,000 грн.
Ліцензії та програмне забезпечення: 150,000 грн.
Витрати на впровадження та інсталяцію: 100,000 грн.
Навчання персоналу: 50,000 грн.
Загальні початкові витрати:
500,000+150,000+100,000+50,000=800,000 грн.

КБПЗ_2024

Продовження таблиці 7.3

2. Оцінка щорічних операційних витрат Щорічні витрати на підтримку і експлуатацію системи можуть включати: **Підтримка обладнання**: обслуговування мережевих пристроїв і серверів. **Витрати на електроенергію**: для роботи обладнання. **Оплата ліцензій**: річні витрати на підтримку ліцензійного програмного забезпечення. Підтримка обладнання: 100,000 грн на рік. Електроенергія: 50,000 грн на рік. Оплата ліцензій: 30,000 грн на рік. **Щорічні операційні витрати**: 100,000+50,000+30,000=180,000 грн на рік.
3. Оцінка економії від впровадження системи (вигоди) Вигоди можуть проявлятися в зниженні витрат на телекомунікації, підвищенні швидкості передачі даних, зменшенні часу простою та покращенні ефективності обробки даних. Ось кілька прикладів економії: **Зниження витрат на телекомунікації**: зменшення витрат на традиційні телефонні лінії, зниження вартості обслуговування старих мереж. **Покращення ефективності**: зменшення часу на передачу даних, швидший доступ до ресурсів для співробітників. **Зменшення часу простою**: знижена ймовірність збоїв в системі. Зниження витрат на зв'язок: 200,000 грн на рік. Покращення ефективності обробки даних: 150,000 грн на рік. Зменшення часу простою: 50,000 грн на рік. **Загальна економія**: 200,000+150,000+50,000=400,000 грн на рік.

КБПЗ_2024

Продовження таблиці 7.3

4. Розрахунок терміну окупності (Payback Period) Термін окупності – це період, за який система покриває свої початкові витрати завдяки отриманій економії. Для цього потрібно поділити початкові витрати на річну економію: **Термін окупності**: 2 роки.
5. Оцінка рентабельності інвестицій (ROI) **ROI** – це показник, що визначає вигоду або прибуток від інвестицій у систему. ROI=Початкові витратиЕкономія−Початкові витрати×100% ROI=(400,000−800,000)/800,000×100%=50% **ROI**: 50% на рік, що є гарним показником для інвестицій у сучасну телекомунікаційну технологію.

Економічна ефективність від впровадження системи IP-ATM для інтегрованих телекомунікацій може бути досить високою, враховуючи значні вигоди від зниження витрат на зв'язок, підвищення ефективності обробки даних та скорочення часу простою системи. Термін окупності: 2 роки. Після цього система почне приносити чистий прибуток. Рентабельність: 50% щорічно.

КБПЗ_2024

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Алгоритм просування проекту програмної реалізації системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM включає в себе кілька етапів, спрямованих на забезпечення успішної реалізації, залучення потенційних клієнтів і створення стійкої репутації на ринку (рисунок 7.3). Цей алгоритм дозволить ефективно просувати проект і забезпечити успішну реалізацію системи інтегрованого телекомунікаційного рішення на основі IP-ATM на ринку.

1. Визначення цільової аудиторії та сегментування ринку

- **Оцінка ринку**: Аналіз потенційних клієнтів (корпорації, телекомунікаційні компанії, державні організації, великі підприємства), що можуть бути зацікавлені в інтегрованих телекомунікаційних рішеннях.
- **Сегментація ринку**: Визначення ключових сегментів: великі компанії, які потребують високошвидкісного зв'язку, інтернет-провайдери, фінансові установи, урядові органи тощо.
- **Створення портрета клієнта (Customer Persona)**: Визначення потреб і болей клієнтів для надання саме того рішення, яке відповідає їх вимогам.

2. Оцінка конкурентів і унікальних переваг продукту

- **Аналіз конкурентів**: Дослідження ринку і визначення основних конкурентів, їхніх продуктів, цінових політик і маркетингових стратегій.
- **Визначення УТП (унікальної торгової пропозиції)**: Описати, чим ваша система IP-ATM відрізняється від інших рішень на ринку (наприклад, висока надійність, масштабованість, простота інтеграції, зниження витрат на інфраструктуру).

3. Розробка маркетингової стратегії

- **Створення бренду та позиціонування**: Розробка бренду та стратегії позиціонування на ринку. Визначення, як ви хочете бути сприйняті клієнтами (як інноваційне, надійне або високопродуктивне рішення).
- **Опис продукту**: Створення технічних і маркетингових матеріалів, які детально пояснюють переваги продукту: опис характеристик IP-ATM, переваги інтеграції, продуктивність, економія витрат.
- **Розробка цінової стратегії**: Визначення конкурентоспроможної ціни, що відповідає вартості продукту і очікуванням клієнтів.

4. Прямий контакт з потенційними клієнтами

- **Тематичні заходи та виставки**: Участь у галузевих конференціях, виставках і форумах, де можна продемонструвати продукт безпосередньо потенційним клієнтам.
- **Пошук партнерів та інвесторів**: Пошук можливих стратегічних партнерів і інвесторів, які можуть бути зацікавлені в інтеграції або використанні IP-ATM рішень.
- **Ключові презентації**: Проведення вебінарів, презентацій продукту для ключових гравців ринку (включаючи технічні та управлінські зустрічі).

5. Створення інформаційних матеріалів і контент-маркетинг

- **Розробка контенту**: Створення статей, блогів, технічних звітів, white papers, які демонструють технічні переваги системи та практичні кейси її застосування.
- **Відео-презентації та демонстрації**: Запис відео, які показують, як працює система в реальних умовах, її інтеграція з іншими рішеннями, переваги використання.
- **SEO та онлайн-реклама**: Оптимізація контенту для пошукових систем, реклама в інтернеті (Google Ads, LinkedIn, таргетована реклама для IT-менеджерів, телекому та ін.).

6. Використання соціальних мереж і професійних платформ

- **Робота в LinkedIn**: Створення сторінки компанії на LinkedIn, публікація кейсів, статей, результатів успішних інтеграцій для підвищення довіри до компанії та її продуктів.
- **Facebook та Twitter**: Публікації анонсів нових продуктів і функцій, новини, оновлення, що підвищують впізнаваність бренду серед ширшої аудиторії.
- **Публікація кейсів на професійних форумах і порталах**: Публікація успішних кейсів, а також статей про технології IP-ATM на спеціалізованих форумах і платформах.

7. Формування команди продажу та каналів дистрибуції

- **Навчання команди продажу**: Підготовка команди з продажу до активних переговорів з клієнтами, навчання з обробки запитів і демонстрації продукту.
- **Пошук дистриб'юторів і партнерів**: Укладання партнерських угод із телекомунікаційними компаніями, які можуть включити рішення IP-ATM до своїх продуктів або пропонувати його своїм клієнтам.

8. Підтримка і розвиток клієнтів

- **Післяпродажна підтримка**: Організація підтримки клієнтів, надання сервісу та допомоги в інтеграції та налаштуванні продукту.
- **Розширення функціоналу**: Постійне вдосконалення продукту, додавання нових функцій і можливостей відповідно до запитів ринку.

Рисунок 7.3 – Алгоритм просування проєкту

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту та шляхів реалізації проєкту програмної реалізації системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM є критично важливими для забезпечення успішного впровадження і максимізації прибутку (рисунок 7.4).

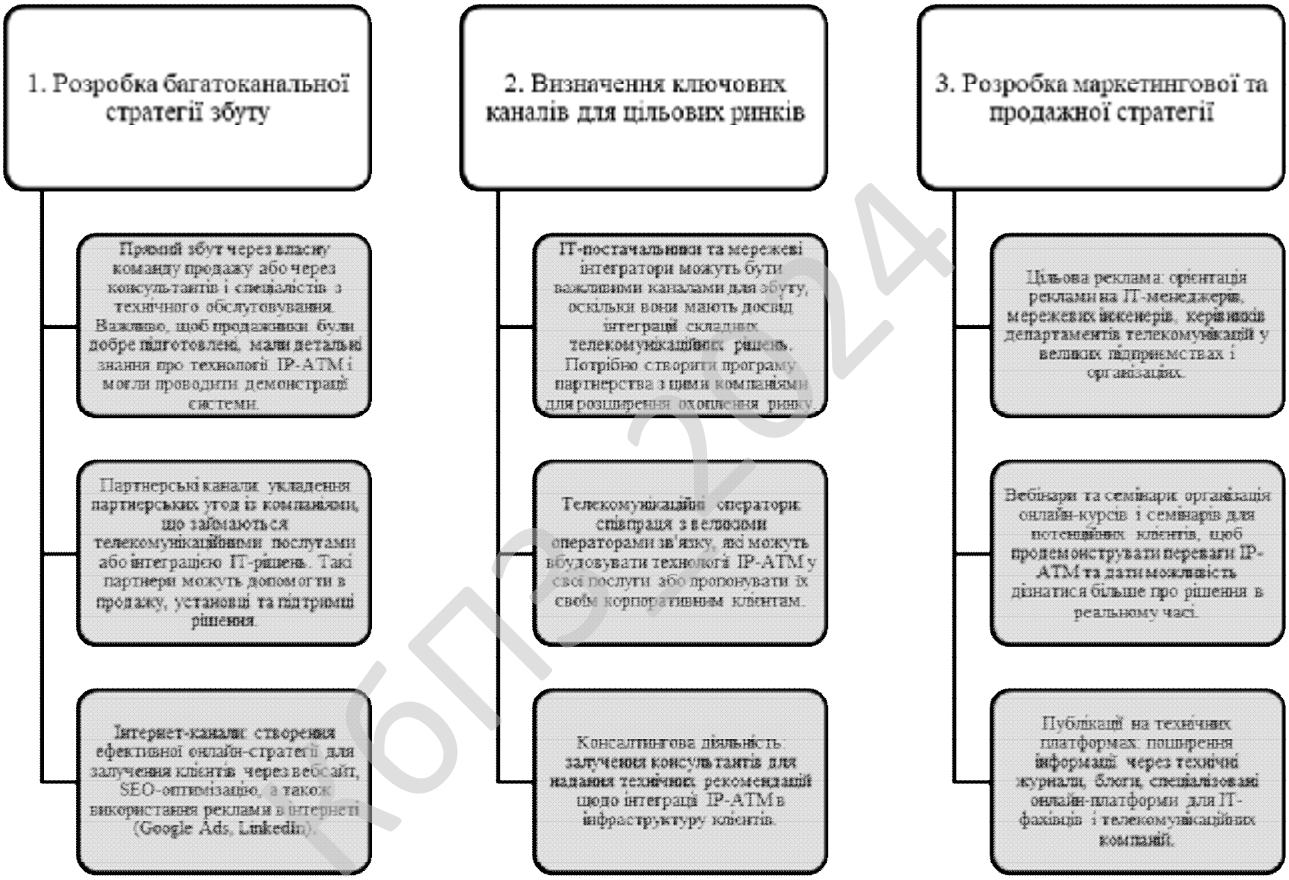

Рисунок 7.4 – Алгоритм оптимізації збуту

Ці стратегії дозволяють не тільки оптимізувати канали збуту, але й підвищити ефективність впровадження проекту, знижуючи витрати та максимізуючи прибуток від реалізації системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

КБПО 2024

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM включають різноманітні технічні, організаційні та комерційні аспекти, які мають вплив на ефективність реалізації та впровадження проекту (рисунок 7.5).

Рисунок 7.5 – Ключові фактори успіху проєкту

КБПЗ – 2024

Успіх проєкту залежить від комплексного підходу до всіх цих факторів, і їх належне управління та інтеграція дозволяють забезпечити ефективне впровадження та максимальний результат від реалізації системи IP-ATM.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

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

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

КБПЗ_2024

Умови праці програміста включають наступні фактори:

– параметри повітряного середовища в приміщенні;

– вентиляція приміщення;

– освітлення приміщення;

– параметри повітряного середовища в приміщенні, тощо.

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

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машин (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Оскільки робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у тому числі програмісти) необхідно створити належний мікроклімат, праметри якого регламентуються Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

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

– ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.

– ризик виникнення пожежі;

– негативний вплив на органи зору людини;

– ризики ураження електричним струмом;

– недостатня, або надмірна освітленість робочого місця;

– електромагнітні (у тому числі високочастотні) випромінювання (коливання);

– несприятливі мікрокліматичні умови;

– нервово-емоційна напруженість праці;

КБПЗ_2024

– інтелектуальні навантаження;

– монотонність праці;

– невідповідність ергономічних показників робочого місця діючим вимогам;

– шум;

– статичні навантаження на кістково-м’язовий апарат.

праці.

Лікарі вважають, що з'явився новий тип захворювання – синдром стресу оператора дисплея, дія якого проявляється у вигляді головного болю, підвищеної стомлюваності та слабкості, порушення сну, болю в області спини, шиї та зап'ястків, напруженого стану м'язів, відчуття скутості, запалення сухожиль, запальних процесів м'язів, що керують рухами передпліч і пальців, неприємних відчуттів В області очей, що супроводжуються різзю, алергії. Можливі астматичні прояви, пригніченість, млявість, депресія. про аналітичного паралічу, тобто. нездатності до аналізу одержуваної інформації. Напружений характер праці, висока вартість помилки оператора, перебування у стані тривожно-томного очікування за відсутності емоційної розрядки ведуть до невротизації особистості, що може несприятливо позначатися на спадковості як невротизації потомства.

КбПЗ_2024

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1. У зазначеному приміщенні працюють двоє людей. За даними, які наведено у таблиць 8.1-8.2, можна зробити висновок, що площа та об’єм приміщення у розрахунку на одне робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [5], але відповідають

нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	3
Довжина	4,6
Висота	3

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	$м^2$	не менше 6.0	6,9
Об'єм, V	$м^3$	не менше 20.0	20,7

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

КБПЗ_2024

Таким чином, можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість

атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Темпера-тура, °С	Воло-гість,%	Швидкість повітря, м/с	Темпера-тура,°С	Воло-гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-24	40-55	0,12
Тепла	23-25	50-70	0,1	24-25	50-65	0,9

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

КБПЗ_2024

У приміщенні знаходяться наступні джерела шуму: принтер Xerox WorkCentre 3025BI (3025VBI), електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [1], у яких прописані вимоги до використання всіх освітлювальних приладів, у тому числі світлодіодних.

Працю працівника, який постійно працює за комп’ютером, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [1], Крім того все поле зору повинне бути освітлено достатньо рівномірно – це основна гігієнічна вимога. Оскільки яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

КБПЗ_2024

8.4 Розробка заходів з умов поліпшення охорони праці

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

– розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних допустимих значень.

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

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга). Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В. Оскільки при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

КБПЗ_2024

8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 3 м, довжина – 4,6 м, висота – 3 м.

У зазначеному приміщенні працює 1 особа.

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

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

де F – світловий потік, що розраховується, Лм;

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

S – площа освітлюваного приміщення (у нашому випадку S = 3x4,6 = 13,8 $м^2$);

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

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

n – коефіцієнт використання світлового потоку (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп, обчислюється в долях одиниці; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{стін.}$) і стелі ($\rho_{стелі}$), значення коефіцієнтів дорівнюють $\rho_{стін}$ = 50% і $\rho_{стелі}$ = 50%.

Обчислимо індекс приміщення за формулою:

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

де S – площа приміщення, S = 13,8 $м^2$;

h – розрахункова висота підвісу, h = 3 м (співпадає з висотою стелі, оскільки лампи освітлення закріплюються на стелі);

A – ширина приміщення, A = 3 м;

B – довжина приміщення, B = 4,6 м.

КБПЗ-2024

Підставимо всі значення у формулу та визначимо індекс приміщення: i=0,43.

Знаючи індекс приміщення, знаходимо n = 0,23 (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом ламп). Підставимо всі значення у формулу, визначимо світловий потік: F=29700 Лм.

Для розрахунку будемо використовувати світлодіодні панелі LED панель 42 Вт 6000 К SUNLED 000000127, світловий потік яких $F_л$ = 3990 Лм.

Число ламп визначається за формулою:

$$N=F/F_л$$

де F – світловий потік,

$F_л$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначений індекс приміщення:

$$N = 29700/ 3990 =7,4 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 8 шт.

Висновки до розділу

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

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

КБПЗ_2024

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

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 інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

– Досліджена система інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

– На основі отриманих результатів досліджень створена програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

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.

КБПЗ_2024

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

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 7624:2014.

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

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

КБПЗ_2024

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Найдьонов І.В. Дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. ITU-T Recommendation G.723.1. Dual Rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kit / sec. – 1996.
3. ITU-T Recommendation G.729. Speech codec for multimedia telecommunications transmitting at 8 / 13 kbit / s. – 1996.
4. ITU-T Recommendation H.225.0. Call signaling protocols and media stream packetization for packet-based multimedia communication systems. -Geneva, 1998.
5. ITU-T Recommendation H.245. Control protocol for multimedia communication. -Geneva, 1998
6. ITU-T Recommendation H.248. Gateway control protocol. – Geneva, 2000.
7. ITU-T Recommendation H.320. Narrow-band Visual Telephone Systems and Terminal Equipment. – 1996.
8. ITU-T Recommendation H.321. Adaptation of H.320 Visual Telephone Terminals to B-ISDN Environments. – 1996.
9. ITU-T Recommendation H.322. Visual Telephone Systems and Terminal Equipment for Local Area Networks which Provide a Guaranteed Quality of Service. – 1996.
10. ITU-T Recommendation H.323. Packet based multimedia communication systems. – Geneva, 1998.
11. ITU-T Recommendation H.324. Terminal for Low Bit Rate Multimedia Communications. -1996.
12. ITU-T Recommendation Q.931. ISDN User-Network Interface Layer 3

КБПЗ-2024

Specification for Basic Call Control. – 1993.

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

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

15. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

16. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447

17. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

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

19. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». Lecture Notes on Data Engineering and Communications Technologies, 2023, 178, pp. 208–223.

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

21. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh

КБПЗ_2024

«Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.

22. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w

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

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

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

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

27. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.

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

КБПЗ_2024

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

30. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

31. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

32. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

33. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

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

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

КБПЗ – 2024

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

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

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

39. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

40. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

41. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

42. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

КБПЗ-2024

43. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

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

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

46. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології,* 2024, № 13, с. 28-35.

47. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи,* 2023, том 7, № 2, С. 49-56.

48. . О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

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

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

КБПЗ-2024

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

КБПЗ_2024

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

2 Підстава для розробки

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

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM.

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;

КБПЗ_2024

– 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 інтегрованого телекомунікаційного рішення на основі технології IP-ATM;

– 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.

КбПЗ_2024

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

– температура повітря: 19-20 град. по Цельсію;

– відносна вологість повітря до 80%;

– атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Python.

КБПЗ_2024

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз санітарно-гігієнічних умов праці на робочому місці програміста.

КБПЗ_2024

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 – 107 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 20.12.2024 р.

КБПЗ_2024

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.

КБПЗ_2024

Дослідження та програмна реалізація системи інтегрованого телекомунікаційного рішення на основі технології IP-ATM

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 15

Літера: РП

Кропивницький – 2024 року

Основна програма

```
import socket
import threading
import time
from cryptography.fernet import Fernet

# Генерація ключа шифрування
encryption_key = Fernet.generate_key()
cipher = Fernet(encryption_key)

class IPATMNetwork:
    def __init__(self):
        self.devices = {}
        self.switching_table = {}
        self.logs = []
        self.failed_routes = {}

    def register_device(self, ip_address, device, auth_token):
        # Реєстрація нового пристрою у мережі після перевірки токена
        if auth_token != "secure_token":
            self.logs.append(f"Authorization failed for device {ip_address}")
            print(f"Authorization failed for device {ip_address}")
            return False
        self.devices[ip_address] = device
        self.logs.append(f"Device {ip_address} registered successfully.")
        return True

    def register_route(self, source_ip, destination_ip, route):
        # Реєстрація маршруту у таблиці комутації
        self.switching_table[(source_ip, destination_ip)] = route
        self.logs.append(f"Route from {source_ip} to {destination_ip}
registered.")

    def send_data(self, source_ip, destination_ip, data):
        # Відправка зашифрованих даних через зареєстрований маршрут
        if (source_ip, destination_ip) in self.switching_table:
            encrypted_data = cipher.encrypt(data.encode())
            route = self.switching_table[(source_ip, destination_ip)]
            route.send(encrypted_data)
            self.logs.append(f"Encrypted data sent from {source_ip} to
{destination_ip}.")
        else:
            self.logs.append(f"Error: No route found from {source_ip} to
{destination_ip}")
            print(f"No route found from {source_ip} to {destination_ip}")
```

КБПЗ_2024

```
    def reserve_route(self, source_ip, destination_ip, backup_route):
        # Реєстрація резервного маршруту
        self.failed_routes[(source_ip, destination_ip)] = backup_route
        self.logs.append(f"Backup route from {source_ip} to {destination_ip} registered.")

    def send_with_failover(self, source_ip, destination_ip, data):
        # Відправка даних з резервним маршрутом
        try:
            self.send_data(source_ip, destination_ip, data)
        except Exception:
            if (source_ip, destination_ip) in self.failed_routes:
                backup_route = self.failed_routes[(source_ip, destination_ip)]
                encrypted_data = cipher.encrypt(data.encode())
                backup_route.send(encrypted_data)
                self.logs.append(f"Data sent using backup route from {source_ip} to {destination_ip}.")
            else:
                self.logs.append(f"Critical error: No primary or backup route available for {source_ip} to {destination_ip}.")

    def monitor_traffic(self):
        # Моніторинг трафіку в реальному часі
        traffic_data = {}
        for key, route in self.switching_table.items():
            traffic_data[key] = len(route.data_queue)
        self.logs.append(f"Traffic monitoring: {traffic_data}")
        return traffic_data

    def view_logs(self):
        # Перегляд логів мережі
        for log in self.logs:
            print(log)

class ATMRoute:
    def __init__(self, bandwidth):
        # Ініціалізація маршруту з вказаною пропускною здатністю
        self.bandwidth = bandwidth
        self.data_queue = []

    def send(self, data):
        # Додавання зашифрованих даних до черги для відправки
        self.data_queue.append(data)
        threading.Thread(target=self.process_data).start()
```

КБПЗ_2024

```
    def process_data(self):
        # Обробка даних у черзі з урахуванням пропускної здатності
        if self.data_queue:
            time.sleep(len(self.data_queue[0]) / self.bandwidth)
            print(f"Processed encrypted data: {self.data_queue.pop(0)}")

class Device:
    def __init__(self, ip_address, network, auth_token):
        # Ініціалізація пристрою
        self.ip_address = ip_address
        self.network = network
        if not network.register_device(ip_address, self, auth_token):
            raise    PermissionError(f"Authorization    failed    for    device
{ip_address}.")

    def send_data(self, destination_ip, data):
        # Відправка даних до іншого пристрою
        self.network.send_data(self.ip_address, destination_ip, data)

    def receive_data(self, encrypted_data):
        # Розшифровка та прийом даних
        decrypted_data = cipher.decrypt(encrypted_data).decode()
        print(f"Device {self.ip_address} received data: {decrypted_data}")

class Server(Device):
    def __init__(self, ip_address, network, auth_token):
        # Ініціалізація серверу
        super().__init__(ip_address, network, auth_token)
        self.data_storage = []

    def receive_data(self, encrypted_data):
        # Розшифровка та збереження даних сервером
        decrypted_data = cipher.decrypt(encrypted_data).decode()
        self.data_storage.append(decrypted_data)
        print(f"Server {self.ip_address} stored data: {decrypted_data}")

    def provide_service(self, client_ip, service_message):
        # Надавання сервісу клієнту
        self.send_data(client_ip, f"Service response: {service_message}")

class Client(Device):
    def __init__(self, ip_address, network, auth_token):
        # Ініціалізація клієнта
        super().__init__(ip_address, network, auth_token)

    def request_data(self, destination_ip, request_message):
```

КБПЗ_2024

```
        # Відправка запиту до іншого пристрою
        self.send_data(destination_ip, request_message)

# Створення IP-ATM мережі
network = IPATMNetwork()

# Створення серверу, клієнтів та маршрутів
server = Server("192.168.0.1", network, "secure_token")
client1 = Client("192.168.0.2", network, "secure_token")
client2 = Client("192.168.0.3", network, "secure_token")
atm_route_primary = ATMRoute(bandwidth=10)
atm_route_backup = ATMRoute(bandwidth=5)

# Реєстрація маршрутів у мережі
network.register_route("192.168.0.2", "192.168.0.1", atm_route_primary)
network.register_route("192.168.0.3", "192.168.0.1", atm_route_primary)

# Реєстрація резервного маршруту
network.reserve_route("192.168.0.2", "192.168.0.1", atm_route_backup)

# Клієнти відправляють зашифровані запити до серверу
client1.request_data("192.168.0.1", "Hello Server from Client 1!")
client2.request_data("192.168.0.1", "Hello Server from Client 2!")

# Сервер надає послугу клієнту
server.provide_service("192.168.0.2", "Here is your encrypted requested data!")

# Моніторинг трафіку
traffic_status = network.monitor_traffic()
print("Traffic Status:", traffic_status)

# Перегляд логів мережі
network.view_logs()
```

КБПЗ_2024

Файл ip_atm_advanced.py

```
# ip_atm_advanced.py
import time
from cryptography.fernet import Fernet

# Генерація ключа шифрування для можливих додаткових операцій
encryption_key = Fernet.generate_key()
cipher = Fernet(encryption_key)

class QoSManager:
    def __init__(self):
        # Ініціалізація менеджера QoS
        self.traffic_classes = {}
        self.logs = []

    def define_traffic_class(self, class_name, priority):
        # Визначення класу трафіку та його пріоритету
        self.traffic_classes[class_name] = priority
        self.logs.append(f"Defined traffic class '{class_name}' with priority
{priority}.")

    def prioritize_traffic(self, data, class_name):
        # Пріоритизація трафіку на основі класу
        if class_name in self.traffic_classes:
            priority = self.traffic_classes[class_name]
            self.logs.append(f"Data prioritized with class '{class_name}'
(priority {priority}).")
            return (priority, data)
        else:
            self.logs.append(f"Error: Undefined traffic class '{class_name}'.")
            raise ValueError(f"Traffic class '{class_name}' is not defined.")

    def view_qos_logs(self):
        # Перегляд логів QoS
        return self.logs

class NetworkVisualization:
    def __init__(self, network):
        # Ініціалізація візуалізації для заданої мережі
        self.network = network

    def visualize_topology(self):
        # Виведення топології мережі
        print("Network Topology:")
        for source, device in self.network.devices.items():
            print(f"Device {source} connected.")

class RouteAlerts:
    def __init__(self, network):
        # Ініціалізація системи оповіщень
        self.network = network
        self.alerts = []
```

КБПЗ_2024

```
    def check_routes(self):
        # Перевірка доступності маршрутів та створення оповіщень
        for route in self.network.switching_table.keys():
            # Симуляція перевірки доступності маршруту
            if time.time() % 2 == 0:  # Для тестування: випадкове виявлення
проблем
                self.alerts.append(f"Route issue detected between {route[0]} and
{route[1]}.")

    def view_alerts(self):
        # Перегляд оповіщень
        return self.alerts

class DynamicRouting:
    def __init__(self, network):
        # Ініціалізація динамічної маршрутизації
        self.network = network

    def update_routing_table(self, new_routes):
        # Оновлення таблиці маршрутів
        for source, destination, route in new_routes:
            self.network.register_route(source, destination, route)

    def simulate_dynamic_update(self):
        # Симуляція оновлення таблиці маршрутів
        print("Dynamic routing updates simulated.")

class MulticastSupport:
    def __init__(self, network):
        # Ініціалізація підтримки multicast-з'єднань
        self.network = network
        self.groups = {}

    def create_group(self, group_id, devices):
        # Створення multicast-групи
        self.groups[group_id] = devices
        print(f"Multicast group '{group_id}' created with devices {devices}.")

    def send_to_group(self, group_id, data):
        # Відправка даних до multicast-групи
        if group_id in self.groups:
            for device in self.groups[group_id]:
                if device in self.network.devices:
                    encrypted_data = cipher.encrypt(data.encode())
                    self.network.devices[device].receive_data(encrypted_data)
        else:
            print(f"Error: Group '{group_id}' not found.")
```

КБПЗ_2024

Файл ip_atm_management.py

```
# ip_atm_management.py
import json
import time

class QoSAdjuster:
    def __init__(self, network):
        # Ініціалізація налаштування QoS
        self.network = network
        self.adjustments = []

    def adjust_bandwidth(self, source_ip, destination_ip, new_bandwidth):
        # Динамічна зміна пропускної здатності маршруту
        if (source_ip, destination_ip) in self.network.switching_table:
            route = self.network.switching_table[(source_ip, destination_ip)]
            old_bandwidth = route.bandwidth
            route.bandwidth = new_bandwidth
            self.adjustments.append(f"Adjusted bandwidth from {old_bandwidth} to
{new_bandwidth} for route {source_ip} -> {destination_ip}.")
        else:
            print(f"Error: Route {source_ip} -> {destination_ip} not found.")

    def view_adjustments(self):
        # Перегляд змін пропускної здатності
        return self.adjustments

class MulticastEnhancer:
    def __init__(self, network):
        # Ініціалізація підтримки multicast-з'єднань
        self.network = network
        self.groups = {}

    def create_multicast_group(self, group_id, device_ips):
        # Створення multicast-групи
        self.groups[group_id] = device_ips
        print(f"Multicast group '{group_id}' created with devices
{device_ips}.")

    def send_multicast(self, group_id, data):
        # Відправка даних до multicast-групи
        if group_id in self.groups:
            for ip in self.groups[group_id]:
                if ip in self.network.devices:
                    encrypted_data = cipher.encrypt(data.encode())
                    self.network.devices[ip].receive_data(encrypted_data)
        else:
            print(f"Error: Group {group_id} not found.")

class BillingSystem:
    def __init__(self):
        # Ініціалізація білінгової системи
        self.usage_records = {}
```

КбПЗ_2024

```
    def log_usage(self, source_ip, destination_ip, data_size):
        # Логування використання мережі
        if (source_ip, destination_ip) not in self.usage_records:
            self.usage_records[(source_ip, destination_ip)] = 0
        self.usage_records[(source_ip, destination_ip)] += data_size
        print(f"Usage logged: {data_size} bytes from {source_ip} to
{destination_ip}.")

    def generate_report(self):
        # Генерація звіту про використання
        report = json.dumps(self.usage_records, indent=4)
        print("Usage Report:")
        print(report)

class DeviceHealthChecker:
    def __init__(self, network):
        # Ініціалізація перевірки стану пристроїв
        self.network = network
        self.health_status = {}

    def check_health(self):
        # Симуляція перевірки доступності пристроїв
        for ip, device in self.network.devices.items():
            # Для тесту: випадковий статус доступності
            status = "Online" if time.time() % 2 == 0 else "Offline"
            self.health_status[ip] = status
        print("Device health check completed.")

    def view_health_status(self):
        # Перегляд стану пристроїв
        for ip, status in self.health_status.items():
            print(f"Device {ip}: {status}")

class NetworkLoadSimulator:
    def __init__(self, network):
        # Ініціалізація симуляції навантаження
        self.network = network

    def simulate_load(self, source_ip, destination_ip, load_size):
        # Симуляція створення навантаження на маршрут
        if (source_ip, destination_ip) in self.network.switching_table:
            route = self.network.switching_table[(source_ip, destination_ip)]
            for _ in range(load_size):
                route.send(b"Simulated Load")
            print(f"Simulated load of size {load_size} created from {source_ip}
to {destination_ip}.")
        else:
            print(f"Error: Route {source_ip} -> {destination_ip} not found.")
```

КОПЗ_2024

Файл ip_atm_advanced_tools.py

```
# ip_atm_advanced_tools.py
import time
import random

class ErrorLogger:
    def __init__(self, log_file="error_log.txt"):
        # Ініціалізація системи логування помилок
        self.log_file = log_file

    def log_error(self, error_message):
        # Запис помилки до файлу
        timestamp = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
        with open(self.log_file, "a") as log:
            log.write(f"[{timestamp}] {error_message}\n")
        print(f"Error logged: {error_message}")

    def view_logs(self):
        # Перегляд логів помилок
        try:
            with open(self.log_file, "r") as log:
                print(log.read())
        except FileNotFoundError:
            print("Error log file not found.")

class AnalyticsIntegration:
    def __init__(self):
        # Ініціалізація системи інтеграції з аналітичними сервісами
        self.analytics_data = {}

    def record_event(self, event_type, details):
        # Запис події до аналітики
        timestamp = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
        if event_type not in self.analytics_data:
            self.analytics_data[event_type] = []
        self.analytics_data[event_type].append({"timestamp": timestamp,
"details": details})
        print(f"Event recorded: {event_type} - {details}")

    def generate_analytics_report(self):
        # Генерація аналітичного звіту
        print("Analytics Report:")
        for event_type, events in self.analytics_data.items():
            print(f"Event Type: {event_type}")
            for event in events:
                print(f"  - {event['timestamp']}: {event['details']}")

class DynamicBandwidthAllocator:
    def __init__(self, network):
        # Ініціалізація динамічного виділення пропускної здатності
        self.network = network
```

КБПЗ_2024

```
    def allocate_bandwidth(self, source_ip, destination_ip, minimum_bandwidth):
        # Динамічне виділення пропускної здатності на маршруті
        if (source_ip, destination_ip) in self.network.switching_table:
            route = self.network.switching_table[(source_ip, destination_ip)]
            route.bandwidth = max(route.bandwidth, minimum_bandwidth)
            print(f"Bandwidth for route {source_ip} -> {destination_ip} set to
{route.bandwidth}.")
        else:
            print(f"Error: Route {source_ip} -> {destination_ip} not found.")

class MulticastService:
    def __init__(self, network):
        # Ініціалізація multicast-сервісу
        self.network = network
        self.groups = {}

    def create_group(self, group_id, members):
        # Створення multicast-групи
        self.groups[group_id] = members
        print(f"Multicast group '{group_id}' created with members: {members}")

    def send_multicast(self, group_id, message):
        # Відправка повідомлення до multicast-групи
        if group_id in self.groups:
            members = self.groups[group_id]
            for member in members:
                if member in self.network.devices:
                    encrypted_message = cipher.encrypt(message.encode())
                    self.network.devices[member].receive_data(encrypted_message)
            print(f"Message sent to multicast group '{group_id}'.")
        else:
            print(f"Error: Group '{group_id}' not found.")
```

КБПЗ-2024

Файл ip_atm_simulation_tools.py

```
# ip_atm_simulation_tools.py
import time
import json
import random

class BillingSystem:
    def __init__(self):
        # Ініціалізація білінгової системи
        self.usage_records = {}

    def log_usage(self, source_ip, destination_ip, data_size):
        # Логування використання мережі
        if (source_ip, destination_ip) not in self.usage_records:
            self.usage_records[(source_ip, destination_ip)] = 0
        self.usage_records[(source_ip, destination_ip)] += data_size
        print(f"Usage logged: {data_size} bytes from {source_ip} to
{destination_ip}.")

    def generate_billing_report(self):
        # Генерація звіту про використання мережі
        report = json.dumps(self.usage_records, indent=4)
        print("Billing Report:")
        print(report)
        return report

class DeviceAvailabilityChecker:
    def __init__(self, network):
        # Ініціалізація перевірки доступності пристроїв
        self.network = network
        self.status_report = {}

    def check_availability(self):
        # Перевірка доступності пристроїв
        for ip, device in self.network.devices.items():
            # Симуляція статусу доступності (випадковий результат для
тестування)
            status = "Online" if random.choice([True, False]) else "Offline"
            self.status_report[ip] = status
        print("Device availability check complete.")

    def generate_availability_report(self):
        # Генерація звіту про доступність пристроїв
        print("Device Availability Report:")
        for ip, status in self.status_report.items():
            print(f"Device {ip}: {status}")
        return self.status_report

class NetworkLoadSimulator:
    def __init__(self, network):
        # Ініціалізація симуляції навантаження
        self.network = network
```

КБПЗ_2024

```
    def simulate_load(self, source_ip, destination_ip, load_size):
        # Симуляція створення навантаження на маршрут
        if (source_ip, destination_ip) in self.network.switching_table:
            route = self.network.switching_table[(source_ip, destination_ip)]
            for _ in range(load_size):
                route.send(b"Simulated Load")
            print(f"Simulated load of size {load_size} created from {source_ip}
to {destination_ip}.")
        else:
            print(f"Error: Route {source_ip} -> {destination_ip} not found.")

    def simulate_peak_load(self, source_ip, destination_ip, duration,
peak_load_size):
        # Симуляція пікового навантаження на певний час
        if (source_ip, destination_ip) in self.network.switching_table:
            route = self.network.switching_table[(source_ip, destination_ip)]
            end_time = time.time() + duration
            while time.time() < end_time:
                for _ in range(peak_load_size):
                    route.send(b"Peak Load")
                time.sleep(1)
            print(f"Peak load simulation complete for route {source_ip} ->
{destination_ip}.")
        else:
            print(f"Error: Route {source_ip} -> {destination_ip} not found.")
```

КБПЗ_2024

Файл ip_atm_extended_features.py

```
# ip_atm_extended_features.py
import time
import json
import threading

class TrafficAnalyzer:
    def __init__(self, network):
        # Ініціалізація аналізатора трафіку
        self.network = network
        self.traffic_data = {}

    def analyze_traffic(self):
        # Аналіз трафіку у мережі
        for (source, destination), route in
self.network.switching_table.items():
            traffic_volume = len(route.data_queue)
            self.traffic_data[(source, destination)] = traffic_volume
        print("Traffic analysis completed.")

    def generate_traffic_report(self):
        # Генерація звіту про трафік
        print("Traffic Analysis Report:")
        report = json.dumps(self.traffic_data, indent=4)
        print(report)
        return report

class LoadBalancer:
    def __init__(self, network):
        # Ініціалізація балансувальника навантаження
        self.network = network
    def balance_traffic(self, source_ip, destination_ips):
        # Балансування трафіку між декількома маршрутами
        for destination_ip in destination_ips:
            if (source_ip, destination_ip) in self.network.switching_table:
                route = self.network.switching_table[(source_ip,
destination_ip)]
                if len(route.data_queue) > 5:  # Умова для перерозподілу
                    print(f"High traffic detected on route {source_ip} ->
{destination_ip}. Redistributing traffic.")
                    for alternative_ip in destination_ips:
                        if alternative_ip != destination_ip and (source_ip,
alternative_ip) in self.network.switching_table:
                            alternative_route =
self.network.switching_table[(source_ip, alternative_ip)]
                            while route.data_queue:
                                alternative_route.send(route.data_queue.pop(0))
class SecurityEnhancer:
    def __init__(self, network):
        # Ініціалізація підвищення безпеки мережі
        self.network = network
        self.blacklist = []
```

КБПЗ_2024

```
    def block_ip(self, ip_address):
        # Додавання IP до чорного списку
        self.blacklist.append(ip_address)
       print(f"IP {ip_address} has been blocked.")
    def monitor_and_block(self, ip_address, suspicious_threshold):
        # Моніторинг трафіку і блокування за підозрілу активність
        if ip_address in self.network.devices:
            device = self.network.devices[ip_address]
            if len(device.network.switching_table) > suspicious_threshold:
                self.block_ip(ip_address)
                print(f"Suspicious activity detected for IP {ip_address}. IP has
been blocked.")
class RealTimeMonitoring:
    def __init__(self, network):
        # Ініціалізація реального моніторингу
        self.network = network
        self.running = False

    def start_monitoring(self):
        # Запуск моніторингу в окремому потоці
        self.running = True
        thread = threading.Thread(target=self.monitor_network)
        thread.start()

    def stop_monitoring(self):
        # Зупинка моніторингу
        self.running = False
        print("Real-time monitoring stopped.")

    def monitor_network(self):
        # Реалізація циклу моніторингу
        while self.running:
            for (source, destination), route in
self.network.switching_table.items():
                print(f"Monitoring: Route {source} -> {destination}, Queue size:
{len(route.data_queue)}")
            time.sleep(5)

class HistoricalDataRecorder:
    def __init__(self):
        # Ініціалізація запису історичних даних
        self.data_logs = []

    def record_event(self, event):
        # Запис події
        timestamp = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
        self.data_logs.append({"timestamp": timestamp, "event": event})

    def generate_history_report(self):
        # Генерація звіту про історичні дані
        print("Historical Data Report:")
        for log in self.data_logs:
            print(f"{log['timestamp']}: {log['event']}")
```

КБПЗ_2024