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
“____” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи керування розподіленою мережею на базі Zyxel”

Виконав здобувач вищої освіти
IV курсу, групи КІ-22-1
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Лисяк А.А.
«____» _____ 2026 р.

Керівник проекту
доктор філософії (PhD)
_____ Усік П.С.
«____» _____ 2026 р.
Рецензент _____
_____

м. Кропивницький

Центральноукраїнський національний технічний університет
Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Освітній ступінь *бакалавр*
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
« 27 » лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Лисяку Артему Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи керування розподіленою мережею на базі Zyxel*

2. Керівник роботи *Усік Павло Сергійович, доктор філософії (PhD)*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 133-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту *23.05.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи*: Метою роботи є розробка програмного забезпечення системи керування розподіленою мережею на базі Zyxel*

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

1. Призначення та область використання.
2.Перегляд аналогічних існуючих систем.
3. Опис і обґрунтування проектних рішень.
4. Етапи програмування системи.
5. Впровадження системи в промислову експлуатацію.
6. Висновки

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

Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи додатку	*2 аркуша*

7. Дата видачі завдання « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв’язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника
_____ Усік П.С.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача
_____ Лисяк А.А.
(прізвище та ініціали)

АНОТАЦІЯ

Лисяк А.А. Програмне забезпечення системи керування розподіленою мережею на базі Zyxel. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування розподіленою мережею на базі Zyxel.

Метою розробки є програмне забезпечення системи керування розподіленою мережею на базі Zyxel.

Результат роботи – програмна реалізація системи керування розподіленою мережею на базі Zyxel.

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, розподілена мережа, Zyxel

ABSTRACT

Lysiak A.A. Software for a distributed network control system based on Zyxel. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed that is intended for a distributed network control system based on Zyxel.

The purpose of the development is software for a distributed network control system based on Zyxel.

The result of the work is a software implementation of a distributed network control system based on Zyxel.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

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

The program was developed in the Python environment.

Keywords: computer engineering, distributed network, Zyxel

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БНМ	–	балансування навантаження мережі, Network Load Balancing
ЕЦП	–	електронний цифровий підпис
ІВК	–	інфраструктура відкритих ключів
ІТ	–	інформаційні технології
ЛОМ	–	локальна обчислювальна мережа
ПЗ	–	програмне забезпечення
СКЗІ	–	система комплексного захисту інформації
УК	–	управляючі компоненти
УЦ	–	удостовірюючий центр
IGMP	–	Internet Group Management Protocol
NLB	–	Network Load Balancing, балансування мережевого навантаження
PKI	–	Public Key Infrastructure
VPN	–	віртуальна приватна мережа

ВСТУП

Актуальність теми. Завдяки Zyxel хмарне керування розподіленою мережею стає доступним для компаній малого й середнього бізнесу. Під впливом користувальницьких переваг корпоративні IT перетерпіли за останні роки серйозні зміни. Характерний приклад – дозвіл на підключення до корпоративної мережі пристроїв, що належать співробітникам (Bring Your Own Device, BYOD). Втім, хоча багато хто в повсякденному житті (і тайкома на роботі) користуються самими різними хмарними сервісами, зокрема хмарними сховищами, компанії дотепер насторожено відносяться до таких можливостей.

Тим часом хмарні сервіси допомагають ефективно вирішувати багато завдань, і одна з них – віддалене керування мережевим устаткуванням у компанії з розподіленою структурою. Перші подібні пропозиції були націлені на досить вузький сегмент – керування бездротовими точками доступу. У даній роботі розробляється система керування розподіленою мережею на базі Zyxel, за допомогою яких можна контролювати не тільки точки доступу, але й інше мережеве устаткування.

Із системи керування розподіленою мережею на базі Zyxel можна здійснювати моніторинг і налаштування системи хмарного керування розподіленою мережею на базі пристроїв Zyxel. На даний момент у лінійці сумісного устаткування чотири точки доступу з підтримкою 802.11ac (плюс три гібридні точки доступу, які можуть працювати й автономно), п'ять моделей комутаторів (три з них з підтримкою PoE+) і три моделі шлюзів з функціями виявлення й запобігання вторгнень.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи керування розподіленою мережею на базі Zyxel.

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

– Огляд існуючих систем керування розподіленою мережею на базі Zyxel.

– Дослідження системи керування розподіленою мережею на базі Zyxel.

– Програмна реалізація системи керування розподіленою мережею на базі Zyxel.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі керування розподіленою мережею на базі Zyxel.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи керування розподіленою мережею на базі Zyxel, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Zyxel Networks – провідний постачальник безпечних мережевих рішень на базі штучного інтелекту для малого та середнього бізнесу (МСБ) та корпоративного рівня. Ми пропонуємо безперебійне мережеве підключення та гнучкі рішення з використанням найновіших технологій, що підтримуються надійними та безпечними платформами управління.

Компанія Zyxel Networks має багату історію в мережевій індустрії та понад 30 років об'єднує бізнес та домашніх користувачів. У 1989 році це означало аналогові модеми. Сьогодні це означає використання хмари для забезпечення спрощеного, але потужного уніфікованого мережевого досвіду, використання штучного інтелекту та машинного навчання для забезпечення кібербезпеки та мережевого підключення, які постійно розвиваються, щоб відповідати вимогам світу, що постійно змінюється.

Входячи до складу Zyxel Group, Zyxel Networks є світовим лідером на ринку мереж з неперевершеною міжнародною присутністю на 150 ринках. Ми поставили понад 100 мільйонів пристроїв і користуємося довірою понад 1 мільйона компаній, які працюють розумніше з нашими рішеннями.

Zyxel Networks надає постачальникам керованих послуг (MSP) більше можливостей для масштабування та зростання

Останній випуск платформи керування хмарою Zyxel Nebula спрощує налаштування та керування кількома орендарськими локаціями, дозволяючи партнерам використовувати більше бізнес-можливостей.

Zyxel Networks, лідер у розробці безпечних хмарних мережевих рішень на базі штучного інтелекту, спрощує для постачальників управляних послуг масштабування та розвиток бізнесу завдяки серії вдосконалень своєї платформи

управління хмарою Zyxel Nebula.

Оновлення надають нові можливості для реплікації конфігурацій сайтів, скорочуючи час, необхідний для налаштування нових локацій, з кількох годин до хвилин. Розширена панель інструментів надає постачальникам керованих послуг (MSP) кращу видимість на кількох сайтах, а додаткові функції безпеки допомагають зменшити ризик отримання неавторизованими особами доступу до мережевих налаштувань. Нові функції вже доступні всім партнерам, які використовують Zyxel Nebula від Zyxel Networks для надання своїм клієнтам послуг керованого мережевого моніторингу та моніторингу безпеки.

Тисячі партнерів-постачальників керованих послуг (MSP) у регіоні EMEA використовують Zyxel Nebula як центральну основу своєї ціннісної пропозиції щодо керованих послуг, і в міру їх зростання та розвитку ключовим завданням, з яким вони стикаються, є здатність робити більше з тими ж ресурсами. Розширені можливості Zyxel Nebula дозволять MSP оптимізувати ефективність, використовувати більше можливостей та надавати вищий рівень обслуговування. Вони також роблять Zyxel Nebula ще кращим інструментом для управління клієнтами, яким потрібна узгодженість на всіх локаціях для дотримання вимог та мінімізації ризиків.

Крім того, нові функції зміцнюють позиції постачальників керованих послуг (MSP), які орієнтуються на ринок малого та середнього бізнесу (SMB). SMB зараз є сектором ринку керованих послуг, який найшвидше розвивається, і завдяки новій міжсайтовій реплікації, підтримці управління командами, консолідованому перегляду панелі інструментів та іншим новим функціям Zyxel Nebula, партнери Zyxel Networks мають ще більше можливостей повною мірою скористатися цією величезною можливістю зростання.

Понад 100 продуктів Zyxel Networks, включаючи точки доступу WiFi 7 та уніфіковані шлюзи безпеки, тепер можна налаштовувати та керувати ними через платформу Zyxel Nebula, що надає постачальникам керованих послуг (MSP) значну можливість розширити свої можливості та збільшити продажі як

продуктів, так і послуг. Серед них – остання оновлена версія популярного маршрутизатора безпеки USG LITE 60AX, який тепер має функцію Dual-WAN для резервного перемикання між мережами, а також комплексний пакет засобів безпеки, що забезпечує малий бізнес бюджетним універсальним рішенням.

1.2 Область застосування

Цілком очевидно, що робота будь-якої системи, навіть найпростішої, стає неможлива без її контролю на різних рівнях. Адміністрування програми може вироблятися в ручному напівавтоматичному або автоматичному режимі. Автоматизація програмного забезпечення дозволяє знизити експлуатаційні витрати й збільшити доходи через ріст пропускної здатності мережі.

Менеджмент роботи застосунків, тестування й аналіз функціонування окремих вузлів системи дозволяє вчасно реагувати на відхилення в системі, сигналізувати у випадку непередбачених ситуацій, у тому числі аварійних.

Автоматизована система керування мережами необхідна для реалізації декількох функцій:

- Введення в експлуатацію мереж.
- Забезпечення функціонування мереж.
- Технічне обслуговування.
- Контроль якості програмного комплексу.
- Регулювання трафіку.
- Модернізація мереж.
- Керування розрахунками зі споживачами.

Аспекти системи керування й моніторингу мережі

Виділяють кілька основних характеристик систем керування:

Архітектура

Архітектура дозволяє аналізувати потенційні можливості функціонування системи й принципи адміністрування. У цей час однією з найбільш затребуваних

програм, є архітектура, що базується на агент-менеджері (концепція TMN). Незважаючи на бездоганну роботу зазначеної системи, усе більше явним стає її нездатність застосування стосовно складних систем керування телекомунікаційними мережами. Даний термін має на увазі створення складного ієрархічного комплексу програм.

Структура

Структура програмного забезпечення визначається винятково розроблювачами програми. Однак існують модулі, що є невід'ємною частиною будь-якої системи:

– Графічний інтерфейс.

– Інформаційна база адміністрування.

– Функціональна модель.

– Інформаційна модель.

– Передісторія.

– Математичний комплекс по створенню повідомлень.

– Система сигналізації.

Всі системи керування можна також розділити на два типи:

– централізовані (що мають єдиний центр керування);

– децентралізовані (не мають єдиного центра керування; функції адміністрування в цьому випадку розподілені між безліччю систем).

Рівні

Виділяють 4 рівні керування мережами:

– Керування елементами (містить у собі тестування, тех. обслуговування й угруповання окремих компонентів програмного комплексу).

– Керування мережею (має на увазі контроль і менеджмент мережі в цілому й взаємозв'язок всіх її компонентів).

– Керування послугами (забезпечує взаємодія споживача з постачальниками послуг, дозволяє здійснювати контроль якості надаваних послуг, їхню своєчасну пролонгацію або припинення).

– Адміністрування (ставиться до керування системами самого оператора зв'язку).

Напрямки

У ряді основних напрямків, на які націлена будь-яка програма керування мережами, можна вказати:

– Пошук помилок.

– Надання доступу до інформаційної бази даних.

– Підвищення продуктивності.

– Безпека інформаційних джерел.

Протоколи керування

Інтерфейс системи керування мережами передачі даних являє собою сукупність протоколів, тобто набір прийомів і процедур, націлених на функціонування системи протягом усього строку існування. Виділяють кілька основних протоколів, на яких базуються сучасні моделі програмного забезпечення цього типу: SNMP, CMIP, WBEM, TMN, LNMP, ANMP, WMI і ін.

Системи керування локальною мережею

Керування комп'ютерною мережею, як правило, здійснюється представниками керівного состава окремих компаній і виконує трохи значимих для розвитку функцій:

– Моніторинг комп'ютерної мережі.

– Координація робота комп'ютерних ресурсів.

– Контроль продуктивності.

– Регулювання трафіку.

– Безпека.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи керування розподіленою мережею на базі Zyxel, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

SmartView – Система керування елементами мережі

Переваги системи керування елементами мережі SmartView:

– обробки аварійних сигналів у режимі реального часу;

– забезпечення всіх функцій налаштування й обслуговування для пристрою зв'язку;

– доступу до функцій EMS сервера здійснюється за допомогою CORBA;

– використовує SNMP протокол для моніторингу й керування мережевими пристроями.

EMS сервер призначений для забезпечення всіх функцій налаштування й обслуговування для пристрою зв'язку. Метод доступу до функцій EMS сервера здійснюється за допомогою CORBA протоколу через технічні характеристики OMG CORBA. Коли користувач завантажує програмне забезпечення EMS Client і встановлює посилання на сервері EMS, можна відслідковувати й контролювати всі мережеві пристрої за допомогою CORBA. EMS сервер використовує SNMP протокол для моніторингу й керування мережевими пристроями за допомогою SET GET і TRAP SNMP.

Система керування мережевими елементами:

– Платформа централізованого керування мережею.

– Візуальні подання й обробки аварійних сигналів у режимі реального часу.

– Проста й зручна експлуатація інтерфейсу.

Характеристики**:**

– Дистанційне керування доступом для ефективної конфігурації.

– Трафік/моніторинг і керування продуктивністю.

– Сигнал тривоги й керування файлами реєстрації подій.

– Автоматичне виявлення й перегляд пристроїв.

– Дозволяє вхід до 25 адміністраторам.

– Стосовно до основного видам продукції CTM Union.

Система керування оператора первинної мережі на базі TMN

Демонополізація й лібералізація в області зв'язку, поява безлічі операторів на території України, високі вимоги по якості зв'язку з боку споживачів породили конкурентне середовище, виживання в якій представляє складне завдання для операторів.

Необхідність знижувати тарифи в найближчому майбутньому, збільшення капітальних вкладень на впровадження нових технологій з метою надання нових послуг збільшують витрати операторів на експлуатацію мереж.

У цих умовах ефективна підтримка функціонування мереж зв'язку можливо лише за допомогою автоматизації ручних процесів експлуатації мереж, застосування новітніх систем керування, які дозволяють максимізувати доходи, мінімізувати витрати, забезпечити високу якість обслуговування користувачів, підтримуючи тим самим потрібний рівень конкуренції. Сучасні системи керування операторів зв'язку дозволяють не тільки управляти мережами зв'язку, але також послугами й бізнесом в інтересах замовника. У цьому розділі розглянуті системи керування (СК) мережами операторів із загальних позицій Концепції TMN (Telecommunication Management Network – телекомунікаційна мережа керування).

Основними завданнями TMN є керування мережами електрозв'язку (різних розмірів і типів) і взаємодією з мережами інших операторів. Відповідно, реальні СК повинні розроблятися для виконання цих завдань. У зв'язку з наявністю мереж різної природи (наприклад, мереж передачі, мереж комутації,

рухливих мереж, мереж доступу й ін.) повинні існувати й різні СК з урахуванням особливостей мереж операторів. Оскільки потрібне взаємодія всіх СК й мереж між собою й із центральними органами керування, то для цього необхідно використовувати однакові архітектурні принципи побудови систем керування. Крім того, щоб елементи СК оператора могли обмінюватися інформацією керування, вони повинні приєднуватися до лінії зв'язку, і кожний елемент на ній повинен підтримувати однаковий інтерфейс. Для спрощення проблем інформаційної взаємодії на мережі з різними мережевими технологіями й устаткуванням різних постачальників варто використовувати сумісні інтерфейси. Сумісний інтерфейс визначає множину протоколів, процедур, форматів і семантики повідомлень, які передаються відповідно до його й використовуються з метою керування. Він повинен бути об’єктно-орієнтованим, щоб на основі переданих повідомлень можна було здійснювати «маніпуляції» з об'єктами керування.

Загальна архітектура TMN має три складові: функціональну, інформаційну й фізичну. Функціональна описує розподіл функцій між елементами TMN з метою створення систем будь-якої складності. Визначення функціональних блоків і опорних точок для їхньої взаємодії приводить до специфікації типів інтерфейсів TMN. Інформаційна архітектура заснована на об’єктно-орієнтованому підході й дає способи реалізації керування й взаємодії відкритих систем (ВВС).

Фізична архітектура описує реалізовані інтерфейси й інші речовинні елементи (технічні засоби), які утворять TMN.

З урахуванням складності керування системою електрозв'язку функції керування мережею в TMN розділяються на чотири рівні: керування бізнесом – керування послугами – керування мережею – керування елементами мережі. Кожний рівень обмежує набір завдань керування у відповідності зі своїм рангом.

На всіх рівнях завдання й функції керування виконуються відповідними операційними системами. Операційні системи (OS) є основним технічним

засобом СК й використовуються для збору й обробки різного виду інформації, підтримки прикладних програм, ведення баз даних і т.п. Архітектура OS залежить від місця в мережі й виконуваних функціях. Наприклад, OS на рівні елемента мережі управляє їм на індивідуальній основі й не знає топології мережі; OS на рівні керування послугами здійснює керування однойменними послугами, надаваними за допомогою різних мереж, і реалізує інтерфейс із замовником послуг. Операційна система рівня керування бізнесом повинна вирішувати наступні завдання: підтримки прийняття рішень для оптимізації інвестиційного процесу й використання нових ресурсів мережі, підтримки керування бюджетом експлуатації, підтримки системи матеріально-технічного постачання й вимог технічного персоналу, обробки й ведення даних про доходи й витрати оператора.

Елементи мережі (NE) повинні виконувати функції електрозв'язку, наприклад передачі й комутації, і функції підтримки, які потрібні для керованості мережі.

Для виконання функцій названих вище блоків систем керування TMN створюються відповідні підсистеми: підсистема керування елементами (EMS), підсистема керування мережею (NMS), підсистема керування послугами (SMS) і підсистема керування бізнесом (BMS), а також центри експлуатації й технічного обслуговування (OMC), центри керування мережею (NMC), центри керування послугами (SMC) і центри керування бізнесом (BMC). Керування послугами й бізнесом може бути об'єднане в загальну підсистему керування (загальний центр).

Інформація для керування групується на базі об’єктно-орієнтованого підходу в термінах керованих об'єктів, які є концептуальним (абстрактним) поданням таких ресурсів (або їхніх взаємин), які можуть бути керованими. Відповідно до рекомендації МСЕ-Т X.722 керований об'єкт характеризується атрибутами, видимими на його границі, що управляють діями, які можуть застосовуватися до об'єкта, повідомленнями, які генеруються керованим об'єктом, поводженням, що він демонструє у відповідь на керуючі впливи або інші типи подій.

Інакше кажучи, немає необхідності робити однозначне відображення між керованими об'єктами й реальними ресурсами, які можуть бути фізичними або логічними. Більше того, той самий фізичний (логічний) ресурс може бути представлений декількома об'єктами керування. Керовані об'єкти більшою мірою є логічними, ніж фізичними ресурсами мережі.

Розглянемо систему керування оператора первинної мережі. Кожна із вхідних у неї підсистем керування елементом мережі (EMS) виконує функції «Менеджера елемента», які відносяться до керування окремим елементом мережі. Менеджер елемента звичайно виконує наступні типові функції:

– Контроль підмножини NE по наступних питаннях:

а) конфігурації NE, наприклад: зміна конфігурації устаткування; конфігурація ресурсів каналів; керування пристроями синхронізації; доповнення конфігурації;

б) керування програмним забезпеченням (ПЗ) NE (включаючи його завантаження);

в) керування тривожною сигналізацією й помилками, наприклад: фільтрація подій і тривожних повідомлень, візуалізація, завантаження; ведення списку поточних тривожних повідомлень; доступ до завантаження й видача повідомлень;

г) моніторингу якості роботи NE, наприклад: конфігурація точки виміру; керування лічильниками якості роботи (старт, зупинка, обнуління, відновлення); доступ до завантаження й видача повідомлень; конфігурація порогів тривожних повідомлень для якості роботи.

– Керування безпекою у функціонального й мережевого доступу.

– Одержання інформації про несправності (наприклад, розташування ушкоджень, тип ушкодження, процедури ремонту).

– Шлюзові функції рівня керування мережею й функції робочої станції (WS). Робоча станція являє собою систему, що повинна мати у своєму розпорядженні засоби для інтерпретації інформації керування в людино-

машинній мові для користувача й навпаки. У загальному значенні WS можуть розглядатися як термінали, які приєднуються через мережу передачі даних до операційних систем.

Коли EMS управляє групою елементів, які утворять підмережа (наприклад, кільце СЦІ), те EMS називають системою керування підмережею (SNMS).

Наступний рівень – підсистема керування мережею (NMS). Вона виконує роль «Менеджера мережі», що ставиться до процесів керування на рівні мережі в цілому. Менеджер мережі зберігає й обробляє дані, отримані на рівні EMS, і може виконувати наступні типові функції:

– Контроль мережі або підмережі й керування ними:

а) контроль конфігурації мережі: фізичної мережі; мережі синхронізації; трактів від точки до точки (односпрямованих і двонаправлених); трактів від точки до декількох точок (конфігурація мережі розподілу віщання); резервної мережі; засобів відновлення;

б) керування несправностями й аварійними повідомленнями: фільтрація аварійних повідомлень, обробка, завантаження й візуалізація; кореляція аварійних повідомлень трактів; ведення поточного списку трактів з аварійними повідомленнями;

в) моніторинг якості роботи трактів на відповідність Рекомендаціям G.821, G.826: керування вимірами трактів; керування лічильниками якості роботи трактів (старт, зупинка, обнуління, відновлення); доступ до завантаження трактів і видача повідомлень; керування порогами аварійних повідомлень.

– Керування безпекою функціонального й мережевого доступу.

– Збір даних по взаєморозрахунках між операторами за пропуск трафіку й приєднання до мережі.

– Шлюзова функція рівня керування послугами й робочої станції (WS).

Рівень керування мережею може містити деяке число підрівнів відповідно до ієрархії мереж, наприклад, складатися з місцевої, внутрізонової, регіональної й

національної мереж. У цьому випадку підрівень із вищою ієрархією управляє всією мережею, що складається з підмереж, розташованих на більше низьких підрівнях

Підсистема керування послугами (SMS) виконує функцію менеджера послуг, що ставиться до керування на рівні керування послугами, і може виконувати наступні типові функції: керування орендованими каналами; задоволення запитів замовників; надання й припинення надання послуг; фіксація повідомлень про скарги й несправності.

EMS різних типів передають інформацію контролю в систему керування мережею (NMS), що збирає інформацію від всіх EMS, обробляє й представляє цю інформацію для всієї мережі в цілому. Знання загального виду мережі дозволяє NMS найбільше точно визначати проблеми мережі й оптимізувати керуючі дії для всієї мережі в цілому. NMS взаємодіє з операційними системами наступних рівнів: керування послугами (SMS) і бізнесу (BMS).

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це потужна мова програмування, яка проста у вивченні. Він має ефективні структури даних високого рівня та простий, але ефективний підхід до об’єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованим характером роблять його ідеальною мовою для створення сценаріїв і швидкої розробки додатків у багатьох сферах на більшості платформ.

Інтерпретатор Python і обширна стандартна бібліотека доступні у вихідному або двійковому вигляді для всіх основних платформ на веб-сайті Python https://www.python.org/ і можуть вільно поширюватися. Цей же сайт також містить дистрибутиви та вказівники на багато безкоштовних сторонніх модулів Python, програм і інструментів, а також додаткову документацію.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускну кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи керування розподіленою мережею на базі Zyxel.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) 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.

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Zyxel Vantage CNM

Централізоване рішення для управління безпекою та звітності

Централізоване керування мережею Vantage (CNM) – це інструмент керування на основі браузера для постачальників послуг та IT-персоналу. Розподілені мережеві компоненти можна налаштовувати, аналізувати та контролювати централізовано. Цей «головний офіс управління» спрощує отримання огляду мережевої активності та підвищує ефективність резервних служб.

– Інтуїтивне створення VPN та прості інструменти діагностики.
– Централізований контроль UTM та управління ліцензіями.
– Експлуатація та обслуговування групових пристроїв.
– Активний моніторинг та комплексна звітність.

Переваги

Просте керування VPN та можливості діагностики

Zyxel Vantage CNM – це централізоване рішення для керування, яке дозволяє IT-менеджерам адмініструвати серію ZyWALL Unified Security Gateway (USG). Конструкція управління VPN допомагає скоротити час, витрати та складність, пов'язані з конфігурацією VPN. Мережеві адміністратори можуть легко створювати мережі VPN та змінювати параметри VPN (наприклад, попередньо спільний ключ, IP-адресу та ідентифікатор шлюзу тощо) на кількох пристроях у мережі, а ця функція також дозволяє уникнути проблем із конфігурацією, а також потенційної людської помилки. Функція моніторингу VPN дозволяє мережевим адміністраторам активно контролювати стан тунелю VPN та усувати проблеми VPN за допомогою діагностичних інструментів.

Повне управління політикою безпеки та єдиним управлінням загрозами (UTM)

Vantage CNM забезпечує єдиний інтегрований інтерфейс керування, який дозволяє керувати всіма пристроями з централізованого місця. За допомогою централізованої консолі мережеві адміністратори можуть легко керувати послугами UTM для підписки, оновлення та моніторингу стану. Сповіщення про закінчення терміну дії активної ліцензії може генеруватися автоматично, щоб нагадати адміністраторам і клієнтам про поновлення, водночас ретельно дотримуючись політик безпеки та зменшуючи складність управління. Мережеві атаки та загрози легко виявляються за допомогою панелі інструментів; корисні комплексні звіти та можливість деталізації також допомагають адміністратору аналізувати джерела та тенденції атак.

Низька сукупна вартість володіння (TCO) для масового розгортання та обслуговування пристроїв

Vantage CNM забезпечує швидке налаштування одного пристрою або групи пристроїв, що підвищує ефективність управління та зменшує накладні витрати для масових розгортань. Він дозволяє адміністраторам проводити оновлення прошивки для кількох пристроїв та мінімізувати перебої в обслуговуванні кінцевих користувачів завдяки опції планування. Налаштування пристроїв можна зберігати та відновлювати регулярно або за потреби, а групова робота зменшує зусилля адміністратора на обслуговування мережі. Використовуючи CNM, адміністратори можуть навіть відображати та сортувати статуси (онлайн, офлайн, тривога або заплановані завдання) численних ZyWALL шляхом активного моніторингу пристроїв.

Активний моніторинг, оповіщення та вичерпні графічні звіти

Централізована функція ведення журналу Vantage CNM дозволяє адміністраторам аналізувати журнали пристроїв з однієї консолі та надає багато корисних попередньо визначених шаблонів, включаючи звіти про мережевий трафік, мережеві атаки, використання VPN, використання Інтернету та політику

безпеки, щоб керівництво могло легко аналізувати поведінку мережі та користувачів. Можна створювати автоматичні щоденні/тижневі заплановані звіти, а налаштовувані шаблони звітів дозволяють постачальникам керованих послуг (MSP) надавати своїм клієнтам індивідуальні звіти.

Хмарне мережеве управління

У певному сенсі, надання мережевих рішень для малого та середнього бізнесу схоже на космічні подорожі. Обидва види діяльності пов'язані з технологіями. Для успішної роботи в обох випадках потрібен багаторічний досвід. І обидва варіанти дійсно варто залишити експертам. Але іноді, навіть якщо ви справді знаєте свою справу, важко зрозуміти, що робити далі. Візьмемо, наприклад, хмарне управління мережею. Дехто каже, що це найкращий спосіб задовольнити потреби сучасного бізнесу. А інші вважають, що ще зарано приєднуватися до цього міжгалактичного тренду. Якщо ви налаштовували та керували мережами локально, ви знаєте, що це велике та складне завдання. Однак врахуйте ось що: за допомогою правильних інструментів налаштування нових мереж за допомогою хмари не просто простіше, а швидше за швидкість світла. Це також масштабовано, тому, коли ви захочете розширити їхню ємність, ви можете зробити це швидше, ніж можете собі уявити. Ви, мабуть, добре знаєте про накладні витрати, пов'язані з управлінням локальною мережею. Технічне обслуговування, поїздки, робоча сила та час простою можуть бути дорогими. Річ у тім, що завдяки хмарному управлінню мережею ці витрати можна або значно зменшити, або повністю виключити. Це означає, що ви можете залишати собі більшу частину зароблено, а зайву суму можете витратити на міжзоряні ракетні прискорювачі або інвестувати у свій бізнес. Звісно, ви знаєте, що неважливо, скільки у вас модних пристроїв, якщо вони всі не поєднуються в одне ефективне рішення. Тож, якщо ви все ж таки перейдете на хмарні технології, вам знадобиться інструмент керування, який працює з бездротовими, комутаторними та безпековими продуктами. Незалежно від того, чи залишаєтесь ви локально, чи інвестуєте в хмару, ви можете продовжувати прагнути до зірок і за їх межі.

Час розпочати хмарне управління мережею

Якщо ви думаєте про грандіозний крок у розвитку свого бізнесу, подумайте про використання хмарного інструменту керування мережею Zyxel, Zyxel Nebula.

Легке налаштування

Функція Plug-n-play спрощує налаштування всіх точок доступу, комутаторів та шлюзів Zyxel Nebula. Просто відскануйте QR-код пристрою за допомогою мобільного додатку Zyxel Nebula на своєму смартфоні та одразу почніть керувати своїм пристроєм через Zyxel Nebula Control Center.

1. Один вид. Отримайте повний огляд вашої мережі за допомогою інтуїтивно зрозумілого інтерфейсу керування. Скористайтеся перевагами централізованого адміністрування мереж з кількома об'єктами, не витрачаючи ресурси на витрати на вантажівки, робочу силу тощо.

2. Витрачайте розумніше. Вбудована масштабованість, резервування та оплата за зростання забезпечують підвищену гнучкість, що дозволяє легко переходити від моделі управління капітальними витратами до моделі управління операційними витратами.

3. Залишайтеся в робочому стані. Насолоджуйтесь надійною безвідмовною роботою 99,99% та доступом до керування будь-коли та будь-де.

4. Підтримуйте безпеку. Zyxel Nebula не передає жодних даних користувачів через хмару та шифрує все за протоколом NETCONF.

3.2 Розробка структурної схеми

Ключові нові функції Zyxel Nebula включають:

– Шаблони конфігурації, які постачальники керованих послуг (MSP) можуть використовувати для синхронізації налаштувань на кількох сайтах однієї організації клієнта, з автоматичним відображенням майбутніх змін.

– Повна реплікація сайту дозволяє постачальнику керованих послуг (MSP)

миттєво клонувати свої налаштування «найкращих практик» або налаштування існуючого сайту на нове місце, що потенційно заощаджує години або навіть дні, необхідні для складного процесу ручного адаптації.

– Оновлені можливості резервного копіювання та відновлення дозволяють копіювати кілька сайтів масово, автоматизувати планування та швидко відновлювати або відкатувати попередні версії.

– Оновлена центральна панель інструментів MSP тепер відображає стан організації клієнта, сайту та пристроїв в одному вікні, надаючи партнерам повний контроль над усіма клієнтами.

– Зведений огляд інвентаризації пристроїв та ліцензій для всіх клієнтів.

– Журнали змін на рівні організації містять детальні хронологічні записи, які відстежують, хто які зміни вніс і коли, що забезпечує відповідність вимогам та потреби аудиту, а також дозволяє швидше виправляти та усувати несправності.

– Контроль доступу на основі ролей дозволяє призначати права доступу відповідно до ролей співробітників за допомогою функції «Адміністратори та команди» Zyxel Nebula.

– Миттєве адаптування та виключення означає, що новим співробітникам можна миттєво надати успадкований, попередньо визначений доступ, а тих, хто звільнився, можна видалити за один крок.

– Групування клієнтів означає, що постачальники керованих послуг (MSP) тепер можуть організовувати клієнтів зі схожими вимогами, щоб забезпечити узгодженість та зменшити адміністративні накладні витрати.

У сучасному гіперпов’язаному бізнес-ландшафті управління мережами перетворилося з технічної необхідності на стратегічний імператив. З огляду на те, що світовий ринок систем управління мережами оцінювався в 11,86 мільярда доларів у 2025 році та, за прогнозами, досягне 25,94 мільярда доларів до 2034 року, організації в усьому світі визнають, що ефективне управління мережами має вирішальне значення для операційної досконалості, безпеки та конкурентної переваги.

Сучасні компанії покладаються на дедалі складніші ІТ-інфраструктури, що охоплюють численні місця розташування, хмарні середовища та екосистеми пристроїв. Зростання віддаленої роботи, поширення Інтернету речей та ініціатив цифрової трансформації створили мережі, які стали більш розподіленими та вимогливими, ніж будь-коли раніше. 57% мережевих адміністраторів називають брак прозорості своєю головною проблемою, тоді як 45% ІТ-фахівців визнають, що вони не до кінця розуміють конфігурацію своєї мережі. Ця складність, у поєднанні з тим фактом, що мережеві адміністратори витрачають майже 50% свого часу на гасіння пожеж в ІТ, підкреслює нагальну потребу в комплексних стратегіях управління мережею.

Управління мережею – це вже не просто підтримка працездатності систем, а забезпечення гнучкості бізнесу, забезпечення безпеки, оптимізація продуктивності та надання надійного цифрового досвіду, якого очікують сучасні клієнти та співробітники.

Управління мережею охоплює комплексні процеси, інструменти та практики, що використовуються для моніторингу, адміністрування та оптимізації комп'ютерних мереж та їх компонентів. Воно включає нагляд за всіма аспектами мережевої інфраструктури, включаючи апаратні пристрої, програмні системи, протоколи підключення, заходи безпеки та оптимізацію продуктивності, щоб забезпечити доступність, безпеку та ефективність мережевих ресурсів.

Основні компоненти управління мережею:

– Нагляд за інфраструктурою. Управління мережею охоплює адміністрування фізичних та віртуальних мережевих компонентів, включаючи маршрутизатори, комутатори, брандмауери, точки доступу та сервери. Це включає моніторинг стану пристроїв, керування конфігураціями та забезпечення оптимальної продуктивності всіх мережевих елементів.

– Оптимізація продуктивності. Безперервний моніторинг та налаштування показників продуктивності мережі, таких як використання пропускної здатності, затримка, пропускна здатність та втрата пакетів. Це включає виявлення вузьких

місць, оптимізацію потоків трафіку та забезпечення дотримання стандартів якості обслуговування (QoS).

– Адміністрація безпеки. Впровадження та управління політиками мережевої безпеки, контролем доступу, системами виявлення загроз та процедурами реагування на інциденти. Цей критичний аспект часто інтегрується з комплексними практиками аудиту кібербезпеки для підтримки надійних систем безпеки.

– Управління ресурсами. Розподіл та моніторинг мережевих ресурсів, включаючи управління IP-адресами, розподіл пропускної здатності та планування потужностей для підтримки поточних потреб та підготовки до майбутнього зростання.

Системи управління мережею: Командний центр

Система управління мережею (NMS) слугує централізованою платформою для моніторингу, контролю та оптимізації мережевої інфраструктури. Сучасні рішення NMS забезпечують видимість продуктивності мережі в режимі реального часу, автоматизують рутинні завдання та дозволяють проактивно вирішувати проблеми.

Типи систем управління мережею:

– Хмарні системи управління мережею. Хмарні рішення NMS розміщуються на віддалених серверах і доступ до них здійснюється через Інтернет, зазвичай вони пропонуються як продукти «програмне забезпечення як послуга» (SaaS). Ці системи пропонують кілька переваг, включаючи зниження витрат на інфраструктуру, автоматичні оновлення, масштабованість і віддалений доступ. Вони особливо популярні серед малого та середнього бізнесу та організацій з розподіленими мережами.

– Локальні системи управління мережею. Локальні рішення NMS встановлюються та експлуатуються у власній інфраструктурі організації, забезпечуючи повний контроль над системою та даними. Ці рішення пропонують підвищену безпеку для чутливих середовищ, ширші можливості налаштування та

дотримання суворих вимог щодо місця зберігання даних. Їм часто надають перевагу великі підприємства та організації в регульованих галузях.

– Гібридні підходи до управління мережею. Багато організацій застосовують гібридні підходи, що поєднують хмарні та локальні елементи, подібно до гібридних хмарних стратегій, що дозволяє їм збалансувати контроль, безпеку та гнучкість на основі конкретних вимог.

Як працює управління мережею: повний процес

Управління мережею здійснюється за допомогою систематичного підходу, який поєднує автоматизований моніторинг, інтелектуальний аналіз та проактивні механізми реагування:

– Виявлення та картографування мережі. Основа ефективного управління мережею починається з комплексного виявлення мережі. Цей процес включає:

– Автоматизоване виявлення пристроїв: сканування мережі для ідентифікації всіх підключених пристроїв, від серверів і комутаторів до датчиків Інтернету речей і мобільних пристроїв.

– Топологічне відображення: створення візуальних представлень архітектури мережі, що показують, як пристрої підключаються та як відбуваються потоки даних.

– Інвентаризація активів: каталогізація специфікацій пристроїв, версій програмного забезпечення та деталей конфігурації.

– Аналіз залежностей: Розуміння того, як різні компоненти мережі залежать один від одного.

– Безперервний збір та моніторинг даних.

Після того, як мережа буде відображена, починається безперервний моніторинг через кілька каналів:

– Опитування SNMP: регулярні запити до мережевих пристроїв щодо оновлень стану та показників продуктивності.

– Аналіз потоку: моніторинг моделей трафіку та використання пропускної здатності в усіх сегментах мережі.

– Агрегація журналів: Збір та аналіз даних журналів з різних мережевих пристроїв та програм.

– Метрики в режимі реального часу: відстеження показників ефективності, таких як затримка, втрата пакетів та доступність.

– Інтелектуальний аналіз та оповіщення.

Сучасні системи управління мережею використовують штучний інтелект та машинне навчання для аналізу зібраних даних:

– Виявлення аномалій: Виявлення незвичайних закономірностей, які можуть свідчити про проблеми або загрози безпеці.

– Прогнозна аналітика: прогнозування потенційних проблем до того, як вони вплинуть на операції.

– Автоматизована кореляція: З'єднання пов'язаних подій між різними мережевими компонентами.

– Управління порогами: встановлення динамічних базових значень та генерування сповіщень, коли показники виходять за межі нормальних діапазонів

– Автоматизація та реагування.

Розширені платформи управління мережею включають можливості автоматизації для реагування на проблеми:

– Самовідновлювальні мережі: автоматичне перенаправлення трафіку в обхід несправних компонентів.

– Керування конфігурацією: розгортання узгоджених налаштувань на всіх мережевих пристроях.

– Керування виправленнями: планування та застосування оновлень безпеки протягом періодів технічного обслуговування.

– Реагування на інциденти: Запуск автоматизованих дій щодо усунення поширених проблем.

Ключові завдання та обов'язки з управління мережею

Ефективне управління мережею охоплює широкий спектр поточних завдань, необхідних для підтримки оптимальної продуктивності та безпеки

мережі:

Налаштування та забезпечення мережі.

Розподіл ресурсів: розподіл пропускної здатності, IP-адрес та мережевої потужності для підтримки різних служб та груп користувачів. Це передбачає ретельне планування, щоб забезпечити наявність достатніх ресурсів там, де це необхідно, уникаючи водночас марнотратства:

– Конфігурація пристроїв: налаштування та конфігурація нових мережевих пристроїв у міру розширення або розвитку інфраструктури. Це включає маршрутизатори, комутатори, точки доступу та пристрої безпеки, що забезпечує їх безперешкодну інтеграцію з існуючою інфраструктурою.

– Розгортання послуг: впровадження нових мережевих послуг або програм, включаючи політики якості обслуговування (QoS), віртуальні локальні мережі (VLAN) та засоби контролю доступу.

Управління безпекою та контроль доступу

Управління мережевою безпекою включає впровадження та підтримку комплексних заходів безпеки:

– Системи контролю доступу: керування автентифікацією та авторизацією користувачів, включаючи інтеграцію з протоколами SSO для оптимізованого та безпечного управління доступом.

– Керування брандмауером: налаштування та підтримка правил брандмауера, систем запобігання вторгненням та політик сегментації мережі.

– Моніторинг загроз: безперервне спостереження за загрозами безпеці, часто інтегроване з рішеннями SIEM проти SOC для комплексного виявлення загроз та реагування на них.

– Оцінка вразливостей: регулярні аудити безпеки та тестування на проникнення для виявлення та усунення потенційних слабких місць.

Моніторинг та оптимізація продуктивності

Управління пропускною здатністю: моніторинг та оптимізація використання пропускної здатності в мережі для запобігання перевантаженню та

забезпечення отримання критично важливими програмами достатніх ресурсів:

– Оптимізація затримки: Виявлення та усунення факторів, що сприяють затримкам у мережі, забезпечення швидкої роботи програм реального часу.

– Планування потужностей: аналіз тенденцій використання та моделей зростання для планування майбутнього розширення та модернізації мережі.

– Якість обслуговування (QoS): впровадження та управління політиками пріоритезації трафіку для забезпечення отримання критично важливими бізнес-додатками відповідних мережевих ресурсів.

Технічне обслуговування та оновлення

Профілактичне обслуговування: регулярний огляд та очищення фізичної інфраструктури, включаючи кабелі, стійки та засоби контролю навколишнього середовища:.

– Оновлення програмного забезпечення: керування оновленнями прошивки, патчами безпеки та оновленнями програмного забезпечення на всіх мережевих пристроях.

– Управління життєвим циклом обладнання: планування та виконання циклів оновлення обладнання, забезпечення заміни застарілого обладнання до того, як воно стане загрозою його надійності.

– Документація: Ведення точної мережевої документації, включаючи схеми топології, файли конфігурації та операційні процедури.

Протоколи управління мережею: структура зв'язку

Протоколи мережевого управління служать стандартизованими методами зв'язку між мережевими пристроями та системами управління, що забезпечує ефективний моніторинг, налаштування та керування.

Простий протокол керування мережею (SNMP)

SNMP залишається найпоширенішим протоколом керування мережею, забезпечуючи основу для:

– Моніторинг пристроїв: збір інформації про стан та показників продуктивності з мережевих пристроїв.

– Керування конфігурацією: Дистанційне налаштування параметрів та налаштувань пристрою.

– Виявлення несправностей: отримання сповіщень та сповіщень, коли пристрої стикаються з проблемами.

– Відстеження ефективності: Збір історичних даних для аналізу тенденцій та планування потужностей.

Потокова телеметрія

Сучасні мережі все частіше використовують потокову телеметрію для збору даних у режимі реального часу:

– Безперервний потік даних: Пристрої безперервно передають дані про продуктивність, а не чекають на запити опитування.

– Вища деталізація: Більш детальний та частий збір даних дозволяє краще зрозуміти поведінку мережі.

– Зменшення накладних витрат: Усуває накладні витрати на опитування, пов'язані з традиційним моніторингом SNMP.

– Аналітика в режимі реального часу: Забезпечує негайний аналіз та реагування на стан мережі.

Протокол керування інтернет-повідомленнями (ICMP)

Хоча ICMP використовується переважно для діагностики мережі, він також відіграє важливу роль в управлінні мережею:

– Тестування підключення: ping-тести для перевірки доступності пристрою та часу відгуку.

– Виявлення шляхів: операції трасування маршрутів для визначення мережевих шляхів та локалізації вузьких місць.

– Звітування про помилки: сповіщення про помилки мережі та недоступні пункти призначення.

Основні переваги комплексного управління мережею

Впровадження надійних практик управління мережею забезпечує значні операційні та стратегічні переваги, які безпосередньо впливають на успіх бізнесу.

Покращена видимість та контроль мережі

Керування мережею забезпечує безпрецедентну прозорість мережевих операцій, вирішуючи критичну проблему, з якою стикаються 57% мережевих адміністраторів через відсутність прозорості у своїх поточних інструментах моніторингу. Такий комплексний нагляд дозволяє:

– Моніторинг у режимі реального часу: безперервний контроль продуктивності мережі на всіх локаціях та пристроях.

– Централізоване керування: уніфікований інтерфейс керування для розподіленої мережевої інфраструктури.

– Детальна аналітика: глибоке розуміння моделей трафіку, тенденцій використання та показників продуктивності.

– Проактивне виявлення проблем: раннє виявлення потенційних проблем, перш ніж вони вплинуть на користувачів.

Значне скорочення простоїв

Ефективне управління мережею значно зменшує кількість перебоїв у роботі мережі та мінімізує час відновлення після виникнення проблем. Організації, які впроваджують комплексне управління мережею, зазвичай спостерігають:

– Проактивне вирішення проблем: виявлення та усунення проблем до того, як вони спричинять перебої в обслуговуванні.

– Швидше реагування на інциденти: Автоматизовані сповіщення та діагностика забезпечують швидке усунення несправностей.

– Покращений MTTR: скорочення середнього часу ремонту завдяки кращій видимості та автоматизованому усуненню недоліків.

– Безперервність бізнесу: Підвищена надійність підтримки критично важливих бізнес-операцій.

Дослідження показують, що сучасні платформи управління мережею, що використовують аналітику та автоматизацію на основі штучного інтелекту, можуть виявляти та вирішувати проблеми до їх ескалації, що призводить до

зменшення простоїв та покращення роботи.

Оптимізована продуктивність мережі

Технології управління мережею систематично оптимізують продуктивність мережі шляхом:

– Виявлення вузьких місць: Точне визначення обмежень продуктивності та впровадження рішень.

– Оптимізація трафіку: інтелектуальна маршрутизація та балансування навантаження для максимальної ефективності.

– Управління пропускною здатністю: забезпечення критично важливих програм достатніми мережевими ресурсами.

– Забезпечення якості: Підтримка послідовних стандартів продуктивності в усіх сегментах мережі.

Організації повідомляють, що оптимізація мережі за допомогою сучасних платформ дозволяє безперешкодно інтегрувати нові пристрої, програми та користувачів без зниження продуктивності.

Посилена безпекова політика

Управління мережею відіграє вирішальну роль у підтримці надійної кібербезпеки, що особливо важливо, оскільки у 2023 році сталося 2365 кібератак. Ключові переваги безпеки включають:

– Безперервний моніторинг: цілодобове спостереження за загрозами безпеці та незвичайною активністю.

– Застосування політик: послідовне впровадження політик безпеки в усіх сегментах мережі.

– Реагування на інциденти: швидке виявлення порушень безпеки та реагування на них.

– Підтримка відповідності: Дотримання нормативних вимог шляхом проведення комплексних аудиторських перевірок та звітності.

Інтеграція управління мережею з практиками аудиту мережевої безпеки забезпечує комплексний захист від загроз, що постійно змінюються.

Підвищення продуктивності та операційної ефективності

Комплексне управління мережею сприяє значному підвищенню продуктивності:

– Зменшення накладних витрат на ІТ: Автоматизація усуває ручні завдання та зменшує адміністративне навантаження.

– Покращений користувацький досвід: надійна мережева продуктивність забезпечує безперебійну роботу.

– Стратегічний фокус: ІТ-команди можуть зосередитися на інноваціях, а не на гасінні пожеж.

– Оптимізація ресурсів: Ефективне використання мережевих ресурсів зменшує втрати та витрати.

Дослідження показують, що організації можуть розраховувати на 50% зниження витрат на інструменти для багатоточкових рішень , що призведе до економії близько 1 мільйона доларів протягом трьох років, тоді як мережеві фахівці спостерігають підвищення ефективності на 30%, що призведе до економії 1,5 мільйона доларів протягом трьох років.

Галузеві програми управління мережами

Різні галузі стикаються з унікальними проблемами та вимогами до управління мережею, що робить індивідуальні підходи важливими для досягнення оптимальних результатів.

Галузь охорони здоров'я

Організаціям охорони здоров'я потрібне спеціалізоване управління мережею для підтримки:

– Відповідність HIPAA: забезпечення конфіденційності даних пацієнтів та дотримання нормативних вимог.

– Підтримка телемедицини: надійне з’єднання для дистанційних консультацій пацієнтів.

– Інтеграція медичних пристроїв: управління спеціалізованим медичним обладнанням та пристроями Інтернету речей.

– Надійність у надзвичайних ситуаціях: критично важливі вимоги до безвідмовної роботи для життєзабезпечувальних застосувань.

Фінансові послуги

Фінансові установи стикаються з суворими вимогами, зокрема:

– Відповідність нормативним вимогам: дотримання SOX, PCI-DSS та інших фінансових норм.

– Високочастотна торгівля: вимоги до наднизької затримки для торгових систем.

– Виявлення шахрайства: моніторинг підозрілої мережевої активності в режимі реального часу.

– Безперервність бізнесу: Мінімальна толерантність до перерв у наданні послуг.

Виробництво та логістика

Виробничі середовища потребують рішень для управління мережею, які враховують:

– Промисловий Інтернет речей: управління датчиками, робототехнікою та автоматизованими системами.

– Зв'язок ланцюга поставок: З'єднання складів, розподільчих центрів та виробничих потужностей.

– Операції в режимі реального часу: Підтримка виробничих процесів, чутливих до часу.

– Прогнозне обслуговування: Використання мережевих даних для прогнозування відмов обладнання.

Роздрібні операції

Роздрібні організації отримують вигоду від управління мережею завдяки:

– Надійність точок продажу: забезпечення працездатності систем обробки транзакцій.

– Wi-Fi для клієнтів: керування доступом гостей та підключенням клієнтів.

– Управління запасами: підтримка RFID та автоматизованих систем інвентаризації.

– Інтеграція електронної комерції: поєднання онлайн- та офлайн-роздрібних операцій.

Сучасні тенденції управління мережами, що формують 2026 рік

Ландшафт управління мережами швидко розвивається під впливом технологічного прогресу та змінних бізнес-вимог.

Мережеві операції на основі штучного інтелекту

Штучний інтелект революціонізує управління мережами завдяки:

– Прогнозна аналітика: прогнозування проблем мережі до їх виникнення.

– Автоматизоване виправлення: самовідновлювальні мережі, які автоматично вирішують проблеми.

– Інтелектуальна оптимізація: маршрутизація трафіку та розподіл ресурсів на основі штучного інтелекту.

– Виявлення аномалій: алгоритми машинного навчання, які виявляють незвичайну поведінку мережі.

Очікується, що ринок систем управління мережами зростатиме зі середньорічним темпом зростання (CAGR) на рівні 9,6% з 2025 по 2032 рік, значною мірою завдяки інтеграції штучного інтелекту та машинного навчання.

Хмарне мережеве управління

Організації все частіше впроваджують хмарні підходи, які пропонують:

– Масштабованість: Еластичне масштабування для задоволення зростаючих потреб мережі.

– Гнучкість: швидке розгортання та зміни конфігурації.

– Економічна ефективність: моделі ціноутворення «оплата за використання» та зниження витрат на інфраструктуру.

– Глобальна доступність: керування розподіленими мережами з будь-якого місця.

Ця тенденція узгоджується з ширшими стратегіями впровадження

хмарних обчислень.

Архітектура мережі нульової довіри

Підходи «нульової довіри» стають поширеними, зокрема:

– Безперервна перевірка: ніколи не довіряємо, завжди перевіряємо доступ користувачів та пристроїв.

– Мікросегментація: обмеження доступу до певних сегментів та ресурсів мережі.

– Безпека, орієнтована на ідентифікацію: зосередження уваги на ідентичності користувача та пристрою, а не на мережевому розташуванні.

– Динамічний контроль доступу: налаштування дозволів доступу на основі оцінки ризиків у режимі реального часу.

Мережа як послуга (NaaS)

Перехід до моделей NaaS забезпечує:

– Споживання на основі передплати: перехід від капітальних витрат до операційних витрат.

– Керовані послуги: аутсорсинг управління мережею спеціалізованим постачальникам.

– Швидке розгортання: Швидший вихід на ринок для нових локацій та послуг.

– Стабільний досвід: Стандартизовані мережеві послуги в усіх локаціях.

Інтеграція периферійних обчислень

Управління мережею адаптується для підтримки вимог периферійних обчислень:

– Розподілений моніторинг: управління продуктивністю на периферійних локаціях.

– Локальна обробка: зменшення затримки завдяки аналітиці на периферії.

– Автономні операції: Самокеровані граничні вузли з мінімальним централізованим наглядом.

– Інтеграція 5G: підтримка бездротових технологій наступного покоління.

Розрахунок рентабельності інвестицій: фінансовий вплив управління мережею

Розуміння рентабельності інвестицій (ROI) для ініціатив з управління мережею має вирішальне значення для забезпечення підтримки керівництва та виправдання інвестицій у технології.

.**Пряма економія коштів**

Операційна ефективність: Організації, які впроваджують комплексне управління мережею, зазвичай спостерігають зниження загальних операційних витрат на IT на 5-15% завдяки автоматизації та підвищенню ефективності.

Зменшення часу простою: Завдяки миттєвій окупності інвестицій від інструментів моніторингу мережі, компанії уникають значних витрат, пов'язаних з перебоями в роботі мережі, які можуть коливатися від тисяч до мільйонів доларів на годину залежно від організації.

Продуктивність персоналу: Мережеві фахівці відзначають підвищення ефективності на 30%, що дозволяє IT-командам зосередитися на стратегічних ініціативах, а не на реактивному вирішенні проблем.

Стратегічні переваги

Масштабованість: Сучасні платформи управління мережею дозволяють організаціям масштабувати операції без пропорційного збільшення управлінських витрат.

Відповідність: Автоматизований моніторинг та звітність щодо відповідності зменшують витрати та ризики, пов'язані з порушеннями нормативних вимог.

Сприяння інноваціям: Надійна мережева інфраструктура підтримує ініціативи цифрової трансформації та впровадження нових технологій.

Конкурентна перевага: Вища продуктивність мережі може виділити організації на конкурентних ринках.

Кількісна оцінка рентабельності інвестицій в управління мережею

Щоб розрахувати рентабельність інвестицій (ROI), організаціям слід

враховувати:

– Витрати на технології: ліцензування програмного забезпечення, обладнання та послуги з впровадження.

– Економія на операційній діяльності: зменшення потреб у персоналі, нижчі витрати на технічне обслуговування та підвищення ефективності.

– Зменшення ризиків: Уникнення витрат через перебої в роботі, порушення безпеки та порушення нормативних вимог.

– Стратегічна цінність: Розширені можливості, що відкривають нові бізнес-можливості.

Дослідження показують, що організації можуть досягти рентабельності інвестицій у розмірі 160% протягом трьох років завдяки комплексному впровадженню мережевого управління.

Найкращі практики впровадження управління мережею

Успішне управління мережею вимагає ретельного планування, належного впровадження та постійної оптимізації.

.Планування та оцінювання

Інвентаризація мережі: проведення комплексної оцінки існуючої мережевої інфраструктури, документування всіх пристроїв, з'єднань та конфігурацій.

Аналіз вимог: Визначення конкретних бізнес-вимог, цілей продуктивності та потреб у відповідності.

Аналіз розривів: Визначення відмінностей між поточними можливостями та бажаними результатами.

Узгодження інтересів зацікавлених сторін: Переконайтеся, що всі зацікавлені сторони розуміють цілі та критерії успіху.

Вибір технології

Оцінка платформи: Оцінка платформ управління мережею на основі масштабованості, функцій та можливостей інтеграції.

Оцінка постачальника: Оцініть стабільність постачальника, можливості

підтримки та довгострокові плани.

Підтвердження концепції: проведення пілотних впроваджень для перевірки можливостей платформи.

Планування інтеграції: забезпечення сумісності з існуючими системами та майбутніми технологічними ініціативами.

Стратегія впровадження

Поетапне розгортання: Поетапно впроваджуйте можливості управління мережею, щоб мінімізувати ризики та збої в роботі.

Навчальні програми: Забезпечення комплексного навчання IT-персоналу новим інструментам та процесам.

Документація: Ведіть детальну документацію щодо конфігурацій, процедур та посібників з усунення несправностей.

Управління змінами: Дотримуйтесь структурованих процесів управління змінами для всіх модифікацій мережі.

Безперервна оптимізація

Моніторинг ефективності: Регулярно перевіряйте ефективність управління мережею та визначайте можливості для покращення.

Удосконалення процесів: Постійне вдосконалення операційних процедур на основі досвіду та передового досвіду.

Оновлення технологій: Будьте в курсі оновлень платформи та випусків нових функцій.

Узгодження бізнесу: Регулярно переглядайте пріоритети управління мережею на основі змінних потреб бізнесу.

Структурна схема системи зображена на рисунку 3.1.

Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

Незалежно від об'єкта керування, бажано, щоб система керування виконувала ряд функцій, які визначені міжнародними стандартами, що узагальнюють досвід застосування систем керування в різних областях. Існують рекомендації ITU-T X.700 і близький до них стандарт ISO 7498-4, які ділять завдання системи керування на п'ять функціональних груп:

- керування конфігурацією мережі й іменуванням;
- обробка помилок;
- аналіз продуктивності й надійності;
- керування безпекою;
- облік роботи мережі.

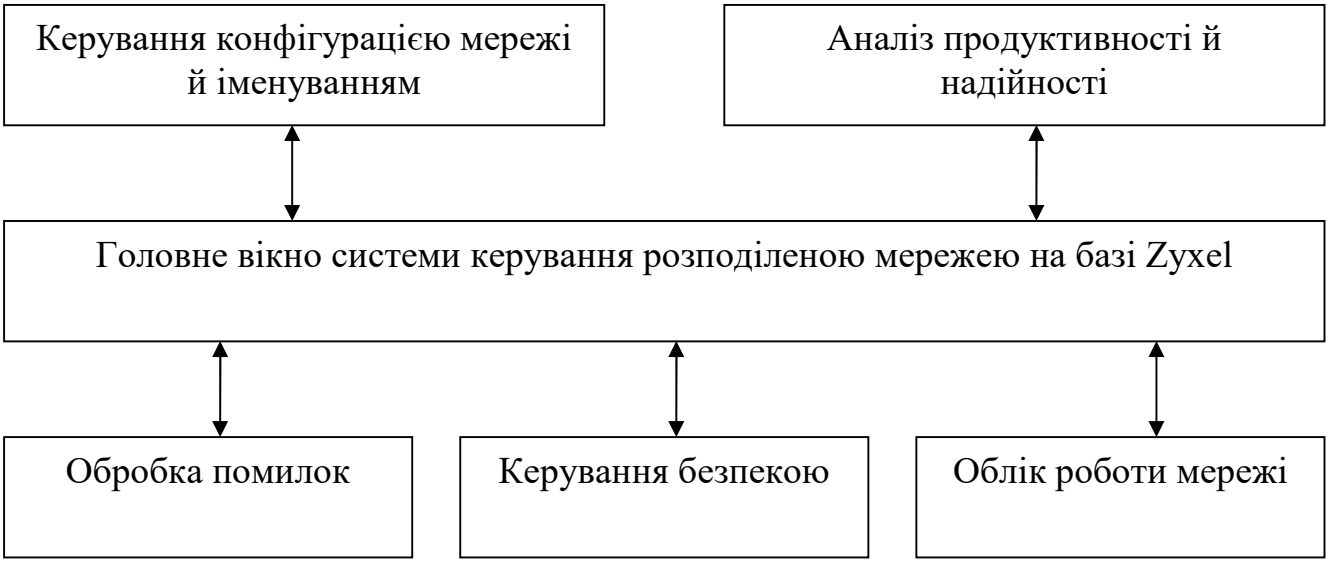

Рисунок 3.2 – Функціональна схема системи

Відповідно розроблювальне програмне забезпечення реалізує дані рекомендації. Розглянемо завдання цих функціональних областей керування стосовно до систем керування мережами.

Керування конфігурацією мережі й іменуванням (Configuration Management)

Ці завдання полягають у конфігуруванні параметрів як елементів мережі (Network Element, NE), так і мережі в цілому. Для елементів мережі, таких як маршрутизатори, мультиплексори й т.п., за допомогою цієї групи завдань визначаються мережеві адреси, ідентифікатори (імена), географічне положення та ін. Для мережі в цілому керування конфігурацією звичайно починається з побудови карти мережі, тобто відображенні реальних зв'язків між елементами мережі й зміні зв'язків між елементами мережі – утворення нових фізичних або логічних каналів, зміна таблиць комутації й маршрутизації.

Керування конфігурацією (як і інші завдання системи керування) можуть виконуватися в автоматичному, ручному або напівавтоматичному режимах. Наприклад, карта мережі може складатися автоматично, на підставі зондування реальної мережі пакетами-дослідниками, а може бути уведена оператором

системи керування вручну. Найчастіше застосовуються напівавтоматичні методи, коли автоматично отриману карту оператор підправляє вручну. Методи автоматичної побудови топологічної карти, як правило, є фірмовими розробками.

Більше складним завданням є налаштування комутаторів і маршрутизаторів на підтримку маршрутів і віртуальних шляхів між користувачами мережі. Погоджене ручне налаштування таблиць маршрутизації при повній або частковій відмові від використання протоколу маршрутизації (а в деяких глобальних мережах, наприклад, X.25, такого протоколу просто не існує) являє собою складне завдання. Багато систем керування мережею загального призначення її не виконують, але існують спеціалізовані системи конкретних виробників, наприклад, система NetSys компанії Cisco Systems, що вирішує її для маршуртизаторов цієї ж компанії.

Обробка помилок (Fault Management)

Ця група завдань включає виявлення, визначення й усунення наслідків збоїв і відмов у роботі мережі. На цьому рівні виконується не тільки реєстрація повідомлень про помилки, але і їхня фільтрація, маршрутизація й аналіз на основі деякої кореляційної моделі. Фільтрація дозволяє виділити з досить інтенсивного потоку повідомлень про помилки, що звичайно спостерігається у великій мережі, тільки важливі повідомлення, маршрутизація забезпечує їхню доставку потрібному елементу системи керування, а кореляційний аналіз дозволяє знайти причину, що породила потік взаємозалежних повідомлень (наприклад, обривши кабелю може бути причиною великої кількості повідомлень про неприступність мереж і серверів).

Усунення помилок може бути як автоматичним, так і напівавтоматичним. У першому випадку система безпосередньо управляє устаткуванням або програмними комплексами й обходить елемент, що відмовив, за рахунок резервних каналів і т.п. У напівавтоматичному режимі основні рішення й дії по усуненню несправності виконують люди, а система керування тільки допомагає в організації цього процесу – оформляє квитанції на виконання робіт і

відслідковує їхнє поетапне виконання (подібно системам групової роботи). У цій групі завдань іноді виділяють підгрупу завдань керування проблемами, маючи на увазі під проблемою складну ситуацію, що вимагає для дозволу обов'язкового залучення фахівців з обслуговування мережі.

Аналіз продуктивності й надійності (Performance Management)

Завдання цієї групи пов'язані з оцінкою на основі накопиченої статистичної інформації таких параметрів, як час реакції системи, пропускна здатність реального або віртуального каналу зв'язку між двома кінцевим і абонентами мережі, інтенсивність трафіку в окремих сегментах і каналах мережі, імовірність перекручування даних при їхній передачі через мережу, а також коефіцієнт готовності мережі або її певної транспортної служби.

Функції аналізу продуктивності й надійності мережі потрібні як для оперативного керування мережею, так і для планування розвитку мережі.

Результати аналізу продуктивності й надійності дозволяють контролювати угода про рівень обслуговування (Service Level Agreement, SLA), що міститься між користувачем мережі і її адміністраторами (або компанією, що продає послуги). Звичайно в SLA обмовляються такі параметри надійності, як коефіцієнт готовності служби протягом року й місяця, максимальний час усунення відмови, а також параметри продуктивності, наприклад, середньої й максимальної пропускної здатності при з'єднанні двох точок підключення користувальницького устаткування, час реакції мережі (якщо інформаційна служба, для якої визначається час реакції, підтримується усередині мережі), максимальна затримка пакетів при передачі через мережу (якщо мережа використовується тільки як транзитний транспорт). Без засобів аналізу продуктивності й надійності постачальник послуг публічної мережі або відділ інформаційних технологій підприємства не зможе не проконтролювати, ні тим більше забезпечити потрібний рівень обслуговування для кінцевих користувачів мережі.

Керування безпекою (Security Management)

Завдання цієї групи містять у собі контроль доступу до ресурсів мережі (даним і устаткуванню) і збереження цілісності даних при їхньому зберіганні й передачі через мережу. Базовими елементами керування безпекою є процедури автентифікації користувачів, призначення й перевірка прав доступу до ресурсів мережі, розподіл і підтримка ключів шифрування, керування повноваженнями й т.п. Часто функції цієї групи не включаються в системи керування мережами, а реалізуються або у вигляді спеціальних продуктів (наприклад, системи автентифікації й авторизації Kerberos, різних захисних екранів, систем шифрування даних), або входять до складу операційних систем і системних застосунків.

Облік роботи мережі (Accounting Management)

Завдання цієї групи займаються реєстрацією часу використання різних ресурсів мережі – пристроїв, каналів і транспортних служб. Ці завдання мають справа з такими поняттями, як час використання служби й плата за ресурси – billing. Через специфічний характер оплати послуг у різних постачальників і різних форм угоди про рівень послуг, ця група функцій звичайно не включається в комерційні системи й платформи управління типу HP Open View, а реалізується в замовлених системах, розроблювальних для конкретного замовника.

Модель керування OSI не робить розходжень між керованими об'єктами – каналами, сегментами локальних мереж, мостами, комутаторами й маршрутизаторами, модемами й мультиплексорами, апаратним і програмним забезпеченням комп'ютерів, СУБД. Всі ці об'єкти керування входять у загальне поняття “система”, і керована система взаємодіє з керуючою системою по відкритих протоколах OSI.

Розгортаємо мережу

Якщо у вас уже є обліковий запис системи керування розподіленою мережею на базі Zyxel, ним можна скористатися для реєстрації на сайті системи керування розподіленою мережею на базі Zyxel. Спочатку треба створити

організацію (у нашому випадку це вже було зроблено). Під одним обліковим записом дозволяється реєструвати кілька організацій, що зручно для сервіс-провайдерів, але без необхідності краще їх не плодити, оскільки цю операцію скасувати не можна. Створення окремої філії (площадка або сайт – у термінології системи керування розподіленою мережею на базі Zyxel) для конкретної організації здійснюється в парі кліків – досить указати ім'я нової площадки й вартовий пояс.

Устаткування можна додати відразу при створенні сайту, указавши MAC-адресу й серійний номер. Тим, хто не знає, де їх знайти, допоможуть підказки – як і вся процедура розгортання мережі, вони явно розраховані на недосвідчених користувачів мережевого устаткування. Конфігурацію устаткування можна клонувати шляхом копіювання параметрів кожної з існуючих площадок. Таким чином, якщо офіси мають типове оснащення, налаштування досить зробити один раз. Зробити це можна до фізичної установки пристроїв за умови реєстрації устаткування. Для прискорення процесу реєстрації відразу декількох пристроїв можна скористатися шаблонами, а вже наявні пристрої швидко перереєструвати – зняти з реєстрації на одному сайті й додати на іншій.

Реєстрацію устаткування можна відкласти й зробити це пізніше за допомогою мобільного додатка – для когось такий варіант простіше. Після входу в додаток треба вибрати організацію й офіс, клацнути на значок «додати» («+») і відсканувати QR-код на коробці або пристрої. Пристрій автоматично додасться до зазначеного сайту. QR-код – невеликого розміру, і, щоб піймати його у фокус, потрібна деяка вправність – увести MAC-адресу й серійний номер вручну було б ненабагато довше. Але зате відсканувати код може будь-який користувач на віддаленій площадці, якщо до відправлення устаткування не було зареєстровано.

Крім реєстрації, додаток можна використовувати для перегляду статусу пристроїв (online/offline) і рівня їхнього завантаження. Поки устаткування не підключене фізично, його статус відображається як offline; після підключення

треба небагато почекати (як затверджують в Zyxel, у середньому близько 3-5 хв), поки він зміниться на online.

У нашій тестовій конфігурації було встановлено три системи керування розподіленою мережею на базі Zyxel-сумісні пристрої: шлюз NSG200, комутатор NSW 100-10P і точка доступу NWA 1123-AC HD. Online-статусу точки доступу довелося чекати більше десяти хвилин. Використовувана модель передбачає роботу у двох режимах – автономному (звичайному) і хмарному. За замовчуванням передбачається перший, тому зміна прошивання (її завантаження із хмари, установка й перезавантаження ТД) зайняла якийсь час.

Втім, 10 хв первісного очікування – не така більша плата за можливість віддаленого адміністрування в порівнянні з необхідністю виїзду фахівця на місце. До того ж при наступному відключенні й включенні пристрої ставали доступними набагато швидше. (Зміна статусу відбувається з деякою затримкою й впливає тільки на виконання моніторингу й адміністрування пристроїв, але не відбивається на їхньому функціонуванні – так, до точки доступу можна було підключитися ще до того, як її статус змінився на online.)

Функціональність мобільного додатка обмежується реєстрацією пристроїв і переглядом статусу устаткування, тому для повноцінного керування й моніторингу прийдеться повернутися в Command Center на сайті системи керування розподіленою мережею на базі Zyxel.

Огляд інтерфейсу системи керування розподіленою мережею на базі Zyxel

Портал системи керування розподіленою мережею на базі Zyxel має простий і приємний дизайн. Адміністратор може переглянути зведену інформацію з кожної конкретної організації (якщо їх декілька) або площадці й більш детально по кожному виді устаткування (точка доступу, комутатор, шлюз). Керуюча панель дозволяє одним поглядом оцінити загальний стан і завантаженість устаткування й з'єднань на конкретній площадці. Тут же відображається інформація про підключених клієнтів, споживаному ними трафік

й використовуваних додатках (статистика по трафіку застосунків доступна тільки при наявності ліцензії на IDP), а також про виділений бюджет PoE.

Меню містить п'ять основних пунктів:

– організація;

– сайт;

– ТД;

– комутатор;

– шлюз.

Кожний з них, у свою чергу, ділиться на два підменю – моніторингу й налаштування. У режимі моніторингу можна переглядати статистику не тільки на сучасний момент, але й за різні попередні періоди аж до одного року. Для визначення місцезнаходження пристрою можна подивитися його розташування на карті Google (якщо прив'язка по IP-адресі була здійснена некоректно, його можна розташувати в потрібнім місці вручну). Для більше точного позиціонування передбачена можливість додавання планів будинків.

Підменю конфігурації надає множину опцій налаштування. Так, наприклад, для точок доступу можна налаштувати до 8 SSID, задати розклад і прив'язати до кожної SSID портал реєстрації (captive portal). Для автентифікації на точці доступу й порталі, крім стандартних WPA2 і WPA2 Enterprise, пропонується задіяти різні механізми, у тому числі систему керування розподіленою мережею на базі Zyxel Cloud Authentication (це хмарна база користувачів зі своїми обліковими записами).

Точки доступу підтримують режим балансування точок навантаження. Ця опція корисна, коли підключається багато клієнтів, число яких можна обмежити – наприклад, від 1 до 127 (на один радіомодуль). При відключеній опції дисоціації й у випадку перевищення заданого ліміту клієнт буде спрямований на іншу точку доступу. При включеної – відбудеться видалення неактивних клієнтів (залежно від періоду їхньої неактивності) і клієнтів з низьким рівнем сигналу.

На комутаторах налаштувань менше. Для кожного порту можна налаштувати підтримувані VLAN, режим автоузгодженняя швидкості каналу, контроль широкомовних штормів, розклад PoE і т.д. Загальний бюджет PoE розрахований на живлення відповідних пристроїв, але може бути розподілений між устаткуванням PoE і PoE+. Порт можна відключити/включити для перезавантаження підключеного пристрою. Разом з тим комутатори підтримують і не самий необхідний для них функціонал: фільтрацію по IP-адресах і автентифікацію клієнтів на сервері RADIUS.

Найбільш широкий вибір налаштувань доступний для шлюзу. Крім системи керування розподіленою мережею на базі Zyxel, пристроями можна управляти й через Web-інтерфейс (це може придатися у випадку неприступності сервісу системи керування розподіленою мережею на базі Zyxel), але в ньому підтримується обмежений набір налаштувань.

Патруль застосунків

Шлюз працює у двох режимах – с підтримкою NAT і як звичайний маршрутизатор. Локальні порти шлюзу можна розбити на дві групи й у такий спосіб створити дві підмережі – кожну зі своїми правилами. Контроль доступу в Інтернет для кожної з локальних підмереж настроюється аналогічно точкам доступу. Портал реєстрації включається й на самому шлюзі. Для автентифікації можна використовувати системи керування розподіленою мережею на базі Zyxel Cloud Authentication, RADIUS-сервер або Active Directory.

У хмару направляється тільки службова інформація, а користувальницький трафік передається прямо по місцю призначення. До того ж системи керування розподіленою мережею на базі Zyxel дозволяє легко й просто створити VPN між сайтами, так що про захищеність користувальницького трафіку турбуватися не доводиться. VPN підтримуються на всіх шлюзах, але в кожної моделі свій ліміт на кількість каналів: 10 у випадку NSG50 і 200 для NSG200.

Можливі дві топології VPN – site-to-site (з'єднання прямо) і hub-and-spoke (через центральний офіс). Для створення VPN досить указати шлюзи, між якими будуть створюватися VPN, а також задати, що підключаються через нього порт WAN, зовнішній IP-адреса шлюзу й внутрішню локальну підмережу(-і). Після цього тунелі будуть створені автоматично (шлюзи, що не підтримують системи керування розподіленою мережею на базі Zyxel, можна додати вручну).

Шлюз може виконувати функції міжмережевого екрана. Поки можливе налаштування тільки вихідних правил. Функція Application Patrol («патруль застосунків») сканує трафік аж до сьомого рівня моделі OSI і перевіряє його на відповідність різним сигнатурам – наприклад, можна заблокувати певні мессенджери (Telegram не блокується, сигнатури поки не знайшли), клієнтів однорангових мереж, торренти. Усього доступно близько 3000 сигнатур, тобто шлюз здатний розпізнавати до 3000 застосунків.

Крім того, на шлюзі є система виявлення й запобігання вторгнень (для її використання потрібна окрема ліцензія). Її можна включити в пункті меню Security Filtering у режимі моніторингу (detection) і/або блокування (prevention).

Розмежування прав

Система керування розподіленою мережею на базі Zyxel дозволяє здійснювати детальне розмежування прав адміністраторів, яким можна призначати різні повноваження – для всієї організації або тільки для конкретного сайту, з повними правами або тільки переглядом. Окрема роль (Guest Ambassador) передбачена для налаштування гостьових облікових записів. Крім цього, можна надати інсталятору права для реєстрації устаткування. Всі дії адміністраторів (зроблені ними зміни в конфігурації) протоколюються.

При віддаленому вході система керування розподіленою мережею на базі Zyxel пропонує налаштувати двофакторну автентифікацію. У якості другого контрольного пункту можна використовувати додаток Google Authenticator або електронну пошту, на яку приходить посилання для верифікації.

За замовчуванням всі попередження відключені, їх треба налаштувати вручну. Попередження будуть відсилатися по зазначених адресах електронної пошти при настанні таких подій, як неприступність пристроїв протягом заданого інтервалу часу (від 5 хв до 1 ч), установленні або розриві каналу VPN, а також при будь-яких змінах конфігурації.

Сесія не завершується автоматично. Це з'ясувалося, коли я, не вийшовши з панелі керування, знову відкрив її через кілька годин. Час неактивності, після якого сесія автоматично переривається, можна налаштувати, але за замовчуванням ця опція відключена.

Хмарна незалежність

Система керування розподіленою мережею на базі Zyxel призначений для організацій з декількома невеликими офісами. Це рішення розроблялося розраховуючи на те, що підключенням устаткування будуть займатися не мережеві адміністратори, а звичайні користувачі. Крім того, підключення однотипних офісів значно спрощується завдяки можливості клонування налаштувань.

Звичайно, для керування мережею знадобиться досвідчений адміністратор, але не потрібно тримати окремого фахівця в кожному офісі. Zyxel не пропонує спеціальних курсів по керуванню мережею за допомогою системи керування розподіленою мережею на базі Zyxel: будь-який адміністратор, що знає англійську мову на базовому рівні, може без проблем розібратися в налаштуваннях, однак рекомендується пройти курси Zyxel Certified Network Engineer. Крім того, завжди можна поставити запитання на форумі системи керування розподіленою мережею на базі Zyxel, де відповідь дається досить оперативно.

Для використання переваг хмарного керування не обов'язково встановлювати всі три типи пристроїв у кожному офісі. Наприклад, можна обмежитися бездротовими точками доступу, причому їх не обов'язково

підключати до Інтернету прямо – вони можуть розташовуватися за MME або маршрутизатором.

У випадку системи керування розподіленою мережею на базі Zyxel наявність постійного доступу в хмару не настільки критично, як для інших хмарних сервісів. Мережа буде працювати в кожному разі, але не можна здійснювати налаштування й міняти конфігурацію устаткування. (Втім, як відзначалося, підтримується віддалений доступ через Web-інтерфейс із обмеженими можливостями керування).

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей.

Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

При детальному її розгляді можна побачити як саме проходе взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

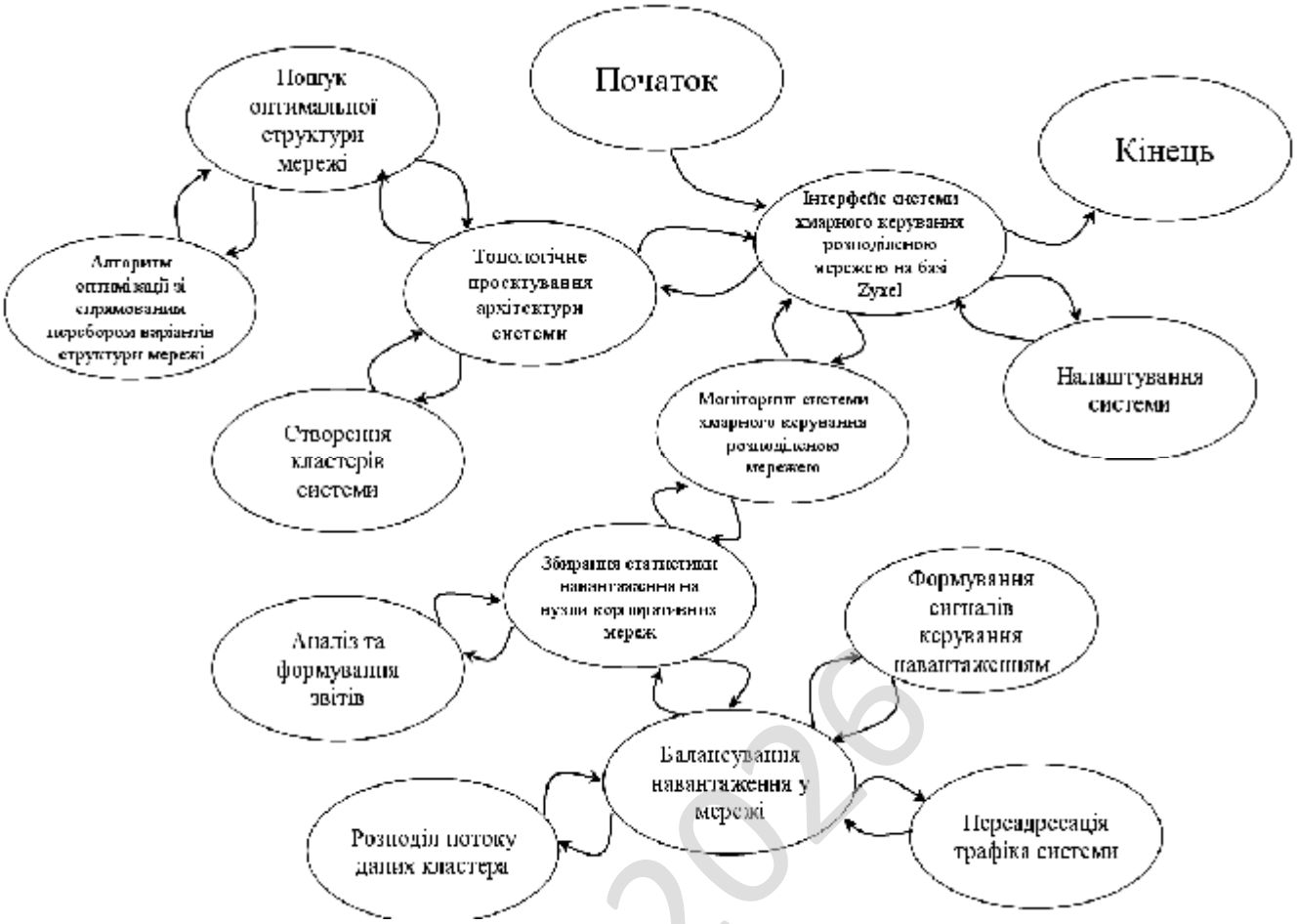

Рисунок 3.3 – Діаграма взаємодії процесів

Діаграми потоків даних містять чотири типи елементів:

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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

– Сховища даних (репозиторії).

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

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень. 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 підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання. Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз’яснення який саме тип блок-схем використовується.

Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об’єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

Було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу).

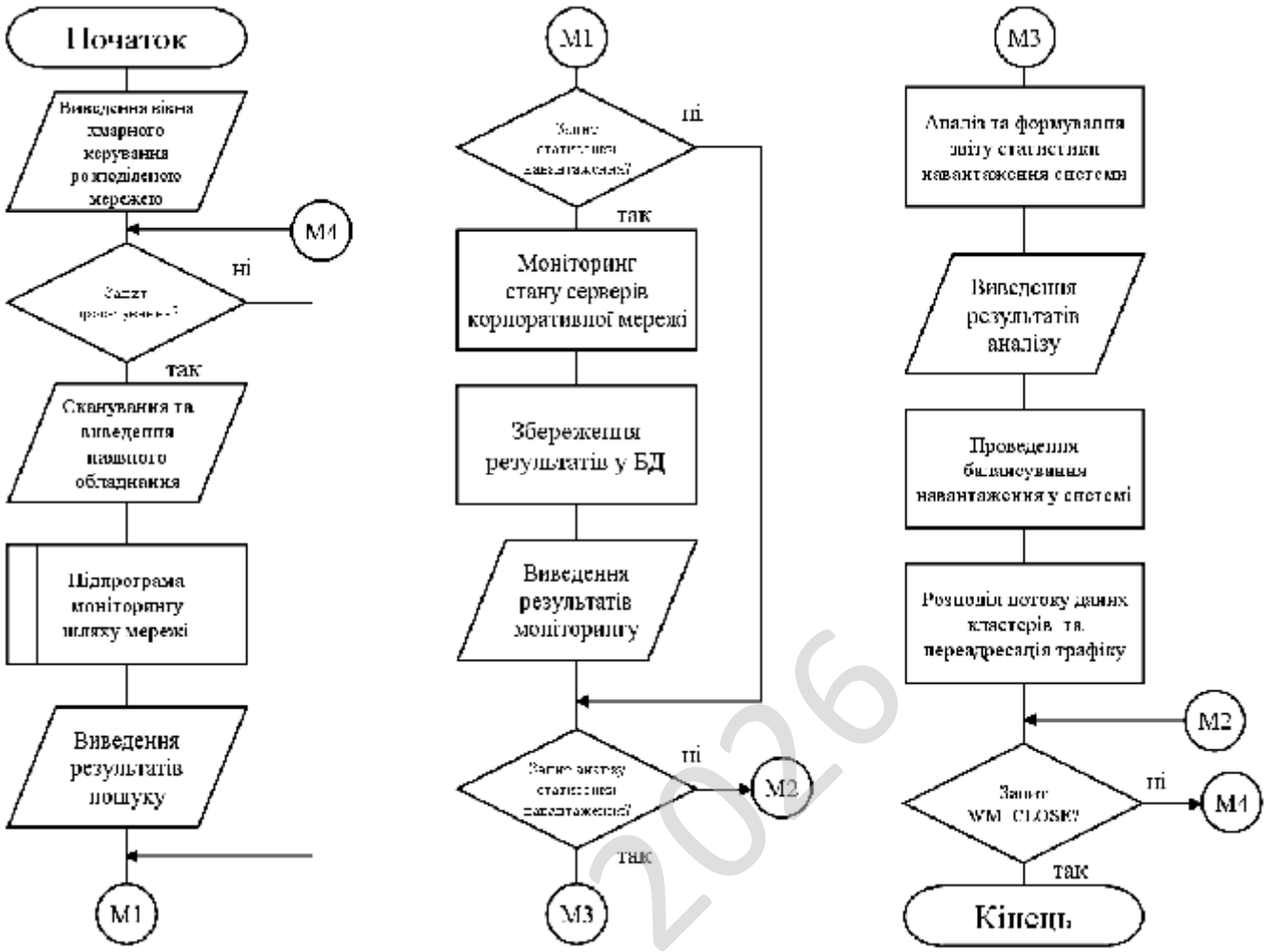

Рисунок 4.1 – Блок-схема основної програми

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

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії

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

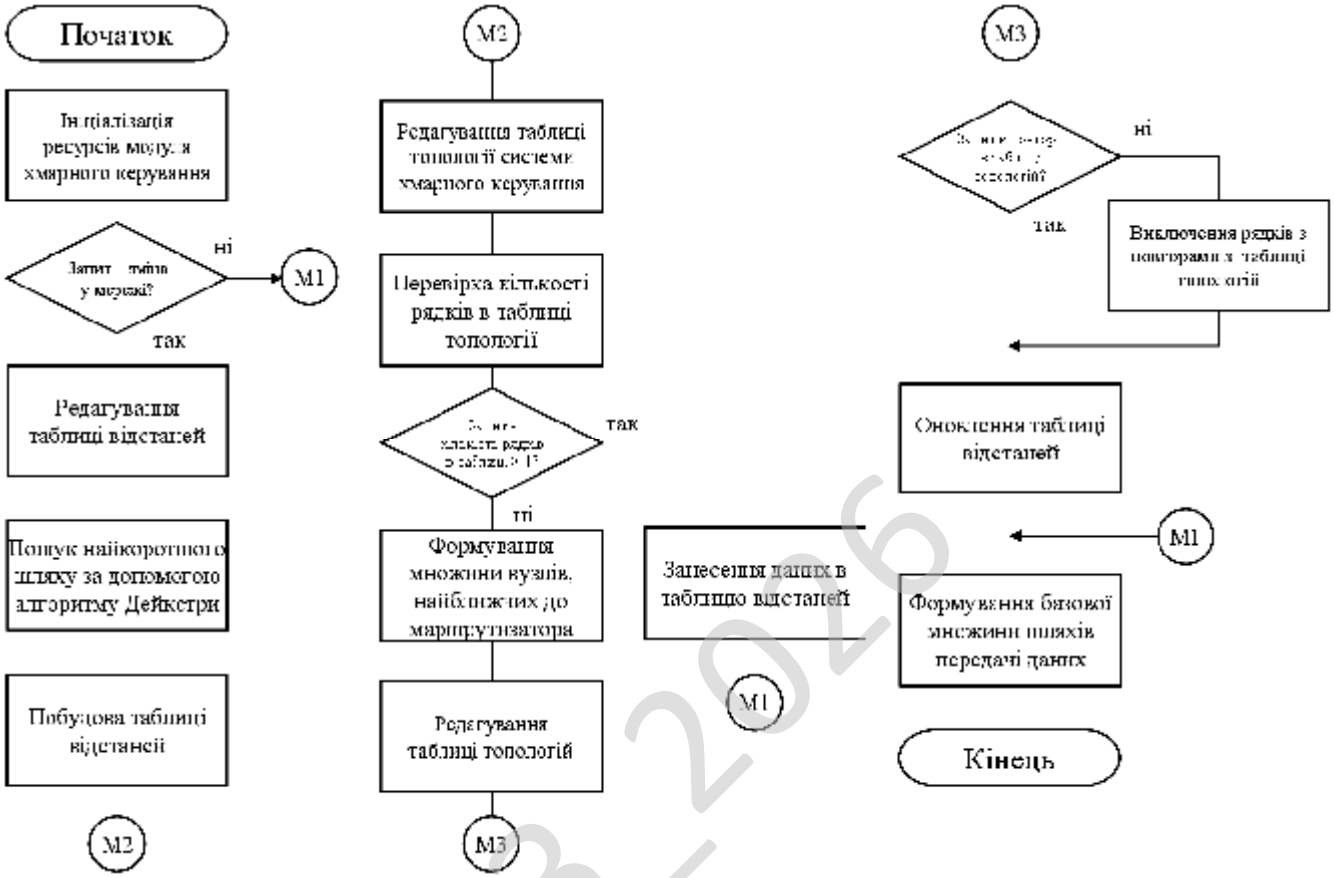

Рисунок 4.2 – Блок-схема роботи підпрограми

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

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

– 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 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 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).

Розглянемо визначення API. Це прикладни́й програ́мний інтерфе́йс (Application Programming Interface, API) – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

Спрощено - це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для малювання вікна чи іконок на екрані. API є абстрактним поняттям – програмне забезпечення, що пропонує деякий API, часто називають реалізацією (implementation) даного API. У багатьох випадках API є частиною набору розробки програмного забезпечення, водночас, набір розробки може включати як API, так і інші інструменти/апаратне забезпечення, отже ці два терміни не є взаємозамінювані.

Високорівневі API часто програють у гнучкості. Виконання деяких функцій нижчого рівня стає набагато складнішим, або навіть неможливим.

В об'єктно-орієнтованих мовах, прикладний програмний інтерфейс зазвичай включає в себе опис набору визначень класу, з набором форм поведінки, пов'язаних з цими класами. Це абстрактне поняття пов'язане з реальними функціями, які надані або надаватимуться, класами, які реалізуються в методах класу.

Прикладний програмний інтерфейс в даному випадку можна розглядати як сукупність всіх методів, які публічно доступні в класах (зазвичай званий інтерфейс класу). Це означає, що прикладний програмний інтерфейс вказує методи, за допомогою яких взаємодіє з об'єктами, отриманими з визначень класів і обробляє їх.

У більш загальному плані можна визначити Прикладний Програмний Інтерфейс як сукупність усіх видів об'єктів, які можна вивести з визначення класу, і пов'язаних з ними можливих варіантів поведінки.

Наприклад: клас, що представляє Stack, може просто виставити публічно два методи Push() (для додавання нового елемента в стек) і Pop() (для вилучення останнього пункту, ідеально розташований на вершині стека).

У цьому випадку Прикладний Програмний Інтерфейс може бути інтерпретованим як два методи pop() і push(), або, більш широко, використовується варіант, коли можна використовувати елемент типу Stack, який реалізує поведінку стека, надаючи йому можливість для додавання / видалення елементів з вершини. Друга інтерпретація видається більш доречною в дусі об'єктно-орієнтованого підходу.

Якість документації, пов'язаної з Прикладним Програмним Інтерфейсом, є часто ключовим фактором, що визначає його успішність з точки зору простоти використання.

ППІ, як правило, пов'язаний із бібліотеками програмного забезпечення: ППІ описує і вказує очікувану поведінку в той час, як бібліотека є фактичною реалізацією даного набору правил. Один ППІ може мати декілька реалізацій (або

жодної, будучи абстрактним) у вигляді різних бібліотек, які мають такий же інтерфейс.

Прикладний програмний інтерфейс також може бути пов'язаним з платформами програмування: платформа може бути заснована на кількох бібліотеках реалізує декілька інтерфейсів ППІ, але на відміну від звичайного використання ППІ, доступ до поведінки вбудований в платформу опосередкований шляхом розширення його змісту новими класами і вставлений в саму платформу. Крім того, загальний потік управління програми може бути під контролем абонента.

Прикладний програмний інтерфейс може бути також реалізацією протоколу.

Коли ППІ реалізує протокол, він може бути заснованим на проксі-методах віддалених викликів, що засновані на протоколі зв'язку. Роль ППІ може заключатися саме в тому, щоб приховати деталі транспортного протоколу. Наприклад: RMI є ППІ, який реалізує протокол або JRMP IIOP як RMI-IIOP.

Протоколи, як правило, розподіляються між різними технологіями і зазвичай дозволяють різним технологіям обмінюватися інформацією, діючи як абстракція між двома світами. ППІ, як правило, є специфічним для конкретної технології: звідси, інтерфейси даної мови не можуть бути використані на інших мовах, якщо виклики функції не будуть перетворені з конкретної адаптації бібліотеки.

Деякі мови, серед яких такі, що працюють на віртуальних машинах (наприклад: мови, сумісні з NET CLI середовища CLR і JVM сумісних мов у віртуальній машині Java) можуть ділитися програмними інтерфейсами.

У цьому випадку віртуальна машина дозволяє мові взаємодії завдяки спільному знаменнику віртуальної машини, що абстрагується від конкретної мови, використовувати проміжний байт-код і його мову.

При використанні прикладного програмного інтерфейсу в контексті веб-розробки, як правило, ППІ визначається набором повідомлень запиту HTTP,

також визначається структура повідомлень-відповідей, зазвичай у розширенні мови розмітки XML або в форматі об'єктного запису JavaScript (JSON). У той час як прикладний програмний інтерфейс у Web історично був практично синонімом для веб-служби, останнім часом тенденція змінилась (так званий Web 2.0) на відхід від Simple Object Access Protocol (SOAP) на основі веб-сервісів і сервіс-орієнтованої архітектури (SOA) на більш прямі передачі репрезентативного стану (REST) стилів веб-ресурсів та ресурсів-орієнтованої архітектури (ROA).

Частина цієї тенденції пов'язана з рухом Семантичного веб-ресурсу до Опису Платформ (RDF), Концепції розвитку веб-технологій інженерних онтологій. Прикладні програмні інтерфейси у Web, що дозволяють комбінувати декількома прикладними програмними інтерфейсами в нові додатки називають гібридними.

NetBIOS (Network Basic Input/Output System) – протокол для роботи в локальних мережах на персональних ЕОМ типу IBM/PC, розроблений у вигляді інтерфейсу, який не залежить від фірми–виробника. Був розроблений фірмою Sytek Corporation за замовленням IBM в 1983 році. Він включає в себе інтерфейс сеансового рівня (англ. NetBIOS interface), в якості транспортних протоколів використовує TCP і UDP.

Особливістю NetBIOS є можливість його роботи поверх різних протоколів, найпоширенішими/відомими з яких є NetBEUI, IPX і стек протоколів TCP/IP; причому якщо старі версії Windows орієнтувалися на більш легкі в реалізації і менш ресурсомісткі NetBEUI і IPX, то сучасні Windows орієнтуються на TCP/IP.

При використанні NetBEUI і IPX NetBIOS сам забезпечує надійність доставки даних (функціональність SPX не використовувати), а при використанні TCP/IP надійність доставки забезпечує TCP, за що удостоївся окремого імені «NBT».

Інтерфейс NetBIOS являє собою типовий інтерфейс взаємодії програм (API) для забезпечення мережевих операцій введення–виведення і управління транспортним протоколом.

Програми, що використовують NetBIOS API інтерфейс, можуть працювати тільки при наявності протоколу, що допускає використання такого інтерфейсу.

NetBIOS також визначає протокол, що функціонує на сеансовому/транспортному рівнях моделі OSI. Цей протокол використовується протоколами нижчих рівнів, такими як NBFP (NetBEUI) і NetBT для виконання мережевих запитів вводу–виводу і операцій, описаних в стандартному інтерфейсному наборі команд NetBIOS.

Тобто, NetBIOS сам не підтримує виконання файлових операцій. Ця функція покладається на протоколи нижчих рівнів, а сам NetBIOS забезпечує тільки зв'язок з цими протоколами і NetBIOS API–інтерфейс.

Пакет NETBIOS створений для використання групою ЕОМ. Підтримує як режим сесій (робота через з'єднання), так і режим дейтаграм (без встановлення з'єднання). 16–символьні імена об'єктів в NETBIOS розподіляються динамічно.

NETBIOS має власну DNS, яка може взаємодіяти з інтернетівський. Ім'я об'єкта при роботі з NETBIOS не може починатися з символу *.

Додатки можуть знайти через NETBIOS потрібні їм ресурси, встановити зв'язок і послати або отримати інформацію. NETBIOS використовує для служби імен порт 137, для служби дейтаграм – порт 138, а для сесій – порт 139. Будь–яка сесія починається з NETBIOS–запиту, завдання IP-адреси і визначення TCP–порту віддаленого об'єкта, далі йде обмін NETBIOS–повідомленнями, після чого сесія закривається.

Сесія здійснює обмін інформацією між двома NETBIOS – додатками. Довжина повідомлення лежить в межах від 0 до 131 071 байт. Припустимо одночасне встановлення декількох сесій між двома об'єктами.

При організації IP-транспорту через NETBIOS IP-дейтаграмма вкладається в NETBIOS-пакет. Інформаційний обмін відбувається в цьому випадку без встановлення зв'язку між об'єктами. Імена NETBIOS повинні містити в собі IP-адреси. Так, частина NETBIOS–адреси може мати вигляд IP.XX.XX. XX. XX, де IP вказує на тип операції (IP через Netbios), а XX. XX. XX.XX – IP-адреса.

Система NETBIOS має власну систему команд (call, listen, hang up, send, receive, session status, reset, cancel, adapter status, unlink, remote program load) і примітивів для роботи з дейтаграммами (send datagram, send broadcast datagram, receive datagram , receive broadcast datagram).

Всі кінцеві вузли NETBIOS діляться на три типи:

– широкомовні («b») вузли;

– вузли точка–точка («p»);

– вузли змішаного типу («m»).

IP-адреса може асоціюватися з одним із зазначених типів. B–вузли встановлюють зв'язок зі своїм партнером за допомогою широкомовних запитів. P– і M–вузли для цієї мети використовують netbios сервер імен (NBNS) і сервер розподілу дейтаграм (NBDD).

NetBIOS забезпечує:

– реєстрацію і перевірку мережевих імен;

– встановлення і розрив з'єднань;

– зв'язок з підтвердженням доставки інформації;

– зв'язок без підтвердження доставки інформації;

– підтримку управління і моніторингу драйвера і мережевої карти.

Розглянемо формат що використовується – **JSON** (JavaScript Object Notation, укр. запис об'єктів JavaScript, вимовляється джейсон) – це текстовий формат обміну даними між комп'ютерами.

JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати об'єкти та інші структури даних. Цей формат головним чином використовується для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією). Розробив і популяризував формат Дуглас Крокфорд.

JSON знайшов своє головне призначення у написанні веб-програм, а саме при використанні технології AJAX. JSON, що використовується в AJAX, виступає як заміна XML (використовується в AJAX) під час асинхронної передачі

структурованої інформації між клієнтом та сервером. При цьому перевагою JSON перед XML є те, що він дозволяє складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript в об'єкти.

JSON з'явився через необхідність обміну даними з сервером у реальному часі без використання плагінів для браузерів, flash-додатків або Java-аплетів, які використовувались скрізь на початку 2000-х років. Дуглас Крокфорд був тим, хто активно просував новий на той час формат. Він з колегами хотів створити технологію, яка використовувала б можливості звичайного браузера давала б веб-розробникам можливість створювати веб-додатки із постійним двостороннім зв'язком із веб-сервером.

JSON вперше був використаний в проекті в Communities.com для Cartoon Network, він дозволяв обмінюватись повідомленнями і одночасно маніпулювати DHTML-елементами.

Веб-сайт JSON.org було запущено 2002 року. У грудні 2005-го року Yahoo! почав переводити деякі зі своїх веб-сервісів на роботу з JSON. Google взялася до роботи з технологією у своєму веб-протоколі GData у грудні 2006-го року.

Використання

За рахунок своєї лаконічності в порівнянні з XML, формат JSON може бути більш придатним для серіалізації складних структур.

Якщо говорити про веб-застосунки, в такому ключі він доречний в задачах обміну даними як між браузером і сервером (AJAX), так і між самими серверами (програмні HTTP-інтерфейси).

Формат JSON так само добре підходить для зберігання складних динамічних структур в реляційних базах даних або файловому кеші.

Приклад використання JSON (JavaScript)

```
const ajaxData = '{"name": "wiki", "fname": "pedia", "rates": [1,4,5,6]}'
const ajaxObj = JSON.parse(ajaxData)
console.log(`${ajaxObj.name}_${ajaxObj.rates[2]}`) /* Виведе "wiki_5"*/
```

Синтаксис

JSON будується на двох структурах:

1. Набір пар назва/значення. У різних мовах це реалізовано як об'єкт, запис, структура, словник, хеш-таблиця, список з ключем або асоціативним масивом.

2. Впорядкований список значень. У багатьох мовах це реалізовано як масив, вектор, список, або послідовність.

Це універсальні структури даних. Теоретично всі сучасні мови програмування підтримують їх у тій чи іншій формі. Оскільки JSON використовується для обміну даними між різними мовами програмування, то є сенс будувати його на цих структурах.

У JSON використовуються такі їхні форми:

1. Об'єкт – це послідовність пар назва/значення. Об'єкт починається з символу { і закінчується символом } . Кожне значення слідує за : і пари назва/значення відділяються комами.

2. Масив – це послідовність значень. Масив починається символом [і закінчується символом]. Значення відділяються комами.

3. Значення може бути рядком в подвійних лапках, або числом, або логічними true чи false, або null, або об'єктом, або масивом. Ці структури можуть бути вкладені одна в одну.

4. Рядок – це послідовність з нуля або більше символів юнікода, обмежена подвійними лапками, з використанням escape-послідовностей, що починаються зі зворотної косої риски \. Символи представляються простим рядком. Тип Рядок (String) дуже схожий на String в мовах C і Java. Число теж дуже схоже на C- або Java-число, за винятком того, що вісімкові та шістнадцяткові формати не використовуються. Пропуски можуть бути вставлені між будь-якими двома лексемами.

Наступний приклад показує JSON представлення об'єкта, що описує людину. У об'єкті є рядкові поля імені і прізвища, об'єкт, що описує адресу, і масив, що містить список телефонів.

```
{
   "firstName": "Іван",
   "lastName": "Коваленко",
   "address": {
       "streetAddress": "пр. Університетський 8, каб.502",
       "city": "Кропивницький",
       "postalCode": 21000
   },
   "phoneNumbers": [
       "044 123-1234",
       "050 123-4567"
   ]
}
```

Використання JSON в AJAX

Наступний фрагмент коду JavaScript показує, як клієнт може використати XMLHttpRequest для запиту об'єктів в форматі JSON з серверу. (Серверна частина коду пропущена, вона просто повертає на запит URL рядок у JSON форматі).

Треба відзначити, що тут використання XMLHttpRequest не є кросс-браузерним (за деталями звертайтеся на сторінку XMLHttpRequest), і код зазнаватиме незначних модифікацій в різних версіях оглядачів Internet Explorer, Opera, Safari або Mozilla. Використання запиту XMLHttpRequest обмежено правилом одного джерела (same origin policy): URL, що відповідає на запит, має посилатися на той же сайт, що обслуговує сторінку, що ініціювала запит.

Оглядачі можуть також використовувати тег <iframe> для асинхронного запиту JSON-даних в кросс-браузерному варіанті, або використати просте перенаправлення

```
<form action="url_to_cgi_script" target="name_of_hidden_iframe">.
```

Такий підхід був поширений до приходу популярного нині запиту XMLHttpRequest.

Динамічний тег <script> також можна використати для підвантаження JSON-даних. Ця техніка можлива, щоб обійти надто суворе правило одного

джерела, але вона не є безпечною. Запит JSONRequest пропонується, як безпечніша альтернатива.

Питання безпеки

Хоча JSON призначений для передачі даних в серіалізованому вигляді, його синтаксис відповідає синтаксису JavaScript і це створює низку проблем безпеки. Часто для обробки даних, отриманих від зовнішнього джерела у форматі JSON, до них застосовується функція eval() без якої-небудь попередньої перевірки.

JavaScript eval()

Оскільки JSON представляється синтаксично правильним фрагментом коду JavaScript, природним способом розбору JSON-даних в JavaScript-програмі є використання вбудованої в JavaScript функції eval(), яка призначена для обчислення JavaScript-виразів. При цьому підході відпадає необхідність у використанні додаткових парсерів.

Техніка використання eval() робить систему вразливою, якщо джерело JSON-даних, що використовуються, не відноситься до надійних. Такими даними може виступати шкідливий JavaScript-код для атак за допомогою ін'єкції коду. За допомогою цієї вразливості можливо здійснювати крадіжку даних, підробку автентифікації. Проте, вразливість можна усунути за рахунок використання додаткових засобів перевірки даних на коректність. Наприклад, до виконання eval() отримані від зовнішнього джерела дані можуть перевірятися за допомогою регулярних виразів. У RFC, що визначає JSON пропонується використовувати такий код для перевірки його відповідності формату JSON

```
const my_JSON_object = !(/[^,:{}\[\]0-9.\-+Eaeflnr-u \n\r\t]/.test(
text.replace(/"(\\.|[^"\\])*"/g, '''))) &&
eval('(' + testedData + ')');
```

Як безпечніша альтернатива eval() була запропонована нова функція parseJSON(), здатна обробляти тільки JSON-дані.

Вбудований JSON

Останні версії веб-браузерів мають вбудовану підтримку JSON і здатні його обробляти без виклику функції eval(), що призводить до описаної проблеми. Обробка JSON у такому разі зазвичай здійснюється швидше. Так у червні 2009 року вбудовану підтримку JSON мали такі браузери: Mozilla Firefox 3.5+; SeaMonkey 2; Thunderbird 3; Microsoft Internet Explorer 8; Opera 10.5; Браузери, засновані на WebKit (наприклад, Google Chrome, Apple Safari).

Принаймні дві популярні бібліотеки JavaScript використовують вбудований JSON у разі його доступності: jQuery; Dojo.

Підробка крос-доменного запиту

Непродумане використання JSON робить сайти вразливими до підробки міжсайтових запитів (CSRF або XSRF). Оскільки тег <script> допускає використання джерела, що не належить до того ж домену, що і використовуваний ресурс, це дозволяє виконувати код даних, представлених у форматі JSON, в контексті довільної сторінки, що робить можливою компрометацію паролів або іншої конфіденційної інформації користувачів, що пройшли авторизацію на іншому сайті.

Це є проблемою тільки у разі вмісту в JSON-даних конфіденційної інформації, яка може бути компрометована третьою стороною і якщо сервер розраховує на політику одного джерела, блокуючи доступ до даних при виявленні зовнішнього запиту. Це не є проблемою, якщо сервер визначає допустимість запиту, надаючи дані тільки у разі його коректності. HTTP cookie не можна використовувати для визначення цього. Виключне використання HTTP cookie використовується підробкою міжсайтових запитів.

Розширення, JSONP

JSONP або «JSON з підкладкою» є розширенням JSON, коли назва функції зворотного виклику вказується як вхідний аргумент. Спочатку ідея була запропонована в блозі MacPython в 2005 році, і в наш час використовується

багатьма Web 2.0 застосунками, такими, як Dojo Toolkit Applications, Google Toolkit Applications і zanox Web Services.

Подальші розширення цього протоколу були запропоновані з урахуванням введення додаткових аргументів, як, наприклад, у разі JSONPP за підтримки S3DB вебсервісів.

Оскільки JSONP використовує скрипт-теги, виклики по суті відкриті світові. З цієї причини JSONP може бути недоречними для зберігання конфіденційних даних.

Включення скриптових тегів від віддалених сайтів дозволяє їм передати будь-який контент на сайті. Якщо віддалений сайт має вразливості, які дозволяють виконати ін'єкції JavaScript, то початковий сайт також може бути ними зачеплений.

Розширення, BSON

BSON – це бінарна форма представлення простих структур даних і асоціативних масивів (які називають об'єктами або документами). Назва «BSON» заснована на визначенні JSON і неофіційно означає «Binary JSON» (бінарний JSON).

JSON Reference

Стандарт JSON не описує посилання на інші об'єкти або частини, але існує чернетка стандарту IETF для посилань на об'єкти на основі JSON. Посилання дозволяють здійснювати трансклюзію - вставляти одні документи в інші.

JSON Reference - це JSON-об'єкт з ключем "$ref" (всі інші ключі ігноруються) і значенням стрічкового типу що містить URI, наприклад:

```
{ "$ref": "http://example.com/example.json#/foo/bar" }
```

Якщо URI містить ідентифікатор фрагмента (в прикладі вище "/foo/bar"), він інтерпретується як JSON Pointer.

Модуль dojox.json.ref в Dojo toolkit, забезпечує підтримку декількох форм JSON Reference.

4.2 Захист розробленого програмного забезпечення

Tiny Encryption Algorithm (TEA) [1] – блочний алгоритм шифрування типу «Мережі Фейстеля». Алгоритм був розроблений на факультеті комп'ютерних наук Кембриджського університету Девідом Вілером (David Wheeler) і Роджером Нідгемом (Roger Needham) та вперше представлений в 1994 році [2] на симпозіумі зі швидкими алгоритмами шифрування в Льовені (Бельгія).

Шифр не патентований, широко використовується в ряді криптографічних додатків і широкому спектрі апаратного забезпечення, завдяки вкрай низькими вимогами до пам'яті й простоті реалізації. Алгоритм має як програмну реалізацію на різних мовах програмування, так і апаратну реалізацію на інтегральних схемах типу FPGA.

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму TEA, який заснований на бітових операцій з 64-бітним блоком, має 128-бітний ключ шифрування. Стандартна кількість раундів мережі Фейстеля біля 64 (32 циклу), однак, для досягнення найкращої продуктивності або шифрування, число циклів можна варіювати від 8 (16 раундів) до 64 (128 раундів). Мережа Фейстеля несиметрична через використання в якості операції накладення додавання за модулем 2^{32}.

Перевагами шифру є його простота в реалізації, невеликий розмір коду й досить висока швидкість виконання, а також можливість оптимізації виконання на стандартних 32-бітних процесорах, так як в якості основних операцій використовуються операції виключна «АБО» (XOR), побітового зсуву й додавання за модулем 2^{32}. Оскільки алгоритм не використовує таблиць підстановки і раундова функція досить проста, алгоритму потрібно не менше 16 циклів (32 раундів) для досягнення ефективної дифузії, хоча повна дифузія досягається вже через 6 циклів (12 раундів).

Алгоритм має відмінну стійкість до лінійного криптоаналізу і досить гарну до диференціального криптоаналізу. Головним недоліком цього алгоритму шифрування є його вразливість до атак «на пов'язаних ключах» (англ. Related-key attack). Через простий розклад ключів кожен ключ має 3 еквівалентних ключа. Це

означає, що ефективна довжина ключа складає всього 126 біт [3] [4], тому даний алгоритм не слід використовувати в якості геш-функції.

Опис алгоритму

Вихідний текст розбивається на блоки по 64 біта кожен. 128-бітний ключ K ділиться на чотири 32-бітних підключа K [0], K [1], K [2] і K [3]. На цьому підготовчий процес закінчується, після чого кожен 64-бітний блок шифрується протягом 32 циклів (64 раундів) за нижченаведеним алгоритмом. [5]

Припустимо, що на вхід n-го раунду ($1 \leq n \leq 64$) надходять права й ліва частини (L_n, R_n), тоді на виході n-го раунду будуть ліва й права частини (L_{n+1}, R_{n+1}), які обчислюються за такими правилами:

$$L_{n+1} = R_n.$$

Якщо $n = 2 * i - 1$ для $1 \leq i \leq 32$ (непарні раунди), то:

$$R_{n+1} = L_n (\{ [R_n 4] K [0] \} \{ R_n i * \delta \} \{ [R_n 5] K [1] \}).$$

Якщо $n = 2 * i$ для $1 \leq i \leq 32$ (парні раунди), то:

$$R_{n+1} = L_n (\{ [R_n 4] K [2] \} \{ R_n i * \delta \} \{ [R_n 5] K [3] \}).$$

Де

X Y – операція додавання чисел X і Y за модулем 232.

X Y – побітове виключне АБО» (XOR) чисел X і Y, яке в мові програмування Сі позначається як X ^ Y

X Y і X Y – операції побітового зсуву числа X на Y біт вліво й вправо відповідно.

Константа δ була виведена з Золотого перерізу:

$$\delta = (-1) * 2^{31} = 2654435769 = 9E3779B9_h.$$

У кожному раунді константа множиться на номер циклу i. Це було зроблено для запобігання простих атак, заснованих на симетрії раундів.

Також очевидно, що в алгоритмі шифрування TEA немає як такого алгоритму розкладу ключів. Замість цього в непарних раундах використовуються підключі K [0] та [1], у парних – K [2] і [3].

Так як це блочний шифроалгоритм, де довжина блоку 64-біт, а довжина даних може бути не кратна 64-біт, значення всіх байтів, які доповнюють блок до кратності в 64-біт, встановлюється в 0x01.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі система керування розподіленою мережею на базі Zyxel. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи: Функціональних кнопок ПЗ; Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Верхнього меню; Розділу обрання групи; Розділу виведення результату роботи системи; Функції представлені у графічному вигляді.

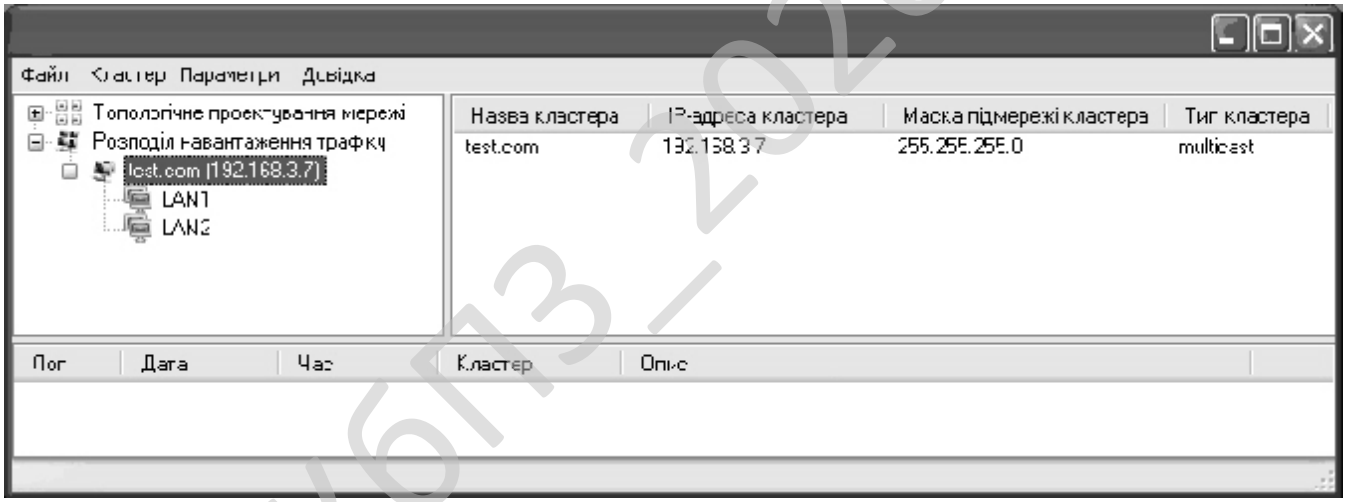

Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і 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.

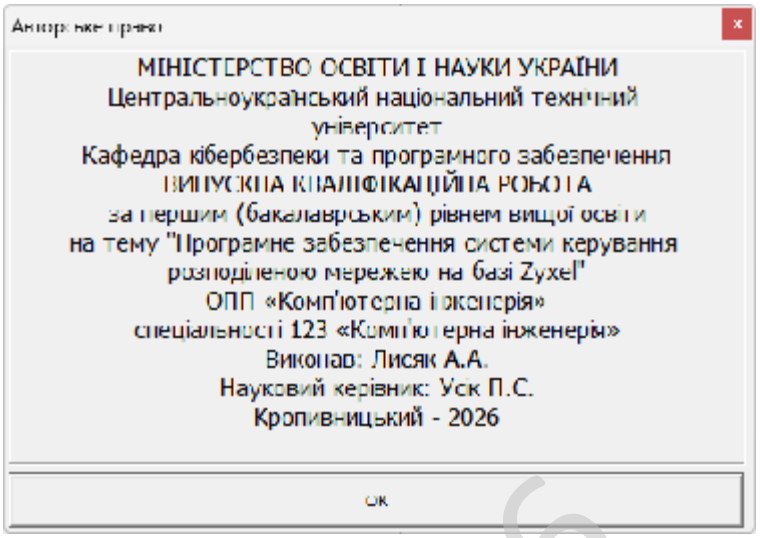

Рисунок 5.2 – Авторське право

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.

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

У цьому випадку формуються тестові варіанти, в яких:

– Гарантується перевірка всіх незалежних маршрутів програми.

– Знаходяться гілки True, False для всіх логічних рішень.

– Виконуються всі цикли (у межах їхніх кордонів та діапазонів).

– Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

– Кількість незалежних маршрутів може бути дуже велика.

– Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

– У програмі можуть бути пропущені деякі маршрути.

– Не можна виявити помилки, поява яких залежить від даних.

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

– Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

– Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

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

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

Проводилось тестування чорної скриньки.

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

Ці тести демонструють:

– Як виконуються функції програми.

– Як приймаються вихідні дані.

– Як виробляються результати.

– Як зберігається цілісність зовнішньої інформації.

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

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

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

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

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

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

– Помилок ініціалізації та завершення.

Обрано умови розповсюдження – commercial software.

Програмне забезпечення, створене комерційною організацією з метою отримання прибутку від його використання іншими, наприклад, шляхом продажу копій.

Найважливішою особливістю комерційний програмних продуктів є підтримка великих компаній, прямо зацікавлених у поширенні програм. Багато організацій надають виключно платну підтримку своїх продуктів, такий підхід, як правило, використовують організації надають відкриті вихідні коди. Для продуктів, що розповсюджуються на комерційній основі діють зазвичай безкоштовні служби підтримки, покликані збільшити рівень довіри у клієнтів і потенційних покупців.

Далеко не завжди, але як правило терміни критично важливих змін в комерційних продуктах значно менше, ніж у некомерційних проектів. Це пов'язано з тим, що над комерційним продуктом працюють цілі групи розробників і ця робота є їх основним заняттям. Розробникам-початківцям як правило доводиться шукати додаткові способи заробітку, і це збільшує час, що витрачається на доповнення і зміни програм. Так як основним рушійним фактором створення комерційного ПЗ є одержання прибутку, то комерційні програмні продукти першими заповнюють вільні ніші та пропонують варіанти вирішення завдань відразу по мірі виявлення вакууму в будь-якому секторі ринку.

Окремий вид комерційних програм, коли їх розробка оплачується безпосередньо замовником. Такі програми найчастіше позбавлені всіх переваг комерційних продуктів, оскільки мають обмежений бюджет, але більш адаптовані до вимог замовника, ніж аналоги.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи керування розподіленою мережею на базі Zyxel.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем керування розподіленою мережею на базі Zyxel.

– Досліджена система керування розподіленою мережею на базі Zyxel.

– На основі отриманих результатів досліджень створена програмна реалізація системи керування розподіленою мережею на базі Zyxel.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання керування розподіленою мережею на базі Zyxel.

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 призначені для системи керування розподіленою мережею на базі Zyxel. 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 TEA.

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.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.

2. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.

3. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.

4. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.

5. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.

6. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.

7. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка»* (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.). Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.

8. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.

9. 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.

10. 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.

11. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

12. 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.

13. 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.

14. 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.

15. 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.

16. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». CEUR Workshop Proceedings Volume 3156, 2022, Pages 390-399.

17. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління,* № 2(70). 2022. С. 28-37.

18. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку,* 2022, № 3(69). С. 93-98.

19. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»,* № 2 (307). С. 46-52. 2022.

20. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

21. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

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

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

24. Smirnov O., Kuznetsov A., 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.

25. 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.

26. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T),* Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

33. 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.

34. 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.

35. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

36. 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.

37. 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.

38. Smirnov A.A., Kuzne tsov 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.

39. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

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

41. 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.

42. Смірнов О.А., Конопліцька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

43. 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.

44. 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.

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

46. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan

D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

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

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

49. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дрєєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

50. Смірнов О.А., Стасєв Ю.В. Бараннік В.В. Захист інформації в автоматизованих системах управління. Навчальний посібник – Харків: ХУПС, 2015. – 264 с.

51. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник. – Кіровоград: КНТУ 2012. – 250 с.

52. Смірнов О.А., Євсеєв С.П., Жукарев В.Ю., Король О.Г., Сорокін В.Є., Мелешко Є.В. Технології і стандарти комп'ютерних мереж. Навчальний посібник. – Кіровоград: КНТУ 2012. – 454 с