

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи керування
Ethernet-фабрикою Brocade VCS”

КБПЗ - 2024

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

Керівник проекту
кандидат технічних наук, доцент
_____ Смірнов С.А.
« ____ » _____ 2024 р.
Рецензент _____

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

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

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

Шуляку Максиму Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи керування Ethernet-фабрикою Brocade VCS*

2. Керівник роботи *Смірнов Сергій Анатолійович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

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

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

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи керування Ethernet-фабрикою Brocade VCS*

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

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

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

- | | |
|--|-----------------|
| <i>Наукова новизна</i> | <i>1 аркуш</i> |
| <i>Структурна схема системи</i> | <i>1 аркуш</i> |
| <i>Функціональна схема системи</i> | <i>1 аркуш</i> |
| <i>Діаграма процесів</i> | <i>1 аркуш</i> |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> |
| <i>Показники економічної ефективності</i> | <i>1 аркуш</i> |

6. Консультанти розділів роботи

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

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

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2024 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Шуляк М.В. Дослідження та програмна реалізація системи керування Ethernet-фабрикою Brocade VCS. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування Ethernet-фабрикою Brocade VCS.

Метою розробки є дослідження та програмна реалізація системи керування Ethernet-фабрикою Brocade VCS.

Об'єктом дослідження є процес керування Ethernet-фабрикою Brocade VCS.

Предметом дослідження є методи керування Ethernet-фабрикою Brocade VCS.

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

Результат роботи – програмна реалізація системи керування Ethernet-фабрикою Brocade VCS.

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, Ethernet-фабрика Brocade VCS

ABSTRACT

Shulyak M.V. Research and software implementation of the Brocade VCS Ethernet factory control system. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this final qualification work for the second (master's) level of higher education, software has been developed, which is intended for the Brocade VCS Ethernet factory control system.

The purpose of the development is the research and software implementation of the Brocade VCS Ethernet factory control system.

The object of the research is the Brocade VCS Ethernet factory control process.

The subject of the research is the Brocade VCS Ethernet factory control methods.

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

The result of the work is the software implementation of the Brocade VCS Ethernet factory control system.

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 user-friendly interface has been developed. Instructions for working with software tools are provided.

The program can be used on PCs with Windows 10/11.

The program is developed in the Python environment.

Keywords: computer engineering, Brocade VCS Ethernet factory

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	12
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	12
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	30
2.3 Розгорнута постановка завдання	34
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	36
3.1 Опис функціонування системи	36
3.2 Розробка структурної схеми.....	40
3.3 Розробка функціональної схеми	44
3.4 Розробка діаграми процесів.....	58
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	60
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	60
4.2 Захист розробленого програмного забезпечення.....	76
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	80
6 НАУКОВА НОВИЗНА	86

					ВКРМ-123.24.0048.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Дослідження та програмна реалізація системи керування Ethernet-фабрикою Brocade VCS	Літ.	Аркуш	Аркушів
Розроб.	Шуляк М.В.					М	1	113
Перев.	Смірнов С.А.					ЦНТУ КІ-23М		
Н.контр.	Коваленко А.С.							
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	87
7.1	Визначення цільової аудиторії кінцевого готового продукту	87
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	88
7.3	Вибір методу оцінки вартості ПЗ	89
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	90
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	93
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	94
7.7	Визначення ключових факторів успіху конкретного проєкту.....	94
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	96
8.1	Вступ.....	96
8.2	Шкідливі і небезпечні фактори при роботі з комп'ютером.....	97
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	99
8.4	Розробка заходів з умов поліпшення охорони праці	102
8.5	Розрахункова частина	103
9	ОСНОВНІ ВИСНОВКИ.....	105
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	107

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЕОМ	–	електронна обчислювальна машина
КВ	–	коефіцієнт варіації
КЗ	–	канал зв'язку
НСД	–	несанкціонований доступ
ПС	–	програмна середа
СВВ	–	система виявлення вторгнень
СеМО	–	експонентна мережа масового обслуговування
СМО	–	система масового обслуговування
СПД	–	система передачі даних
DDoS	–	розподілена відмова від обслуговування
SAN	–	центр обробки даних
SDN	–	програмно-визначаєма мережа
VCS	–	Virtual Cluster Switching

КБПЗ-2024

ВСТУП

Актуальність теми. Питання побудови й експлуатації сучасної віртуалізованої мережі є найважливішими для фахівців в області центрів обробки даних. Значна увага приділяється актуальним тенденціям галузі: віртуалізації й програмно обумовленим мережам (SDN), продуктам для побудови мережної інфраструктури центрів обробки даних – LAN і SAN, а також організації комунікацій між віддаленими площадками й створенню розподілених центрів даних.

Ethernet-фабрика Brocade VCS реалізує віртуальний розподілений комутатор рівня доступу й агрегування. Вона практично не вимагає налаштування, підтримуючи «самоформування» і автовиявлення пристроїв у будь-якій топології, при цьому керування нею здійснюється як одним віртуальним комутатором. Для запуску фабрики досить задати її домен і ID, а потім можна користуватися єдиною консоллю керування всіма комутаторами. Тобто є програмно-визначаємою мережею або програмно-конфігуруємою мережею (англ. software-defined networking, SDN) – мережею передачі даних, у якій рівень керування мережею відділений від пристроїв передачі даних і реалізується програмно. Одна з форм віртуалізації мережі.

SDN уже знаходить практичне застосування. У даній роботі, система, яка розроблена, допомагає знизити негативний вплив атак DDoS. Погрози виявляються за допомогою sFlow-RT (у режимі реального часу) і відправляються на колектор, де обробляються програмним забезпеченням централізованого керування мережею Brocade Network Advisor, здатним виступати також як контролер sFlow. Колектор сповіщає контролер, що потрібно використовувати правило OpenFlow DDoS на маршрутизаторі MLXe для протидії атаці DDoS.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи керування Ethernet-фабрикою Brocade VCS.

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

- Огляд існуючих систем керування Ethernet-фабрикою Brocade VCS.
- Дослідження системи керування Ethernet-фабрикою Brocade VCS.
- Програмна реалізація системи керування Ethernet-фабрикою Brocade VCS.

Об'єктом дослідження є процес керування Ethernet-фабрикою Brocade VCS.

Предметом дослідження є методи керування Ethernet-фабрикою Brocade VCS.

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

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

- Удосконалено метод керування Ethernet-фабрикою Brocade VCS.
- Розроблено вітчизняний продукт керування Ethernet-фабрикою Brocade VCS, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі керування Ethernet-фабрикою Brocade VCS.

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи керування Ethernet-фабрикою Brocade VCS, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2024

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для протидії DDoS-атакам. DDoS – це скорочення англійського вираження Distributed Denial of Service, що перекладається на українську мову як «Розподілена відмова від обслуговування». Це означає відмову від обслуговування мережного ресурсу в результаті численних розподілених (тобто, що відбуваються з різних точок Інтернет-доступу) запитів. Відмінність DoS-атаки (Denial of Service – «Відмова від обслуговування») від DDoS полягає в тому, що в цьому випадку перевантаження відбувається в результаті запитів з якого-небудь певного Інтернет-вузла.

У випадку набагато більше складної й зробленої DDoS-атаки може бути повністю порушена робота будь-якого ресурсу – від невеликого інформаційного сайту до великого Інтернет-магазину або поштового сервера. Під час атаки на сервер сайту-«жертви» надходять мільйони запитів від користувачів, що приводить до його перевантаження й, відповідно, неприступності. Не встигаючи обробляти величезна кількість запитів, сервер спочатку починає просто гальмувати, а потім і зовсім припиняє роботу. Запити найчастіше носять хитромудрий і безглуздий характер, що ще більше ускладнює роботу сервера.

Основна складність для власників сайтів полягає в тому, що величезна частина методів боротьби з DDoS – практично неефективні, адже запити надходять із різних сторін, і перекрити який-небудь вузол, з якого надходять запити (як у випадку в DoS-атаками) – недостатньо. Звичайно атака проводиться за допомогою вірусних троянських програм, що утягують у цей процес мільйони користувачів без них на те згоди й повідомлення. Трояни заражають недостатньо захищені комп'ютери й можуть досить довгий час діяти, взагалі ніяк себе не виявляючи. Зона охопту в такий спосіб стає неймовірно широкою, а запити

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

злочинців. Простежити зворотний зв'язок неможливо, максимум – можна обчислити одну з керуючих консолей, які також, до речі, вважаються потерпілими від атаки.

Складністю у виявленні злочинців є й вільне поширення в мережі програм для проведення атак. Споконвічно подібне програмне забезпечення розроблялося для виявлення ступеня стійкості мережі до зовнішніх навантажень. Однак за роки воно перетерпіло сильні зміни, було сформовано й удосконалене кілька видів атак, які до того ж можуть сполучатися, варіюватися й видозмінюватися. Саме тому захист від DDoS-атак повинна бути професійною, постійною й оновлюваною.

1.2 Область застосування

Областю застосування системи є Ethernet-фабрика Brocade VCS. Інноваційна технологія Brocade Virtual Cluster Switching (VCS) є складовою частиною концепції Brocade One і дозволяє компаніям зменшити складність і знизити вартість їхньої мережної інфраструктури за рахунок використання нової архітектури – Ethernet-фабрики для дата-центрів. VCS збільшує масштабованість і ефективність використання мережі, кардинально спрощує архітектуру й збільшує доступність додатків, що необхідно для сучасних центрів обробки даних що використовують віртуалізацію. VCS включає набір динамічних сервісів розширювального функціонала й інвестицій, що забезпечують захист, замовників що робить цю технологію основою для побудови мереж віртуальних дата-центрів.

Ethernet-фабрика дозволяє обійти такі обмеження традиційної багаторівневої архітектури дата-центрів як:

- утворення петель трафіку при неправильному налаштуванні протоколу STP;
- блокування частини портів і неможливість їхнього використання для передачі трафіку;

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

- затримки й втрати пакетів при змінах топології мережі;
- складність у налаштуванні великої кількості пристроїв.

VCS дозволяє відійти від ідеології налаштування й функціонування комутаторів як самостійних пристроїв, що мають власні рівні керування й передачі даних. Тепер комутатори працюють у складі фабрики, що має єдиний інтерфейс керування й передачі даних у не залежності від кількості підключених до фабрики пристроїв і кількості й типу з'єднань між ними. Нові шасі, що підключаються до фабрики, приєднуються й настраюються автоматично без втручання адміністратора. Зовні фабрика виглядає як єдине логічне шасі з безліччю зовнішніх портів.

Внутрішні з'єднання фабрики VCS завжди активні, об'єднані в агреговані канали й передають трафік без втрат (lossless) і затримок у випадку зміни топології мережі.

Одне із ключових відмінностей фабрики від застарілої багаторівневої архітектури – проходження трафіку між серверами прямо, по найкоротших маршрутах, без обов'язкового проходу через ядро мережі. Це дозволяє істотно знизити затримки й оптимізувати роботу віртуалізованих ЦОД.

Перша лінійка комутаторів, у якій реалізована підтримка VCS, була випущена на початку 2011 року. Це серія Brocade VDX 6720 у якій були доступні моделі на 24 і 60 портів 10 Gb. Серія VDX 6730 дозволила підключити фабрику до традиційних мереж зберігання через FC-інтерфейси. Гігабітні моделі серії VDX 6710 дозволяли підключити до фабрики невеликі сервера з мідними портами Ethernet. На даний момент дані серії комутаторів уже зняті з виробництва (EOL).

Шасійна лінійка VDX 8770 із продуктивністю 4 терабіта на слот дозволяє побудувати ядро VCS-мережі й підключати в одну фабрику до 8000 портів 10 Gb. У дану серію входять 4-х і 8-мі слотові моделі. В 2016 для VDX 8770 вийшли нові лінійні карти з мідними інтерфейсами 10 GBASE-T, оптичними інтерфейсами QSFP+ (40 Gb) і CFP2 (100 Gb).

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

VDX 6740 це комутатори з оптичними портами 10 Gb SFP+, VDX 6740T мають мідні порти 10 Gb 10 BASE-T. Всі моделі оснащені Uplink портами 40 Gb QSFP+. Пізніше була випущена модифікація VDX 6740 T-1G, що має на борті мідні порти 1000 BASE-T, які можна проапгрейдити до 10 GBASE-T за допомогою програмної ліцензії.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи керування Ethernet-фабрикою Brocade VCS, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ_2024

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Для галузі високих технологій характерні мікро- і мегацикли інновацій. Мікроцикли – це те, що відбувається щодня, щотижня, щорічно. Мегацикли набагато більше тривалі й займають приблизно 20 років. За цей час відбуваються масові й фундаментальні зміни, що міняють не тільки технологічну галузь, але й всі інші галузі й компанії – відбувається як би ланцюгова реакція.

Зараз ми перебуваємо на початку нового мегациклу.

Щоб зрозуміти контекст майбутніх технологічних потрясінь, варто небагато поглибитися в історію. Із середини 70-их до середини 90-их платформою для бізнес-інновацій були термінали мейнфреймів і централізована модель обчислень. Аналітики IDC називають це «першою платформою». Базовою мережною інфраструктурою тієї пори була Systems Network Architecture (SNA), в основі якої – мережі з комутацією каналів (circuit-switched private lines) з обмеженими вимогами до кібербезпеки.

New IP: основа для бізнесу нового покоління

Два десятиліття через, на початку 90-их почався новий мегацикл. У цю епоху платформою для інновацій стала модель «клієнт-сервер» і локальні/глобальні мережі – LAN/WAN. У відповідь на нові вимоги до конфіденційності користувальницьких даних вони точково вирішували проблеми забезпечення безпеки за допомогою окремих спеціалізованих пристроїв. В IDC це називають «другою платформою». Саме в даний період почався перехід до мереж IP. А самим глобальним зрушенням був і залишається Інтернет.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

На початку 1995 року, під час розгортання мереж IP, у світі налічувалося 16 млн. користувачів Інтернету, 2700 Web-сайтів, а концепції електронної комерції тільки стояло народитися. Крім того, у світі було менш 100 млн. мобільних абонентів. У ті часи ви не могли підключити свій мобільний пристрій до Інтернету – смартфонів ще не існувало. І в більшості випадків для виходу в Інтернет використовувалися лінія, що комутирується, і настільний ПК.

20 років через мир Інтернету став настільки популярний, що народилася зовсім нова культура – онлайнова культура. Крім того, змінилася й у тім або іншому ступені трансформувалася практично будь-яка галузь.

До 2024 року кількість мобільних пристроїв виросла в 70 разів і досягло 7 млрд. Сьогодні у світі 2 млрд. користувачів Інтернету й 1 млрд. Web-сайтів. Щорічно обсяг даних збільшується більш ніж на 50%. Забезпечення інформаційної безпеки стає, якщо вже не стало, однією із пріоритетних завдань ІТ-директорів. Ясно, що ми вступаємо в епоху «третьої платформи», для якої також характерні знаменні віхи в плані інновацій: хмари, мобільність, соцмережі, Великі Дані – все це її частина. В Gartner ці елементи називають «сполучними силами» і говорять про те, що для виживання в нових умовах компанії повинні перетворитися в «цифрові підприємства».

Гарна новина полягає в тім, що третя платформа розширює можливості для бізнесу й у значній мірі збільшує гнучкість. Погана полягає в наступному: перед ІТ ще ніколи не стояли настільки складні завдання, адже організації буквально накриває лавина даних. Їм доводиться забезпечувати безпеку в режимі реального часу, а часом і з попередженням усувати пробіли в захисті. І при цьому багато хто з них використовують IP-мережі, спроектовані для другої платформи, для реалій 20-літньої давнини. Ці мережі вже не відповідають потребам сьогодення.

З розширенням можливостей, які третя платформа, як очікується, відкриє в наступні 20 років, ця невідповідність буде тільки збільшуватися. За прогнозами Gartner, за цей час кожна компанія, незалежно від галузі, перетвориться в

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

«цифрове підприємство» (джерело: “Digital Business: Implications for CPOs and Supply Management Leaders”).

Вплив цього зрушення на базову мережну інфраструктуру, на ті, хто неї створює й обслуговує, буде дуже глибоким.

Поглиблення невідповідності й старі IP-мережі

За останні три роки витрати на хмари в усьому світі вирости до 65 млрд. доларів у рік. Ми перейшли від взаємодії площадок і людей до взаємодії речей – мільярдів і трильйонів речей. І всі вони будуть генерувати й споживати дані. Проблема в тому, що всіма цими даними потрібно управляти. Щоб перетворити дані в кошовну інформацію, їх потрібно зберігати, захищати, переміщати, аналізувати, не жертвуючи при цьому інформаційною й мережною безпекою.

Темпи інновацій таких хмарних провайдерів як Amazon і Google у сполученні й низкою вартістю доставки сервісів поглиблюють розрив із традиційними провайдерами й IT-департаментами. З кожним днем користувачі споживають усе більше IT-сервісів і додатків безпосередньо із хмари. Причому їхнього очікування від можливостей самообслуговування, швидкої доставки й прискорення темпів інновацій з кожним днем ростуть, адже вони вже одержали такий досвід при роботі зі споживчими додатками.

IT-департаменти багатьох компаній не поспівають за цими змінами. Мережна архітектура застаріла: вона не проектувалася відповідно до даних вимог, і 70% IT-бюджету йде тільки на підтримку колишньої інфраструктури. IT повинні бути рушійною силою корпоративних інновацій і цінностей, і кожний IT-департамент хоче відповідати мінливим і зростаючим вимогам користувачів. IT-підрозділи хочуть надати їм платформу для інновацій – основу для інновацій, побудови конкурентного цифрового бізнесу, таку, яку здатні запропонувати сьогодні хмарні провайдери. Все це стосується й традиційних сервіс-провайдерів. На жаль, якщо покладатися на успадковані мережні архітектури, втілити сказане дуже важко. Багато хто говорять, що «гальмує» інновації саме стара архітектура.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Ці старі IP-мережі – в англomовних країнах їх називають ємним терміном «Old IP» – добре послужили останні два десятиліття. А їхня довговічність говорить про те, наскільки елегантним виявився протокол IP. Звичайно, ніхто не затверджує, що від них треба позбуватися. Але й проходження колишнім курсом – не варіант. Потрібні корінні зміни.

Нова, удосконалена, але мережна архітектура, що зберігає еволюційну наступність, уже тут. Це «New IP» – нові IP-мережі.

New IP

Що ж таке нові IP-мережі – New IP? Це мережі IP, пристосовані для сучасного миру – спроектовані відповідно до потреб хмар, мобільності користувачів, соціальних мереж і Больших Даних. New IP – це новий підхід до архітектури IP-мережі, що сприяє росту бізнесу й швидких змін і при цьому підтримуючий високий рівень безпеки або посилюючий її. New IP відповідає завданням «третьої платформи» і припускає використання відкритих, автоматизованих, програмно-конфігуруємих компонентів, що сприяє гнучкості й зниженню витрат.

Щоб зрозуміти переваги й наслідки даної трансформації, корисно буде зрівняти сьгоднішню ситуацію з тим, що обіцяє New IP. Старі IP-мережі ґрунтуються на закритих системах, і цикли інновацій у них обмежені спеціалізованим устаткуванням, а виділення мережних ресурсів являє собою складне й трудомістке завдання. Сумісність різнорідних систем обмежена. У центрі екосистеми – вендори. Вартість рішень висока, інновації впроваджуються повільно.

В основі New IP – відкриті системи (Open Source) і стандарти, що дозволяють вийти за рамки пропрієтарних рішень із галузевими стандартами. New IP дає IT-підрозділам можливість використовувати стандартне, типове устаткування або системи, розроблені для конкретних завдань. Виділення мережних ресурсів автоматизоване й може здійснюватися по моделі самообслуговування. Ключ до сумісності – відкриті програмні інтерфейси (API).

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

У центрі екосистеми – замовник. Скорочуються капітальні й операційні витрати (CapEx і OpEx), а інновації впроваджуються зі швидкістю бізнесу.

New IP характеризують чотири основних атрибути:

Цілеспрямована відкритість

New IP не просто відкрита архітектура. Її відкритість переслідує певні цілі.

Відкритість прискорює інновації, зменшує «прив'язку до вендору», а також вартість і складність.

Оскільки ніхто не може пророчити, у якому напрямку буде розвиватися цикл інновацій, New IP дозволяє використовувати компоненти й сервіси широкого кола співтовариства інноваторів і тим самим по-новому вирішувати завдання.

Відкриті рішення дозволяють ефективніше забезпечувати захист завдяки обміну інформацією між елементами системи безпеки (фізичними або віртуальними). При цьому створюється цілісна картина й використовуються дані із всіх джерел.

New IP дає можливість комбінувати рішення різних вендорів за допомогою відкритих API, погоджувати свою стратегію й темпи інновацій завдяки волі вибору, швидкості й гнучкості змін.

Фокус на інновації й програмне забезпечення

New IP – це орієнтація на інновації й ПЗ для швидкого досягнення результатів і розширення можливостей.

Сучасні галузі вимагають інновацій – це умова їхнього росту. New IP – це платформа для швидкого впровадження інновацій.

ПЗ – це програмувальне керування складними завданнями. IT-персонал може сфокусувати своя увага стратегічних завданнях, що мають першорядне значення для росту бізнесу, на нових пропозиціях – відповідно до умов ринку.

В New IP система безпеки убудована, а не додана поверх мережі, але при цьому вона абстрагована від базової мережної інфраструктури. У результаті мережа може бути «завжди наготові», захищаючи й збережені, і передані дані.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Екосистема, орієнтована на користувачів

New IP означає створення екосистеми, набору ресурсів для прискорення інновацій.

New IP не обмежується підходом одного вендора й дозволяє компаніям іти в ногу з інноваціями, використовуючи величезний потенціал інших виробників і вносячи в нього свій внесок.

У рамках New IP можливі самі різні джерела інновацій і рішень. Всі члени співтовариства зацікавлені в тім, щоб вносити свій внесок і вирішувати проблеми інформаційної й мережної безпеки.

New IP трансформує ІТ

Застосований в New IP «екосистемний» підхід робить все це можливим, не уступаючи «вендорно-орієнтованої» моделі останнього двадцятиліття.

На ваших умовах

Кроки в напрямку еволюційного переходу до New IP дозволяють вам трансформувати свою інфраструктуру, ІТ-підрозділи й бізнес для досягнення революційних результатів.

Якщо говорити про безпеку, то вона стає більше контрольовані й керованою, оскільки не вимагає покладатися в основному на статичні й ідентифікаційні рішення. При застосуванні політик безпеки мережі New IP можуть урахувати поведінку, а не просто підтримувати ідентифікацію. Така система безпеки безупинно самонавчається й самооптимізується, а тому може проактивно знижувати ризики, а не тільки реагувати на атаки. Завдяки New IP організації можуть приступитися до розгортання додатків на 90% швидше, скоротити операційні витрати на 50% і перетворити мережа дата-центра в Ethernet-фабрику, готову до впровадження SDN. А замінивши апаратні рішення на віртуалізовані мережні функції (NFV), реалізовані на типових серверних платформах або як хмарні сервіси, можна зменшити CapEx і підвищити гнучкість мережної інфраструктури.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Повна заміна інфраструктури для New IP не буде потрібно. Ви можете мислити масштабно, але почати з малого, швидкими темпами нарощуючи гнучкість мережної архітектури.

Мережі з такими атрибутами New IP мають деякі дивні властивості.

ЦОД усюди

New IP дозволяє розміщати робочі навантаження з будь-якого джерела (частка/публічне/гібридна хмара), виходячи із цілей бізнесу й корпоративної політики, і управляти ними. Даний підхід підвищує ефективність, масштабованість і гнучкість, перетворює ІТ- і мережні підрозділи в постачальника послуг для своїх внутрішніх замовників, вендорів і партнерів. Ви зможете працювати швидше й ефективніше, ніж ваші конкуренти.

Ваш бізнес повинен розвиватися, інновації повинні впроваджуватися швидше, ніж коли-або раніше. New IP допомагає прискорити інновації, скорочує операційні витрати й дозволяє вам випереджати конкурентів. У центрі екосистеми New IP – ваші користувачі.

Кожний з ваших користувачів, кожний їхній додаток можуть одержати свою власну мережу з необхідними сервісами, якістю й безпекою. І мережа може сама себе змінювати відповідно до місцеположення пристрою. В New IP ніякої границі мережі в її традиційному розумінні – вона більше не існує. Границя виникає там, де користувач взаємодіє з мережею, сервісами й додатками.

Архітектура New IP

І що в результаті? Користувач одержує найвищу якість і економічну ефективність.

Коротше кажучи, New IP – це мережа, побудована на ваших умовах.

Принципи безпеки в New IP

Із чого почати в дата-центрі?

New IP дає можливість шляхом еволюційних кроків домогтися революційних результатів. Багато хто думають, що починати потрібно з NFV (Network Functions Virtualization) або SDN (Software Defined Networking). Хоча,

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

як описується нижче, це фундаментальні технології для New IP, у більшості випадків точка старту – створення основи мережі, її фундаменту. При побудові надійної основи мережі найкраще використовувати поетапний підхід.

Основа мережі

Як перший крок створення надійної основи потрібно визначити, що ви хочете побудувати на цьому фундаменті, і як швидко. Модель зрілості мережі Brocade Network Maturity Model (5 кроків до New IP) показує еволюцію ІТ – від витратного напрямку до центра інновацій і механізму росту бізнесу. Вона також представляє швидке прискорення в напрямку інтеграції й контролю віртуалізованих мережних ресурсів і підкреслює що попереджає, що самонавчається характер наступних етапів моделі зрілості.

У кожної готовій до New IP мережі в остаточному підсумку потрібно передавати дані й пакети. Завдяки відкритості, ефективним маршрутам передачі даних і автоматизації заснованої на фабриці мережі можна створити добре працююче середовище й відкрити дорогу для інновацій на основі програмного забезпечення. Архітектура на базі мережної фабрики підвищує гнучкість, зменшує складність і збільшує ступінь автоматизації.

Модель зрілості мережі: П'ять кроків до New IP

Фабрики мають високу вертикальну й горизонтальну масштабованість, адаптуються до постійних змін у потоках трафіку, розмірам потоків і пакетів, протоколах.

Така спрощена архітектура підвищує рівень безпеки. Даний підхід до базових принципів побудови мережі цілеспрямовано забезпечує посилений захист. А якщо ви віртуалізуєте своє обчислювальне середовище, то вам, імовірно, буде потрібно архітектура, заснована на мережній фабриці. Вона дозволяє більш результативно використовувати обчислювальну систему. Ця фабрика також підтримує віртуальні машини. Підтримка ВМ, число віртуалізованих сервісів, їхні типи, поведіння ВМ – все це важливо з погляду забезпечення безпеки цих сервісів.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Дуже важливий і правильний вибір заснованої на фабриці архітектури. Найчастіше рекомендують архітектури на базі Ethernet-фабрики, розроблені такими компаніями як VMware, Cisco або Brocade. У віртуальному середовищі потрібно відслідковувати переміщувані віртуальні машини (VM), потрібен моніторинг трафіку між VM і діагностика – як на рівні віртуального середовища, так і на рівні нижележачої фізичної інфраструктури.

У старих IP-мережах це завдання вирішити дуже складно, але нові, готові до впровадження New IP і засновані на фабриці мережні архітектури спрощують її. Однак однієї лише мережної фабрики для віртуалізованих обчислювальних інфраструктур недостатньо. Фабрика повинна бути глибоко інтегрована з віртуалізованим обчислювальним рівнем. Для середовища VMware це означає інтеграцію з vRealize, vCenter і NSX. Це забезпечить наскрізну видимість заснованої на фабриці архітектури й максимальну продуктивність, що дозволяє збільшити щільність VM і звести до мінімуму простої за рахунок високої стійкості й ефективної діагностики.

Архітектура, що трансформує, IP-фабрики Brocade заснована на п'яти принципах

Перелічимо п'ять обов'язкових характеристик будь-якої фабрики:

1. «Справжня демократія». Це означає, що всі комутатори рівноправні. Архітектура мережі плоска – ніякої ієрархії. Тому в неї немає єдиної точки відмови. У результаті утвориться «плоска поверхня» Layer 2 або 3 – що самоформується й самовідновлюється мережа. Всі маршрути в ній рівноцінні, можна сполучати пристрою різної продуктивності й архітектури. Допускається сполучення встаткування шасійної і фіксованої конфігурації.

2. «Розподілений інтелект». Будь-який мережний порт знає про будь-який інший порт. Це означає, що можна переміщати навантаження з асоційованими характеристиками – так звана автоматична міграція профілів портів (Automatic Migration of Port Profiles, AMPP). Ніякої трудомісткої «ручної» реконфігурації, при якій часто трапляються помилки, таке переміщення не

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Завдяки убудованій автоматизації, масштабованості й швидкості фабрик скорочуються операційні витрати, збільшується продуктивність і рівень доступності, заставляються основи для впровадження SDN.

Збільшення гнучкості за допомогою NFV, VNF і SDN

Віртуальні мережні функції (Virtual Network Function, VNF) або віртуалізація мережних функцій (Network Functions Virtualization, NFV) заміняють мережне встаткування – маршрутизатори, межмережні екрани, контролери доставки додатків (Application Delivery Controller, ADC) і інше фізичне встаткування – на програмні версії, виконувані на серверах x86. NFV скорочує витрати (CapEx і OpEx), спрощує нарощування й зменшення ресурсів у міру необхідності. З NFV сервіси стають мобільними.

Можна взяти всі необхідні віртуалізованому додатку мережні сервіси й помістити їх у VM на тому же сервері. Мережний трафік між додатками й сервісами NFV не буде залишати сервер. У результаті зменшиться трафік «північ-південь» і значно скоротяться витрати. За допомогою віртуальних межмережних екранів, що додаються до кожного додатка, можна також підвищити безпека. Цей прошарок безпеки буде переміщатися по інфраструктурі разом з додатком.

Програмно-конфігуруємі мережі (Software-Defined Networking, SDN) надають інструменти для роботи з мережними сервісами (віртуалізованими чи ні) і інфраструктурою й керування ними. Ініціатива Open Source SDN за назвою OpenDaylight дозволяє візуалізувати ресурси й централізовано управляти ними. Це також дає можливість бачити потоки трафіку й динамічно, автоматизовано управляти ними. Будь-який створюваний у мережі сервіс, фізичн або віртуальний, управляється одним контролером SDN у повністю програмувальному режимі.

Оркестрація сервісів

Мережа – лише одна важлива частина інфраструктури. Є ще обчислювальні системи, системи зберігання даних. І тут на сцену виходить оркестрація. Відкритий (Open Source) протокол OpenStack рівня оркестрації може

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

дати ті ж переваги, що й OpenDaylight на мережному рівні, причому охоплює обчислювальні ресурси, ресурси зберігання даних і мережні ресурси. Крім того, оскільки середовище часто є «багатохмарним» (поєднує приватну й публічну хмару), потрібно, щоб ці хмари працювали разом, причому були б передбачуваними, масштабованими й керованими.

Оркестрація OpenStack дає таку можливість – вона поширюється на все середовище. Якщо ви волієте починати з малого, то сфокусуйте увагу на архітектурі, заснованій на фабриці. І будь-яке мережний пристрій, що здобувається вами, повинне підтримувати SDN/OpenFlow. Навіть якщо ви в цей час не використовуєте програмні можливості SDN, у майбутньому вони знадобляться.

Границі мережі

Границя мережі перестала бути якимсь фізичним місцем. Тепер це якісь операції при взаємодії користувачів з мережею. Додаток «живе» у хмарі або в даті-центрі. Користувачі можуть перебувати в будь-якому місці – границя мережі переміщається разом з користувачем.

При взаємодії користувача з додатком потрібно застосовувати сервіси й політики, які управляють доступом і визначають надавані користувачеві можливості. Вони можуть бути різними й залежать від того, хто він, де він перебуває, що він робить, коли й навіщо, яке застосовує пристрій, які має права. На основі цієї інформації можна застосовувати політики рольового доступу (Role-Based Access Control, RBAC). Вони відбиваються в таких мережних сервісах як маршрутизація, межмережне екранування, QoS і ін.

В New IP можна віртуалізувати практично будь-який мережний сервіс, тому політики повинні застосовуватися практично в будь-якому місці. У цю картину відмінно вписуються ПЗ й сервіси віртуальної границі – те, що мовою сервісів-провайдерів часто називається віртуальною користувальницькою границею (virtual Consumer Edge, vCE) або віртуальним устаткуванням на території клієнта (virtual Customer Premise Equipment, vCPE).

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Наприклад, можна подумати про заміну такого комплекту встаткування у віддалених офісах як фізичні маршрутизатори, межмережні екрани й / або балансувальники навантаження на пакет їхніх програмних екземплярів, виконуваних на серверах на будь-якій площадці й керованих віддалено.

Одним з безпосередніх переваг даної архітектури може бути оптимізація шляхів прямо на площадці філії за допомогою віртуалізованого контролера доставки додатків, що значно скорочує витрати на мережі MPLS/Ethernet, що зв'язують компанію з корпоративним центром. Уже одна тільки ця економія з лишком окупає вартість серверів і віртуальних мережних сервісів на кожній площадці. Але крім цього ви одержуєте на границі New IP – платформу для доставки із самообслуговуванням, що ще більше скорочує витрати й створює основу для впровадження інноваційних сервісів.

Таким чином, почати свій шлях до New IP можна й з фабрик у ЦОД, і з NFV, і з SDN, і з оркестрації на новій границі мережі. Кожний варіант створює ту або іншу цінність для бізнесу, або їхню комбінацію.

Чому саме Brocade

Компанія Brocade була заснована в 1995 році, саме тоді, коли почався ці останні великі перетворення в обчисленнях. На відміну від інших мережних компанія, Brocade споконвічно сфокусувала свою діяльність на дата-центрах, маючи чітке розуміння, як максимально ефективно управляти пакетами й переміщати дані.

Сьогодні Brocade – компанія з обортом в 2,3 млрд. доларів і найбільшою часткою світового ринку комутаторів SAN. Вона посідає друге місце в сегменті дата-центрів і є безперечним лідером у цих областях – жоден інший вендор не може настільки ефективно допомогти компаніям у побудові гетерогенних середовищ зберігання й переміщенні даних між різнорідними пристроями за допомогою різних протоколів. А партнерство – обов'язкова умова створення відкритої екосистеми безпеки, що забезпечує сьогодні більше високий ступінь захисту.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

EMC, HP, IBM, Dell, HDS і інші компанії продають продукти й технології Brocade у складі своїх портфелів рішень для дата-центрів, що відбиває ключову роль партнерства в бізнесі Brocade. Практично всі компанії зі списку Fortune 500 і найбільші у світі організації застосовують у своїх ЦОД технологію Brocade.

Для підтримки їхнього бізнесу Brocade фокусує увагу на додатках, що вимагають найвищого рівня продуктивності, безпеці й доступності у відповідальних середовищах. Brocade розуміє, що тиск ринку на вас і на вашу команду росте. Хмари, мобільні додатки, потреба в перетворенні в «цифрове підприємство», проблеми безпеки – все це пред'являє до старих IP-мережам нові вимоги.

Ті ж фактори обмежують інновації в бізнесі, змушуючи компанії переосмислити свої інфраструктурні стратегії. Щоб очолити перехід до New IP, Brocade займає більше вигідні позиції, чим будь-який інший мережний вендор. Чому? Тому що наш бізнес не ґрунтується на старих IP-мережах.

Із самого початку інвестували у відкриті технології й мережне програмне забезпечення й сконцентровані на тому, щоб очолити цей перехід.

New IP – основа стратегії Brocade. Апаратне й програмне забезпечення проектується для реалізації всіх переваг переходу до відкритого й віртуалізованої IT-інфраструктурі. Пропонована Brocade платформа для віртуальних сервісів містить у собі віртуальний маршрутизатор, межмережний екран, і VPN і NAT – все це в єдиному програмному пакеті. Контролери доставки додатків і балансування навантаження від Brocade, призначені для віртуальних і хмарних сервісів, містять масштабовані інструменти забезпечення безпеки на рівні додатків (Layer 7) і керування продуктивністю. Розроблений також перший «мультивендорний» SDN-контролер OpenDaylight. Їх можна скачати із сайту www.brocade.com і почати використовувати у своїй роботі, накопичуючи досвід і одержуючи коштовні результати. І ви будете не самотні: на сучасний момент завантажено вже більше 1,5 млн. віртуальних маршрутизаторів, що проробили в

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

цілому, по оцінках, 100 млн. годин. Це саме широко використовуване й програмне забезпечення, що розгортається у світі, віртуальної маршрутизації.

vRouter і пакет мережного ПЗ Brocade – єдине програмне забезпечення, споконвічно створене для роботи на платформі Intel, а не просто перенесена на неї оригінальна розробка. От чому воно має кращу в галузі продуктивність: відповідно до незалежних тестів, швидкість становить 80 Гбіт/с на канал на одному сервері із трьома обчислювальними ядрами Intel. Такої продуктивності досить для розміщення на одному сервері десятків мережних сервісів, що дозволяє обмежити трафік «схід-захід» між додатками й мережними сервісами однією серверною платформою. Це веде також до значного зменшення трафіку «сервер-південь», що залишає сервер для передачі в транспортну мережу. Результатом буде чимала економія, гнучкість програмної мережної архітектури, управляти якої можна з будь-якої точки.

Контролер доставки додатків Brocade Virtual Application Delivery Controller (vADC) також розроблений для середовища NFV з повним комплектом інструментів безпеки й контролю продуктивності на рівні додатків, що підвищує ефективність і надійність прикладного програмного забезпечення. Завдяки детальному контролю користувачів і сервісів через відкриті API, дане рішення може автоматизувати розгортання, ліцензування й одержання метрик сервісів доставки додатків у багатоарендному високоплотному середовищу.

Brocade пропонує також єдиний у галузі контролер OpenDaylight, що здатний управляти багатовендорним середовищем, використовуючи як серверний інтерфейс відкриті API. Тим самим на рівні оркестрації забезпечується можливість трансляції бізнес-політики на машинну мову. Brocade SDN Controller відіграє невід'ємну роль у реалізації принципів безпеки New IP, безпеки, споконвічно убудованої в архітектуру мережі й націленої на відкритість, контроль поведінки користувачів і доповненої можливостями самонавчання.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

New IP прискорює доставку сервісів, доступ до даних і інновації

Крім того, комутатори для дата-центрів Brocade® VDX®, комутатори для кампусних сірій Brocade ICX®, маршрутизатори ядра Brocade MLXe підтримують OpenFlow і можуть працювати під управлінням контролера Brocade SDN Controller. Компанія Brocade використовує гібридний режим портів: в устаткуванні MLX і ICX порти одночасно підтримують традиційний форвардинг L2/3 і OpenFlow. Все це – через той самий порт і зі швидкістю лінії. Така унікальна можливість відкриває вам практичний шлях переходу до SDN. Адміністратори мережі можуть інтегрувати OpenFlow в існуючі мережі без переривання їхньої роботи, одержуючи можливість програмного керування конкретними потоками, як це робиться в SDN, і зберігаючи колишню модель форвардинга іншого трафіку. Ніяких змін у фізичній інфраструктурі це не зажадає. Для підтримки SDN не потрібна буде окрема мережа. Готові до впровадження SDN мережі підтримують орієнтоване на додатки динамічне виділення ресурсів, тобто ресурси можна динамічно виділяти в реальному часі відповідно до вимог додатків. Додатки SDN можуть одержувати інформацію з різних джерел, включаючи системи безпеки рівня додатків, інструменти контролю відповідності вимогам QoS, мережну статистику, відомості про користувальницьку активність, системи аналізу погроз безпеки. На основі цих даних виділяються мережні ресурси, забезпечується їхній захист, встановлюються правила контролю доступу, пріоритезується трафік. Все це відбувається динамічно, у реальному часі.

Наприклад, додаток Brocade Flow Optimizer дозволяє одержати детальну інформацію про політиків керування трафіком – для цього виявляються більші потоки трафіку на рівнях L2 – L4, забезпечується керування ними. Це SDN-додаток допомагає вирішити реальні проблеми продуктивності в мережі, що особливо важливо у випадку хмарних сервісів. Воно робить мережу видим і керованою, одночасно пропонуючи нові можливості програмного конфігурування й автоматизації. Завдяки такому «мережному інтелекту» замовники можуть

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

підвищити ефективність мережі за рахунок інформованого планування її пропускної здатності й використання ресурсів, зменшити ризики мережних так і усунути перевантаження за допомогою політик на основі регулювання трафіку. Результатом стане підвищення якості сервісу, задоволення користувачів.

І ще кілька цікавих моментів: у сімейство Brocade VDX входить новий комутатор 40Gb високої щільності. Незабаром цей пристрій у корпусі 1U буде підтримувати 100Gb. Шлюзи NSX Gateway забезпечують пряму й зворотну трансляцію протоколів VXLAN, забезпечуючи інтеграцію й стикування з контролером уже встановленого встаткування. (Цей підхід називається VTEP, і Brocade реалізує його краще, ніж будь-який інший вендор. Крім того, Brocade лідирує по мережних фабриках завдяки своїй технології Brocade VCS® Fabric, підтримуваної комутаторами Brocade VDX. Зараз випущене вже п'яте покоління цього ПЗ. У ньому розширена підтримка SDN і віртуальних середовищ на рівнях Layer 2 і 3. Маршрутизатор ядра Brocade MLXe не тільки забезпечує для додатків високу продуктивність маршрутизації й форвардинга між дата-центрами, але й дозволяє створити основу для критично важливих рішень, що забезпечують взаємодію між ЦОД по IP/MPLS. Їх програмувальні мережні процесори – ідеальний варіант для багатьох типів аналітичних додатків, керуючих трафіком. Крім того, Brocade MLXe забезпечує найвищий у галузі рівень безпеки (шифрування AES 256 Suite B) без зниження продуктивності. Шифрування можна застосовувати в глобальних мережах і в комунікаціях між ЦОД.

При всіх обіцянках SDN і високої гнучкості мережа повинна просто працювати. Тобто вона повинна бути надійної, доступної, захищеної, підтримувати нові вимоги без повної інфраструктурної перебудови. От чому багато хто аналітики й вендори рекомендують фабрику. Якщо ви використовуєте VMware, то жоден вендор мережних фабрик не проробив настільки глибоку роботу з інтеграції своїх продуктів і засобів керування з інструментами й середовищем VMware, як Brocade. Це включає інтеграцію з vRealize, vCenter і NSX. Завдяки фабрикам Brocade середовища VMware працюють краще, стійкіше,

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

ефективніше, стають більше простими. Цього неможливо домогтися в старій IP-середовищі. Ви одержуєте максимальну віддачу від інвестицій в VMware. Вам не нудно витратити час на те, щоб робити всі вручну. Зараз як ніколи важливо, щоб мережа була автоматизована, мала можливості самоформування, самовідновлення, управлялася централізовано, ефективно працювала з усіма основними компонентами середовища VMware. Наприклад, ви можете створити в vCenter віртуальну машину, і фабрика VCS автоматично конфігурує профілі портів. При переміщенні VM фабрика VCS автоматично обновляє відповідні порти. Ви розгортаєте NSX – і Brocade VCS Gateway дає можливість бачити потоки трафіку для швидкої діагностики. Простіше говорячи, для динамічних віртуальних мереж потрібні динамічні фізичні мережі. Фабрики Brocade VCS мають необхідну гнучкість, економічність й можливостями автоматизації.

При покупці продуктів Brocade можна скористатися опцією Brocade Network Subscription. Вона дозволить вам платити за мережу по її фактичному використанню. Ви також можете в будь-який час її модернізувати. Для більшості компаній щомісячна економія OpEx по поточних контрактах на обслуговування старих IP-мереж ЦОД або кампуса вже виправдує перехід на якісно нову мережу – мережу, що володіє знайомими властивостями колишніх IP-мереж при явних перевагах New IP.

Тільки Brocade пропонує безконфліктний шлях до New IP. В арсеналі Brocade – лідируюча платформа віртуальної маршрутизації, найефективніший контролер OpenDaylight і провідна в галузі мережна фабрика. Все це допоможе вам в еволюції до New IP, дозволить здійснити даний перехід у намічений термін і на ваших умовах.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – це об'єктно-орієнтована мова програмування високого рівня загального призначення з відкритим кодом. Це визначення може бути важким для новачків, тому розглянемо кожну характеристику окремо, щоб зрозуміти, що вона означає:

- Відкритий вихідний код: це безкоштовно та доступно для подальших покращень, таких як додавання корисних функцій або виправлення помилок.
- Об'єктноорієнтована: заснована не на функціях, але в об'єктах з певними атрибутами й методами.
- Високий рівень: зручний для людини, а не для комп'ютера.
- Загальне призначення: можна використовувати для створення будь-яких програм.

Ця мова використовується в будь-якому програмному забезпеченні, про яке ви тільки можете подумати. Ви можете використовувати його для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу та багато іншого. Також застосовується в науці про дані, аналізі даних, машинному навчанні, інженерії даних, веброзробці, розробці програмного забезпечення та інших галузях.

Переваги та недоліки Python

Переваги:

– Її легко читати, вчити та писати. Це мова програмування високого рівня з англійським синтаксисом. Це полегшує читання та розуміння коду. Її дійсно легко зрозуміти і вивчити, тому багато людей рекомендують Python новачкам. Вам потрібно менше рядків коду для виконання того ж завдання в порівнянні з іншими основними мовами, такими як C/C++ та Java.

– Підвищує продуктивність. Це дуже продуктивна мова. Завдяки її простоті розробники можуть зосередитися на розв'язанні проблеми. Їм не

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

потрібно витратити багато часу на розуміння синтаксису або поведінку мови програмування. Ви пишете менше коду та виконуєте більше завдань.

– Інтерпретована мова. Python мова, що інтерпретується, а це означає, що вона безпосередньо виконує код по рядку. Якщо сталася помилка, вона зупиняє подальше виконання та повідомляє про її виникнення. Вона показує лише одну помилку, навіть якщо у програмі їх кілька. Це спрощує налагодження.

– Динамічно типізована. Python не визначає тип змінної, доки ми не запустимо код. Вона автоматично надає тип даних, коли відбувається процес виконання. Фахівець може не турбуватися про оголошення змінних та типи даних.

– Безкоштовна та з відкритим вихідним кодом. Ця мова постачається під схваленою OSI ліцензією з відкритим вихідним кодом. Це робить його безкоштовним для використання та розповсюдження. Ви можете завантажити вихідний код, змінити його та навіть розповсюджувати свою версію. Це корисно для організацій, які хочуть використати свою версію для розробки.

– Підтримка великих бібліотек. Стандартна бібліотека Python є величезною, ви можете знайти майже всі функції, необхідні для вашого завдання. Таким чином ви не залежите від зовнішніх бібліотек.

– Портативність. У багатьох мовах, таких як C/C++, потрібно змінити свій код, щоб запустити програму на різних платформах. З Python все інакше. Ви тільки пишете один раз і запускаєте її будь-де.

Недоліки:

– Низька швидкість. Вище ми обговорювали, що це інтерпретована мова з динамічною типізацією. Порядкове виконання коду часто призводить до повільного виконання. Динамічна природа Python також є причиною її низької швидкості, оскільки їй доводиться виконувати додаткову роботу при виконанні коду. Тому вона не підходить для цілей, де швидкість важливий аспект проєкту.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

заснований на ній seaborn. Використовуючи їх, ми можемо створювати буквально всі види візуалізації: від найпростіших до складніших.

– Для машинного навчання. Машинне навчання (ML) є основою більшості завдань науки даних. Він є областю штучного інтелекту, пов'язаною з використанням алгоритмів, що дозволяють машинам вивчати закономірності та тенденції на основі історичних даних, щоб робити прогнози на основі невідомих даних. – Використовуючи методи ML, ми можемо створювати моделі, які можуть точно передбачити швидкість відтоку клієнтів компанії, оцінити ризик виникнення у людини певного захворювання, визначити оптимальне розташування автомобілів таксі й т.д. За допомогою Python ми можемо побудувати модель ML, використовуючи лише три рядки коду.

– Для розробки програмного забезпечення. Крім свого багатостороннього застосування в галузях науки про дані, Python використовується на кожному етапі розробки програмного забезпечення, включаючи контроль складання, автоматичну безперервну компіляцію, прототипування, відстеження помилок, тестування та обслуговування програмного забезпечення. За допомогою цієї мови можемо створювати аудіо- або відеопрограми на основі методів штучного інтелекту, машинного навчання, API (інтерфейсів прикладного програмування), GUI (графічних інтерфейсів) або будь-якого іншого типу програмного забезпечення.

– Для веброзробки. У той час як для створення візуальної частини вебсайту ми переважно будемо використовувати такі мови, як HTML, CSS та JavaScript, для його невидимої частини ми часто вибираємо Python. Серед масштабних вебсайтів та програм, створених за допомогою цієї мови, варто згадати Google, Facebook, Instagram, YouTube, Dropbox та Reddit.

– Для автоматизації задач/скриптингу. Це відмінний інструмент для написання програм для автоматизації різних завдань, що повторюються. Цей процес називається скриптингом. Зокрема, можна робити скрипти для роботи з файлами та папками. Наприклад, можна створювати, перейменовувати,

перетворювати, розділяти, об'єднувати або видаляти файли, перевіряти їх на наявність помилок. Ви також можете використовувати автоматизацію Python для пошуку та завантаження інформації з Інтернету, заповнення та надсилання онлайн-форм та надсилання регулярних повідомлень або електронних листів.

Яким фахівцям потрібно володіти Python:

- Фахівець з даних.
- Аналітик даних.
- Інженер даних.
- Інженер з машинного навчання.
- Журналіст даних.
- Архітектор даних.
- Повний стек веброзробника.
- Backend-розробник.
- DevOps-інженер.
- Інженер-програміст

Можемо зробити висновок, що Python ще довго буде популярною мовою, хоч і має низку недоліків. Цю мову використовують для створення вебсайтів, штучного інтелекту, серверів, програмного забезпечення для бізнесу, аналізу даних, машинного навчання, інженерії даних та для багатьох інших областей. Це перспективна і затребувана навичка, яка необхідна у всіх галузях.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи керування Ethernet-фабрикою Brocade VCS.

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

- а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;
- б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;
- в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;
- г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;
- д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;
- е) провести розрахунки по визначенню економічної ефективності розробленої системи;
- ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;
- з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Віртуалізація від NUAGE

Мережні інфраструктури стають вузьким місцем при «переході» до віртуалізованих ІТ і хмарної моделі одержання ІТ-сервісів. Динамічне середовище віртуальних машин (ВМ) дуже погано співвідноситься зі статичними мережами, у яких використовуються протоколи (наприклад, VLAN і STP), що застосовувалися в них і п'ять, і десять років тому.

Розглянемо типове завдання по розгортанню нового додатка або впровадженню нового сервісу, що складає з декількох функціональних компонентів, розміщених у різних підмережах. Раніше для цього була потрібна установка нових серверів, що займало кілька тижнів. Сучасні системи віртуалізації серверних ресурсів дозволяють виділити необхідні новому додатку або сервісу віртуальні машини з потрібною продуктивністю за лічені хвилини.

Однак відповідна підготовка мережі займає набагато більше часу. Конфігурування різних типів мережних пристроїв часто відбувається вручну й вимагає залучення великої кількості фахівців:

- один займається локальною мережею;
- інший – територіально розподіленою;
- третій – налаштуванням VLAN;
- четвертий – конфігуруванням межмережних екранів і т.д.

Як результат, всі переваги віртуалізації серверів у частині прискорення впровадження додатків зводяться на ні «повільністю» традиційних процедур налаштування мережі.

Вирішити проблему дозволяють нові технології віртуалізації мережі, які зараз активно розробляють і виводять на ринок багато виробників. Одне з

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

найбільш зрілих рішень, що вже розгорнули у своїх мережах кілька закордонних компаній, пропонує Nuage Networks. Воно підтримує більшість основних гіпервізорів (KVM, Xen, ESXi), систем оркестрації (включаючи VMware, OpenStack, CloudStack) і працює поверх мережної інфраструктури, побудованої на встаткуванні будь-якого виробника.

Рішення Nuage Networks Virtualized Services Platform (VSP) складається із трьох основних елементів:

- каталогу віртуалізованих сервісів (Virtualized Services Directory, VSD);
- агентів віртуальної маршрутизації/комутації (Virtual Routing & Switching, VRS);
- контролерів віртуалізованих сервісів (Virtualized Services Controller, VSC).

Каталог VSD забезпечує високий рівень абстракції. Він дозволяє описати мережні структури користувачів хмарної мережі й надати можливість адміністраторам самостійно управляти мережними ресурсами в заданих границях. Найважливішою функцією VSD є створення шаблонів мережних політик, не зв'язаних зі специфікою апаратного пристрою мережі. Надалі, при запуску конкретного екземпляра віртуалізованого додатка, на основі заданого шаблону створюється опис конфігурації його мережі. Наприклад, окремий шаблон можна створити для додатків (віртуальних машин), що поміщаються в «демілітаризовану» зону, доступ у яку може здійснюватися тільки по TCP і тільки через протокольний порт 80.

Агенти VRS відповідають за керування віртуальними мережними інтерфейсами відповідно до заданих шаблонів (політиками) і працюють під керуванням найпоширеніших гіпервізорів. Вони інкапсулюють генеруємий віртуальними машинами трафік і передають його між кінцевими точками (гіпервізорами) по тунелях. Таким чином, поверх фізичної інфраструктури мережі формується накладена мережа типу «фабрика».

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Агенти VRS одержують інструкції щодо того, що робити з конкретним трафіком, від контролера VSC, відповідального за цілісну роботу й подання загальної топології багатокористувальницької мережі. Оскільки цей контролер використовує ту ж операційну систему SROS, що й лінійка сервісних маршрутизаторів Alcatel-Lucent 7750, на базі якої працюють ряд найбільших у світі IP-мереж, він здатний управляти повнофункціональною маршрутизацією (на рівнях L 2-L4) для тисяч користувальницьких підмереж.

При запуску віртуальної машини, що виконує роль одного з функціональних компонентів віртуалізованого додатка, контролер VSC по запиті від VRS одержує з бази даних VSD екземпляр шаблону (instance). Потім цей шаблон завантажується в контролер, що автоматично набудовує всі мережні елементи, які можуть бути задіяні в обслуговуванні даного додатка. Ручне переконфігурування окремих мережних пристроїв виключається.

Шаблон, у якому визначені мережна конфігурація й правила обробки трафіку конкретного додатка, залишається прив'язаний до відповідних віртуальних машин поза залежністю від їхніх переміщень. Вони можуть розташовуватися в одному ЦОД або в декількох, переміщатися між фізичними серверами й територіально віддаленими площадками – віртуальна мережа буде автоматично підлаштовуватися. Якщо традиційна пошта завжди доставляється в конкретне фізичне місце, то електронна сьогодні доставляється на ПК, ноутбук і/або смартфон незалежно від того, де ви перебуваєте.

Прискорення інтеграції віртуалізованого додатка/сервісу в мережу, пояснює він, досягається за рахунок створення шаблону, у якому вказуються вимоги до мережі, включаючи конфігурацію підмереж, параметри якості обслуговування (QoS), налаштування безпеки та ін. Пізніше адміністратор може багаторазово тиражувати його при розгортанні екземплярів віртуалізованого додатка/сервісу.

На практичному семінарі фахівці Nuage Networks розгорнули демонстраційну мережу, до складу якої входили хости із двома типами

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

гіпервізорів: ESXi (під керуванням VMware vCenter) і KVM (під керуванням OpenStack Havana). За допомогою засобів VSD було продемонстроване формування структури накладеної мережі (поділ її на зони, підмережі), після чого подібні дії були виконані із застосуванням системи OpenStack – відповідні зміни автоматично з'явилися в каталозі Nuage VSD.

Потім віртуальні машини були запущені на хостах з гіпервізорами KVM і ESXi (в останньому випадку через vCenter). Вони містилися в ті зони (підмережі), які забезпечували необхідні ним параметри обслуговування. Трафік між гіпервізорами різних типів (KVM і ESXi) передавався в режимі L3 – тобто з маршрутизацією.

Найбільш затребуваний сценарій – «живаючи» міграція віртуальних машин з одного хоста на іншій. Ця процедура була ініційована із системи керування vCenter, а мережа автоматично перенастроїлася для обслуговування VM на новому місці. Шаблон, що визначає правила обробки трафіку додатка, автоматично треба за віртуальною машиною при її переїзді. При цьому ніякого втручання адміністратора не потрібно. Більше того, віртуальну мережу можна розтягти по VPN через територіально розподілену мережу (WAN) до офісної мережі компанії.

З огляду на той факт, що не всі підключені до мережі пристрої піддаються віртуалізації, рішення Nuage VSP передбачає механізми інтеграції апаратних компонентів (таких, наприклад, як сервери баз даних, межмережні екрани, різні шлюзи). Для сценаріїв, що не вимагають високої продуктивності, пропонується програмний модуль VRS Gateway. Для ситуацій з більшими обсягами трафіку розроблений апаратний шлюз 7850 VSG із продуктивністю більше 1 Тбіт/с, що підтримує інтерфейси 1GE, 10GE і 40GE.

3.2 Розробка структурної схеми

Brocade: віртуалізуючі фабрики й мережні функції

З 2008 року в результаті покупки Foundry Networks у лінійці компанії Brocade з'явилися IP-Рішення, і зараз у сферу її інтересів входять класичні IP-маршрутизатори L3, а також SDN, NFV і засобу оркестрації хмар (автоматизація виділення ресурсів у хмарі й керування ними). Останні реалізуються Brocade по SAN і LAN у рамках проекту OpenStack. На семінарі компанії Brocade, для якої рішення, орієнтовані на ЦОД, мають стратегічне значення, були представлені як уже успішно експлуатовані в замовників продукти, так і нові розробки.

З появою в нашій лінійці продуктів Ethernet ми взяли краще з миру FC, об'єднали ці інструменти з технологіями Ethernet і одержали Ethernet-фабрики – віртуальний розподілений Ethernet-комутатор. Фактично це новий підхід до побудови мережі передачі даних – з використанням одного логічного комутатора, керування яким здійснюється з єдиної точки, і при цьому забезпечується високий ступінь автоматизації. Зараз поряд з фабриками FC лінійка Brocade містить у собі Ethernet-фабрики – основні її рішення для ЦОД.

Ethernet-фабрика Brocade VCS

Ethernet-фабрика Brocade VCS реалізує віртуальний розподілений комутатор рівня доступу й агрегування. Вона практично не вимагає налаштування, підтримуючи «самоформування» і автовиявлення пристроїв у будь-якій топології, при цьому керування нею здійснюється як одним віртуальним комутатором. Для запуску фабрики досить задати її домен і ID, а потім можна користуватися єдиною консоллю керування всіма комутаторами.

Для підвищення надійності підтримується топологія Full Mesh, завдяки чому працездатність системи зберігається навіть при виході з ладу одного комутатора. Крім того, фабрика готова до конвергенції (FC 16 / FCo 10 Гбітс), підтримує три рівні балансування трафіку й здатна агрегувати (без попереднього налаштування) велика кількість каналів. Завдяки покадровому балансуванню

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

забезпечується рівномірний розподіл навантаження між портами комутаторів. За допомогою протоколу ECMP здійснюється балансування між еквівалентними за вартістю шляхами, що з'єднують кінцеві точки (незалежно від числа транзитних вузлів). А балансування навантаження між шлюзами VRRP-E L3 (до чотирьох шлюзів) поліпшує масштабованість і надійність рішення.

Недавно з'явилася в Brocade технологія дає можливість створювати «віртуальні фабрики» усередині VCS. Виділення логічних фабрик для кожного користувача в рамках поділюваної фізичної фабрики на основі TRILL Fine-Grained Labels (IETF draft) дозволяє повторно задіяти VLAN. У рамках кожної фізичної фабрики підтримується до 8000 віртуальних фабрик VCS.

Завдяки інтеграції з VMware vCenter, Ethernet-фабрика може одержувати інформацію про віртуальні машини, їх MAC- і IP-адресах, групах VLAN. Ця інформація автоматично заноситься в Ethernet-фабрику, і за замовчуванням створюються «профілі портів», до яких можна застосовувати різні налаштування – наприклад, задавати якість обслуговування QoS. Система настроюється однократно – надалі фабрика сама застосовує задані політики по MAC-адресі, на якому б з портів він не був ідентифікований. Одна фабрика поєднує до 48 комутаторів.

В останніх версіях фабрики Brocade реалізований VXLAN-шлюз VCS (VTEP) для VMware NSX – міст між віртуальними й фізичними ресурсами. Його можна використовувати для інтеграції із класичними мережами, що не підтримують VXLAN. Повна підтримка VCS реалізована в комутаторах Brocade VDX серії 6740 і маршрутизаторах 8770. У серії 6740 (SDN-ready) можна використовувати протокол OpenFlow 1.3. Відповідно до планів Brocade, контролер SDN з'явиться на ринку під кінець року. Він буде сполучати функції контролера й комутатора SDN, продовжуючи лінійку VDX 6740. Протокол OpenFlow підтримується також продуктами Brocade Fibre Channel.

На думку аналітиків, у найближчій перспективі будуть розгортатися переважно гібридні рішення, що сполучають традиційну мережну

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

інфраструктуру й SDN, тому більшість комутаторів SDN допускають програмне перемикання між SDN і комутацією L2/L3. Тим самим знижуються ризики впровадження нової технології. Brocade пропонує своє рішення. На відміну від інших вендорів компанія розробила комутатори й маршрутизатори з «гібридними портами», тобто вони можуть працювати в змішаному режимі, підтримуючи як традиційне керування, так і керування за допомогою контролера SDN. Такий режим зараз реалізується маршрутизаторами Brocade MLXe. SDN уже знаходить практичне застосування.

У даній роботі, система, що розроблена, допомагає знизити негативний вплив атак DDoS. Погрози виявляються за допомогою sFlow-RT (у режимі реального часу) і відправляються на колектор, де обробляються програмним забезпеченням централізованого керування мережею Brocade Network Advisor, здатним виступати також як контролер sFlow. Колектор сповіщає контролер, що потрібно використовувати правило OpenFlow DDoS на маршрутизаторі MLXe для протидії атаці DDoS. Все це відображено на структурній схемі системи (рисунок 3.1).

Поряд з SDN, у мережній галузі набирає популярність технологія NFV. Вона дозволяє за допомогою віртуалізації консолідувати різні мережні пристрої (межмережне встаткування, комутатори, маршрутизатори, балансувальники навантаження й т.п.) на стандартних серверах x86, що веде до зменшення вартості й збільшенню «живучості» систем. Перераховані пристрої стають віртуальними (функціонують у вигляді VM), так що здобувати дороге «залізо» не потрібно. Тим часом Brocade уже пропонує віртуальний маршрутизатор Vyatta vRouter з функціями концентратора VPN із шифруванням IPSec і межмережного екрана. Це відносно недороге й гнучке рішення для невеликої мережі Ethernet дає можливість надавати мережні функції «по запиті» і реалізує маршрутизацію, межмережний екран, VPN. Для розгортання таких VM можна використовувати існуючі стандартні сервери x86, здатні успішно замінити спеціалізовані системи, у тому числі рішення для інфраструктури зв'язку. vRouter інтегрований зі стандартними оркестраторами (OpenStack).

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

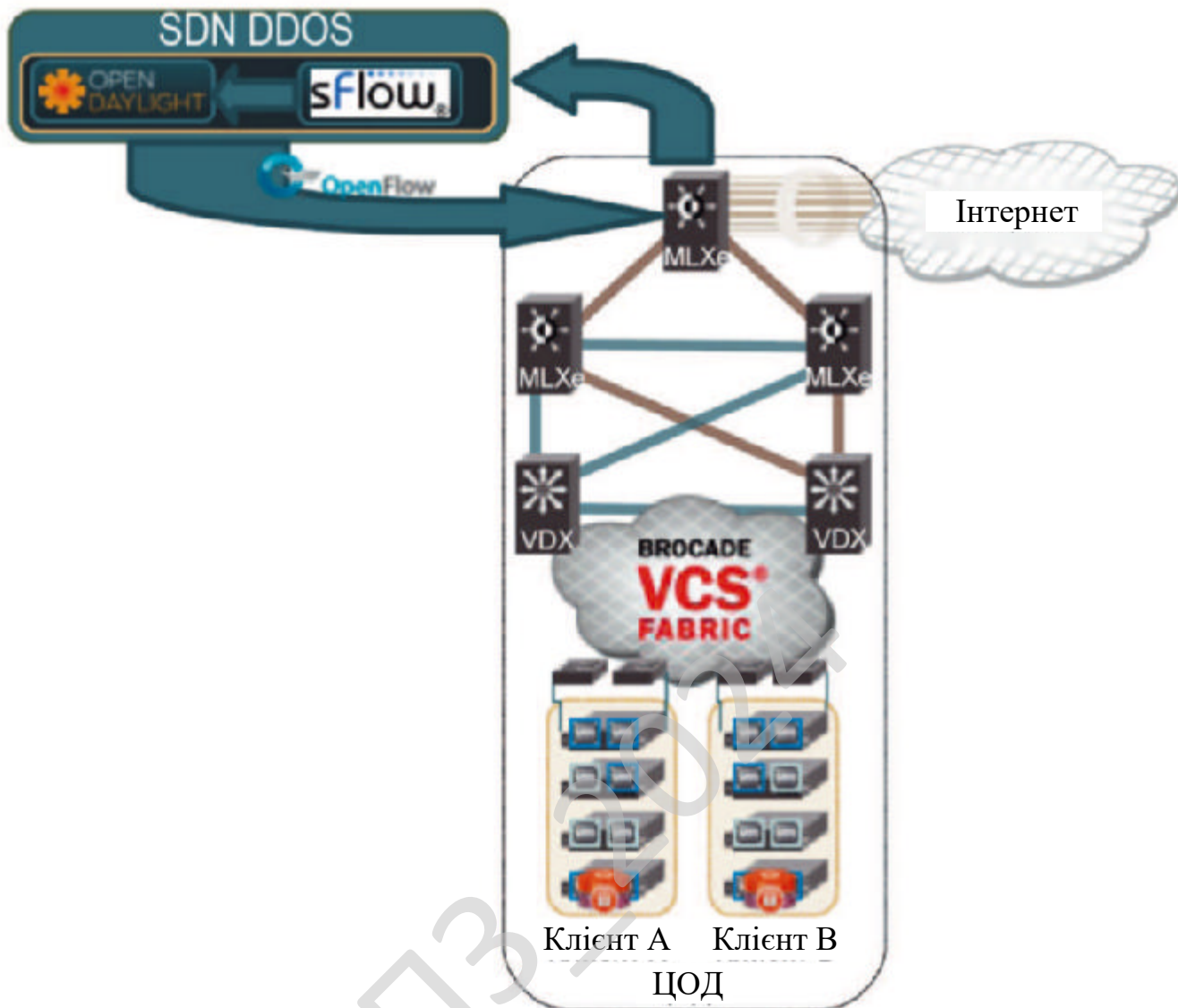


Рисунок 3.1 – Структурна схема системи

Ще один віртуальний продукт Brocade – повнофункціональний балансувальник трафіку додатків ADX. Будучи ключовим компонентом L 4-L7 в NFV, він динамічно балансує трафік додатків і прискорює запуск сервісів. Для цілей централізованого керування передбачається підтримка OpenStack і API. Віртуальний ADX можна розгортати у вигляді VM на сервері з гіпервізором або без гіпервізора, виділяючи йому весь сервер. Ці продукти дозволяють гнучко (по запиті) впроваджувати різні сервіси, однак пропускна здатність у них трохи нижче, ніж у фізичних пристроїв, – від 10 Мбіт/с до 3 Гбіт/с (обмеження залежить

від ліцензії, а не від швидкодії сервера). Продуктова лінійка Brocade, включаючи віртуальні пристрої, підтримує оркестрацію по протоколі OpenStack (через оркестратори сторонніх вендорів). Компанія бере активну участь у розробці стандартів OpenStack і пропонує розширення Dynamic Network Resource Management (DNRM) у рамках проекту Neutron, щоб спростити побудова мультивендорних мереж. Зі збільшенням обсягів трафіку й кількості підключених до мережі пристроїв конфігурування мереж і керування ними перетворюється в складне завдання. Щоб неї полегшити, потрібні серйозні зміни в підходах до побудови, експлуатації мереж і керуванню ними. Мережна інфраструктура розвивається й еволюціонує повільно, відстаючи від бурхливого розвитку засобів віртуалізації серверів і систем зберігання, однак саме мережі повинні стати надійною й гнучкою основою для середовища хмарних обчислень. В останні рік-два намітився явний прогрес, а розробки в області NFV відкривають для замовників перспективи переходу від апаратних до віртуальних мережних рішень зі збереженням всіх функцій, властивим апаратним продуктам.

3.3 Розробка функціональної схеми

Функціональна схема розробленої системи зображена на рисунку 3.2. Так, як функціональна схема є більш подібним описом функціональних можливостей структурної схеми, то вона буде представляти собою, більш детальний варіант структурної схеми.

З рисунку видно, що розроблена система складається з наступних частин:

- Блок моніторингу мережі Ethernet-фабрикою Brocade VCS.
- Блок аналізу мережної статистики.
- Блок визначення топології мережі Ethernet-фабрикою Brocade VCS.
- Блок виявлення аномального поведіння трафіку.
- Блок визначення виду DDoS-атаки Ethernet-фабрикою Brocade VCS.
- Блок зберігання результатів.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

вкладок для обміну пакетами за допомогою Ethernet, IP, TCP та UDP. «Діалог» між комп'ютерами відображає трафік між двома активними хостами. Номер, зазначений на вкладці після назви протоколу, означає кількість «діалогів» між хостами. Номер, зазначений на вкладці після назви протоколу, означає кількість «діалогів» між хостами, наприклад, «Ethernet:6».

4. Точки призначення. Точки призначення забезпечують статистику даними про відправку та прийом пакетів. Номер, зазначений на вкладці після назви протоколу, вказує на кількість точок призначення. Наприклад, «Ethernet:6».

5. Графіки I/O. Основний графік може бути отриманий за допомогою команди «IO graphs» (Графіки I/O). Ще декілька графіків можуть бути додані у тому ж вікні на основі фільтрів відображення.

6. Час чекання відповіді від сервісу. 13 протоколів доступні для глибокого аналізу.

7. RTP. RTP (Real-time Transport Protocol, протокол передачі у реальному часі, RFC 3550) – це протокол для передачі звука та відео через IP-мережу. Він працює у початку протоколу дейтаграм користувача (User Datagram Protocol, UDP). Він часто використовується у сукупності з протоколами SIP або H.233, забезпечуючи виконання сигнальних завдань.

8. SIP. SIP (Session Initiation Protocol, протокол встановлення сесії, RFC 3261) – це сигнальний протокол, який оголошує відео– або VoIP-сесії. Він працює разом із протоколом RTP, який використовується для передачі мультимедійних даних.

9. Виклики VoIP.

VoIP (Voice over IP, голосовий зв'язок за допомогою Інтернету) взагалі використовує два типи протоколів:

- сигнальні протоколи, такі, як SIP або H.323
- переносні протоколи, наприклад, RTP

10 Призначення. Відображення усіх IP-адрес призначення мережевих пакетів.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

11. Графік потоків. Графіки потоків забезпечує послідовний аналіз TCP-з'єднань. Перші три строки містять оголошення TCP-з'єднання з послідовностями «SYN», «SYN ACK» та «ACK».

12 HTTP. HTTP (Hypertext Transfer Protocol, протокол передачі гіпертексту) – це протокол типу «клієнт-сервер», який використовується для передачі HTML-файлів. HTTP-клієнт (у більшості випадків це web-браузер) відсилає HTTP-запит до web-серверу із полем «URL», який допомагає знайти потрібний файл. Web-сервер відповідає HTTP-пакетом та забезпечує клієнт необхідною web-сторінкою.

Меню «HTTP» містить три підменю:

- «Load Distribution» (Розподіл пакетів).
- «Packet Counter» (Лічильник пакетів).
- «Requests» (Запити).

14 IP-адреса. Відображення IP-адреси джерела або призначення мережевих пакетів.

15. Довжина пакету.

16. Тип порту. Відображення статистики портів TCP або UDP.

Блок визначення топології мережі Ethernet-фабрикою Brocade VCS

Блок визначення топології мережі Ethernet-фабрикою Brocade VCS:

– Блок використання відомостей із загальної системи моніторингу мережі Ethernet-фабрикою Brocade VCS, а не опитування пристрою додатково.

– Блок складання списку пристроїв у мережі Ethernet-фабрикою Brocade VCS, автоматично, ґрунтуючись на дані системи моніторингу.

– Блок побудови топології мережі Ethernet-фабрикою Brocade VCS, за станом на задану дату й відстеження змін у топології протягом часу.

– Блок автоматичного визначення рівнів ієрархії пристроїв у мережі Ethernet-фабрикою Brocade VCS, з виділенням периферійних, проміжних і центральних вузлів.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

– Блок побудови топології мережі Ethernet-фабрикою Brocade VCS, незалежно від використовуваної системи моніторингу й програмно-апаратних платформ.

– Блок комбінувати показників, на основі яких визначаються зв'язки між пристроями, і при їхньому обчисленні виконувати перевірку на значимість із використанням статистичних критеріїв.

Блок виявлення аномального поведження трафіку

Блок виявлення аномального поведження трафіку:

- Блок визначення профілю поведження нормального трафіку.
- Блок заміни направлення трафіку.
- Блок усунення аномального трафіку.

При первісному розгортанні рішення по DDoS адміністратор створює профіль поведження нормального трафіку. Цей процес іменується навчанням. Компанія використовує додатки звичайним образом протягом 24 годин протягом одного тижня, і трафік додатка проходить через Детектор аномалій трафіку. У період навчання Детектор аномалій трафіку збирає базову інформацію для розуміння нормальної роботи мережі Ethernet-фабрикою Brocade VCS, куди входять:

- Інтенсивність пакетів для кожного типу пакетів, обмірювана як кількість пакетів у секунду (pps).
- Співвідношення пакетів, наприклад, співвідношення пакетів SYN і пакетів FIN.
- Кількість одночасних TCP-з'єднань, відкритих одним джерелом.

Базова інформація збирається по кожній цільовій адресі хост-ПК, цільовій підмережі Ethernet-фабрикою Brocade VCS, вихідній адресі хост-ПК і вихідній підмережі Ethernet-фабрикою Brocade VCS.

Після закінчення періоду навчання Детектор аномалій трафіку переводиться в режим моніторингу, а Блок усунення аномального трафіку – у резервний режим готовності. Доти, поки немає DDoS-атаки Ethernet-фабрикою

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Brocade VCS, що активно розвивається, вхідний трафік з мережі Ethernet-фабрикою Brocade VCS Інтернет проходить через комутатор без якого-небудь втручання з боку Блоку усунення аномального трафіку. Копія вхідного трафіку посилає для аналізу на Детектор аномалій трафіку через зовнішній аналізатор протоколів (SPAN) або віртуальні списки ACL. Якщо Детектор аномалій трафіку виявляє аномальне в порівнянні з базовою інформацією поведження трафіку, починається процес усунення:

– Детектор аномалій трафіку направляє в Блок усунення аномального трафіку команду почати процес зміни напрямку.

– Блок усунення аномального трафіку відхиляє (“захоплює”) трафік, адресований на атакуєму IP-адресу, переадресуючи його на самого себе.

– Блок усунення аномального трафіку піддає трафік багатоступінчастому аналізу й застосовує контрзаходи для відділення благонадійних джерел від джерел DDoS-атаки Ethernet-фабрикою Brocade VCS. Цей процес іменується очищенням або вичищенням.

– Блок усунення аномального трафіку скидає трафік DDoS-атаки Ethernet-фабрикою Brocade VCS й пересилає благонадійний трафік назад на нормальний маршрут проходження трафіку до мети. Цей процес іменується ін'єкцією.

Детектор аномалій трафіку

Детектор аномалій трафіку – це пасивний пристрій моніторингу, що постійно виявляє ознаки, що вказують на присутність DDoS-атаки Ethernet-фабрикою Brocade VCS DDoS, спрямованої проти захищеного місця призначення, також іменованого зоною. Це може бути сервер, інтерфейс міжмережного екрана або інтерфейс маршрутизатора. Детектор аномалій трафіку аналізує копії всього вхідного трафіку, адресуємого в захищені зони, через SPAN або відгалуження пасивної мережі Ethernet-фабрикою Brocade VCS. Цей аналіз включає зіставлення поточного поведження трафіку з базовими граничними параметрами, які також іменуються зональною політикою, для виявлення аномального поведження трафіку. Якщо аномальне поведження виявлене й

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

виглядає як можлива атака, Детектор аномалій трафіку через позаполосну управлінську мережу Ethernet посилає в Блок усунення аномального трафіку сигнал про початок аналізу й усунення DDoS-атаки Ethernet-фабрикою Brocade VCS.

Блок усунення аномального трафіку

Блок усунення аномального трафіку – це автономний пристрій аналізу й фільтрації трафіку. Починаючи прийом трафіку, адресованого в конкретну зону, що, очевидно, піддається атаці, Блок усунення аномального трафіку проводить точний аналіз цього трафіку. Якщо результати аналізу підтверджують, що трафік злочинний, Блок усунення аномального трафіку застосовує контрзаходи, наприклад, механізми анти-спуфінга й фільтрацію різного рівня (таблиця 3.1). Кінцевий результат полягає в тому, що трафік зі злочинних джерел скидається, а трафік із благонадійних джерел пересилається в передбачений пункт призначення.

Можливі варіанти зміни напрямку трафіку

Фахівці з ІТ можуть використовувати описані нижче варіанти зміни напрямку трафіку з його пересиланням з мережі Ethernet-фабрикою Brocade VCS, розташованого вище лежачого оператора зв'язку, на Блок усунення аномального трафіку. Цей процес також іменується “захватом” трафіку:

– Повідомлення прикордонного шлюзового протоколу (Border Gateway Protocol, BGP) із Блок усунення аномального трафіку на маршрутизатори, розташовані у вище лежачого оператора зв'язку, з інформацією про те, що трафік, адресований на захищену адресу призначення, буде переспрямований на Блок усунення аномального трафіку.

– Використання зовнішніх механізмів зміни напрямку трафіку, наприклад, маршрутизаторів віддаленого відновлення BGP.

– Повідомлення про ін'єкцію очищеного трафіку на маршруті (Route Health Injection, RHI) від Блок усунення аномального трафіку для процесу маршрутизації в Catalyst серії 6500 або в систему нагляду серії 7600. Ці

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

повідомлення поміщають статичний маршрут у глобальну таблицю маршрутизації, у якій модуль Блок усунення аномального трафіку позначений як наступний вузол.

DDoS-атаки – Виявлення й усунення Ethernet-фабрикою Brocade VCS

У таблиці 3.1 перераховані типи атак DDoS, які може виявляти й усувати Блок усунення аномального трафіку.

Таблиця 3.1 – Категорії й особливі типи атак DDoS

Категорія атаки	Особливі типи атак
Атаки із заповненням смуги пропусчення	Лавинні атаки зі спуфінгом або без спуфінга: – Прапор TCP (SYN, SYN-ACK, ACK, FIN). – Протокол керування повідомленнями в Інтернет (ICMP). – Протокол користувальницьких датаграм (UDP). Приклади: лавинна атака SYN, smurf, LAND і UDP – лавинні Атаки.
	Атаки зомбі-комп'ютерів/мереж зомбі-комп'ютерів, у яких кожний вихідний зомбі-ПК або мережа відкриває множинні TCP-з'єднання й, у деяких випадках, видає багаторазові запити HTTP.
	Атаки DNS, наприклад, лавинна атака із запитами DNS.
Атаки з дефіцитом ресурсів	Атаки пакетного розміру, характерна риса яких – фрагментованні або великі пакети. Приклади: teardrop і ping-of-death.
	Атаки зомбі-комп'ютерів/мереж зомбі-комп'ютерів з низькою інтенсивністю схожі на Атаки із заповненням смуги пропусчення за тим виключенням, що кожне джерело Атаки посилає множинні запити з невеликим обсягом в одиницю часу.
	Атаки DNS з рекурсивним переглядом DNS.

Можливі варіанти ін'єкції трафіку

Ін'єкція трафіку – це процес, застосовуваний у Блок усунення аномального трафіку для пересилання очищеного благонадійного трафіку в точку призначення, що піддається атаці. Рішення підтримує різні варіанти ін'єкції трафіку. У варіанті 2-ого рівня топології, очищений трафік пересилається із Блок усунення аномального трафіку на статично-конфігуруєму наступну адресу заходу. Ця адреса перебуває на маршрутизаторі, розташованому нижче й з'єднаним з тої ж VLAN або підмережею, що й інтерфейс/VLAN ін'єкції трафіку. Ін'єкцію трафіку на 2-му рівні найпростіше конфігурувати, оскільки тут не потрібно вносити які-небудь істотні зміни в конфігурацію маршрутизатора, розташованого нижче.

Варіанти ін'єкції трафіку 3-го рівня:

- Маршрутизація й пересилання по VPN (VPN Routing and Forwarding, VRF).
- Маршрутизація на основі політики (Policy-Based Routing, PBR).
- Транкінг VLAN (VLAN Trunking).
- Інкапсуляція по загальній маршрутизації (GRE) або інкапсуляція IP у тунелі IP (IPIP).

Блок визначення виду DDoS-атаки Ethernet-фабрикою Brocade VCS

Блок визначення виду DDoS-атаки Ethernet-фабрикою Brocade VCS:

- Атака ARP-spoofing на таблицю mac-адрес комутаторів.
- Широкомовний шторм.
- Додатки, що роблять інтенсивне ширококомвне розсилання, наприклад: ширококомвні чати й мережні ігри.

					БКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52



Рисунок 3.2 – Функціональна схема системи

ARP-spoofing

ARP-spoofing – техніка DDoS-атаки в Ethernet мережах, що дозволяє перехоплювати трафік між хостами. Заснована на використанні протоколу ARP.

При використанні в розподіленій обчислювальній системи (РВМ) алгоритмів віддаленого пошуку існує можливість здійснення в такій мережі Ethernet-фабрикою Brocade VCS типової віддаленої DDoS-атаки «помилковий об'єкт РВМ». Аналіз безпеки протоколу ARP показує, що, перехопивши на атакуючому хості усередині даного сегмента мережі Ethernet-фабрикою Brocade VCS широкомовний ARP-запит, можна послати помилкову ARP-відповідь, у якій оголосити себе шуканим хостом (наприклад, маршрутизатором), і надалі активно контролювати мережний трафік дезінформованного хосту, впливаючи на нього за схемою «помилковий об'єкт РВМ».

Протокол ARP призначений для перетворення IP-адрес в MAC-адреси. Найчастіше мова йде перетворенні в адреси Ethernet, але ARP використовується й у мережах інших технологій: Token Ring, FDDI і інших.

Алгоритм роботи ARP

Протокол може використовуватися в наступних випадках:

1. Хост А хоче передати IP-пакет вузлу В, що перебуває з ним в одній мережі Ethernet-фабрикою Brocade VCS.
2. Хост А хоче передати IP-пакет вузлу В, що перебуває з ним у різних мережах, і користується для цього послугами маршрутизатора R.

У кожному із цих випадку вузлом А буде використовуватися протокол ARP, тільки в першому випадку для визначення MAC-адреси вузла В, а в другому – для визначення MAC-адреси маршрутизатора R. В останньому випадку пакет буде переданий маршрутизатору для подальшої ретрансляції.

Далі для простоти розглядається перший випадок, коли інформацією обмінюються вузли, що перебувають безпосередньо в одній мережі Ethernet-фабрикою Brocade VCS. (Випадок коли пакет адресований вузлу, який знаходиться за маршрутизатором, відрізняється тільки тим, що в пакетах

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

замість справжніх MAC-адрес комп'ютерів А і В коштує MAC-адреса комп'ютера С.

Після того як атака виконана, коли комп'ютер А хоче передати пакет комп'ютеру В, він знаходить в ARP-таблиці запис (він відповідає комп'ютеру С) і визначає з її MAC-адресу одержувача. Відправлений по цьому MAC-адресу пакет приходить комп'ютеру С замість одержувача. Комп'ютер С потім ретранслює пакет тому, кому він дійсно адресований – тобто комп'ютеру В.

Широкомовний шторм

Широкомовний шторм – лавина (сплеск) широкомовних пакетів (на другому рівні моделі OSI – кадрів). Розмноження некоректно сформованих широкомовних повідомлень у кожному вузлі приводить до експонентного росту їхнього числа й паралізує роботу мережі Ethernet-фабрикою Brocade VCS. Звичайно такі пакети використовуються мережними сервісами для оповіщення станцій про свою присутність. Вважається нормальним, якщо широкомовні пакети становлять не більше 10% від загального числа пакетів у мережі Ethernet-фабрикою Brocade VCS.

Також досить часто до шторму приводять кільця в мережі Ethernet-фабрикою Brocade VCS при некоректному налаштуванні протоколу Spanning Tree, оскільки в заголовку пакетів Ethernet немає інформації про час життя кадру, як, наприклад, у пакетів IP. Крім цього широкомовний шторм застосовується (навмисно) зломщиками.

Відповідно до галузевого стандарту де-факто число широкомовних і багатоадресних кадрів у мережі Ethernet-фабрикою Brocade VCS не повинне перевищувати 8-10% від загального числа кадрів.

Широкомовний кадр – це кадр, адресований всім станціям у домені мережі Ethernet-фабрикою Brocade VCS. Багатоадресний кадр – це кадр, адресований групі станцій у домені мережі Ethernet-фабрикою Brocade VCS. Оскільки широкомовний кадр адресований всім станціям, то, одержавши його, станції повинні перервати свою роботу й обробити такий кадр. Це сповільнює роботу всієї мережі Ethernet-фабрикою Brocade VCS.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Якщо відношення числа ширококомовних кадрів до загального числа кадрів більше 10%, то такий ефект називається "широкомовним штормом".

Широкомовний шторм може бути наслідком дефектів устаткування або неправильного настроювання параметрів активного встаткування.

Найчастіше це явище спостерігається в розподілених мережах NetWare, побудованих на основі комутаторів, або коли дані між сегментами або доменами мережі Ethernet-фабрикою Brocade VCS можуть передаватися більш ніж по одному потенційному шляху. Якщо один з комутаторів такої мережі Ethernet-фабрикою Brocade VCS не підтримує протокол Spanning Tree (звичайно IEEE 802.1d) або останній неправильно настроєний або збоїть, то в мережі Ethernet-фабрикою Brocade VCS починається некерована циркуляція ширококомовних кадрів.

Виявлення "широкомовного шторму" є не настільки тривіальним завданням, як це може здатися на перший погляд. Для його виявлення недостатньо взяти загальне число ширококомовних кадрів і поділити його на загальне число кадрів, що пройшли по мережі Ethernet-фабрикою Brocade VCS.

Для цього ви повинні визначити: яку частку становлять ширококомовні кадри в кожний інтервал часу (наприклад, за одну хвилину) і яка при цьому утилізація каналу зв'язку. Якщо, наприклад, за одну хвилину по мережі Ethernet-фабрикою Brocade VCS пройшло 4 кадри, а 2 з них були ширококомовними, то це ще не виходить, що ви спостерігаєте "широкомовний шторм".

Захист від ширококомовних штормів (broadcast storm)

Одна з характерних несправностей мережного програмного забезпечення – мимовільна генерація з високою інтенсивністю ширококомовних пакетів. Широкомовним штормом вважається ситуація, у якій відсоток ширококомовних пакетів перевищує 20% від загальної кількості пакетів у мережі Ethernet-фабрикою Brocade VCS. Звичайний комутатор або міст сліпо передає такі пакети на всі свої порти, як того вимагає його логіка роботи, засмічуючи, таким чином, мережу.

Боротьба із ширококомовним штормом у мережі Ethernet-фабрикою Brocade VCS, з'єднаної комутаторами, жадає від адміністратора відключення портів, що

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

генерують широкомовні пакети. Маршрутизатор не поширює такі ушкоджені пакети, оскільки в коло його завдань не входить копіювання широкомовних пакетів в усі поєднувані їм мережі Ethernet-фабрикою Brocade VCS.

Тому маршрутизатор є прекрасним засобом боротьби із широкомовним штормом, щоправда, якщо мережа розділена на достатню кількість підмереж.

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

3.4 Розробка діаграми процесів

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

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

Діаграми потоків даних містять чотири типи елементів:

- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).
- Зовнішні по відношенню до системи сутності.
- Потoki даних між елементами трьох попередніх типів.



Рисунок 3.3 – Діаграма взаємодії процесів

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

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

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

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

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

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

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

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

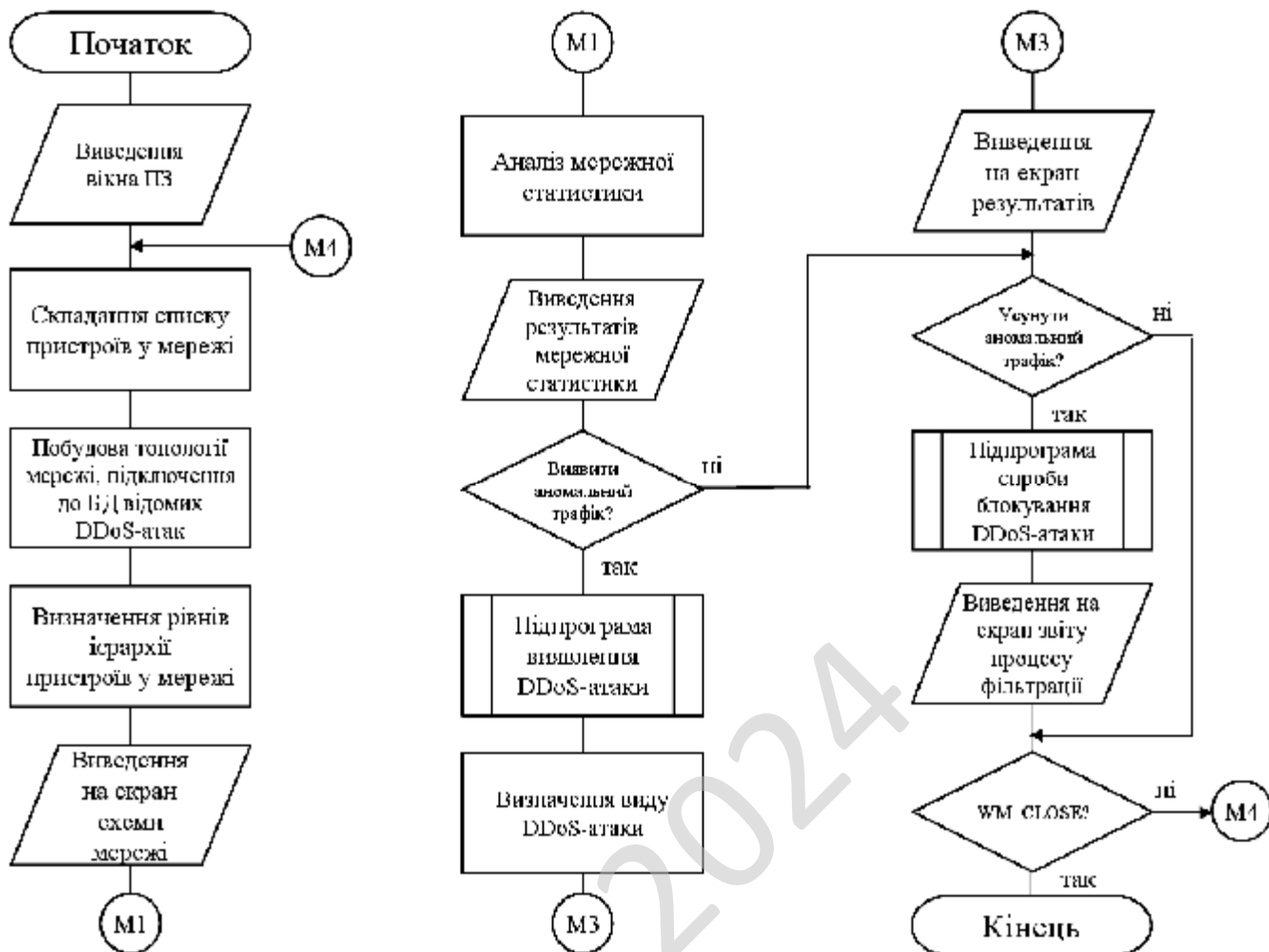


Рисунок 4.1 – Блок-схема основної програми

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

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

UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

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

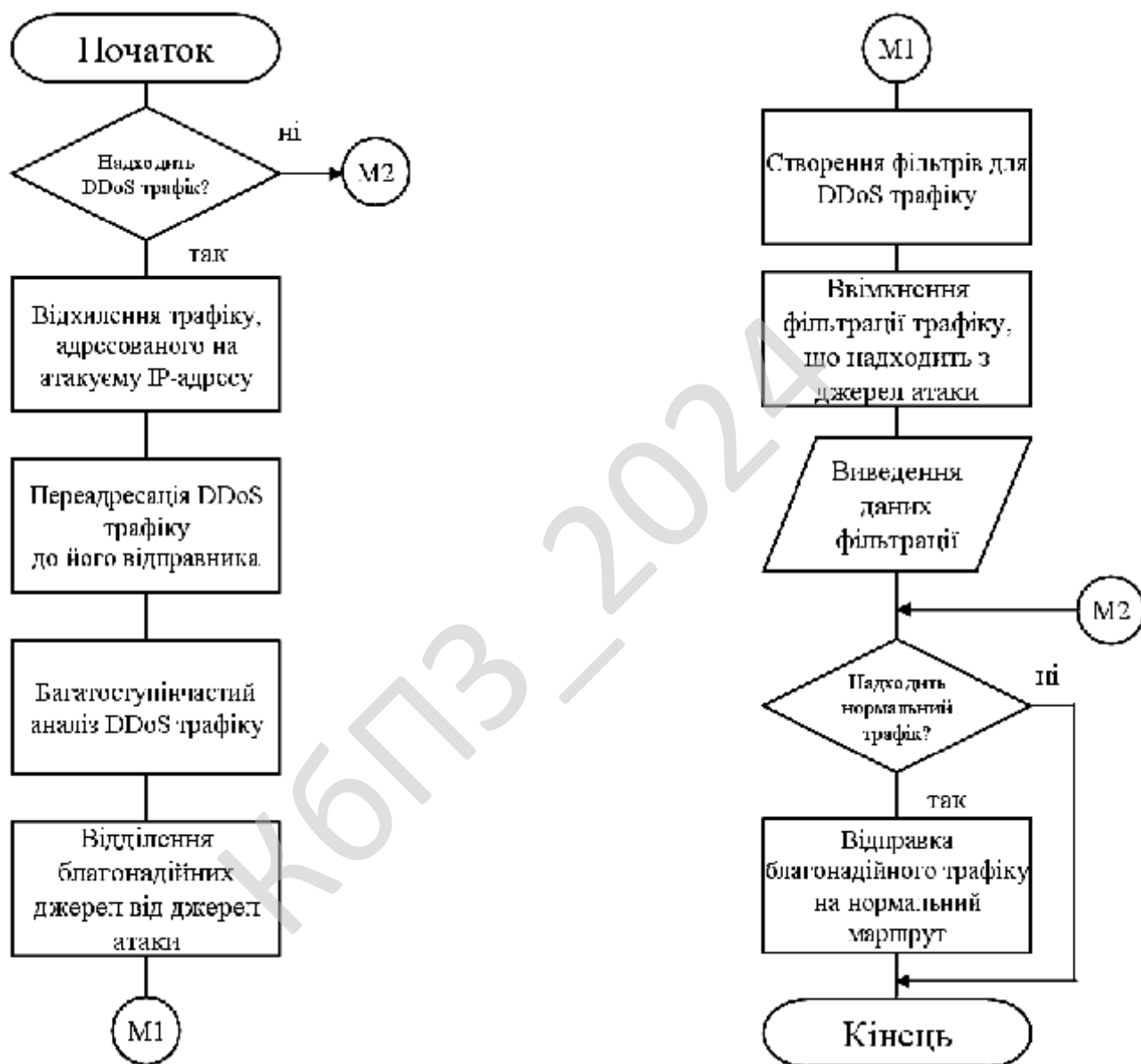


Рисунок 4.2 – Блок-схема роботи підпрограми

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання. Redmine

написано на мові Ruby і є ПЗ розробленим з використанням відомого веб-фреймворку Ruby on Rails, що означає легкість в розгортанні системи та її адаптації під конкретні вимоги. Для кожного проекту можна вести свої вікі та форуми.

Функціональні можливості:

- Ведення декількох проектів.
- Гнучка система доступу з використанням ролей.
- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.
- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.
- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.
- Облік часових витрат.
- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.

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

- Створення записів про помилки на основі отриманих листів
- Підтримка LDAP автентифікації.
- Можливість самореєстрації нових користувачів.
- Багатомовний інтерфейс (у тому числі українська мова).
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі.

Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

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

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

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

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

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

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

Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

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

- номер (ідентифікатор) дефекту;
- хто повідомив про дефект;
- дата і час виявлення дефекту;
- версія продукту, в якій виявлено дефект;
- серйозність (критичність) дефекту та пріоритет рішення;
- опис кроків для відтворення дефекту (неправильної поведінки програми);
- відповідальний за усунення дефекту;
- обговорення можливих рішень та їх наслідків;
- поточний стан виправлення дефекту;
- версії продукту, в якій дефект виправлений.

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

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

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

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

– 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.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

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 браузерями.

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.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

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).

При розробці ПЗ було використано підходи ризик-менеджменту – це система управління ризиками, яка включає в себе стратегію та тактику управління, направлені на досягнення основних цілей. Ефективний ризик-менеджмент включає:

- систему управління;
- систему ідентифікації і вимірювання;
- систему супроводження (моніторингу та контролю).

Сучасна наука представляє ризик як вірогідну подію, в результаті настання якої можуть відбутися позитивні, нейтральні або негативні наслідки. Якщо ризик припускає наявність як позитивних, так і негативних результатів, він відноситься до спекулятивних ризиків. Якщо ж наслідки негативні, або відсутні взагалі, такий ризик іменується чистим.

Мета ризик-менеджменту – підвищення конкурентоспроможності господарюючих суб'єктів за допомогою захисту від реалізації чистих ризиків.

Теорія ризик-менеджменту ґрунтується на трьох базових поняттях: корисності, регресії і диверсифікації.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

За ключовий етап ризик-менеджменту вважається етап вибору методів і інструментів управління ризиком.

Методи і інструментарій ризик-менеджменту

Базовими методами ризик-менеджменту є відмова від ризиків, зниження, передача і ухвалення.

Ризик-інструментарій значно ширший. Він включає політичні, організаційні, правові, економічні, соціальні інструменти, причому ризик-менеджмент як система допускає можливість одночасного застосування декількох методів і інструментів ризик-управління.

Найбільш часто вживаним інструментом ризик-менеджменту є страхування. Страхування припускає передачу відповідальності за відшкодування передбачуваного збитку сторонній організації (страхової компанії).

Прикладами інших інструментів можуть бути відмова від надмірно ризикової діяльності (метод відмови), профілактика або диверсифікація (метод зниження), аутсорсинг витратних ризикових функцій (метод передачі), формування резервів або запасів (метод ухвалення).

При розробці ПЗ було використано V-Model (або VEE модель) є моделлю розробки інформаційних систем (ІС), спрямованої на спрощення розуміння складнощів, пов'язаних з розробкою систем. Вона використовується для визначення єдиної процедури розробки програмного забезпечення, апаратного забезпечення та людино-машинного інтерфейсу.

Концепція V-подібної моделі була розроблена Німеччиною та США в кінці 1980-х років незалежно один від одного:

– Німецька V-модель була розроблена аерокосмічної компанією IAVG в Оттобрунні поряд з Мюнхеном у сприянні з Федеральним департаментом з закупівлі озброєнь в Кобленці, для Міністерства оборони Німеччини.

Модель була прийнята німецькою федеральною адміністрацією для цивільних потреб влітку 1992.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

– Американська V-Model (VEE) була розроблена національною радою з системної інженерії (міжнародна – з 1995 року) для супутникових систем, включаючи обладнання, програмне забезпечення та взаємодію з користувачами.

Сучасною версією V-Model є V-Model XT, яка була затверджена в лютому 2005 року. V-модель використовується для управління процесом розробки програмного забезпечення для німецької федеральної адміністрації.

Зараз вона є стандартом для німецьких урядових і оборонних проектів, а також для виробників ПЗ в Німеччині. V-Model являє собою скоріше набір стандартів у галузі проектів, що стосуються розробки нових продуктів. Ця модель багато в чому схожа з Prince2 і описує методи як для проектного управління, так і для системного розвитку.

Основні принципи

Основний принцип V-подібної моделі полягає в тому, що деталізація проекту зростає при русі зліва направо, одночасно з плином часу, і ні те, ні інше не може повернути назад. Ітерації в проекті виробляються по горизонталі, між лівою і правою сторонами літери.

Стосовно до розробки інформаційних систем V-Model – варіація каскадної моделі, в якій завдання розробки йдуть зверху вниз по лівій стороні букви V, а завдання тестування – вгору по правій стороні букви V. У середині V проводяться горизонтальні лінії, що показують, як результати кожної з фаз розробки впливають на розвиток системи тестування на кожній із фаз тестування.

Модель базується на тому, що приймально-здавальні випробування ґрунтуються, насамперед, на вимогах, системне тестування – на вимогах та архітектури, комплексне тестування – на вимогах, архітектурі та інтерфейсах, а компонентне тестування – на вимогах, архітектурі, інтерфейсах та алгоритмах

Цілі

V-модель забезпечує підтримку у плануванні та реалізації проекту. В ході проекту ставляться такі завдання:

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

1. Мінімізація ризиків: V-подібна модель робить проект більш прозорим і підвищує якість контролю проекту шляхом стандартизації проміжних цілей і опису відповідних їм результатів та відповідальних осіб. Це дозволяє виявляти відхилення в проекті і ризики на ранніх стадіях і покращує якість управління проектом.

2. Підвищення та гарантії якості: V-Model – стандартизована модель розробки, що дозволяє домогтися від проекту результатів бажаної якості. Проміжні результати можуть бути перевірені на ранніх стадіях. Універсальне документування полегшує читаність, зрозумілість та контрольованість.

3. Зменшення загальної вартості проекту: Ресурси на розробку, виробництво, управління і підтримку можуть бути заздалегідь прораховані та проконтрольовані. Отримувані результати також універсальні і легко прогножуються. Це зменшує витрати на подальші стадії та проекти.

4. Підвищення якості комунікації між учасниками проекту: Універсальний опис усіх елементів та умов полегшує взаєморозуміння всіх учасників проекту. Таким чином, зменшуються неточності у розумінні між користувачем, покупцем, постачальником і розробником.

Переваги:

– Користувачі V-Model беруть участь у розробці та підтримці V-моделі. Комітет з контролю за змінами підтримує проект і збирається раз на рік для обробки всіх отриманих запитів на внесення змін до V-Model.

– На старті будь-якого проекту V-подібна модель може бути адаптована під цей проект, так як ця модель не залежить від типів організацій та проектів.

– V-model дозволяє розбити діяльність на окремі кроки, кожен з яких буде включати в себе необхідні для нього дії, інструкції до них, рекомендації та докладне пояснення діяльності.

Обмеження. Наступні моменти не враховуються в V-моделі, але можуть бути розглянуті окремо, або можливо адаптувати модель під них:

– Не регулюється розміщення контрактів на обслуговування.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

– Організація і виконання управління, обслуговування, ремонту та утилізації системи не враховуються в V-моделі. Однак, планування і підготовка до цих операцій моделлю розглядаються.

– V-подібна модель більше стосується розробки програмного забезпечення в проекті, ніж всієї організації процесу.

Переваги:

– У моделі особливе значення надається плануванню, спрямованому на верифікацію та атестацію розроблювального продукту на ранніх стадіях його розробки. Фаза модульного тестування підтверджує правильність деталізованого проектування. Фази інтеграції та тестування реалізують архітектурне проектування або проектування на вищому рівні. Фаза тестування системи підтверджує правильність виконання етапу вимог до продукту і його специфікації.

– У моделі передбачені атестація та верифікація всіх зовнішніх і внутрішніх отриманих даних, а не тільки самого програмного продукту.

– У V-подібної моделі визначення вимог виконується перед розробкою проекту системи, а проектування ПО – перед розробкою компонентів.

– Модель визначає продукти, які повинні бути отримані в результаті процесу розробки, причому кожен отриманий дані повинні піддаватися тестуванню.

– Завдяки моделі менеджери проекту можуть відслідковувати хід процесу розробки, так як в даному випадку цілком можливо скористатися тимчасовою шкалою, а завершення кожної фази є контрольною точкою.

Недоліки:

– Модель не передбачає роботу з паралельними подіями.

– У моделі не передбачено внесення вимоги динамічних змін на різних етапах життєвого циклу.

– Тестування вимог в життєвому циклі відбувається занадто пізно, внаслідок чого неможливо внести змін, не вплинувши при цьому на графік виконання проекту.

– У модель не входять дії, спрямовані на аналіз ризиків.

– Деякий результат можна отримати тільки при досягненні низу букви V.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

- Q_{1024} – перетворення Q для 1024-бітного внутрішнього стану;
- r – кількість ітерацій у перетвореннях P і Q ($r = t$ в ДСТУ 7564:2014);
- S – внутрішній стан геш-функції;
- t – кількість блоків m , з яких складається повідомлення M , включаючи доповнення;
- v_i – i -й біт вхідної послідовності;
- Ω – завершальне перетворення;
- Купина- n – режим використання функції гешування з усіченням обчисленого гешзначення до розміру n біт.

Загальні положення

Під функцією гешування H розуміється залежне від вектора ініціалізації IV відображення послідовності біт M у геш-значення $H(M)$ фіксованої довжини n . ДСТУ 7564:2014 визначає функцію гешування, яка виконує перетворення «Купина-256» або «Купина-512», що забезпечують обчислення геш-значення з довжинами 256 або 512 біт відповідно. Геш-значення довжиною 256 бітів додатково може бути усічено до бітової послідовності довжиною від 8 до 248 біт з кроком у 8 біт, 512 бітів може бути усічене до бітової послідовності довжиною від 264 до 504 біт з кроком у 8 біт. Режим роботи для формування геш-значення довжиною n біт позначається як «Купина- n ». Основними режимами роботи функції гешування, що рекомендуються до застосування, є «Купина-256», «Купина-384» і «Купина-512».

Структура перетворення

Функція гешування, визначена в ДСТУ 7564:2014, формує геш-значення для повідомлення, що складається з бітової послідовності довжини від 0 біт (порожній рядок) до $2^{96}-1$ біт.

При формуванні геш-значення повідомлення доповнюється, далі поділяється на l -бітні блоки m_0, \dots, m_t , після чого виконується обробка кожного блоку шляхом ітеративного виконання функції стиснення φ .

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

При цьому формуються значення $h_i = \varphi(h_{i-1}, m_i)$ де $i = 1, \dots, t$, а початкове значення $h_0 = IV$.

Після обробки останнього блоку повідомлення результуюче геш-значення обчислюється як $H(M) = \Omega(h_t)$, де Ω – завершальне перетворення, що повертає n - бітне значення, кратне 8 ($0 < n \leq l/2$).

КБПЗ_2024

					VKPM-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ керування Ethernet-фабрикою Brocade VCS яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Навігаційне меню: Файл; Вид; Інструменти; Довідка.
- Функції представлені у графічному вигляді (іконки).
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ.

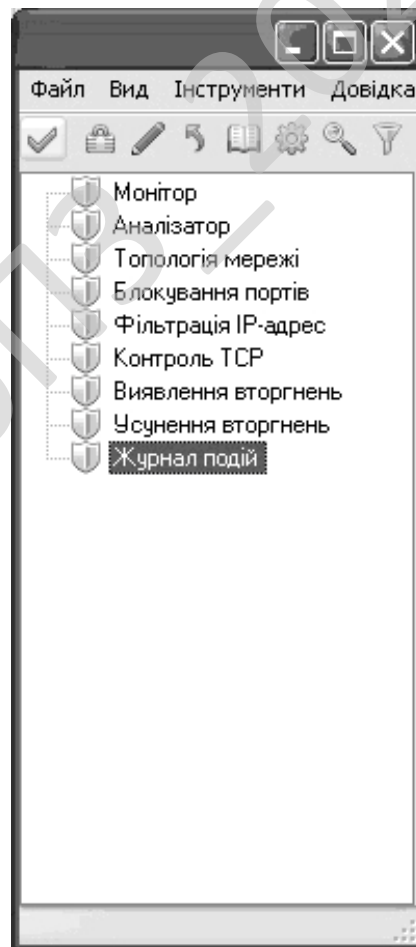


Рисунок 5.1 – Головне вікно ПЗ

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

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

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

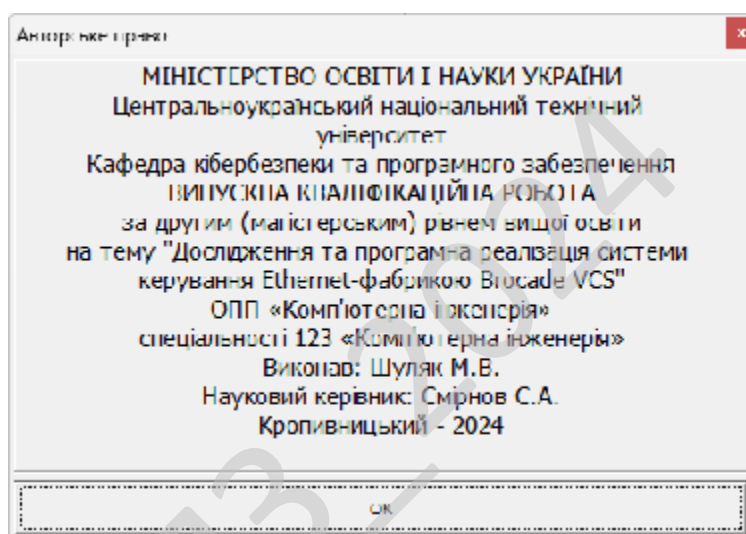


Рисунок 5.2 – Авторське право

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

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 ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ 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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

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

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

Проводилась оцінка:

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

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

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

Проводилось тестування чорної скриньки.

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

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

– Помилки ініціалізації та завершення.

Обрано умови розповсюдження – Shareware.

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

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

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

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

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи керування Ethernet-фабрикою Brocade VCS.

Метою розробки є дослідження та програмна реалізація системи керування Ethernet-фабрикою Brocade VCS.

Об'єктом дослідження є процес керування Ethernet-фабрикою Brocade VCS.

Предметом дослідження є методи керування Ethernet-фабрикою Brocade VCS.

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

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

- Удосконалено метод керування Ethernet-фабрикою Brocade VCS.
- Розроблено вітчизняний продукт керування Ethernet-фабрикою Brocade VCS, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінити привабливість проєкту за допомогою експертних оцінок для визначення його перспективності на ринку та ефективності впровадження можливо розпочати з визначення ключових критеріїв оцінки (рисунок 7.2):

Технічна привабливість (ТА)

- Інтеграція з існуючими мережевими рішеннями.
- Відповідність сучасним технологічним стандартам (SDN, автоматизація).
- Надійність і продуктивність системи.

Економічна привабливість (ЕА)

- Очікуване скорочення операційних витрат клієнта.
- Потенційна економія часу адміністрування.
- Рівень окупності інвестицій (ROI).

Ринкова привабливість (МА)

- Наявність попиту на ринку (рівень конкуренції).
- Відповідність потребам різних сегментів клієнтів (ЦОД, телекомунікації, корпорації).
- Можливість виходу на міжнародні ринки.

Інноваційність (ІН)

- Унікальність пропонованих рішень.
- Потенціал для впровадження нових технологій (наприклад, AI/ML).

Операційна привабливість (ОА)

- Простота впровадження та інтеграції.
- Необхідність навчання користувачів.
- Масштабованість та можливість адаптації під конкретні потреби клієнта.

Рисунок 7.2 – Критерії оцінювання

Далі триває формування експертної групи, залучаються 5-7 експертів із різних галузей: спеціалісти з мережових технологій, бізнес-аналітики, експерти з маркетингу, розробники та інтегратори.

Кожен експерт оцінює критерії в діапазоні від 1 (низька привабливість) до 5 (висока привабливість). Та відбувається зважування критеріїв за ваговими коефіцієнтами критеріїв залежно від їхньої важливості для проєкту: технічна

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

привабливість (ТА) – 30%, економічна привабливість (ЕА) – 25%, ринкова привабливість (МА) – 20%, інноваційність (ІН) – 15%, операційна привабливість (ОА) – 10%.

Таблиця 7.1 – Зведені результати експертних оцінок

Критерій	Експерт	Експерт	Експерт	Експерт	Експерт	Середній бал	Вага (%)	Зважений бал
	1	2	3	4	5			
ТА (технічна)	5	4	5	4	5	4.6	30	1.38
ЕА (економічна)	4	4	4	5	3	4.0	25	1.00
МА (ринкова)	4	3	4	4	4	3.8	20	0.76
ІН (інноваційна)	5	5	4	5	4	4.6	15	0.69
ОА (операційна)	3	4	4	3	3	3.4	10	0.34

Загальний зважений бал = 4.17. Середній зважений бал 4.17 свідчить про високу привабливість проекту. Найвищу оцінку отримали технічна привабливість та інноваційність, що підтверджує його конкурентоспроможність та відповідність сучасним вимогам. Рекомендується додатково посилити ринкову позицію шляхом маркетингових досліджень і збільшення операційної ефективності.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи керування Ethernet-фабрикою Brocade VCS найкраще використати комбінований підхід, що поєднує кілька методів, залежно від етапу проекту, особливостей реалізації та доступних даних.

На рисунку 7.3. наведено рекомендації щодо методів, які доцільно застосувати.

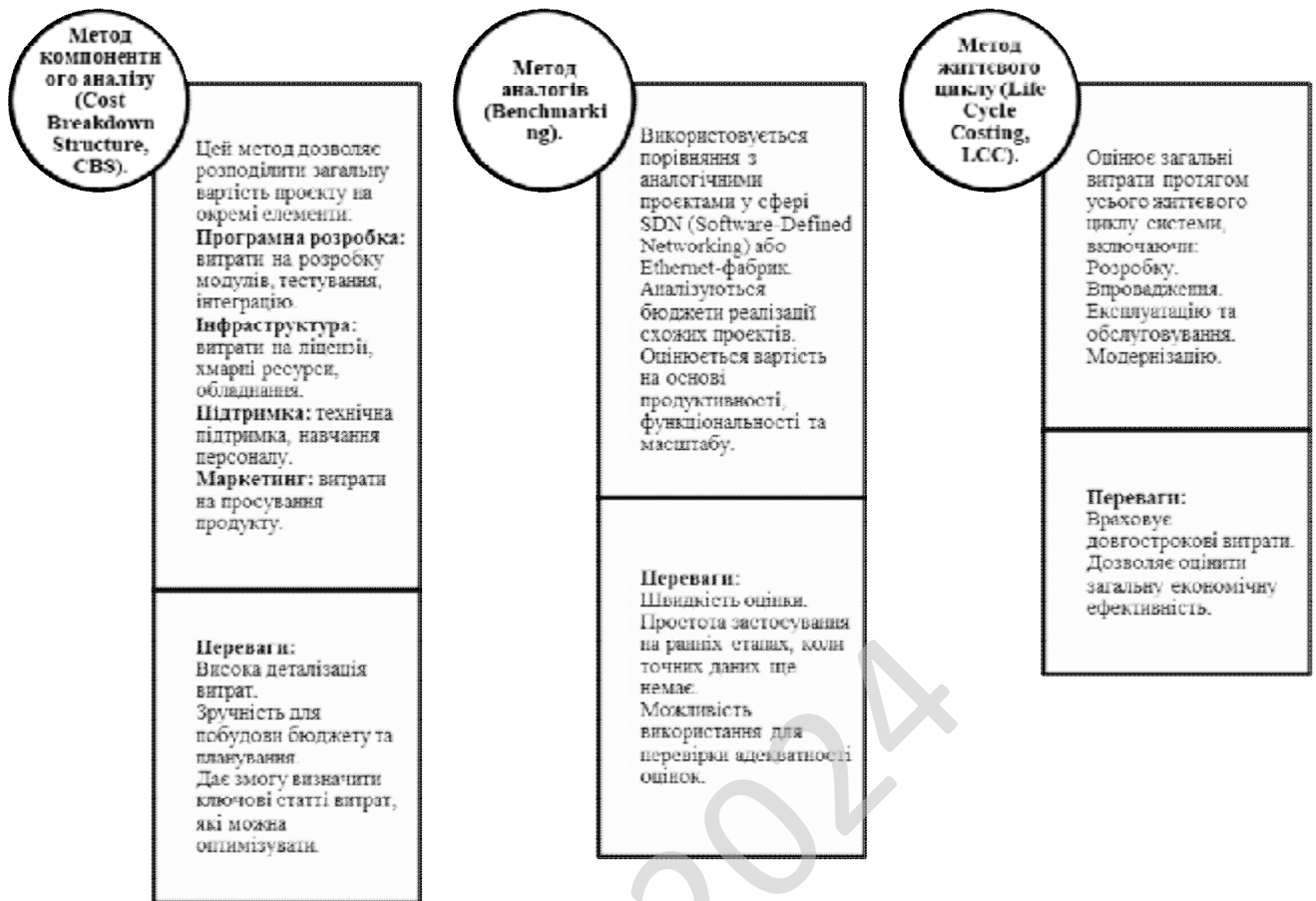


Рисунок 7.3 – Методи оцінювання проєкту

На ранніх етапах проєкту використовуйте метод аналогів для попередніх оцінок. Для детальної оцінки бюджету найкраще підходить метод компонентного аналізу (CBS). Для обґрунтування економічної вигоди перед інвесторами застосуйте LCC.

Таке поєднання підходів забезпечить точність оцінки та дозволить приймати обґрунтовані рішення.

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

Підприємство планує впровадити систему керування Ethernet-фабрикою Brocade VCS для модернізації мережі. Система дозволить покращити

Продовження таблиці 7.3

Розрахунок економічної ефективності

a. Зниження витрат на адміністрування та обслуговування

Після впровадження Brocade VCS, підприємство зможе зменшити витрати на адміністрування мережі на 40%, оскільки нова система надає централізоване управління та автоматизацію багатьох процесів.

Витрати на адміністрування після впровадження:

$$500,000 \times (1 - 0.40) = 300,000$$

Зниження витрат на адміністрування:

$$500,000 - 300,000 = 200,000 \text{ на рік}$$

b. Зниження витрат на обслуговування обладнання

Завдяки інтеграції Brocade VCS, зменшується необхідність у великій кількості окремих пристроїв і зменшується кількість технічних проблем, що дозволяє знизити витрати на обслуговування обладнання.

Витрати на обслуговування після впровадження:

$$300,000 \times (1 - 0.50) = 150,000$$

Зниження витрат на обслуговування:

$$300,000 - 150,000 = 150,000 \text{ на рік}$$

c. Вартість модернізації застарілого обладнання

Із впровадженням Brocade VCS не буде потреби у постійній модернізації застарілого обладнання, оскільки система має гнучкість для масштабування та підтримки різних технологій.

Зниження витрат на модернізацію: 200,000 одноразово

Оцінка річної економії після впровадження

Економія на адмініструванні та обслуговуванні мережі (щорічно):

$$200,000 + 150,000 = 350,000 \text{ на рік}$$

Початкові витрати на впровадження (одноразові):

$$600,000 + 150,000 + 50,000 = 800,000 \text{ одноразово}$$

Операційні витрати після впровадження (щорічно):

$$300,000 + 150,000 = 450,000 \text{ на рік}$$

Продовження таблиці 7.3

Розрахунок окупності інвестицій (ROI)

Тепер розрахуємо час окупності і ROI для оцінки економічної ефективності:

Час окупності: 2.29 роки

ROI (Return on Investment):-12.5%

7.5 Пропозиція алгоритму просування проєкту розробки ПЗ

Просування проєкту впровадження системи керування Ethernet-фабрикою Brocade VCS потребує стратегічного підходу, щоб забезпечити максимальну видимість, привабливість і досягнення бізнес-цілей (рисунок 7.4):



Рисунок 7.4 – Алгоритм для просування проєкту

Алгоритм просування проєкту включає в себе детальне планування стратегії просування, розробку та впровадження маркетингових матеріалів,

організацію демо та пілотних проєктів, а також активну підтримку клієнтів. Такий комплексний підхід дозволяє забезпечити успішне просування системи керування Ethernet-фабрикою Brocade VCS, збільшення попиту та максимізацію прибутку від її впровадження.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Оптимізація каналів збуту та шляхів реалізації проєкту програмної реалізації системи керування Ethernet-фабрикою Brocade VCS вимагає чіткої стратегії, яка дозволяє ефективно дистрибуувати продукт, досягати цільових аудиторій та забезпечувати максимальну видимість.

Пропонуємо окремо переглянути оптимізацію збуту та маркетингові канали підтримки збуту відповідно на рисунках 7.5 та 7.6.

Оптимізація каналів збуту для програмної реалізації системи керування Ethernet-фабрикою Brocade VCS повинна базуватися на чітко визначених стратегічних цілях, аналізі ринку, сегментації клієнтів, ефективному використанні онлайн і офлайн каналів продажу, а також гнучкому підході до ліцензування та пакуванню продукту. Важливо також постійно моніторити і адаптувати стратегію в залежності від результатів та потреб клієнтів.

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи керування Ethernet-фабрикою Brocade VCS можуть включати такі аспекти (рис. 7.7).

Успіх реалізації проєкту залежить від комплексного підходу до всіх цих аспектів. Кожен з них має критичне значення для забезпечення ефективної роботи системи керування Ethernet-фабрикою Brocade VCS у довгостроковій перспективі.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці є головною умовою безпеки та збереження здоров'я працівників. Згідно закону України “Про охорону праці” [3] кожна ІТ-компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

- Інструкцій з охорони праці по кожній професії і загальні.
- Положення про охорону праці.
- Накази з охорони праці.
- Журнали реєстрації та інструктажу.

Роботодавець створює відділ який працює відповідно до типового положення, яку затверджується центральним органом виконавчої влади і забезпечує виконання вимог державної політики у сфері охорони праці.

За недотриманням вимог, керівники ІТ компаній можуть бути притягнуті до відповідальності, яка виглядає у виді накладання штрафу. Якщо в результаті порушення умов охорони праці є постраждалі працівники то керівні особи ІТ компаній притягуються до кримінальної відповідальності.

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

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

Ці шкідливі фактори можуть привести до професійних захворювань.

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

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

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

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Оскільки робота програміста характеризується істотним зоровим навантаженням, то вимагає

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

належного освітлення. У приміщенні, в якому працюють програмісти, необхідно створити належний мікроклімат, параметри якого регламентуються нормативними документами – Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98. При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території;
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня або надмірна освітленість робочого місця;
- електромагнітні (у тому числі високочастотні) випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат.

Працю користувачів ЕОМ відносять до психічних форм праці з високим ступенем навантаження. Ця діяльність пов'язана зі сприйняттям зображення на екрані, постійним стеженням за його динамікою, розрізненням картин, схем, читанням тексту рукописних та друкованих матеріалів, введенням інформації з клавіатури, необхідністю підтримувати активну увагу високою ціною помилки. Будь-яка діяльність із застосуванням ЕОМ супроводжується необхідністю активації уваги та інших вищих психічних функцій, а організм людини, крім того, піддається впливу кількох десятків різноманітних факторів.

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

гігієнічні умови праці на робочому місці програміста відповідають вимогам. Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови №42 від 01.12.1999 Головного державного санітарного лікаря України робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

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

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії важкості робіт Іа, так і для розглянутого приміщення.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Вологість, %	Швидкість повітря, м/с	Температура, °С	Вологість, %	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-58	0,11
Тепла	23-25	50-70	0,1	24-25	50-65	0,2

У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

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

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

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

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

лампи освітлення закріплюються на стелі);

A – ширина приміщення, A = 5 м;

B – довжина приміщення, B = 6,2 м.

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

Знаючи індекс приміщення, за знаходимо $n = 0,23$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом лампам). Підставимо всі значення у формулу, визначимо світловий потік:
 $F=66717$ Лм.

Для розрахунку будемо використовувати світлодіодні панелі LED панель PL PFM 600 30W/3000K, світловий потік яких $F_{л} = 3000$ Лм.

Число ламп визначається по формулі:

$$N=F/F_{л}$$

де F – світловий потік,

$F_{л}$ – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо потрібну кількість світильників

$$N= 66717/ 3000=22,18 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 23 шт.

Висновки до розділу

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

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи керування Ethernet-фабрикою Brocade VCS.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів керування Ethernet-фабрикою Brocade VCS.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем керування Ethernet-фабрикою Brocade VCS.

– Досліджена система керування Ethernet-фабрикою Brocade VCS.

– На основі отриманих результатів досліджень створена програмна реалізація системи керування Ethernet-фабрикою Brocade VCS.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання керування Ethernet-фабрикою Brocade VCS.

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

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

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

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 7564:2014.

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

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шуляк М.В. Дослідження та програмна реалізація системи керування Ethernet-фабрикою Brocade VCS // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.
3. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.
4. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
5. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
6. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
7. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
8. 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.
9. 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.
10. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
11. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

12. 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.

13. 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.

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

15. 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.

16. 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.

17. 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.

18. 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.

19. 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.

20. 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.

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

22. 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.

23. 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.

24. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyzy, 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

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

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

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

38. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

39. 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.

40. 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.

41. Смірнов О.А., Коноплицька-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.

42. 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.

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

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

45. 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).

					ВКРМ-123.24.0048.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

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

47. Дреєва Г.М., Смірнов О.А., Дреєв 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.

48. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв 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.

49. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

50. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

51. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-123.24.0048.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Шуляк М.В.				Літ.	Аркуш	Аркушів
Перевірів	Смірнов С.А.						
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М		
Затв.	Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи керування Ethernet-фабрикою Brocade VCS.

2 Підстава для розробки

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

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи керування Ethernet-фабрикою Brocade VCS.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.24.0048.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи керування Ethernet-фабрикою Brocade VCS;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-123.24.0048.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0048.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

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

					ВКРМ-123.24.0048.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 113 аркушів.

10 Етапи розробки

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

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

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

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

					ВКРМ-123.24.0048.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти
_____ Смірнов С.А.

*Дослідження та програмна реалізація
системи керування Ethernet-фабрикою Brocade VCS*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 20

Літера: РП

Кропивницький – 2024 року

Основна програма

```
import random
import time
import string

# Генеруємо випадковий MAC-адрес
def generate_mac_address():
    return ':'.join(
        ''.join(random.choices('0123456789ABCDEF', k=2))
        for _ in range(6)
    )

# Емуляція пристроїв у фабриці
class Device:

    # Ініціалізація пристрою
    def __init__(self, name):
        self.name = name
        self.mac_address = generate_mac_address()
        self.status = "Disconnected"
        self.ports = {}
        self.logs = []

    # Підключення пристрою до мережі
    def connect(self):
        self.status = "Connected"
        self.log_event("Device connected to network")

    # Відключення пристрою від мережі
    def disconnect(self):
        self.status = "Disconnected"
        self.log_event("Device disconnected from network")

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

    # Додавання порту
    def add_port(self, port_name, speed):
        self.ports[port_name] = speed
        self.log_event(f"Added port {port_name} with speed {speed} Gbps")

# Фабрика Ethernet Brocade VCS
class EthernetFactory:
```

```

# Ініціалізація фабрики
def __init__(self):
    self.devices = []
    self.network_map = {}

# Додавання пристрою
def add_device(self, device_name):
    new_device = Device(device_name)
    self.devices.append(new_device)
    self.network_map[new_device.mac_address] = []
    new_device.log_event("Device added to Ethernet Factory")
    return new_device

# Підключення пристроїв один до одного
def connect_devices(self, device1, device2):
    if device1.status != "Connected":
        device1.connect()
    if device2.status != "Connected":
        device2.connect()
    self.network_map[device1.mac_address].append(device2.mac_address)
    self.network_map[device2.mac_address].append(device1.mac_address)
    device1.log_event(f"Connected to {device2.name}")
    device2.log_event(f"Connected to {device1.name}")

# Друк статусу мережі
def print_network_status(self):
    for device in self.devices:
        print(f"Device: {device.name}")
        print(f"  MAC Address: {device.mac_address}")
        print(f"  Status: {device.status}")
        print(f"  Ports: {device.ports}")
        print(f"  Logs:")
        for log in device.logs:
            print(f"    {log}")
        print("")

# Генеруємо Ethernet-фабрику
def main():
    factory = EthernetFactory()

    # Додаємо пристрої
    device1 = factory.add_device("Switch 1")
    device2 = factory.add_device("Router 1")
    device3 = factory.add_device("Server 1")

```

```
# Додаємо порти
device1.add_port("Port1", 10)
device1.add_port("Port2", 40)
device2.add_port("Port1", 10)
device3.add_port("Port1", 100)

# Підключаємо пристрої
factory.connect_devices(device1, device2)
factory.connect_devices(device2, device3)

# Виводимо стан мережі
factory.print_network_status()

if __name__ == "__main__":
    main()
```

КБПЗ_2024

Файл TrafficMonitor.py

```
# Додавання функцій моніторингу трафіку
class TrafficMonitor:

    # Ініціалізація моніторингу
    def __init__(self):
        self.traffic_data = {}

    # Реєстрація пристрою для моніторингу
    def register_device(self, device):
        self.traffic_data[device.mac_address] = 0
        device.log_event("Device registered for traffic monitoring")

    # Оновлення трафіку
    def update_traffic(self, device, bytes_transferred):
        if device.mac_address in self.traffic_data:
            self.traffic_data[device.mac_address] += bytes_transferred
            device.log_event(f"Transferred {bytes_transferred} bytes")

    # Вивід статистики трафіку
    def print_traffic_statistics(self):
        print("Traffic Statistics:")
        for mac, traffic in self.traffic_data.items():
            print(f" MAC Address: {mac}, Traffic: {traffic} bytes")
        print("")

# Додавання функцій VLAN
class VLANManager:

    # Ініціалізація VLAN менеджера
    def __init__(self):
        self.vlans = {}

    # Створення VLAN
    def create_vlan(self, vlan_id, vlan_name):
        if vlan_id not in self.vlans:
            self.vlans[vlan_id] = {
                "name": vlan_name,
                "devices": []
            }
            print(f"VLAN {vlan_id} - {vlan_name} created")

    # Додавання пристрою до VLAN
    def add_device_to_vlan(self, vlan_id, device):
        if vlan_id in self.vlans:
            self.vlans[vlan_id]["devices"].append(device)
            device.log_event(f"Added to VLAN {vlan_id} - {self.vlans[vlan_id]['name']}")
        else:
            print(f"VLAN {vlan_id} does not exist")

    # Друк VLAN конфігурації
```

```
def print_vlan_configuration(self):
    print("VLAN Configuration:")
    for vlan_id, vlan_data in self.vlans.items():
        print(f"  VLAN {vlan_id}: {vlan_data['name']}")
        print("    Devices:")
        for device in vlan_data["devices"]:
            print(f"      - {device.name} ({device.mac_address})")
    print("")

# Головна функція для тестування додаткових можливостей
def extended_main():
    factory = EthernetFactory()
    traffic_monitor = TrafficMonitor()
    vlan_manager = VLANManager()

    # Додаємо пристрої
    switch = factory.add_device("Core Switch")
    router = factory.add_device("Edge Router")
    server = factory.add_device("Main Server")
    client = factory.add_device("Client PC")

    # Реєструємо пристрої для моніторингу трафіку
    traffic_monitor.register_device(switch)
    traffic_monitor.register_device(router)
    traffic_monitor.register_device(server)
    traffic_monitor.register_device(client)

    # Оновлюємо трафік
    traffic_monitor.update_traffic(switch, 500)
    traffic_monitor.update_traffic(router, 1200)
    traffic_monitor.update_traffic(server, 750)
    traffic_monitor.update_traffic(client, 300)

    # Створюємо VLAN
    vlan_manager.create_vlan(10, "Management VLAN")
    vlan_manager.create_vlan(20, "Production VLAN")

    # Додаємо пристрої до VLAN
    vlan_manager.add_device_to_vlan(10, switch)
    vlan_manager.add_device_to_vlan(10, router)
    vlan_manager.add_device_to_vlan(20, server)
    vlan_manager.add_device_to_vlan(20, client)

    # Друк статусу фабрики
    factory.print_network_status()

    # Друк статистики трафіку
    traffic_monitor.print_traffic_statistics()

    # Друк конфігурації VLAN
    vlan_manager.print_vlan_configuration()
```

```
# Менеджер якості обслуговування (QoS)
class QoSManager:

    # Ініціалізація менеджера QoS
    def __init__(self):
        self.qos_policies = {}

    # Створення політики QoS
    def create_qos_policy(self, policy_name, priority_level):
        self.qos_policies[policy_name] = priority_level
        print(f"QoS Policy '{policy_name}' created with priority
{priority_level}")

    # Призначення політики QoS пристрою
    def assign_qos_policy(self, device, policy_name):
        if policy_name in self.qos_policies:
            device.log_event(f"Assigned QoS Policy: {policy_name} with priority
{self.qos_policies[policy_name]}")
        else:
            print(f"QoS Policy '{policy_name}' does not exist")

# Симуляція помилок у мережі
class NetworkSimulator:

    # Ініціалізація симулятора
    def __init__(self, factory):
        self.factory = factory
        self.failed_devices = set()

    # Випадкова симуляція відключення пристрою
    def simulate_failure(self):
        if self.factory.devices:
            failed_device = random.choice(self.factory.devices)
            failed_device.disconnect()
            self.failed_devices.add(failed_device)
            print(f"Simulated failure on device: {failed_device.name}")

    # Відновлення пристрою
    def recover_device(self, device):
        if device in self.failed_devices:
            device.connect()
            self.failed_devices.remove(device)
            device.log_event("Recovered from failure")
            print(f"Device {device.name} recovered successfully")

# Головна функція з розширеною логікою
def advanced_main():
    factory = EthernetFactory()
    qos_manager = QoSManager()
    simulator = NetworkSimulator(factory)
```

```
# Додаємо пристрої
switch = factory.add_device("Core Switch")
router = factory.add_device("Edge Router")
server = factory.add_device("Main Server")
client = factory.add_device("Client PC")

# Створюємо QoS політики
qos_manager.create_qos_policy("High Priority", 1)
qos_manager.create_qos_policy("Low Priority", 5)

# Призначаємо QoS політики
qos_manager.assign_qos_policy(switch, "High Priority")
qos_manager.assign_qos_policy(client, "Low Priority")

# Симулюємо помилку
simulator.simulate_failure()

# Друк статусу мережі
factory.print_network_status()

# Відновлюємо пристрої
simulator.recover_device(switch)

# Повторний друк статусу мережі після відновлення
factory.print_network_status()

if __name__ == "__main__":
    advanced_main()
```

Файл DynamicRouter.py

```

# Маршрутизатор для динамічної маршрутизації
class DynamicRouter:

    # Ініціалізація маршрутизатора
    def __init__(self):
        self.routing_table = {}

    # Додавання маршруту
    def add_route(self, source_mac, destination_mac, next_hop):
        if source_mac not in self.routing_table:
            self.routing_table[source_mac] = {}
        self.routing_table[source_mac][destination_mac] = next_hop
        print(f"Route added: {source_mac} -> {destination_mac} via {next_hop}")

    # Видалення маршруту
    def remove_route(self, source_mac, destination_mac):
        if source_mac in self.routing_table and destination_mac in
self.routing_table[source_mac]:
            del self.routing_table[source_mac][destination_mac]
            print(f"Route removed: {source_mac} -> {destination_mac}")
        else:
            print("Route not found")

    # Пошук маршруту
    def find_route(self, source_mac, destination_mac):
        if source_mac in self.routing_table and destination_mac in
self.routing_table[source_mac]:
            return self.routing_table[source_mac][destination_mac]
        else:
            return None

    # Виведення таблиці маршрутизації
    def print_routing_table(self):
        print("Routing Table:")
        for source, destinations in self.routing_table.items():
            for destination, next_hop in destinations.items():
                print(f" {source} -> {destination} via {next_hop}")
        print("")

# Автоматичне виявлення пристроїв у мережі
class DeviceDiscovery:

    # Ініціалізація системи виявлення
    def __init__(self, factory):
        self.factory = factory

    # Пошук нових пристроїв
    def discover_devices(self):
        discovered_devices = []
        for _ in range(random.randint(1, 3)):
            new_device_name = f"DiscoveredDevice-{random.randint(1000, 9999)}"

```

```
        new_device = self.factory.add_device(new_device_name)
        discovered_devices.append(new_device)
    print(f"Discovered {len(discovered_devices)} new devices")
    return discovered_devices

# Розширена функція для управління всіма компонентами
def ultimate_main():
    factory = EthernetFactory()
    router = DynamicRouter()
    discovery = DeviceDiscovery(factory)

    # Додаємо пристрої
    switch = factory.add_device("Core Switch")
    router_device = factory.add_device("Main Router")
    server = factory.add_device("Database Server")

    # Створюємо таблицю маршрутизації
    router.add_route(switch.mac_address, server.mac_address,
router_device.mac_address)
    router.add_route(server.mac_address, switch.mac_address,
router_device.mac_address)

    # Виведення таблиці маршрутизації
    router.print_routing_table()

    # Додаємо нові пристрої через автоматичне виявлення
    new_devices = discovery.discover_devices()

    # Додаємо маршрути для нових пристроїв
    for device in new_devices:
        router.add_route(switch.mac_address, device.mac_address,
router_device.mac_address)
        router.add_route(device.mac_address, switch.mac_address,
router_device.mac_address)

    # Виведення оновленої таблиці маршрутизації
    router.print_routing_table()

    # Друк статусу мережі
    factory.print_network_status()

if __name__ == "__main__":
    ultimate_main()
```

Файл PacketProcessor.py

```
# Модуль обробки пакетів
class PacketProcessor:

    # Ініціалізація обробника пакетів
    def __init__(self):
        self.processed_packets = 0
        self.dropped_packets = 0

    # Обробка пакета
    def process_packet(self, packet_size, device):
        if random.random() < 0.9:
            self.processed_packets += 1
            device.log_event(f"Processed packet of size {packet_size} bytes")
        else:
            self.dropped_packets += 1
            device.log_event("Packet dropped due to overload")

    # Друк статистики обробки
    def print_statistics(self):
        print(f"Processed packets: {self.processed_packets}")
        print(f"Dropped packets: {self.dropped_packets}")
        print("")

# Модуль безпеки
class SecurityModule:

    # Ініціалізація модулю безпеки
    def __init__(self):
        self.encryption_keys = {}
        self.detected_threats = 0

    # Генерація ключа шифрування
    def generate_encryption_key(self, device):
        key = ''.join(random.choices(string.ascii_letters + string.digits,
k=16))
        self.encryption_keys[device.mac_address] = key
        device.log_event("Encryption key generated")

    # Перевірка трафіку на загрози
    def scan_traffic(self, traffic_data):
        if random.random() < 0.05: # 5% шанс знайти загрозу
            self.detected_threats += 1
            print("Threat detected in traffic!")
        else:
            print("No threats found")

    # Виведення статистики загроз
    def print_security_statistics(self):
        print(f"Detected threats: {self.detected_threats}")
        print("")
```

```
# Симуляція перевантаження мережі
class NetworkOverloadSimulator:

    # Ініціалізація симулятора перевантаження
    def __init__(self, factory, packet_processor):
        self.factory = factory
        self.packet_processor = packet_processor

    # Генерація трафіку для перевантаження
    def generate_traffic(self):
        for device in self.factory.devices:
            for _ in range(random.randint(5, 20)):
                packet_size = random.randint(64, 1500)
                self.packet_processor.process_packet(packet_size, device)

# Головна функція з додатковими елементами
def overload_main():
    factory = EthernetFactory()
    packet_processor = PacketProcessor()
    security = SecurityModule()
    overload_simulator = NetworkOverloadSimulator(factory, packet_processor)

    # Додаємо пристрої
    switch = factory.add_device("Core Switch")
    server = factory.add_device("Application Server")
    client = factory.add_device("User PC")

    # Генерація ключів шифрування
    security.generate_encryption_key(switch)
    security.generate_encryption_key(server)
    security.generate_encryption_key(client)

    # Генерація мережевого трафіку
    overload_simulator.generate_traffic()

    # Перевірка трафіку на загрози
    traffic_sample = [random.randint(100, 1500) for _ in range(10)]
    for packet in traffic_sample:
        security.scan_traffic(packet)

    # Виведення статусу фабрики
    factory.print_network_status()

    # Виведення статистики обробки пакетів
    packet_processor.print_statistics()

    # Виведення статистики безпеки
    security.print_security_statistics()

if __name__ == "__main__":
    overload_main()
```

```
# Журналювання фабрики
class FactoryLogger:

    # Ініціалізація журналу
    def __init__(self):
        self.logs = []

    # Додавання запису до журналу
    def add_log(self, message):
        timestamp = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
        log_entry = f"[{timestamp}] {message}"
        self.logs.append(log_entry)

    # Друк журналу
    def print_logs(self):
        print("Factory Logs:")
        for log in self.logs:
            print(log)
        print("")

# Модуль для симуляції конфліктів
class ConflictSimulator:

    # Ініціалізація симулятора конфліктів
    def __init__(self, factory, logger):
        self.factory = factory
        self.logger = logger

    # Симуляція конфлікту
    def simulate_conflict(self):
        if len(self.factory.devices) >= 2:
            device1, device2 = random.sample(self.factory.devices, 2)
            conflict_type = random.choice(["IP Conflict", "MAC Conflict", "Port
Overload"])
            self.logger.add_log(f"Conflict detected between {device1.name} and
{device2.name}: {conflict_type}")
            print(f"Conflict simulated: {device1.name} vs {device2.name}
({conflict_type})")

# Хмарний сервер для управління
class CloudServer:

    # Ініціалізація хмарного сервера
    def __init__(self):
        self.connected_factories = {}

    # Реєстрація фабрики
    def register_factory(self, factory_name, factory):
        self.connected_factories[factory_name] = factory
        print(f"Factory '{factory_name}' registered with Cloud Server")
```

```
# Моніторинг фабрики
def monitor_factory(self, factory_name):
    if factory_name in self.connected_factories:
        factory = self.connected_factories[factory_name]
        print(f"Monitoring Factory: {factory_name}")
        factory.print_network_status()
    else:
        print(f"Factory '{factory_name}' not found")

# Розширена головна функція
def ultimate_extension_main():
    factory = EthernetFactory()
    logger = FactoryLogger()
    conflict_simulator = ConflictSimulator(factory, logger)
    cloud_server = CloudServer()

    # Додаємо пристрої
    switch = factory.add_device("Core Switch")
    router = factory.add_device("Main Router")
    server = factory.add_device("File Server")

    # Додаємо події до журналу
    logger.add_log(f"Device {switch.name} added")
    logger.add_log(f"Device {router.name} added")
    logger.add_log(f"Device {server.name} added")

    # Симулюємо конфлікти
    for _ in range(3):
        conflict_simulator.simulate_conflict()

    # Реєстрація фабрики на хмарному сервері
    cloud_server.register_factory("Main Factory", factory)

    # Моніторинг фабрики через хмару
    cloud_server.monitor_factory("Main Factory")

    # Виведення журналу подій
    logger.print_logs()

if __name__ == "__main__":
    ultimate_extension_main()
```

```
# Управління енергоспоживанням
class PowerManager:

    # Ініціалізація менеджера енергії
    def __init__(self):
        self.device_power = {}

    # Налаштування споживання енергії
    def set_power_usage(self, device, watts):
        self.device_power[device.mac_address] = watts
        device.log_event(f"Power usage set to {watts}W")

    # Обчислення загального енергоспоживання
    def calculate_total_power(self):
        total_power = sum(self.device_power.values())
        print(f"Total Power Consumption: {total_power}W")
        return total_power

    # Перехід у режим енергозбереження
    def enable_power_saving(self, device):
        if device.mac_address in self.device_power:
            self.device_power[device.mac_address] *= 0.5
            device.log_event("Power-saving mode enabled")
        else:
            print(f"Device {device.name} not found in power management list")

# Інтеграція IoT-пристроїв
class IoTIntegration:

    # Ініціалізація IoT інтеграції
    def __init__(self):
        self.iot_devices = []

    # Додавання IoT-пристрою
    def add_iot_device(self, device_name):
        iot_device = {
            "name": device_name,
            "status": "Online",
            "data": {}
        }
        self.iot_devices.append(iot_device)
        print(f"IoT Device '{device_name}' added")

    # Оновлення даних IoT-пристрою
    def update_iot_data(self, device_name, data_key, data_value):
        for device in self.iot_devices:
            if device["name"] == device_name:
                device["data"][data_key] = data_value
                print(f"IoT Device '{device_name}' updated: {data_key} =
{data_value}")
```

```
# Виведення статусу IoT-пристроїв
def print_iot_status(self):
    print("IoT Devices Status:")
    for device in self.iot_devices:
        print(f"  Name: {device['name']}, Status: {device['status']}, Data:
{device['data']}")
    print("")

# Оптимізація мережі
class NetworkOptimizer:

    # Ініціалізація оптимізатора
    def __init__(self, factory):
        self.factory = factory

    # Аналіз і оптимізація топології
    def optimize_topology(self):
        print("Optimizing network topology...")
        for device in self.factory.devices:
            if len(device.ports) < 2:
                device.add_port(f"AutoPort-{len(device.ports) + 1}", 10)
                device.log_event("Additional port added during optimization")
        print("Network topology optimized")

    # Автоматичне налаштування трафіку
    def optimize_traffic(self):
        print("Optimizing traffic flow...")
        for device in self.factory.devices:
            if device.status == "Connected":
                device.log_event("Traffic flow optimized")
        print("Traffic optimization completed")

# Головна функція з новими модулями
def hyper_advanced_main():
    factory = EthernetFactory()
    power_manager = PowerManager()
    iot_integration = IoTIntegration()
    optimizer = NetworkOptimizer(factory)

    # Додаємо пристрої
    switch = factory.add_device("Core Switch")
    server = factory.add_device("Database Server")
    iot_gateway = factory.add_device("IoT Gateway")

    # Налаштовуємо енергоспоживання
    power_manager.set_power_usage(switch, 150)
    power_manager.set_power_usage(server, 300)
    power_manager.set_power_usage(iot_gateway, 200)
    power_manager.enable_power_saving(switch)

    # Додаємо IoT-пристрої
    iot_integration.add_iot_device("Temperature Sensor")
    iot_integration.add_iot_device("Motion Detector")
```

```
iot_integration.update_iot_data("Temperature Sensor", "Temperature", "22°C")
iot_integration.update_iot_data("Motion Detector", "Motion", "No motion
detected")

# Оптимізація мережі
optimizer.optimize_topology()
optimizer.optimize_traffic()

# Друк енергоспоживання
power_manager.calculate_total_power()

# Друк статусу IoT
iot_integration.print_iot_status()

# Друк статусу мережі
factory.print_network_status()

if __name__ == "__main__":
    hyper_advanced_main()
```

КБПЗ_2024

```
# Модуль резервного копіювання
class BackupManager:

    # Ініціалізація менеджера резервних копій
    def __init__(self):
        self.backups = {}

    # Створення резервної копії
    def create_backup(self, factory):
        backup_data = {
            "devices": [(device.name, device.mac_address, device.status) for
device in factory.devices],
            "network_map": factory.network_map.copy()
        }
        backup_id = len(self.backups) + 1
        self.backups[backup_id] = backup_data
        print(f"Backup {backup_id} created")

    # Відновлення резервної копії
    def restore_backup(self, factory, backup_id):
        if backup_id in self.backups:
            backup_data = self.backups[backup_id]
            factory.devices = [
                Device(name) for name, mac, status in backup_data["devices"]
            ]
            factory.network_map = backup_data["network_map"]
            print(f"Backup {backup_id} restored")
        else:
            print(f"Backup {backup_id} not found")

    # Друк списку резервних копій
    def print_backups(self):
        print("Available Backups:")
        for backup_id, data in self.backups.items():
            print(f" Backup ID: {backup_id}, Devices: {len(data['devices'])}")
        print("")

# Модуль звітності про продуктивність
class PerformanceReporter:

    # Ініціалізація репортера
    def __init__(self, factory):
        self.factory = factory

    # Генерація звіту про продуктивність
    def generate_report(self):
        print("Performance Report:")
        for device in self.factory.devices:
            print(f"Device: {device.name}")
            print(f" MAC Address: {device.mac_address}")
```

```

        print(f" Status: {device.status}")
        print(f" Ports: {len(device.ports)}")
        print(f" Logs:")
        for log in device.logs:
            print(f"     {log}")
    print("")

# Штучний інтелект для прогнозування збоїв
class AIFailurePredictor:

    # Ініціалізація предиктора
    def __init__(self, factory):
        self.factory = factory

    # Прогнозування збоїв на основі випадкових факторів
    def predict_failures(self):
        print("Predicting potential failures...")
        for device in self.factory.devices:
            failure_risk = random.random()
            if failure_risk > 0.8: # 20% ризику збою
                print(f"Warning: Device {device.name} has a high risk of failure
({failure_risk:.2f})")
            else:
                print(f"Device {device.name} is operating within safe limits
({failure_risk:.2f})")

# Головна функція з новими модулями
def enterprise_main():
    factory = EthernetFactory()
    backup_manager = BackupManager()
    reporter = PerformanceReporter(factory)
    ai_predictor = AIFailurePredictor(factory)

    # Додаємо пристрої
    switch = factory.add_device("Core Switch")
    router = factory.add_device("Router")
    server = factory.add_device("Web Server")

    # Створюємо резервну копію
    backup_manager.create_backup(factory)

    # Прогнозуємо потенційні збої
    ai_predictor.predict_failures()

    # Генеруємо звіт про продуктивність
    reporter.generate_report()

    # Додаємо ще один пристрій і створюємо нову резервну копію
    client = factory.add_device("Client PC")
    backup_manager.create_backup(factory)

    # Відновлюємо попередню версію мережі
    backup_manager.restore_backup(factory, 1)

```

```
# Друкуємо звіт після відновлення
reporter.generate_report()

if __name__ == "__main__":
    enterprise_main()
```

КБПЗ_2024