

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

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи інтелектуальної трансляції
даних з застосунків на фізичну мережу за рахунок застосування
AgileController”

Виконав здобувач вищої освіти
IV курсу, групи КН-22
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Вакацієнко Н.Я.
« ____ » _____ 2026 р.

Керівник проекту
кандидат технічних наук
_____ Лисенко І.А.
« ____ » _____ 2026 р.
Рецензент _____

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 06 » лютого 2026 року

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

Вакацієнку Нікіті Ярославовичу

(прізвище, ім’я, по батькові)

1. Тема роботи *Програмне забезпечення системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController*

2. Керівник роботи *Лисенко Ірина Анатоліївна, канд. техн. наук*

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 116-02 від 06.02.2026 року

3. Строк подання студентом роботи до захисту 23.05.2026 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController*

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання « 06 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

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

Дата видачі завдання
« 06 » лютого 2026 р.

Підпис керівника

Лисенко І.А.
(прізвище та ініціали)

Завдання прийнято до виконання
« 06 » лютого 2026 р.

Підпис здобувача

Вакацієнко Н.Я.
(прізвище та ініціали)

АНОТАЦІЯ

Вакацієнко Н.Я. Програмне забезпечення системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

Метою розробки є програмне забезпечення системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

Результат роботи – програмна реалізація системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

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'ютерні науки, інтелектуальна трансляція даних, AgileController

ABSTRACT

Vakatsiienko N.Ya. Software for the system of intelligent data transmission from applications to the physical network through the use of AgileController. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for the system of intelligent data transmission from applications to the physical network through the use of AgileController.

The purpose of the development is software for the system of intelligent data transmission from applications to the physical network through the use of AgileController.

The result of the work is the software implementation of the system of intelligent data transmission from applications to the physical network through the use of AgileController.

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

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

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

The program is developed in the Python environment.

Keywords: computer science, intelligent data translation, AgileController

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	14
2.3 Розгорнута постановка завдання	19
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	20
3.1 Опис функціонування системи	20
3.2 Розробка структурної схеми.....	32
3.3 Розробка функціональної схеми	39
3.4 Розробка діаграми процесів.....	44
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	46
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	46
4.2 Захист розробленого програмного забезпечення.....	62
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	66
6 ОСНОВНІ ВИСНОВКИ.....	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	74

					ВКРБ-122.26.0005.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController</i>	Лім.	Аркуш	Аркушів
Розроб.	Вакацієнко Н.Я.					Б	1	80
Перев.	Лисенко І.А.							
Н.контр.	Коваленко А.С.					ЦНТУ КН-22		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БД	—	база даних
УЛММР	—	узагальнена лінійна модель множинної регресії
МНК	—	метод найменших квадратів
СУБД	—	система управління базою даних
УМНК	—	узагальнений метод найменших квадратів
ОПР	—	особа яка приймає рішення

К6ПЗ_2026

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Контролер Huawei Agile Controller SDN Software являє собою ефективний інструмент мережевої організації. З його допомогою власник може сформувати конфігурацію:

- кампусних мереж;
- WAN;
- центра інформаційної обробки;
- Інтернету речей.

Споживачами пристрою також виступають оператори зв'язку, що одержують можливість поліпшувати надаваний клієнтам сервіс. Крім цього контролер сприяє перекладу інфраструктури в хмару, створюючи незалежність від традиційних фізичних середовищ.

Найкраще потенціал розкривається в корпоративних мережах, дозволяючи компаніям поступово нарощувати ресурси.

Особливості устаткування:

- Функціонал контролера Huawei Agile Controller SDN Software дозволяє резервувати мережеві ресурси, а потім активувати їх на вимогу користувача;
- У продаж модель надходить у настроєному виді, що передбачає автоматичне розгортання, де більшість процесів виконується по заданих шаблонах;
- Одним із властивостей пристрою виступає корекція пропускну здатності, коли канал розширюється відповідно до встановлених параметрів;
- Можливостей устаткування досить для формування комерційної платформи мережевого керування з наступним наданням клієнтських послуг.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Переваги устаткування:

– Контролери Huawei Agile Controller SDN Software дозволяють завершити цифрову трансформацію організації, повністю переводячи її на сучасні рейки;

– Реалізована з їхньою допомогою інфраструктура буде гнучкою, що вчасно реагує на зміни. Керування здійснюється за допомогою інтелектуальних програм;

– Мережеве середовище перетворюється з допоміжного елемента у виробничий базис, консолідуючи й розподіляючи наявні ресурси.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

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

– Огляд існуючих систем інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

– Дослідження системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

– Програмна реалізація системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Agile Controller є базовим компонентом хмарного сервісного рішення Huawei Cloud Fabric, призначеного для керування мережами ЦОД. Робота Agile Controller заснована на ієрархічно розділеній структурі програмно-визначаємих мереж (SDN).

Можливості Agile Controller-DCN дозволяють автоматизувати схематичний розподіл (маппінг) застосунків на фізичну мережу, швидко розгорнути пул ресурсів і візуалізувати дані по керуванню й обслуговуванню (O&M), що дозволяє IT-фахівцям реалізувати функції по динамічному розподілі мережевих сервісів. Корисною можливістю є взаємодія Agile Controller-DCN із цілим рядом хмарних платформ і обчислювальних ресурсів. Для цього застосовуються стандартизовані й відкриті південні й північні інтерфейси. Завдяки цьому забезпечується гнучка взаємодія між мережевими й обчислювальними ресурсами. У такий спосіб Agile Controller-DCN допомагає зробити мережа ЦОДа більше гнучкою для хмарних сервісів, оскільки дає можливість IT-фахівцям швидко й гнучко планувати й розгортати мережеві ресурси, як того вимагають використовувані застосунки.

1.2 Область застосування

З основними хмарними платформами Huawei Agile Controller-DCN взаємодіє за допомогою API, розроблених на основі архітектури «передачі стану подання» (Representational State Transfer REST), а для взаємодії зі сторонніми застосунками застосовуються драйвери, модульні плагіни або агенти. Так, для підключення інтерфейсів Neutron (Quantum) на хмарних платформах на базі

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

OpenStack до Agile Controller-DCN застосовуються драйвери ML2 для конфігурування мережі рівня L2, підмережі й портів. Підтримка Neutron також дозволяє реалізувати взаємодію через інтерфейс агентів маршрутизатора L3 для конфігурування Elastic IP (EIP) і перетворення мережесих адрес джерел (SNAT). Крім того, підтримується інтеграція OpenStack Neutron з такими плагінами як:

- «віртуальна приватна послуга-сітка-як-послуга» (VPNaaS);
- «загальна інкапсуляція послуга-маршрут-як-послуга» (GREaaS);
- «балансувальник послуга-навантаження-як-послуга» (LbaaS);
- «файрвол-як-послуга» (FwaaS);
- а також рівень захищених сокетів-як-послуга (SSLaaS) що необхідно для повноцінної конфігурації сервісів.

Huawei Agile Controller-DCN прекрасно працює й зі сторонніми обчислювальними платформами при рішенні завдань з оркестрації мережі й обчислювальних ресурсів. ІТ-фахівцям досить здійснити яку-небудь операцію (створення, видалення, міграція) з ресурсами віртуальної машини на базі Microsoft SystemCenter або VMware vCenter через інтеграцію по API, після чого Agile Controller-DCN автоматично визначить, що статус був змінений, і надасть оновлені параметри мережевої конфігурації, що відповідають новому статусу даної віртуальної машини. Мережеві ресурси, таким чином, звільняються автоматично після виводу з експлуатації віртуальних машин, а мережеві конфігурації самостійно настроюються для готовності до міграції.

Що стосується стандартних інтерфейсів, які використовує Huawei Agile Controller-DCN для підключення до різних фізичних і віртуальних пристроїв (комутаторам, маршрутизаторам, файрволам і балансувальникам навантаження), те тут слід зазначити OpenFlow, Network Configuration Protocol (NETCONF) і Open vSwitch Database Management Protocol (OVSDB).

Відкрита архітектура дозволяє Huawei Agile Controller-DCN успішно переборювати обмеження стандартних мережевих архітектур і автоматично розгортати мережі, забезпечені повним набором можливостей по експлуатації й підтримці. Все це не тільки спрощує процеси роботи й технічної підтримки мережі хмарного ЦОДа, але й скорочує час підготовки нових сервісів і послуг.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

К6ПЗ_2026

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Огляд SDN-рішень: Cisco ACI, VMware NSX і Nuage VSP

SDN – напевно ви чуєте цей термін не в перший раз. Інтерес компаній до теми програмно-визначаємих мереж неухильно росте, в ІТ-фахівців уже складається розуміння концепції SDN. Але от розібратися в «тонкостях» імплементацій можуть далеко не все. За останні кілька років ми протестували досить багато SDN-рішень, при цьому постаралися приділити увагу не тільки продуктам відомих виробників, але й утворам численних стартапів і Open Source-розробкам.

Ми розглянемо 3 рішення SDN, які, на наш погляд, варті уваги більшою мірою: що вони із себе представляють і які завдання можуть вирішувати.

Ми враховували безліч аспектів і характеристик рішень, але в першу чергу їхня надійність. Полею бою для SDN є ЦОД, тому ключові вимоги до подібним до розробок – це стабільність і виживаність системи. Оскільки більша частина SDN-Продуктів – це стартапи або проекти відкритих співтовариств із короткою історією, при тестуваннях ми зустрічали безліч багів ПЗ й випадків непрацездатності системи в нетипових сценаріях використання.

Вищезгадані рішення – це розробки «акул» ринку, і цілком логічно, що саме вони найбільше технологічно зрілі на сьогоднішній день. Вони мають найбільш широкого функціонала і являють собою цілісні й самодостатні рішення. З їхніх технологічних можливостей варто відзначити реалізацію більшості мережевих функцій «з коробки», здатність інтегруватися з різними системами CMS, 3rd-party мережевими пристроями.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Нижче ми коротко охарактеризуємо ці продукти, оскільки у свій час уже порівнювали функціонала Cisco ACI і VMware NSX, упор же зробимо на завданнях, які їм під силу вирішити.

Cisco ACI

Cisco ACI (Application Centric Infrastructure, інфраструктура, орієнтована на застосунки) – це архітектура мережі ЦОД нового покоління, як позиціонує рішення вендор.

Інфраструктура ACI складається із двох компонентів – контролера APIC і мережевої фабрики, побудованої на комутаторах Nexus 9K (поки підтримується тільки ця лінійка).

Таким чином, Cisco інтерпретувала одну з головних ідей концепції SDN – «відкритість» системи й взаємозамінність апаратних компонентів – по-своєму.

У той же час за рахунок використання власного заліза вдалося уникнути проблем інтеграції з мережевою фабрикою стороннього виробника (є маса нюансів, пов'язана з реалізаціями того ж стандартизованого протоколу OpenFlow на устаткуванні різних вендорів).

Відповідно, характерною рисою Cisco ACI є необхідність використання у фабриці обмеженого кола «залозок».

Втім, з урахуванням високого ступеня довіри замовників до рішень Cisco цей факт не став проблемою.

Тому, на наш погляд, Cisco ACI доцільно розглядати в першу чергу для побудови інфраструктури ЦОД «з нуля» або ж при плановому відновленні всієї існуючої мережі ЦОД.

VMWare NSX

Якщо ви не готові піти на заміну всієї мережі ЦОД, має сенс подивитися у бік overlay-рішень (коли дані передаються по тунелях поверх існуючої мережевої фабрики).

VMWare NSX – ще один погляд на програмно-визначаємі мережі, але через призму бачення вендора з миру серверної віртуалізації.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Фактично історія NSX веде свій відлік з 2007 року, по суті, це саме «вікове» рішення SDN на ринку.

З погляду архітектури продукт схожий з Nuage VSP – всі ті ж VXLAN-Тунелі для передачі трафіку по існуючій апаратній мережі.

Відмінною рисою NSX є найбільш щільна інтеграція з екосистемою рішень для керування серверної віртуалізацією від VMware.

У результаті замовник одержує можливість керування всією інфраструктурою ЦОД з одного вікна.

Nuage VSP

Альтернативним рішенням може бути платформа мережевий віртуалізації Nuage VSP компанії Nuage Networks (належить Alcatel Lucent, з недавніх пор – компанії Nokia).

При розробці продукту був використаний значний досвід роботи Alcatel Lucent на ринку операторських мереж, а control-plane рішення реалізований на базі операційної системи флагманських маршрутизаторів ALU SROS.

Як відомо, мережі операторів відрізняються географічно розподіленою структурою, тому Nuage VSP разом з SROS успадкував здатність легко поєднувати мережі декількох площадок – одну зі своїх сильних сторін.

Кейси

Незважаючи на розходження реалізації, всі 3 рішення виконують ті самі, традиційні для SDN, завдання:

- автоматизація керування мережею;
- перехід від централізованої до більш гнучкої, розподіленої моделі надання мережевих сервісів;
- підвищення рівня мережевої безпеки ЦОД;
- подолання архітектурних обмежень традиційних мереж.

Але в той же час необхідно враховувати особливості кожного продукту.

Які питання потрібно собі задати, якщо ви дивитесь убік переходу до програмно-визначаємих мереж?

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Перший – що являє собою ваша мережа ЦОД? Якщо мережева фабрика побудована на устаткуванні Cisco, а конкретно на комутаторах Cisco Nexus 9000 (які, до слова, набрали певну популярність за останні кілька років завдяки ціновій політиці вендора), те, зрозуміло, Cisco ACI – це те, що потрібно розглянути в першу чергу. В обмін на свідоме обмеження себе у виборі вендора для майбутнього розвитку мережі ви одержите ряд переваг. У випадку з Cisco ACI ні «розшарування» мережі на кілька рівнів – і апаратні платформи, і програмні компоненти функціонують як одна система. ACI інформована про стан фізичної мережі, про статистику переданого трафіку, наявності помилок на інтерфейсах і т.п. Якщо ж мережа побудована на устаткуванні різних вендорів, при цьому відповідає вашим вимогам по надійності й продуктивності, ефективніше використовувати розв'язання overlay-рішення. За допомогою NSX або Nuage ви створите відсутній «інтелектуальний» шар мережевої фабрики, зберігши інвестиції в «залізо». Фізичну мережу ви зможете продовжувати будувати на комутаторах будь-якого вендора – досить підтримки протоколу IP. Але, як наслідок, мережевим інженерам фактично прийде адмініструвати 2 мережі. І хоч фізична мережа стане статичною й зміни в неї буде потрібно вносити вкрай рідко, процес траблшутинга ускладниться. Крім того, overlay-рішення можуть сильно спростити життя якщо буде потреба злиття двох незалежних інфраструктур. Таке завдання останнім часом дуже часто встають перед компаніями фінансового сектора у зв'язку з поглинаннями й злиттями. Як правило, мережевим інженерам у цьому випадку доводиться вирішувати проблеми пересічної адресації, розходжень логічної архітектури існуючих мереж. Overlay-рішення будуть використовувати старі мережі лише як «транспорт», просто створюючи необхідні топології поверх них, у результаті вони дозволять здійснити плавну міграцію на цільову архітектуру. У випадку будівництва нового ЦОД варто розглянути й перший, і другий підхід. При використанні overlay-мереж можна істотно заощадити на апаратної складової фабрики. У загальному ж, залежно від реальних вимог, економічно більш привабливо може виявитися кожний з варіантів.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Друге питання має відношення до фізичної архітектури мережі: потрібно чи забезпечення географічної розподіленості? Наявність, як мінімум, одного резервного ЦОД – основна умова побудови відказостійкості ІТ-інфраструктури. Тому переважна більшість компаній буде власний РЦОД або орендує інфраструктуру в комерційних дата-центрах. Якщо площадок ЦОД більше, ніж одна, і вам потрібно побудувати розподілену мережеву фабрику з «розтягуванням» логічних підмереж, встає питання організації DCI (Data Center Interconnect, модуль забезпечення взаємодії географічно рознесених ЦОД). Сама по собі Cisco ACI не вирішує це завдання, і DCI потрібно реалізовувати окремо. Для цього вендор рекомендує побудова виділеного мережевого модуля й використання технології OTV. А от overlay-рішення можуть бути як не можна до речі. У силу своєї природи – передачі трафіку по тунелях – вони вирішують завдання забезпечення L2-зв'язності без додаткових архітектурних ускладнень. DCI можна побудувати по будь-яких каналах зв'язку, у тому числі, недорогим L3 VPN. Відзначимо, що донедавна продукт VMware NSX мав деякі складності в частині побудови розподіленої між декількома фізичними площадками фабрики, і ця сторона залишалася ахіллесовою п'ятою рішення. Але з початком підтримки Cross vCenter Server vMotion на гіпервізорах vSphere с версії 6.0 частина обмежень була переборена. Що стосується Nuage VSP, використання протоколу MP-BGP забезпечило дуже широкі можливості географічного масштабування рішення убудованими засобами.

Третє важливе питання: яка серверна інфраструктура? Як ви, напевно, уже зрозуміли, використання overlay-рішень доречно в ЦОД з високою часткою віртуалізованих серверів. І в даному аспекті рішення Cisco ACI більш універсально, тому що воно хоч і інтегрується з гіпервізорами віртуалізації, але не базується на них. Тому якщо з якихось причин віртуалізація серверів ще не накрила ваш ЦОД «з головою», розглянете Cisco. Цілком логічно, що NSX найкраще «ляже» на інфраструктуру, у якій використовується віртуалізація від VMware. Рішення забезпечить найкращу продуктивність вашої віртуалізованої

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

мережі за рахунок максимальної інтеграції з гіпервізором: основна частина мережевих операцій виробляється на dvSwitch в kernel space ESXi. Але у випадку інфраструктури з різними гіпервізорами варто розглядати Nuage або ACI. Є ще ряд кейсів, які можуть вирішуватися за допомогою наших сьогоднішніх об'єктів уваги. Варто відзначити, що VMware NSX має розвинутого функціонала мережевої безпеки віртуального середовища – ви можете створювати гнучкі політики безпеки, ідентифікуючи віртуальні машини по множині критеріїв. Крім того, є можливість інтеграції з 3rd-party рішеннями, наприклад, від Symantec, і NSX може самостійно переміщати заражені вірусами віртуальні машини в спеціальні ізольовані мережеві сегменти. Це перевага особливо привабливо для компаній з високими ІБ-вимогами, зокрема, для банків. Cisco ACI і Nuage VSP надають більше широкі можливості автоматизації й візуалізації мережі. Рішення дозволяють створювати шаблони мережевих топологій і політик під типові застосунки й багаторазово використовувати їх у роботі. В NSX автоматизація можлива за допомогою додаткових рішень vCloud або vRealize. Ще одна відмінна риса Nuage – можливість використання рішення за межами ЦОД. Вендор пропонує зробити програмно-визначаємою і всю корпоративну мережу. Не секрет, що багато компаній не готові витратити гроші на зміст великого штату ІТ-фахівців у філіях, така ситуація характерна для ритейла. Так, у ритейлера є безліч магазинів у різних регіонах, при цьому виконувати прості дії по адмініструванню або рішенню проблем з мережею на місцях найчастіше просто комусь. Розгортання й керування філіальною мережею можна значно спростити за допомогою розширення платформи – Nuage VNS. У результаті навіть при уведенні в експлуатацію можна обійтися без мережевих інженерів на вилученій площадці. А при адмініструванні й мережа дата-центра, і локальні мережі вилучених офісів будуть видні через один інтерфейс керування.

Безсумнівно, рішення, що втілюють концепцію програмно-визначаємих мереж, забезпечують безліч переваг у порівнянні із традиційними підходами. Але в той же час всі вони мають свою специфіку й обмеження. Сподіваємося, що нам

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

удалося прояснити основні особливості розглянутих рішень, і ця інформація допоможе вам зорієнтуватися у виборі вектора подальшого руху в напрямку SDN.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

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

Швидкість виконання коду Python

Один з можливих недоліків Python – швидкість виконання коду. Python не є компільованою мовою. Код на Python спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. У більшості випадків при використанні Python виходять програми повільніші в порівнянні з такими мовами, як С. Втім, сучасні комп'ютери мають таку обчислювальну потужність, що для більшості застосунків швидкість розробки важливіша швидкості виконання, а програми на Python зазвичай пишуться набагато швидше.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

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

Використання Python

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

Застосування мови програмування Python:

1. BitTorrent – протокол для обміну даними.
2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.
3. Blender – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.
4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.
5. World of Tanks.
6. Вільна енциклопедія Вікіпедія.
7. Пошукова система Google.
8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.
9. YouTube – популярне відеосховище.

Версії Python

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

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

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

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

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

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

Коли процесор виконує інструкції програми, він бере участь у процесі, який є відомим як цикл `fetch – decode – execute` (отримати – декодувати – виконати). Цей цикл виконується для кожної інструкції у програмі і складається з трьох кроків:

Отримати

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

Декодувати

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

Виконати

Останній крок циклу – виконати операцію.

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

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

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

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

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

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

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

1. Компілятор.
2. Інтерпретатор.

Завантаження Python

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

Середовище програмування для Python

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

Текстові редактори та інтегровані середовища програмування для Python:

- IDLE – стандартний редактор Python. Встановлюється разом з Python для користувачів Windows, окремим пакунком для користувачів Linux.
- Notepad++ – безкоштовний текстовий редактор початкового коду, який підтримує велику кількість мов, в тому числі і Python. Лише для користувачів Windows.
- Visual Studio Code – це легкий, але потужний редактор початкового коду, який розповсюджується безкоштовно і доступний у версіях для платформ Linux, Windows і macOS.
- PyScripter – інтегроване середовище розробки для мови програмування Python. Для користувачів Windows. Поширюється безкоштовно.
- Wing IDE 101 – вільне інтегроване середовище для Python, розроблене для навчання програмістів-початківців. Для користувачів Linux, Windows і macOS. Поширюється безкоштовно.
- Geany – вільний текстовий редактор з базовими елементами інтегрованого середовища розробки, доступний для операційних систем Linux, Windows і macOS.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

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

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

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

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

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

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

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

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

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

д) розробити рекомендації по організаційних та методичних заходах, які

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

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.

К6ПЗ_2026

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

SDN стрімко ввірвалася в мир мережевих технологій, ставши хедлайнером багатьох конференцій і форумів. І хоча на слух багато хто ще плутають її з ISDN, це тільки підвищує градус інтересу. Так що таке SDN? Геніальна ідея університетських дослідників, що під силу реалізувати тільки таким гігантам, як Google, Facebook або Yahoo? Або «другий Ethernet», що незабаром стане основою мереж різного масштабу й призначення?

Використовувані в українськомовній літературі переклади терміна Software-Defined Networking, такі як «програмно обумовлені» або «програмно конфігуруємі» мережі, не відбивають основну ідею даної концепції. Погодитися, конфігурування мереж уже давно здійснюється за допомогою програмного забезпечення, тому що ж тут нового й тим більше революційного?! Тому за відсутністю адекватного українськомовного терміна будемо використовувати англійську аббревіатуру.

Труднощі перекладу

Переваги, які дає SDN, очевидні, причому не тільки для мереж у центрах обробки даних, але й для інших типів мережевих інфраструктур. Централізоване керування мультивендорним середовищем, значне спрощення обслуговування й модернізації, скорочення часу на відновлення програмних кодів комутаторів/маршрутизаторів і впровадження нових сервісів – все перераховане важливо як для корпоративних мереж, так і для інфраструктур операторів зв'язку. Однак це не привід разом відмовлятися від переваг традиційного підходу, що розвивається десятиліттями, коли кожний мережевий пристрій наділяється «інтелектом», достатнім для автономного функціонування.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Думаю, елементи SDN будуть впроваджуватися поступово, при цьому тільки компанії рівня Google зможуть дозволити собі перейти на комутатори, повністю керовані із центрального контролера. Швидше за все, частина функціонала, наприклад по конфігуруванню й завданню політики безпеки, будуть виносити в контролери, при цьому самі комутатори залишаться досить «розумними», принаймні для того щоб брати участь у роботі протоколів визначення маршрутів або балансування трафіку.

Що ж стосується українськомовного терміна, то SDN – це, скоріше, «централізовано програмувальна мережа». І треба скоріше прийти до консенсусу щодо такого терміна. Нагадаю, що ISDN неформально розшифровувався як «I Still Don't Need it», тобто «мені цього поки не потрібно», причому саме таке відношення й визначило долю цієї технології в Україні. Сподіваюся, дуже розумні ідеї SDN чекає краща участь.

ІТ-менеджери все частіше обирають програмно-визначені мережі (SDN) для розгортання інфраструктури послуг центрів обробки даних, гібридної хмари та хмарних додатків у дуже гнучкий спосіб. У даному розділі надамо огляд галузевих тенденцій, які особливо стимулюють впровадження SDN, коли йдеться про розгортання ІТ-інфраструктури.

Програмно-визначені мережі (SDN) – це мережеве архітектурне рішення, яке допомагає підвищити ефективність та гнучкість мережі шляхом розгортання програмної функціональності та програмного рівня управління мережею поверх фізичної магістралі мережі. Це дозволяє організаціям гнучко та економічно ефективно керувати величезними обсягами даних та мережевого трафіку.

Згідно з даними Allied Market Research, світовий ринок програмно-визначених мереж оцінювався майже в 10 мільярдів доларів у 2025 році та, як очікується, досягне 72,6 мільярда доларів до 2027 року, зростаючи зі середньорічним темпом зростання (CAGR) на 28,2 відсотка між 2025 та 2027 роками. Інша дослідницька компанія, Mordor Intelligence, заявила, що ринок програмно-визначених мереж у 2025 році оцінювався в 12,61 мільярда доларів і,

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

за прогнозами, зростатиме зі середньорічним темпом зростання (CAGR) на 30,75 відсотка до 63,04 мільярда доларів до 2026 року (2025-2026). Швидке впровадження технології програмно-визначених мереж на ринку зумовлене різноманітними технологічними та бізнес-тенденціями, які ми детальніше розглянемо в даному розділі.

Поєднання технологій SDN та NFV

Перш ніж ми глибше заглибимося в ці ринкові тенденції, що лежать в основі зростаючого використання SDN на регіональному та глобальному рівнях, давайте спочатку коротко окреслимо визначення того, що саме передбачає технологія SDN. За допомогою технології SDN апаратний рівень, включаючи маршрутизацію та комутацію мережі, відокремлений від рівня, на якому відбувається управління та контроль мережі. Програмно-визначена мережа керується централізованим контролером SDN. Це фактично ключовий елемент концепції. Він діє як мозок мережі SDN і використовується для проектування та керування фізичними та віртуальними елементами мережі з центрального місця. Замість того, щоб налаштовувати кожен комутатор або маршрутизатор окремо, за допомогою SDN та його централізованого рівня управління мережею можна створити найвищу видимість інфраструктури. Це також забезпечує найвищу масштабованість, гнучкість і навіть автоматизований розподіл мережевих ресурсів.

Поєднуючи SDN з технологією NFV (віртуалізація мережевих функцій), як це зробив Worldstream зі своєю пропозицією Worldstream Elastic Network на основі нашої глобальної магістралі, програмно-визначена мережева інфраструктура здатна додати ще більше гнучкості, коли справа доходить до розгортання IT-інфраструктури. Тобто, коли проектування SDN-контролера виконано правильно. Це технологія, яку розробила команда досліджень і розробок Worldstream для створення оптимальних, унікальних результатів для своїх клієнтів.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Як SDN, так і NFV доповнюють віртуалізацію та абстракцію апаратних ресурсів. У той час як SDN відокремлює керування мережею від функцій переадресації мережі, NFV дозволяє абстрагувати переадресацію мережі, а також інші мережеві функції від обладнання, на якому воно розгорнуто. Підсумок, це означає, що NFV замінює спеціалізоване обладнання віртуалізованим програмним забезпеченням для створення мережевих служб, таких як маршрутизатори, балансувальники навантаження, брандмауери тощо. Ці мережеві апаратні пристрої замінюються програмним забезпеченням віртуальних машин. У традиційній мережевій конфігурації гіпервізор призначається для керування такими віртуалізованими мережевими службами, що є роллю, яку SDN відіграє в цьому випадку. Разом технології NFV та SDN роблять мережеву архітектуру та розгортання IT-інфраструктур надзвичайно гнучкими, як з точки зору пристроїв, так і мережевого трафіку.

Однак, повернімося до головної теми цього розділу – ринкових тенденцій, які стимулюють використання SDN. Наступні 5 галузевих тенденцій стимулюють впровадження SDN: зростання розгортання хмарних додатків; поява 5G; тренд роботи з дому; рух Інтернету речей; а також впровадження гібридних та мультихмарних технологій. Давайте розглянемо це детальніше.

1. Зростання розгортання хмарних застосунків

В організаціях використання «хмарних» застосунків набуло значної популярності протягом останніх кількох років. За даними 451 Research, підприємства можуть працювати швидше, ефективніше та у більших масштабах завдяки хмарним архітектурам програмного забезпечення та послуг. Глобальна дослідницька компанія очікує, що тенденція до хмарних технологій продовжуватиме зростати, частково завдяки перетину із суміжними технологіями та тенденціями, такими як дані та аналітика, штучний інтелект (ШІ) та машинне навчання (МН), безпека та Інтернет речей/периферійні обчислення. SDN-інфраструктура є мережевим аналогом гнучкої природи хмарної архітектури.

Хмарний підхід – це набагато більше, ніж просто реєстрація у хмарного провайдера та запуск хмарних додатків на їхніх платформах. Хмарний підхід до налаштування ІТ-інфраструктури та розгортання хмарних додатків можна використовувати для створення та запуску високомасштабованих додатків у публічних, приватних та гібридних хмарах. Використання мікросервісів, контейнерів, сервісних сіток, незмінної інфраструктури, а також декларативних інтерфейсів прикладного програмування (API) – це деякі з особливостей, які найкраще підкреслюють цей підхід. Використання таких функцій дійсно впливає на проектування, налаштування, розгортання та роботу додатків. Саме це робить їх справді «хмарними».

Хмарні додатки можуть функціонувати високорозподіленим та незалежним чином, водночас легко адаптуючись до динамічних та вимогливих потреб у приватних, публічних та гібридних хмарних середовищах. Залежно від вимог додатків, хмарний підхід дозволяє динамічно розподіляти ресурси ІТ-інфраструктури в цих хмарних середовищах.

Можливості, що надаються базовою програмованою мережевою інфраструктурою, є критично важливими для справності та довговічності такої хмарної архітектури. SDN може допомогти в цьому, а також збільшити швидкість та гнучкість хмарних додатків. Організації можуть використовувати хмарний підхід для створення розподілених систем з високою адаптивністю. SDN спрощує складність управління цими розподіленими фізичними мережевими інфраструктурами завдяки програмованості та централізованому керуванню.

Технологія SDN, особливо в поєднанні з NFV, як це робить Worldstream, робить надзвичайно можливою динамічну підтримку хмарних додатків, використовуючи віртуалізовані еквіваленти фізичних маршрутизаторів, брандмауерів, балансувальників навантаження та інших видів мережевого обладнання, що традиційно використовується в локальних середовищах та в колахованих середовищах центрів обробки даних. Перехід від фізичної залежності до віртуалізованих інфраструктурних послуг можна розглядати як

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

вирішальний крок до справжнього хмарного переходу, що призводить до розгортання справді масштабованих, гнучких додатків за найнижчою можливою ціною. Таким чином, це основна рушійна сила для подальшого впровадження SDN.

2. Поява 5G

Коли йдеться про інфраструктуру мобільного зв'язку, 5G зараз поступово замінює мережі 4G. 5G може забезпечити нам блискавично швидкі та максимально надійні варіанти підключення, дозволяючи підключати величезну кількість приладів з низькою затримкою. Він призначений не лише для підтримки операторів зв'язку у впровадженні першокласних мереж мобільного широкосмугового доступу для підключення смартфонів, iPad тощо. Він також прокладає шлях для інтелектуального розгортання галузевих додатків, що підтримуються приватними мережами 5G, та використання переваг концепції Індустрії 4.0. Крім того, він може забезпечити високошвидкісне мережеве підключення з низькою затримкою, необхідне для додатків Інтернету речей (IoT) та зростаючого розподіленого використання інтелектуальних датчиків. Також він може принести величезні переваги для дорожнього руху, безпілотних автомобілів та управління розумним містом.

Програмно-визначені мережі, зокрема поєднання технологій SDN та NFV, які Worldstream впровадила у свою пропозицію Worldstream Elastic Network, лежать в основі нових мережевих архітектур для 5G у всьому світі. Програмованість мережі є важливим аспектом 5G. Завдяки можливості програмування мережі технологія SDN відіграє вирішальну роль у цій галузі.

Розподіл мережі – це ще одна помітна концепція, яка виникне з використанням 5G. Розподіл мережі передбачає сегментацію мережі на окремі логічні мережі різного розміру та архітектури, кожна з яких призначена для різних типів послуг. Це дозволяє налаштовувати мережеві операції, надаючи кілька рівнів обслуговування для різних користувачів мережі. Кожен з цих мережевих сегментів може мати власні характеристики швидкості, затримки та

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

якості обслуговування (QoS). Щоб скористатися цими потенційними перевагами 5G, SDN необхідно поєднувати з технологією NFV, яка пропонує основу для оркестрації та автоматизації мережевих сегментів.

Отже, 5G – це набагато більше, ніж просто збільшення пропускної здатності та зменшення затримки. Йдеться також про адаптивність, керованість та можливість розробляти нові послуги. Поява приватного 5G є чудовим прикладом цього. Оскільки органи влади надають додатковий спектр підприємствам, щоб вони могли розробляти та експлуатувати власні приватні мережі 5G – мережі, які не обмінюються трафіком з іншими стільниковими мережами в цьому регіоні, – впровадження приватних мереж 5G поступово набирає обертів у всьому світі. Знову ж таки, SDN є вирішальним фактором появи приватної мережі 5G, оскільки дозволяє віртуалізувати частини мережі. Це дає компаніям більше контролю над даними в мережі, а ємність, що розвивається, повністю адаптується до фактичних потреб бізнесу. Технологія SDN є рушійною силою впровадження приватного 5G, оскільки вона допомагає пришвидшити гнучкість та керованість приватного 5G.

3. Тренд роботи з дому

Хоча робота з дому більше не є обов'язковою політикою, все більша кількість працівників у всьому світі перенесли принаймні частину свого робочого часу з офісів додому. Дослідження Євростату за 2025-2026 роки показує, що в середньому 30% європейців регулярно працюють віддалено з дому. За даними Forbes, на початку 2023 року 68% усіх американців воліли б працювати повністю віддалено. Здається, що робота з дому, а точніше віддалено, не є тенденцією, яка збирається зникнути. Організації, зі свого боку, докладають зусиль, щоб впоратися з масовим потоком інтернет-трафіку віддалених працівників, який проходить через брандмауери компанії до VPN (віртуальних приватних мереж). Вони стикаються із завданням підтримувати нормальну роботу в незнайомому середовищі, одночасно відбиваючи нові хвилі фішингових та DDoS-атак. Однак застарілі VPN просто не були розроблені для обробки великого відсотка

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

співробітників організації, або в деяких випадках навіть усіх, які отримують віддалений доступ до мережі компанії.

Пов'язані з цим проблеми з мережею та вузькі місця можуть бути складними для організацій, які досі залежать від застарілої фізичної мережевої інфраструктури. Але не для тих, хто вже переніс свої мережеві можливості з фізичної на віртуальну, включаючи брандмауери, захист від DDoS-атак та хмарні виходи. Це можна зробити за допомогою технологій SDN/NFV, подібних до тих, що Worldstream включила у свої пропозиції інфраструктури як послуги на базі SDN.

Як організація, ви просто не можете ризикувати, забезпечуючи віддаленим працівникам переривання інтернет-з'єднання щоразу, коли вони перебувають на зустрічі в Teams, WebEx, Zoom або Google Meet. Іноді може здаватися, що причина заїкання цифрової зустрічі криється в самій платформі для зустрічей, або що швидкість бізнес-додатків на основі програмного забезпечення як послуги (SaaS) залишає бажати кращого, тоді як справжнім винуватцем може бути власна (традиційно розроблена) мережа організації. Сучасний дизайн мережевої інфраструктури, іншими словами, програмно-визначена, може бути дуже корисним у цьому випадку.

SDN забезпечує гнучке узгодження з потребами бізнесу, що змінюються, включаючи дуже мінливі рівні віддаленої роботи. Завдяки найнижчим значенням затримки, які легко гарантуються, а також можливості плавного масштабування пропускної здатності вгору або вниз, навіть сегментовано, SDN забезпечує інфраструктурну гнучкість, про яку міг би побажати практично кожен ІТ-директор та технічний директор. Також, коли справа доходить до впровадження та управління рішеннями кібербезпеки для гібридних робочих концепцій. Все це економічно ефективним способом.

Традиційна мережа може бути дещо статичною та переважно пов'язаною з офісною будівлею, тоді як SDN як сучасна мережа може динамічно відстежувати всі рухи організації. Як щодо того, як співробітники використовують програми,

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

так і щодо самих програм та інфраструктури, на якій вони працюють – включаючи локальні та хмарні технології. Тому тенденція до роботи з дому є ще одним важливим фактором для сучасного впровадження SDN на ринку.

4. Рух Інтернету речей

За даними IDC, витрати на інфраструктуру Інтернету речей (IoT) у Європі досягли 202 мільярдів доларів у 2026 році та продовжуватимуть зростати двозначними числами до 2027 року. У 2025 році споживчий сектор залишався провідним споживачем IoT у Європі. Виробництво, друга за величиною галузь, використовує рішення IoT для відстеження, моніторингу та обслуговування промислових виробничих пристроїв, які є частиною виробничого ланцюжка створення вартості, шляхом аналізу потоку даних машини в реальному часі.

Оскільки Інтернет речей розвивається в Європі та за її межами, а його вимоги до мережі стають складнішими, SDN (програмно-визначені мережі) є найкращим шляхом для забезпечення оптимального підключення та передачі даних. У традиційних апаратних мережах можуть виникнути серйозні проблеми Інтернету речей, зокрема, щодо гнучкості, затримки та безпеки. Програмно-визначені мережі не мають жодних обмежень у цьому відношенні, а це означає, що SDN має унікальну здатність вирішувати ці проблеми взаємозв'язку для руху Інтернету речей.

Завдяки своїй еластичній природі, SDN значно спрощує для організації управління пропускну здатністю, необхідною для мережі. Розглянемо, наприклад, галузь охорони здоров'я. Найімовірніше, найближчим часом лікарні будуть оснащені великою кількістю обладнання та кінцевих пристроїв на базі Інтернету речей для пацієнтів. Очікується, що всі вони будуть об'єднані в мережу та будуть підтримувати Інтернет речей. Деякі з цих медичних приладів можуть використовувати велику пропускну здатність мережі лікарні. Тому цим лікарням знадобиться гнучка пропускну здатність для обробки раптових навантажень на свої мережі в режимі реального часу, що SDN може забезпечити.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Продовжуючи цей приклад, лікарні мають справу з конфіденційними даними пацієнтів, що вимагає комплексної безпеки даних та дотримання вимог. Використання технології SDN дозволяє лікарням підвищити рівень безпеки до максимуму. Наприклад, SDN дозволяє створювати приватні віртуальні мережі, які можна нашаровувати на фізичну мережу, що дозволяє відокремлювати пристрої Інтернету речей від решти мережі без складнощів, які можуть виникнути через застарілі технології віртуальних приватних мереж. Це також дозволяє дуже гнучко розгортати рішення кібербезпеки, такі як брандмауери, засоби захисту від DDoS-атак тощо.

Під час переміщення лікарняного обладнання по всьому приміщенню, пристрій, можливо, доведеться негайно підключити на новому місці, водночас ті самі мережеві політики та елементи керування повинні динамічно переноситися на нове місце. Ці вимоги до гнучкості та безпеки дійсно вимагають еластичності мережі та можливості гнучко встановлювати найвищі рівні безпеки. SDN забезпечує таку гнучкість в умовах масового передавання даних, спричиненого рухом Інтернету речей.

Концепція Інтернету речей також чутлива до часу, що робить якість підключення до пристроїв Інтернету речей та з'єднань з хмарою головною вимогою.

5. Гібридне та мультихмарне впровадження

Підтримка мультихмарних/гібридних хмар стає ключовим фактором для організацій під час вибору постачальника хмарних послуг для своєї ІТ-інфраструктури, згідно з результатами нещодавнього дослідження, проведеного IDG на замовлення Google, яке базується на глобальному опитуванні понад 2000 ІТ-фахівців. Хоча 30 відсотків респондентів вважають, що підтримка мультихмарних/гібридних хмар є обов'язковою при виборі постачальника хмарних послуг, 47 відсотків називають її важливим фактором.

Від малого та середнього бізнесу до великих міжнародних корпорацій можуть отримати вигоду від гібридного та мультихмарного підходу. Переваги

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

полягають у підвищеній стійкості, гнучкості, свободі вибору та зниженні витрат. Однак ключовим завданням є ефективне поєднання цих різних хмарних та локальних середовищ у регіональному та/або світовому масштабі. SDN пропонує найкращу концепцію підключення для гібридних та мультихмарних рішень.

У традиційних мережах встановлення прямих підключень до кожного хмарного постачальника та розташування центру обробки даних у мережевій конфігурації може бути досить дорогим. Водночас, це може бути складно в управлінні, коли постійно зростаючі обсяги даних та послуг переносяться в таке гібридне, багатохмарне середовище.

Керування різними з'єднаннями з кількома хмарними постачальниками також вимагає знань, ресурсів та часу. Крім того, для належного управління багатохмарною інфраструктурою ІТ-фахівцям знадобиться технічна експертиза в кожній хмарі. Коли кожна з цих хмар обслуговується та підключається окремими мережевими постачальниками, швидке розгортання та розширення нових послуг може бути складним. Крім того, хоча гібридна/багатохмарна інфраструктура надасть більше можливостей для управління робочими навантаженнями, вона також може наражати бізнес на більшу кількість порушень безпеки та кібератак.

Завдяки технології SDN ці проблеми з підключенням та безпекою фактично відійшли в минуле, особливо коли йдеться про встановлення підключень до кількох хмар. SDN забезпечує гнучкість розгортання ІТ-інфраструктури, а також спрощене керування підключенням до хмари та розширені параметри безпеки, водночас знижуючи мережеві витрати та пришвидшуючи впровадження хмари. Завдяки SDN, організації можуть динамічно масштабувати (збільшувати та зменшувати) свою хмарну інфраструктуру, додавати нові хмарні середовища та з'єднувати сайти центрів обробки даних на регіональному та глобальному рівнях. Все це контролюється та керується з одного місця за допомогою технології SDN.

На завершення, SDN дозволяє організаціям подолати розрив у хмарному підключенні, одночасно підвищуючи прозорість, масштабованість, гнучкість та

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

безпеку. Як одна з п'яти галузевих тенденцій, що стимулюють впровадження SDN, вона дозволяє організаціям оптимально використовувати гібридну/багатохмарну інфраструктуру та пришвидшувати своє розширення на основі цієї інфраструктури за допомогою гнучкої стратегії виходу на ринок з новими додатками та послугами.

3.2 Розробка структурної схеми

Agile Controller версії 3.0

5 вересня 2016 року компанія Huawei представила Agile Controller версії 3.0. Контролер орієнтований на чотири сценарії організації мережі (корпоративна кампусна мережа, мережа центра обробки даних, мережа WAN і Інтернет речей).

У функціонала увійшли:

- резервування мережевих ресурсів «на вимогу»,
- автоматичне розгортання,
- інтелектуальна оптимізація й коректування пропускної здатності «на вимогу».

Це допомагає оптимально планувати ресурси в корпоративних хмарних сервісах, підвищуючи ефективність роботи мережі оператора зв'язку й забезпечуючи якість обслуговування.

Ми вважаємо, що ключем для переходу мережі на хмарні технології стають SDN-рішення, а також розробки для віртуалізації мережевих функцій (NFV), ми допомагаємо операторам підвищувати якість послуг у рамках концепції ROADS на базі технології hard pipe, надаючи такі сервіси, як публічна хмара, частка хмара, гібридна хмара й хмарна приватна лінія.

SDN-рішення на платформі контролера AC 3.0 допомагає операторам зв'язку створити електронні комерційні платформи керування мережею для досягнення максимальної швидкості й додаткового налаштування сервісів «на вимогу». Воно допомагає підприємствам завершити цифрову трансформацію,

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

визначаючи високий пріоритет для гнучкості й інтелектуальності, перетворюючи мережі з допоміжних інструментів в основу виробництва.

Контролер AC 3.0 підтримує уніфіковане керування й гнучке розгортання в різних бізнес-сценаріях і умовах експлуатації, відповідаючи вимогам центрів обробки даних і уможливлуючи комерційну експлуатацію надвеликих мереж.

Контролер AC 3.0 на основі відкритої платформи ONOS використовує додаток зі слабозв'язаною архітектурою, що може інтегруватися з ODL і має функціонал «північного» інтерфейсу прикладного програмування.

Agile Controller для кампусної мережі

Контролер нового покоління Agile Controller розробки Huawei призначений для кампусних мереж і мереж філій. Застосування Agile Controller для кампусної мережі в декількох інноваційних рішеннях, таких як автоматичне розгортання мережі, автоматизація політик і SD-WAN, дозволило скоротити операційні витрати, витрати на експлуатацію й обслуговування, прискорити перехід на хмарні технології й цифрове перетворення підприємств, а також зробити керування мережею більше зручним, а процеси експлуатації й обслуговування – більше інтелектуальними.

Agile Controller для кампусної мережі підтримує різні сценарії застосування, такі як CloudCampus, SD-WAN і керування доступом користувачів.

Розглянемо функції.

CloudCampus

Операції планування, настроювання, техобслуговування й профілактичного огляду (Preventive Maintenance Inspection, PMI) у рамках автоматизованої експлуатації мережі. Багатоклієнтська архітектура керування, що допомагає MSP-Провайдерам надавати послуги керування кампусними мережами. Гнучке розширення спектра сервісів кампусних мереж за допомогою відкритих інтерфейсів прикладного програмування.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

SD-WAN

Створення рентабельних, легко корпоративних мереж, що обслуговуються, з урахуванням особливостей використовуваних застосунків і можливістю надання сервісів по запиті, які спростять впровадження інноваційних ідей на підприємствах. Автоматичне настроювання, інтелектуальне керування трафіком і візуалізована система експлуатації й техобслуговування.

Керування доступом користувачів

Уніфікація процесів доступу користувачів і керування співробітниками, відвідувачами й адміністраторами з метою забезпечення дотримання політик обслуговування й необхідної якості (політика перевірки прав користувача на доступ, QoS, політика виділення смуги пропускання). Різні протоколи автентифікації: 802.1x, TACACS, а також перевірка дійсності прав по MAC-адресі й через портал.

Рішення Agile WAN для галузевої взаємодії

Рішення Agile WAN для галузевої взаємодії (AWAN II) забезпечує надійне й безпечне з'єднання мереж, розташованих у різних місцях, які можуть належати або не належати одному підприємству (наприклад, кілька ділянок у транспортній системі компанії або різних компаній, що використовують одну електричну мережу).

У рішенні технологія IP застосовується разом з оптичною технологією, що забезпечує синергію між транспортними рівнями обох, при збереженні гнучкості IP-мереж. Рішення зосереджене на наступних завданнях:

- Продуктивність і універсальність обслуговування
- Еволюція до програмно-конфігуруємої мережі (SDN)
- Удосконалені процеси експлуатації й техобслуговування, що приводять до зниження TCO

Рішення Agile WAN

Huawei розробила рішення Agile WAN для задоволення високих вимог хмарних обчислень в епоху Інтернету речей (IoT). Це рішення дозволяє

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

замовникам задовольняти вимоги організації їхніх мереж, завдяки орієнтованій на майбутнє архітектурі програмно-визначаємої мережі (SDN) за допомогою віртуалізації й технологій оптичної й IP-мережі.

Рішення Huawei Agile WAN дає замовникам наступні переваги:

- Значно підвищується ефективність використання пропускнуєї здатності WAN, що для традиційних WAN становить 20-30%.
- Спрощується експлуатація й техобслуговування (O&M), знижується TCO.
- Прискорюється впровадження послуг для задоволення вимог швидкого надання хмарних обчислень.

Рішення засноване на трьох принципах.

Динамічні послуги

Використовуючи технології вертикальної й горизонтальної віртуалізації, уніфікованої диспетчеризації й розгортання одним натисканням кнопки, рішення усуває перешкоди між IP і оптичними мережами, підтримуючи швидке надання хмарних послуг.

Agile Controller значно поліпшує використання мережі й знижує витрати на оренду ресурсів смуги пропускання для всієї мережі на основі інформації глобальної маршрутизації. Цей метод заміняє традиційні обчислення, засновані на маршрутизації окремих вузлів.

Крім того, Huawei розробила кумулятивне рішення для IP-мереж і оптичних мереж, що реалізує трирівневе забезпечення надійності: на рівні мережі, на рівні елементів і на рівні компонентів.

Динамічна еволюція

Рішення Huawei наближає майбутнє. Архітектура SDN надає відкрите ПЗ для швидкого й ефективного впровадження корпоративних послуг. Розроблені Huawei мікросхеми забезпечують відкритість послуг для апаратних пристроїв на фізичному рівні.

Для забезпечення гнучкості й можливості росту, Huawei розробляє мережі в широкому діапазоні ємності й масштабів. Huawei має можливість надати інтерфейси в діапазоні від 2 Мбіт/с до 100 Гбіт/с; мережі MAN для міст і глобальні мережі в масштабах континенту. Наші виняткові можливості дозволяють підтримувати стійкий розвиток у найближчі 10 років.

Динамічна експлуатація й техобслуговування

Продукція Huawei дозволяє заощадити засоби завдяки інноваційним технологіям контролю якості мережі, наприклад, IP FPM, щоб змінити традиційне сприйняття мережі як «чорного ящика» на візуальну динамічну мережу WAN.

Наша комплексна платформа O&M може одночасно управляти й обслуговувати пристрою на оптичному й IP-рівні. Вона може здійснювати доступ у всій мережі й прогнозувати ємність мережі за допомогою убудованого інтелектуального програмного забезпечення. Це полегшує O&M і знижує TCO.

Довідайтеся більше про те, як рішення Huawei Agile WLAN може допомогти вам наблизити майбутнє. Звернетеся в представництво вже сьогодні.

Agile WAN промислового класу

AWAN II забезпечує наявність масштабованої, надійної, безпечної й стабільно функціонуючої мережі, що підтримує гнучке розгортання послуг і інтелектуальне керування. Спектр послуг, що мають важливе значення для промислових підприємств, простирається від відеоспостереження до офісних і маркетингових послуг, які сильно розрізняються в плані безпеки, надійності, якості, затримок і вимог до пропускну здатності.

AWAN-II надає технологію інтеграції програмних і фізичних каналів, що забезпечує:

- Фізичну ізоляцію послуг.
- Фізичний IP-Канал.
- Коефіцієнт виявлення бітових помилок від 10^{-3} до 10^{-9} .
- Перемикання протягом 50 мс.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

- Затримку на рівні мікро- або мілісекунд.

AWAN II підтримує розширення ємності в режимі онлайн і скорочує строки впровадження від декількох днів до декількох хвилин за рахунок використання:

- Плат від N x 64k до 10 GE/100 GE.
- Плат одножильного оптоволокна 40T.
- Плат лінійної маршрутизації 1T.

Архітектура SDN

В архітектурі SDN здійснюється віртуалізація ресурсів і відділення рівня керування від рівня передачі для програмування основного устаткування. Гнучка архітектура дозволяє збільшити пропускну здатність від 10 Гбіт/с до 8 Тбіт/с для однієї пари волокна.

Новітня технологія дозволяє справлятися із проблемами інтенсивного трафіку й розташування об'єктів у декількох місцях. Технології оптичної мережі з автоматичною комутацією (ASON) і маршрутизації IP/MPLS забезпечують об'єднання керування трафіком, захисту мережі й можливостей експлуатації й техобслуговування для підвищення ефективності й зниження сукупної вартості володіння (TCO).

Точне інтелектуальне керування мережею

AWAN II здатне здійснювати інтелектуальне керування за допомогою потужних інструментів сіткового планування, вилученого тестування й спеціальної бібліотечної системи. Функції візуалізованого, вимірюваного й прогнозованого виміру робочих характеристик і можливість керування трафіком всієї мережі, аналізу ресурсів і виконання перевірки працездатності роблять дане рішення ідеальним вибором, незалежно від галузі.

Huawei Agile Controller (AC)

Huawei Agile Controller (AC) являє собою базовий компонент для програмно-визначаємих мереж (Software-defined networking, SDN). Він може застосовуватися для чотирьох основних бізнес-сценаріїв: для мереж центрів

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

обробки даних, а також кампусних, WAN і IoT-мереж. Завдання АС складається в забезпеченні автоматичної трансляції даних з застосунків на фізичну мережу, повного контролю мережевих ресурсів, а також візуалізацій функцій експлуатації й технічної підтримки. В основі АС лежать відкриті технології, що дає найширший простір для сумісності із продуктами сторонніх розроблювачів і створення на їхній основі комплексних рішень.

Для взаємодії з оточенням Agile Controller використовує кілька інтерфейсів: NBI (NorthBound Interface або Північний інтерфейс) і SBI (SouthBound Interface або Південний інтерфейс).

Розглянемо основні моменти й принципи інтеграції AgileController зі сторонніми системами й устаткуванням, на прикладі сценарію розгортання концепції SDN у центрі обробки даних.

NBI-інтерфейс дозволяє інтегрувати Agile Controller з найпоширенішими хмарними платформами, наприклад розробленими на базі OpenStack. SBI-інтерфейс дозволяє Agile Controller управляти безліччю мережевих елементів, таких як vSwitch, фізичні пристрої (комутатори, маршрутизатори, МСЕ й т.д.). Можливості Agile Controller також дозволяють здійснювати перетворення даних від сервісних API NBI-інтерфейсу в конкретну конфігурацію для автоматизації процесів у мережі через SBI-інтерфейс. Навіть без участі хмарної платформи, АС здатний самостійно інтегруватися із системами керування обчисленнями, такими як VMWare vCenter і Microsoft SystemCenter, для забезпечення спільної мережевої й обчислювальної взаємодії.

Agile Controller підтримує наступні периферійні інтерфейси, які відображені на структурній схемі.

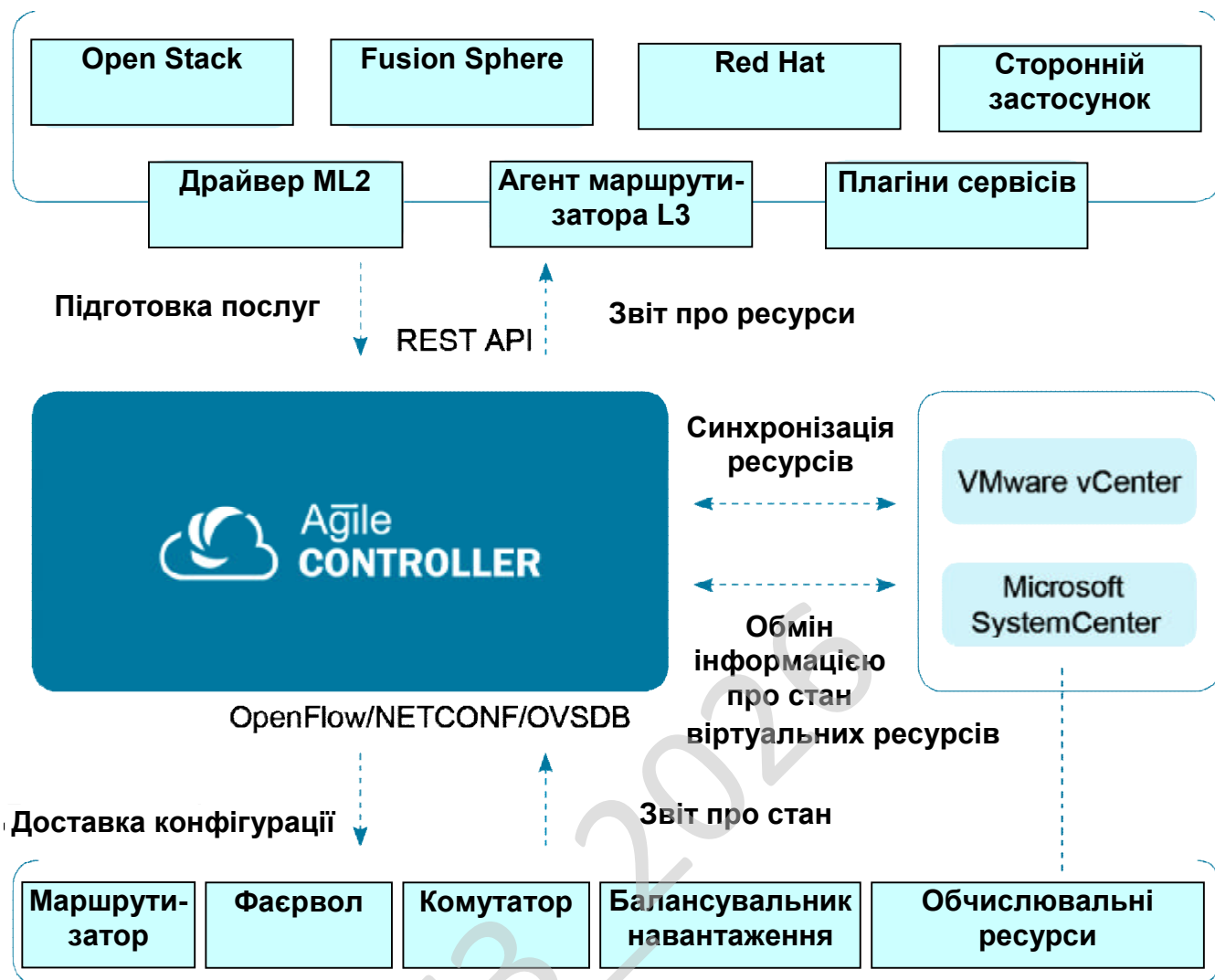


Рисунок 3.1 – Структурна схема системи.

3.3 Розробка функціональної схеми

NBI-інтерфейс для інтеграції з OpenStack

OpenStack Networking являє собою окремий сервіс, що часто використовується для розгортання декількох процесів на декількох вузлах. Ці процеси взаємодіють як один з одним, так і з іншими сервісами OpenStack. Основний процес сервісу OpenStack Networking – neutron-сервер, процес мовою

Python, що активує API OpenStack Networking і передає запити клієнтів системі плагінів для додаткової обробки.

OpenStack Networking, як компонент модуля Neutron, взаємодіє з контролером SDN через API Neutron. Компонент Neutron може автоматично одержувати інформацію про різні події, пов'язаних з віртуальними машинами (створення, видалення, міграція), через компонент NOVA, що відповідає за керування віртуальними компонентами. Контролер SDN одержує оповіщення про зміни статусу віртуальних машин через агента або плагін, убудований у компонент Neutron. Мережева конфігурація фізичної мережі може бути створена повністю автоматично через Agile Controller.

Сервіс neutron-server одержує через власний NBI-інтерфейс запит, потім відправляє запитовані сервісом параметри на АС по протоколі RESTCONF. У такий спосіб neutron-server може через АС управляти комутаторами vSwitch, фізичними комутаторами, а також VAS рівнів L 4-L7.

Інтеграція Layer 2 між АС і OpenStack

OpenStack інтегрується з Agile Controller через драйвер ML2, що дозволяє одержувати різну інформацію про мережу, включаючи підмережі й номери портів, а також здійснювати підготовку конфігурації віртуальної мережі.

Тип підтримуваної віртуалізації Layer2 – VXLAN (Virtual Extensible LAN). Технологія використовує 24-бітні ідентифікатори VN ID віртуальних мережевих сутностей, що дозволяє обійти обмеження в 4096 VLAN, що діє в традиційних мережах. Елемент мережі VTEP (VXLAN Tunnel End Point) забезпечує інкапсуляцію й декапсуляцію VXLAN, він може бути зконфігуrowан на комутаторі TOR або vSwitch. Для рішення VXLAN на апаратній базі вузол VTEP розташовується на TOR, а для рішення VXLAN на програмній базі VTEP розташовується на vSwitch.

Для мережі VXLAN на апаратній базі підтримка ієрархічної прив'язки портів повинна здійснюватися між АС і OpenStack. OpenStack призначає глобальний VN ID, а АС призначає VLAN на uplink-порт сервера, задає зв'язок

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

між призначеної локальної VLAN і глобальним VN ID на OpenStack, а потім задає зіставлення локальної VLAN і глобального VN ID на вузлі VTEP.

Для віртуальної мережі Layer2 також доступна підтримка груп безпеки й функції забезпечення якості обслуговування (QoS). Групи безпеки й QoS використовуються для перевірки безпеки в напрямку схід-захід, а також для контролю смуги пропускання на портах доступу віртуальних машин, відповідно.

Інтеграція Layer3 між AC і OpenStack

Agile Controller підтримує інтеграцію з OpenStack для створення пристроїв Layer3 – vRouter. vRouter використовується для забезпечення зв'язності Layer3 між віртуальними машинами того самого клієнта. Для елемента vRouter може бути використана як модель централізованого шлюзу, так і розподіленого шлюзу або розподіленого віртуального маршрутизатора (Distributed Virtual Router, DVR). Якщо використовується модель розподіленого шлюзу, то функції маршрутизатора можуть розміщатися на vSwitch або TOR. Перший з них являє собою рішення DVR на базі програмної платформи, а останній – рішення DVR на базі апаратної платформи. Якщо використовується модель централізованого шлюзу, то функціонал vRouter, як правило, розміщається на базовому комутаторі/AGG. На фізичному комутаторі кожний екземпляр vRouter зіставляється з екземпляром віртуального маршрутизатора (VRF), що має власну виділену таблицю маршрутизації й комутації.

Інтеграція з VAS

Agile Controller взаємодіє із хмарною платформою керування OpenStack. Взаємодія реалізована через плагіни для керування кінцевим пристроями фаєрволів/балансувальників. Таким чином, AC може управляти на фаєрволі функціями «плаваючих» IP-адрес, ACL і SNAT, IPSEC VPN і т.д.

SBI-інтерфейс

Agile Controller використовує комбінацію протоколів OVSDb і OpenFlow для контролю елементів vSwitch. При цьому OVSDb використовується для статичної конфігурації, а OpenFlow – для динамічного заповнення таблиць

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

потоків трафіку Layer2 і Layer3. Agile Controller може виявляти події створення, видалення й міграції віртуальних машин, а також запускати динамічну синхронізацію таблиць потоків трафіку через різні комутатори vSwitch.

Процес виявлення події створення віртуальної машини відбувається в такий спосіб:

- Оператор через OpenStack Nova створює віртуальну машину на конкретному фізичному сервері.

- vSwitch виявляє створення віртуальної машини за допомогою агента, убудованого в гіпервізор фізичного сервера, а потім посилає повідомлення про створення віртуальної машини на Agile Controller через OVSDB.

- Agile Controller виявляє подія – створення VM.

Agile Controller контролює фізичний комутатор за допомогою комбінації NETCONF і Yang (NETCONF – це протокол конфігурації, а Yang – це мова моделювання даних).

Agile Controller також підтримує протокол BGP EVPN, що діє між AC і апаратними пристроями. Він використовується для синхронізації таблиць комутації й маршрутизації між vSwitch (overlay) і апаратною мережею (underlay) у сценарії гібридного оверлея (hybrid overlay).

На рисунку 3.2 наведена функціональна схема системи.

При відсутності сценаріїв хмарної платформи, AgileController створює віртуальну мережу, використовуючи свій власний інтерфейс порталу. AC може взаємодіяти з додатком Virtual Machine Manager (VMM) через API SOAP/SDK.

AgileController призначає унікальний глобальний VN ID (VXLAN) для кожної віртуальної мережі й класичний VLAN (802.1q) для кожної віртуальної машини в рамках домену кожного VMM (наприклад, VMware vCenter або SystemCenter). Для оповіщення про доступ до VLAN від AC на VMM використовується SOAP API. Коли VMM створює віртуальну машину у віртуальній мережі, та повинна бути асоційована з VLAN, що використовується для ізолювання трафіку на локальному vSwitch.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

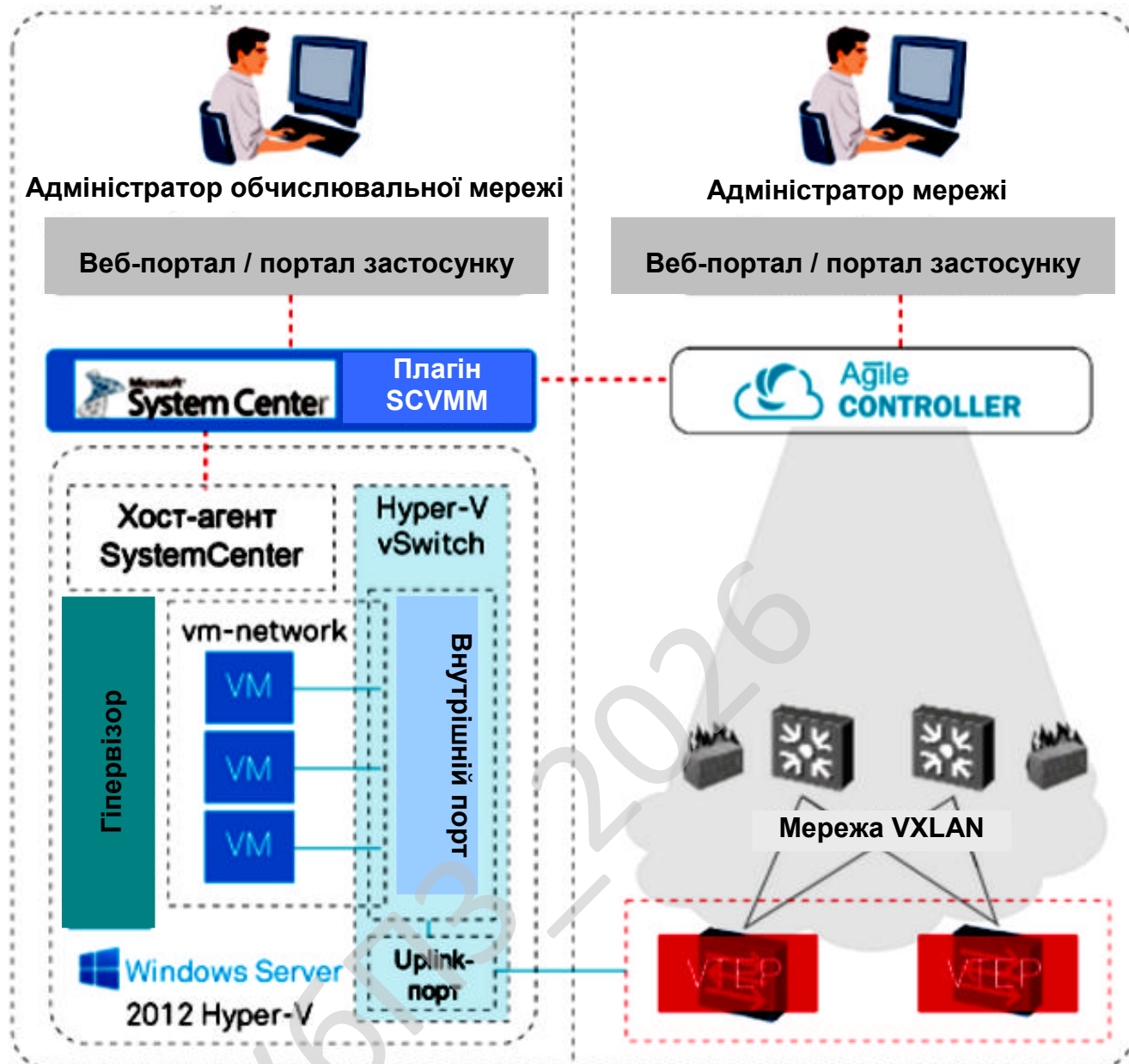


Рисунок 3.2 – Функціональна схема системи

Agile Controller підтримує динамічне виявлення подій створення, видалення й міграції віртуальних машин через API SOAP/SDK, а також автоматичний мапінг конфігурації між VLAN і глобальним VN ID на порту доступу. На закінчення варто сказати, що як центральний елемент керування мережею, якщо хочете «супермозку» мережі ЦОД контролер AgileController може повністю інтегрувати хмарну платформу, апаратний і програмний шари

мережі в рамках ЦОД і забезпечити взаємодія обчислювальних, мережевих і дискових ресурсів за допомогою стандартних відкритих інтерфейсів, включаючи NBI, SBI і інтерфейсу « East-West». Все це допомагає оптимально планувати ресурси в корпоративних хмарних сервісах, підвищувати ефективність мережі операторів зв'язку й підтримувати найвищу якість обслуговування.

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

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

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

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

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

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

Діаграми потоків даних містять чотири типи елементів:

- Зовнішні по відношенню до системи сутності.
- Потoki даних між елементами трьох попередніх типів.

- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).

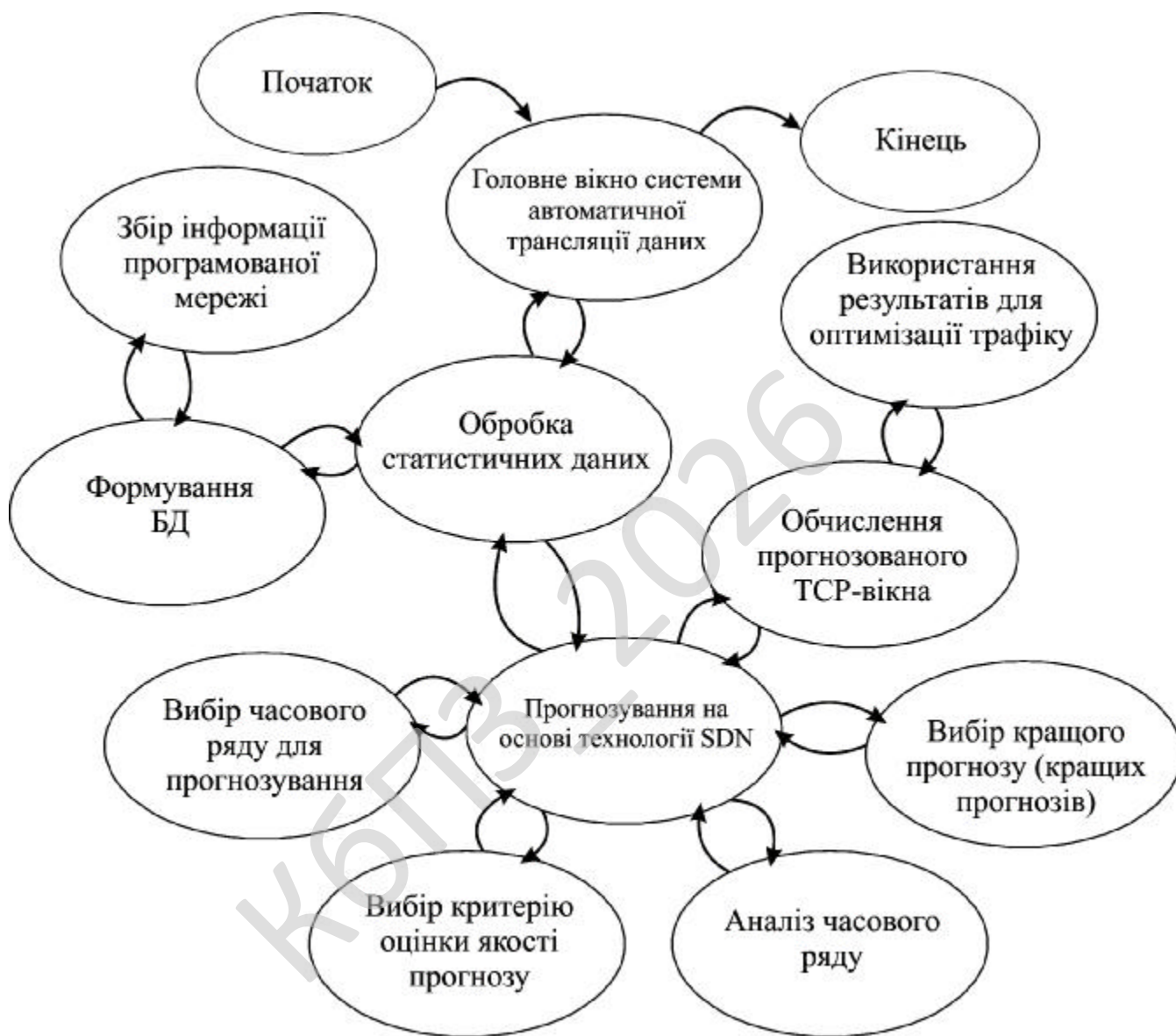


Рисунок 3.3 – Діаграма взаємодії процесів

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

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Для того, щоб оцінити трафік мережі та його зміну з плином часу, тобто побудувати залежність завантаженості трафіку від часу, й отримати часовий ряд, треба взяти дані про рух у мережі з таблиць маршрутизації відповідних маршрутизаторів. Для цього використовується наступна процедура.

```
def validate_ip_address(value: str, field_name: str) -> Optional[str]:
    try:
        ip_address(value)
        return None
    except ValueError:
        return f"Поле '{field_name}' містить некоректну IP-адресу"

def validate_port(value: Any) -> Optional[str]:
    try:
        port = int(value)
        if 1 <= port <= 65535:
            return None
    except (TypeError, ValueError):
        pass
    return "Порт повинен бути числом у діапазоні 1..65535"

def validate_positive_int(value: Any, field_name: str, min_value: int = 1) -> Optional[str]:
    try:
        number = int(value)
        if number >= min_value:
            return None
    except (TypeError, ValueError):
        pass
    return f"Поле '{field_name}' повинно бути цілим числом не менше {min_value}"

def validate_percent(value: Any, field_name: str) -> Optional[str]:
    try:
        percent = float(value)
```

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

```

        if 0 <= percent <= 100:
            return None
        except (TypeError, ValueError):
            pass
        return f"Поле '{field_name}' повинно бути числом у межах 0..100"

def validate_application_payload(data: Dict[str, Any]) -> ValidationResult:
    errors = []
    for field in ("name", "category", "owner", "environment"):
        error = validate_required_text(data.get(field), field)
        if error:
            errors.append(error)
    return ValidationResult(not errors, errors)

def validate_device_payload(data: Dict[str, Any]) -> ValidationResult:
    errors = []
    for field in ("hostname", "device_type", "location", "status"):
        error = validate_required_text(data.get(field), field)
        if error:
            errors.append(error)

    ip_error = validate_ip_address(str(data.get("management_ip", "")).strip(),
    "management_ip")
    if ip_error:
        errors.append(ip_error)

    cpu_error = validate_positive_int(data.get("cpu_capacity"),
    "cpu_capacity", 1)
    if cpu_error:
        errors.append(cpu_error)

    memory_error = validate_positive_int(data.get("memory_capacity"),
    "memory_capacity", 1)
    if memory_error:
        errors.append(memory_error)

    return ValidationResult(not errors, errors)

```

Після вибору критерію оцінки якості відбувається завдання екзаменаційної вибірки.

Для цього виділимо з вихідного часового ряду завантаженості трафіку навчальну вибірку $X_{ob} = \{x(1), x(2), \dots, x(N - \tau)\}$, на підставі спостережень якої побудуємо оцінки значень часового ряду завантаженості трафіку на тактах з 1 по $N - \tau$, і прогнознi значення на тактах з $N - \tau + 1$ по N . $X_{don} = \{x'(1), x'(2), \dots, x'(N - \tau), x'(N - \tau + 1), x'(N - \tau + 2), \dots, x'(N)\}$. Потім, вибравши k довільних крапок вихідного часового ряду, складемо з них екзаменаційну вибірку $X_{icn} = \{x(t_1), x(t_2), \dots, x(t_j), \dots, x(t_k)\}$, де $1 \leq t_j \leq N$.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

З отриманого допоміжного прогнозу X'_{don} виберемо оцінки значень часового ряду на часових тактах, що ввійшли в екзаменаційну вибірку $X'_{icn} = \{ x'(t_1), x'(t_2), \dots, x'(t_j), \dots, x'(t_k) \}$, $1 < t_j < N$. Зіставляючи отримані оцінки значень часового ряду X'_{icn} , зі значеннями екзаменаційної вибірки X_{icn} , оцінимо його якість, використовуючи різні критерії оцінки якості прогнозування. Таким чином, допоміжний прогноз буде перевірений на наявних даних.

Побудуємо прогноз X' , на підставі всіх наявних спостережень часового ряду X , використовуючи при цьому той же метод прогнозування, що й при побудові допоміжного прогнозу.

У зв'язку з тим, що оцінити якість прогнозу X на реальних даних не представляється можливим, будемо припускати, що його якість таке ж, як і якість допоміжного прогнозу.

Таким чином, залежно від критерію оцінки якості прогнозу й обраних прогнозних моделей, експерт вибирає крапки вихідного часового ряду, які необхідно включити в екзаменаційну вибірку, по крапках якої будуть перевірятися побудовані прогнози завантаженості трафіку. За замовчуванням екзаменаційними вважаються t останніх тактів часового ряду.

За завданням екзаменаційної вибірки відбувається суто побудова прогнозів.

Для побудови набору конкуруючих прогнозів необхідно натиснути кнопку «Побудувати прогноз» на панелі інструментів. Після розрахунку в нижній частині вікна системи прогнозування буде виведений ранжируваний список побудованих прогнозів із вказівкою використаної прогнозної моделі і її параметрів, значення критерію якості оцінки прогнозу, а також величина критерію «середня помилка». Цей критерій дозволяє наочно представити й зрівняти якість прогнозів, побудованих на підставі різних по масштабі даних, оскільки в ньому здійснюється перехід до відносних величин, що дозволяє експертові легко інтерпретувати величини помилок.

Після відображення результатів прогнозування відбувається візуальний аналіз часового ряду адміністратором мережі, тобто користувачем розробленої програми.

Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується. Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо.

Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

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

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

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач. Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції. Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента.

Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується.

Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи. Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані. Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи). З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми.

Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

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

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

буде об'єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми. З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

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

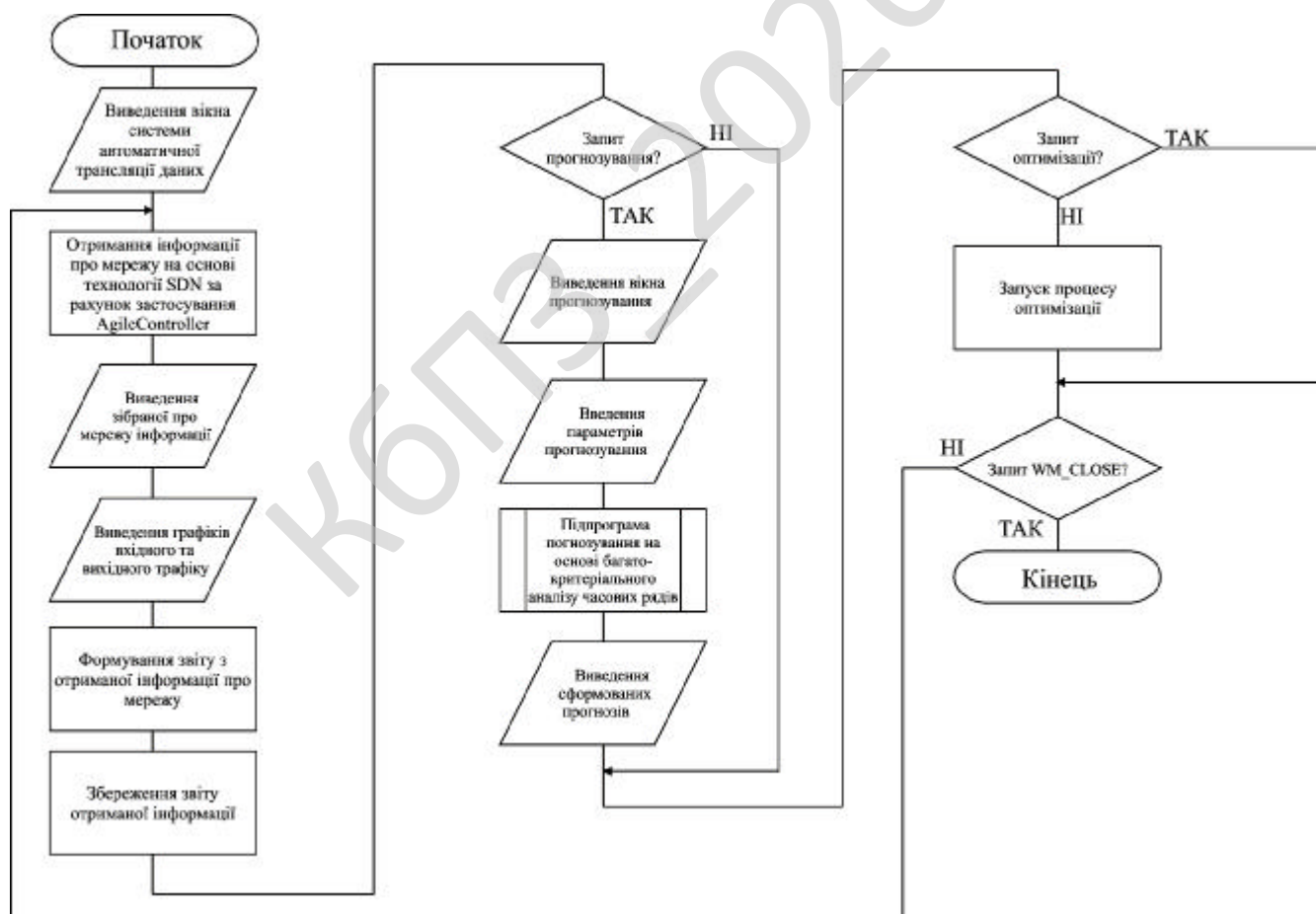


Рисунок 4.1 – Блок-схема основної програми

Незважаючи на те що я працював над ПЗ один в реалізації програми я використовував підходи пришвидшення розробки на основі методологій Agile.

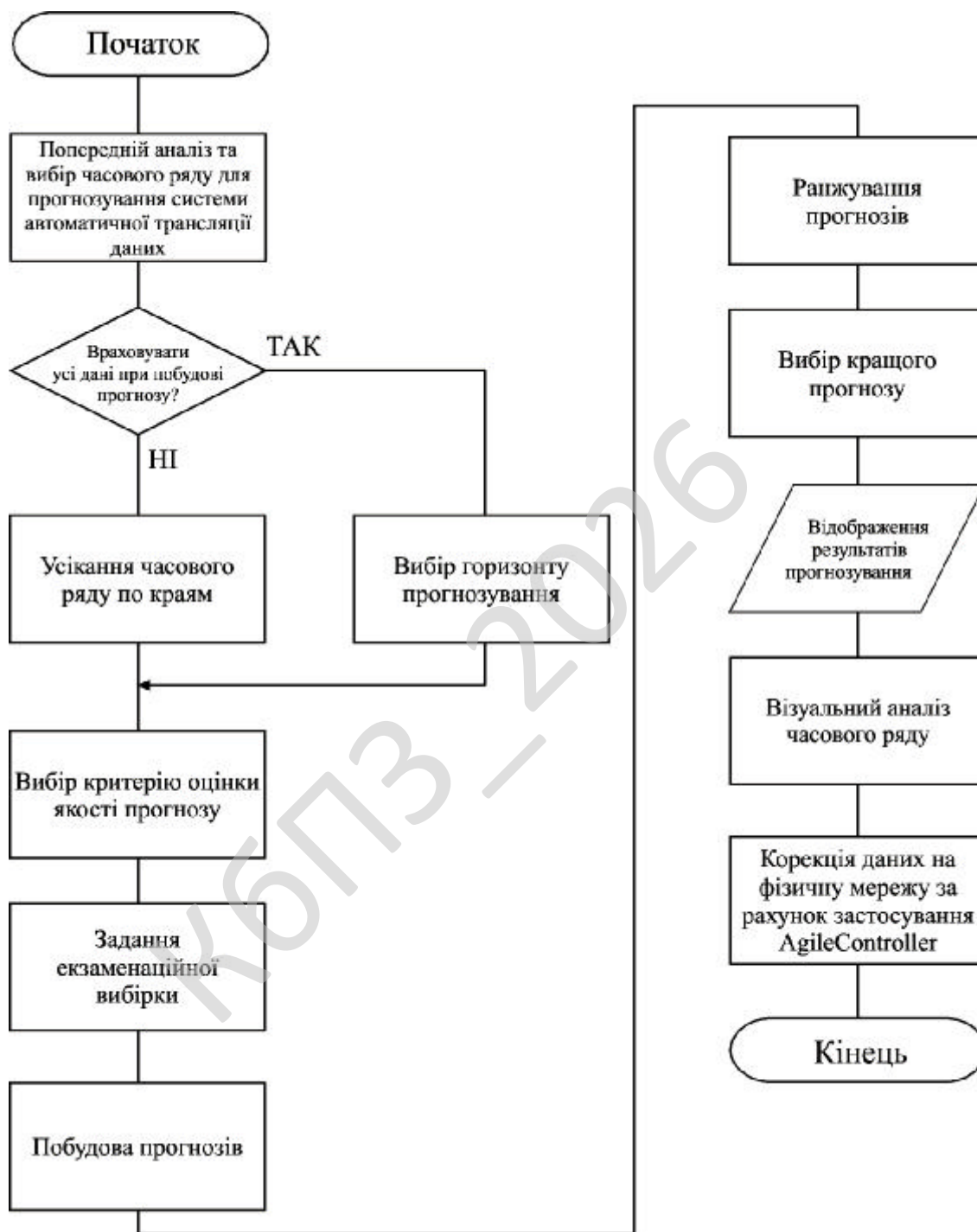


Рисунок 4.2 – Блок-схема роботи підпрограми

Гнучка́ розробка програмного забезпечення (Agile software development, agile-методи) – клас методологій розробки програмного забезпечення, що базується на ітеративній розробці, в якій вимоги та розв'язки еволюціонують через співпрацю між самоорганізовуваними багатофункціональними командами.

Гнучка розробка – найкращий засіб для підвищення продуктивності розробників програмного забезпечення.

Більшість гнучких методологій націлені на мінімізацію ризиків, шляхом зведення розробки до серії коротких циклів, що мають назву ітерацій, які зазвичай тривають один-два тижні. Кожна ітерація сама по собі виглядає як програмний проект в мініатюрі, і включає всі завдання, необхідні для видачі мінімального приросту за функціональністю: планування, аналіз вимог, проектування, кодування, тестування і документування. Хоча окрема ітерація, як правило, недостатня для випуску нової версії продукту, мається на увазі те, що гнучкий програмний проект готовий до випуску наприкінці кожної ітерації. Після закінчення кожної ітерації, команда виконує переоцінку пріоритетів розробки.

Agile акцентує увагу на безпосередньому спілкуванні «віч-на-віч». Більшість agile команд розташовані в одному офісі, його іноді називають bullpen. Як мінімум вона включає і «замовників» (замовники, які визначають продукт, також це можуть бути менеджери продукту, бізнес аналітики або клієнти). Офіс може також включати тестувальників, дизайнерів інтерфейсу, технічних авторів і менеджерів.

Основною метрикою agile методів є робочий продукт. Віддаючи перевагу безпосередньому спілкуванню, agile-методи зменшують обсяг письмової документації в порівнянні з іншими методами. Це привело до критики цих методів як недисциплінованих.

Agile – родина процесів розробки, а не єдиний підхід в розробці програмного забезпечення, і визначається Agile Manifesto. Agile не включає практик, а визначає цінності та принципи, якими керуються успішні команди.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

Agile Manifesto розроблений і прийнятий 17 розробниками 11-13 лютого 2001 року на лижному курорті The Lodge at Snowbird в горах Юти. Маніфест підписали представники наступних методологій Extreme programming, Scrum, DSDM, Adaptive software development, Crystal Clear, Feature driven development, Pragmatic Programming. Agile Manifesto містить 4 основні ідеї та 12 принципів. Примітно, що Agile Manifesto не містить практичних порад.

Основні ідеї:

- Особистості та їхні взаємодії важливіші, ніж процеси та інструменти;
- Робоче програмне забезпечення важливіше, ніж повна документація;
- Співпраця із замовником важливіша, ніж контрактні зобов'язання;
- Реакція на зміни важливіша, ніж дотримання плану.

Принципи, які роз'яснює Agile Manifesto:

- задоволення клієнта за рахунок ранньої та безперебійної поставки коштовного програмного забезпечення;
- вітання змін вимог навіть наприкінці розробки (це може підвищити конкурентоспроможність отриманого продукту);
- часта поставка робочого програмного забезпечення (кожен місяць або тиждень або ще частіше);
- тісне, щоденне спілкування замовника з розробниками впродовж всього проекту;
- проектом займаються мотивовані особистості, які забезпечені потрібними умовами роботи, підтримкою і довірою;
- рекомендований метод передачі інформації – особиста розмова (віч-на-віч);
- робоче програмне забезпечення – найкращий вимірювач прогресу;
- спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп на невизначений термін;
- постійну увагу поліпшенню технічної майстерності та зручному дизайну;
- простота – мистецтво не робити зайвої роботи;

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

- найкращі технічні вимоги, дизайн та архітектура виходять у самоорганізованої команди;
- постійна адаптація до мінливих обставин.

Маніфест та Принципи гнучкої розробки містять високорівневі ідеї щодо того, як потрібно вибудовувати процес розробки програмного забезпечення, щоб успішно завершувати проекти й створювати команди, в яких приємно та цікаво працювати.

Документи визначають, що потрібно для цього зробити, але не говорять, як це зробити. По-іншому й не могло бути, оскільки Маніфест та Принципи народилися внаслідок консенсусу представників різних (хоча й споріднених) напрямів, які могли знайти спільну основу лише на рівні базових цінностей та принципів.

Критика. Багато керівників проектів, що працюють у традиційних методологіях на кшталт «водоспаду», критикують agile-методи.

Один з повторюваних пунктів критики: при agile-підході часто нехтують створенням «дорожньої карти» розвитку продукту, так само як і управлінням вимогами, в процесі якого і формується така «карта». Гнучкий підхід до управління вимогами не має на увазі далекосяжних планів (по суті, управління вимогами просто не існує в даній методології), а має на увазі можливість замовника раптом і несподівано наприкінці кожної ітерації виставляти нові вимоги, що часто суперечать архітектурі вже створеного і поставленого продукту. Таке іноді призводить до катастрофічних «авралів» з масовим рефакторингом і переробками практично на кожній черговій ітерації.

Крім того вважається, що робота в agile мотивує розробників вирішувати всі прибулі завдання найпростішим і найшвидшим можливим способом, при цьому часто не звертаючи уваги на коректність коду з точки зору вимог базової платформи (підхід «працює, та й добре», при цьому не враховується, що може перестати працювати при найменшій зміні або ж породити важкі до відтворення

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

дефекти після реального розгортання у клієнта). Це призводить до зниження якості продукту і накопиченню дефектів.

Методології. Існують методології, які дотримуються цінностей і принципів заявлених в Agile Manifesto, деякі з них:

1. Agile Modeling – набір понять, принципів і прийомів (практик), що дозволяють швидко і просто виконувати моделювання і документування в проектах розробки програмного забезпечення. Не включає в себе детальну інструкцію з проектування, не містить описів, як будувати діаграми на UML.

Основна мета – ефективне моделювання і документування; але не охоплює програмування та тестування, не включає питання управління проектом, розгортання і супроводу системи. Однак включає в себе перевірку моделі кодом.

2. Agile Unified Process (AUP) спрощена версія IBM Rational Unified Process (RUP), розроблена Скоттом Амблером, яка описує просте і зрозуміле наближення (модель) для створення програмного забезпечення для бізнес-додатків.

3 Agile Data Method – група ітеративних методів розробки програмного забезпечення, в яких вимоги та рішення досягаються в рамках співпраці різних крос-функціональних команд.

4. DSDM заснований на концепції швидкої розробки додатків (Rapid Application Development, RAD). Являє собою ітеративний і інкрементний підхід, який надає особливого значення тривалій участі в процесі користувача/споживача.

5. Essential Unified Process (EssUP).

6. Екстремальне програмування (Extreme programming, XP).

7. Feature driven development (FDD) – функціонально-орієнтована розробка. Використовуване в FDD поняття функції або властивості (feature) Системи досить близько до поняття прецеденту використання, використовуваному в RUP, істотна відмінність – це додаткове обмеження: «кожна функція повинна допускати реалізацію не більше, ніж за два тижні». Тобто якщо сценарій використання

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

досить малий, його можна вважати функцією. Якщо ж великий, то його треба розбити на декілька відносно незалежних функцій.

8. Getting Real – ітераційний підхід без функціональних специфікацій, що використовується для веб-додатків. У даному методі спершу розробляється інтерфейс програми, а потім її функціональна частина.

9. OpenUP – це ітераційно-інкрементний метод розробки програмного забезпечення. Позиціюється, як легкий і гнучкий варіант RUP. OpenUP ділить життєвий цикл проекту на чотири фази: початкова фаза, фази уточнення, конструювання та передачі. Життєвий цикл проекту забезпечує надання зацікавленим особам та членам колективу точок ознайомлення і прийняття рішень впродовж усього проекту. Це дозволяє ефективно контролювати ситуацію і вчасно приймати рішення про задовільність результатів. План проекту визначає життєвий цикл, а кінцевим результатом є остаточний додаток.

10. Scrum встановлює правила керування процесом розробки та дозволяє використовувати вже існуючі практики кодування, коректуючи вимоги або вносячи тактичні зміни. Використання цієї методології дає можливість виявляти і усувати відхилення від бажаного результату на більш ранніх етапах розробки програмного продукту.

11. Бережлива розробка програмного забезпечення (lean software development). Використовує підходи з концепції бережливого виробництва.

Було використано наступні підходи UML: Діаграма компонент; Діаграма об'єктів; Діаграма розгортання.

Діаграма компонент в UML це діаграма, на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись.

Модуль програмного забезпечення може бути представлено в якості компоненти. Деякі компоненти існують під час компіляції, деякі – під час компонування, а деякі під час роботи програми.

Діаграма компонент відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортання.

Компоненти об'єднуються разом використовуючи структурні зв'язки (assembly connector) щоб об'єднати інтерфейси двох компонент. Це ілюструє зв'язок типу «клієнт-сервер».

Структурна взаємодія – «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту».

При використанні діаграми компонент щоб показати внутрішню структуру компонента, клієнтські та серверні інтерфейси можуть утворювати пряме з'єднання з внутрішніми. Таке з'єднання називається з'єднанням делегації.

Діаграма об'єктів в UML це діаграма, що відображає об'єкти та їх зв'язки в певний момент часу. Діаграма об'єктів може розглядатись як окремий випадок діаграми класів, на якій можуть бути представлені як класи, так і екземпляри (об'єкти) класів. Схожою за змістом є діаграма взаємодії (collaboration diagram).

Діаграми об'єктів не мають власної нотації. Оскільки діаграми класів можуть відображати об'єкти, то діаграма класів, на якій відображено лише об'єкти, та не відображено класи, може вважатись діаграмою об'єктів.

Діаграма об'єктів відображає об'єкти та зв'язки в певний момент роботи програми. Об'єкти можуть містити інформацію про власні значення а не про описання. Для відображення загальних шаблонів об'єктів та зв'язків, що можуть багаторазово створюватись під час роботи програми, слід використовувати діаграму взаємодії, яка може відображати характеристики об'єктів та зв'язків. Екземпляр діаграми взаємодії створює діаграму об'єктів.

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

Діаграма розгортання (deployment diagram) це діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах.

Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна відобразити на діаграмах компонент.

Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

Розглянемо визначення API. Це прикладний програмний інтерфейс (Application Programming Interface, API) – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

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

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

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

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

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

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

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

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

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

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

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

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

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

Прикладний програмний інтерфейс може бути також реалізацією протоколу.

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

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

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

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

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

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

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

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

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму Camellia – блоковий шифр на основі мережі Фейстеля. У криптографії, Camellia – це симетричний ключ блоковий шифр із розміром блоку 128 біт і розмірами ключа 128, 192 і 256 біт. Він був розроблений спільно Mitsubishi Electric і NTT з Японії. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу NESSIE і Японським CRYPTREC проект. шифр має рівні безпеки й можливості обробки, порівнянні з Advanced Encryption Standard.

Шифр був розроблений, щоб підходити як для програмних, так і для апаратних реалізацій, від недорогих смарт-карти для високошвидкісних мережних систем. Він є частиною криптографічного протоколу Transport Layer Security (TLS), призначеного для забезпечення безпеки зв'язки в комп'ютерній мережі, такий як Інтернет

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Camellia – це шифр Фейстеля з 18 раундами (при використанні 128-бітних ключів) або 24 раундами (при використанні 192- або 256-бітних ключів). Кожні шість раундів застосовується шар логічного перетворення: так звана «FL-функція» або її зворотна. Camellia використовує чотири 8×8 -бітних S-блоку із вхідними й вихідними афіними перетвореннями й логічними операціями. Шифр також використовує введення й вивід відбілювання клавіш. Шар дифузії використовує лінійне перетворення на основі матриці з номером галузей 5.

Аналіз безпеки

Камелія вважається сучасним надійним шифром. Навіть при використанні параметра меншого розміру ключа (128 біт) вважається неможливим зламати його за допомогою атаки грубої сили на ключі за допомогою сучасних технологій. Немає відомих успішних атак, що значно послабляють шифр. Шифр був схвалений для використання ISO / IEC, проектом Європейського Союзу NESSIE і Японським CRYPTREC проект. Японський шифр має рівні безпеки й можливості обробки, порівнянні із шифром AES/Rijndael.

Camellia – це блоковий шифр, який може бути повністю визначені мінімальними системами багатомірних багаточленів:

- Камелія (а також AES) S-блоки можуть бути описані системою 23 квадратних рівнянь в 80 членах.
- Розклад ключів можна описати 1120 рівняннями в 768 змінні з використанням 3328 лінійних і квадратичних членів.
- Увесь блоковий шифр можна описати 5104 рівняннями в 2816 змінні з використанням 14 592 лінійних і квадратичних членів.
- Усього потрібно 6224 рівняння з 3584 змінними з використанням 17 920 лінійних і квадратичних членів.
- Кількість вільних членів становить 11 696, що приблизно таке ж число, що й для AES.

Теоретично, такі властивості можуть дозволити зламати Camellia (і AES) за допомогою алгебраїчної атаки, такий як розширена розріджена лінеаризація, у т Майбутнє за умови, що атака стане можливою.

Хоча Camellia запатентована, вона доступна за безоплатною ліцензією. Це дозволило шифру Camellia стати частиною проекту OpenSSL під ліцензією з відкритим вихідним кодом з листопада 2006 року. Це також дозволило йому стати частиною Mozilla Модуль NSS (Служби мережної безпеки).

Підтримка Camellia була додана в остаточний випуск Mozilla Firefox 3 в 2008 році (за замовчуванням відключене починаючи з Firefox 33 в 2014 році в дусі «Пропозиції по зміні стандартних наборів шифрів TLS, пропонованих браузерами», який був виключено з версії 37 в 2015 році). Pale Moon, відгалуження Mozilla / Firefox, продовжує пропонувати Camellia і розширив свою підтримку, включивши в нього набори Galois / Counter mode (GCM) із шифром, але вилучив GCM знову у випуску 27.2.0, пославшись на очевидну відсутність інтересу до них.

Пізніше, в 2008 році, група розробки релізу FreeBSD оголосила, що цей шифр також був включений в FreeBSD 6.4. Крім того, Йошисато Янагисава додав підтримку шифру Camellia у дисковий клас зберігання geli FreeBSD.

У вересні 2009 року GNU Privacy Guard додала підтримку Camellia у версії 1.4.10.

Veracrypt (відгалуження Truecrypt) включав Camellia як один з підтримуваних алгоритмів шифрування.

Крім того, різні популярні бібліотеки безпеки, такі як Crypto ++, Gnutls, mbed TLS і Openssl також включають підтримку Camellia.

26 березня 2013 р. було оголошено, що Camellia була знову обрана для включення в новий список рекомендованих шифрів для електронного уряду Японії як єдиний 128-бітний алгоритм блокового шифрування, розроблений у Японії. Це збігається з тим, що список CRYPTREC обновляється вперше за 10 років. Вибір був заснований на високій репутації Camellia у плані простоти

придбання, а також характеристик безпеки й продуктивності, порівнянних з такими з Advanced Encryption Standard (AES). Камелія залишається незмінною у своєму повному втіленні. Неможлива диференціальна атака на Camellia з 12 раундами без шарів FL / FL дійсно існує.

Продуктивність

S-блоки, використовувані Camellia, мають структуру, аналогічну S-блоку AES. У результаті можна прискорити реалізацію програмного забезпечення Camellia за допомогою наборів команд ЦП, розроблених для AES, таких як x86 AES-NI.

К6ПЗ_2026

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі програмне забезпечення системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Верхнього меню: Файл; Налаштування; Дані; Довідка.
- Розділу обрання групи.
- Розділу виведення результату роботи системи.
- Функції представлені у графічному вигляді (іконки).

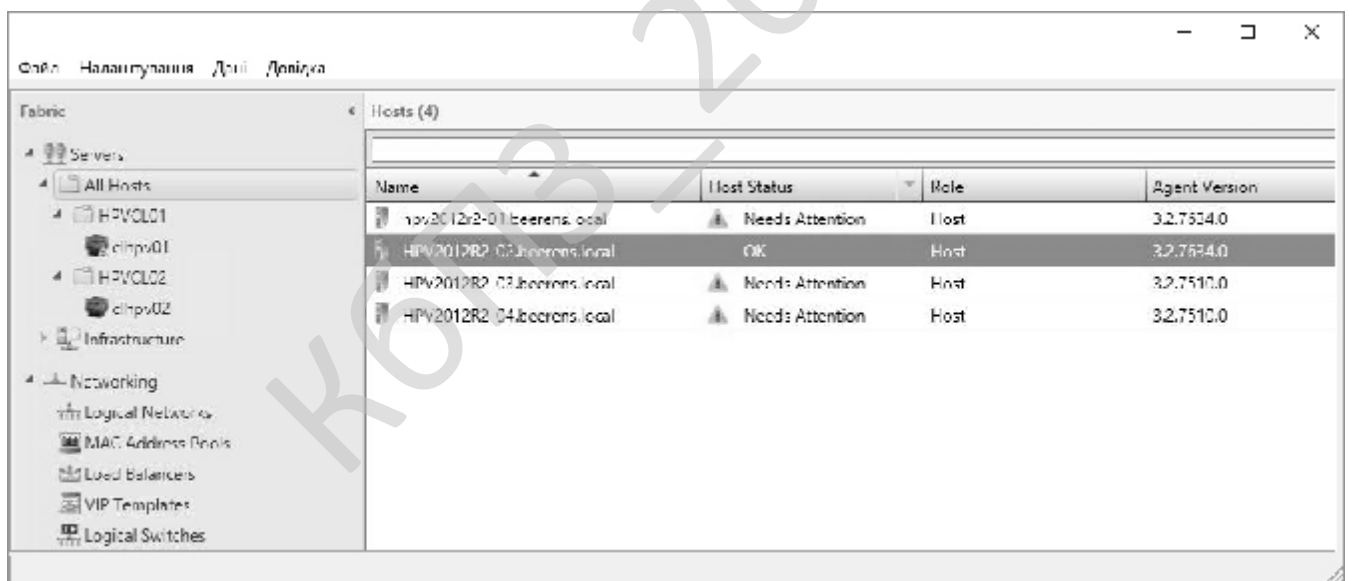


Рисунок 5.1 – Головне вікно ПЗ

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

оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

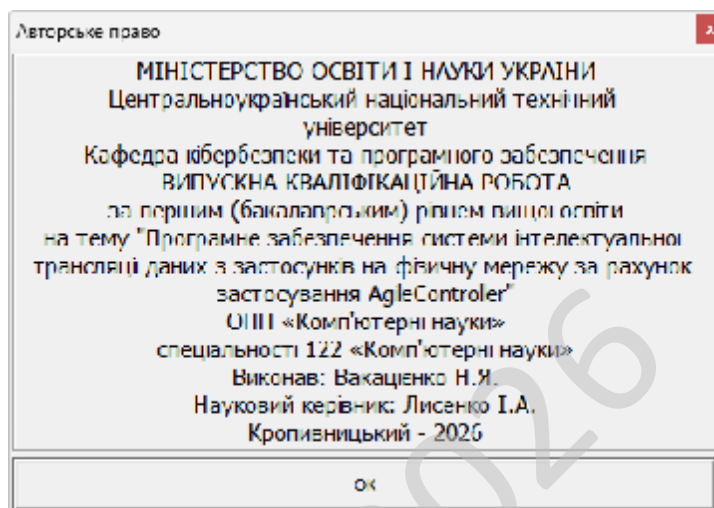


Рисунок 5.2 – Авторське право

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

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних

вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Оновлення.
- 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 в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

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

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

Проводилась оцінка:

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

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

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

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

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.

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

– У програмі можуть бути пропущені деякі маршрути.

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

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

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

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

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

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

Обрано умови розповсюдження – commercial software.

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

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

Далеко не завжди, але як правило терміни критично важливих змін в комерційних продуктах значно менше, ніж у некомерційних проектів. Це

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

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

К6ПЗ_2026

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.
- Досліджена система інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.
- На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController.

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

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

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

системи інтелектуальної трансляції даних з застосунків на фізичну мережу за рахунок застосування AgileController. 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 Camellia .

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.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Е. Таненбаум, Д. Уезеролл «Комп'ютерні мережі». – [5-е вид.]. – 2016. – 960 с.
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. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка» (Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.)*. Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.
9. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.
10. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

11. 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.

12. 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.

13. 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.

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

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

16. 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.

17. 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.

18. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

19. 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.

20. 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.

21. 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.

22. 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.

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

24. 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.

					ВКРБ-122.26.0005.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

25. 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.
26. 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.
27. 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.
28. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.
29. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.
30. Smirnov O. Kuznetsov A., 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.
31. 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.

32. 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.

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

34. 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.

35. 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.

36. 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.

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

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

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

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

41. 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.

42. 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.

43. 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.

44. 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.

45. 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.

46. 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.

47. 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).

48. 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

49. 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.

50. 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.