ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «**Метод впровадження SDN OpenFlow для централізованого керування мультивендорним мережевим обладнанням**»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
(код,найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Давід ТИМОШЕНКО
(підпис) *(ім'я, ПРІЗВИЩЕ здобувача)*

Виконав: здобувач вищої освіти група ІСД-42
Давід ТИМОШЕНКО
(ім'я, ПРІЗВИЩЕ)

Керівник Дмитро КОЗЛОВ
Phd *(ім'я, ПРІЗВИЩЕ)*

Рецензент: _____
(ім'я, ПРІЗВИЩЕ)

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти бакалавр
Спеціальність 126 Інформаційні системи та технології
Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедрою ІСТ
_____ Каміла СТОРЧАК

“____” _____2026 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Тимошенку Давіду Сергійовичу
(прізвище, ім’я, по батькові здобувача)

1.Тема кваліфікаційної роботи: Метод впровадження SDN OpenFlow для централізованого керування мультивендорним мережевим обладнанням
керівник кваліфікаційної роботи: Дмитро КОЗЛОВ Phd
(ім’я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від “20” лютого 2026 р. № 51

2. Строк подання кваліфікаційної роботи «30» травня 2026 р.

3. Вихідні дані кваліфікаційної роботи:
1. Інформаційні ресурси мережевої інфраструктури, що використовуються у процесі централізованого керування в мультивендорному середовищі.
2. програмні компоненти для реалізації SDN-рішення: емулятор мережі Mininet, SDN-контролер Floodlight, програмний комутатор Open vSwitch.
3. нормативні та методичні матеріали у сфері побудови та захисту комп’ютерних мереж, сучасні підходи до використання технологій SDN, принципи реалізації протоколу OpenFlow.

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

1. Аналіз предметної області та постановка задачі централізованого керування мережею у мультивендорному середовищі з використанням технологій SDN.
2. Проєктування архітектури системи централізованого керування мережею на основі SDN.
3. Оцінка ефективності функціонування SDN-рішення у мультивендорному середовищі.

5.Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання «15» лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Аналіз предметної області програмно-визначених мереж та огляд сучасних підходів до централізованого керування мережею	24.02-03.03.26	
2.	Вивчення технічної документації SDN-технологій, протоколу OpenFlow та інструментів реалізації (Mininet, Floodlight, Open vSwitch)	04.03-17.03.26	
3.	Постановка задачі, визначення функціональних вимог до системи централізованого керування мережею та формування технічного завдання	18.03-31.03.26	
4.	Розробка архітектури SDN-рішення: структура мережі, взаємодія контролера з комутаторами, побудова топології	01.04-24.04.26	
5.	Висновки по роботі	25.04-16.05.26	
6.	Розробка демонстраційних матеріалів, доповідь.	19.05-20.05.26	
7.	Оформлення бакалаврської роботи	21.05-30.05.26	

Здобувач вищої освіти _____ Давід ТИМОШЕНКО
(підпис) *(ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Дмитро КОЗЛОВ
(підпис) *(ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступня бакалавр: 56 стор., 19 рис., 8 табл., 24 джерел.

Мета роботи – розробка та обґрунтування методу впровадження SDN OpenFlow для забезпечення централізованого, гнучкого та ефективного керування мультивендорною мережевою інфраструктурою.

Об'єкт дослідження – мережеві інформаційно-комунікаційні системи та процеси їх централізованого керування в мультивендорному середовищі.

Предмет дослідження – методи впровадження технології програмно-визначених мереж (SDN) з використанням протоколу OpenFlow для централізованого керування мережевим обладнанням різних виробників.

Короткий зміст роботи. У бакалаврській роботі проведено аналіз сучасних підходів до побудови та керування мережевою інфраструктурою на основі технології Software-Defined Networking (SDN). Досліджено принципи функціонування програмно-визначених мереж, архітектуру OpenFlow та особливості централізованого керування мережевими пристроями в мультивендорному середовищі. Виконано аналіз проблем традиційних мереж і обґрунтовано доцільність використання SDN для підвищення гнучкості, масштабованості та ефективності адміністрування мереж.

У роботі розроблено архітектуру системи централізованого керування мережею на основі SDN та OpenFlow, реалізовано експериментальне середовище з використанням Mininet, Floodlight і Open vSwitch. Проведено тестування функціонування мережі в умовах нормальної роботи та під час мережевих атак типу DoS і ARP Spoofing.

КЛЮЧОВІ СЛОВА: SDN, OPENFLOW, ПРОГРАМНО-ВИЗНАЧЕНІ МЕРЕЖІ, ЦЕНТРАЛІЗОВАНЕ КЕРУВАННЯ, МУЛЬТИВЕНДОРНА МЕРЕЖА, МЕРЕЖЕВІ КОНТРОЛЕРИ, КОМУТАТОРИ, МЕРЕЖЕВА ІНФРАСТРУКТ

ЗМІСТ

ВСТУП 8
1 ТЕОРЕТИЧНІ ОСНОВИ ПРОГРАМНО-ВИЗНАЧЕНИХ МЕРЕЖ 11
1.1 Сутність та принципи побудови SDN-мереж 11
1.2 Архітектура та компоненти технології OpenFlow 16
1.3 Аналіз традиційних мереж та проблем мультивендорного середовища 20
2 ПРОЄКТУВАННЯ СИСТЕМИ ЦЕНТРАЛІЗОВАНОГО КЕРУВАННЯ МЕРЕЖЕЮ 24
2.1. Аналіз вимог до системи централізованого керування мережею 24
2.2. Розробка архітектури SDN-рішення на основі OpenFlow 28
2.3. Моделювання взаємодії мережевих пристроїв у мультивендорному середовищі 34
3 РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ SDN-РІШЕННЯ 46
3.1 Реалізація централізованого керування мережею з використанням OpenFlow 46
3.2 Тестування та забезпечення захисту від DoS атак 51
3.3 Відтворення атаки ARP Spoofing та оцінка ефективності впровадження SDN у мультивендорному середовищі 52
ВИСНОВКИ 56
ПЕРЕЛІК ПОСИЛАНЬ 57
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ 60

ВСТУП

Актуальність дослідження. У сучасних умовах стрімкого розвитку інформаційно-комунікаційних технологій мережеві інфраструктури стають дедалі складнішими, масштабнішими та більш різнорідними. Особливо це стосується мультивендорних середовищ, у яких використовується обладнання різних виробників із різними підходами до керування та конфігурації. Така різноманітність ускладнює адміністрування мереж, знижує їх гнучкість і створює додаткові труднощі у забезпеченні безпеки та ефективності функціонування.

Технологія програмно-визначених мереж (SDN) пропонує новий підхід до організації мережевої інфраструктури, заснований на відокремленні площини керування від площини передавання даних та централізації управління мережею. Використання протоколу OpenFlow дозволяє уніфікувати взаємодію з мережевими пристроями та забезпечити гнучке налаштування політик маршрутизації незалежно від виробника обладнання.

Водночас впровадження SDN супроводжується появою нових викликів, пов’язаних із безпекою мереж, зокрема ризиками реалізації атак на контролер, канали керування та механізми виявлення топології. Це зумовлює необхідність дослідження можливостей застосування SDN у мультивендорному середовищі, а також розробки ефективних підходів до забезпечення його надійності та захищеності.

Актуальність даної роботи визначається потребою у вдосконаленні методів централізованого керування мережами з використанням SDN-технологій, а також необхідністю підвищення рівня безпеки сучасних мережевих інфраструктур.

Об’єкт дослідження – мережеві інформаційно-комунікаційні системи та процеси їх централізованого керування в мультивендорному середовищі.

Предмет дослідження – методи впровадження технології програмно-визначених мереж (SDN) з використанням протоколу OpenFlow для централізованого керування мережевим обладнанням різних виробників.

Мета роботи – розробка та обґрунтування методу впровадження SDN OpenFlow для забезпечення централізованого, гнучкого та ефективного керування мультивендорною мережевою інфраструктурою.

Наукові завдання:

1. Проаналізувати сучасний стан розвитку програмно-визначених мереж та особливості їх застосування у мультивендорних середовищах.
2. Дослідити архітектурні принципи побудови SDN та механізми взаємодії мережевих компонентів за допомогою протоколу OpenFlow.
3. Визначити основні вразливості та загрози безпеці SDN-мереж, зокрема атаки на контролер, канали керування та механізми формування топології.
4. Розробити модель централізованого керування мережею на основі SDN-технологій для мультивендорного середовища.
5. Реалізувати експериментальну SDN-мережу з використанням інструментів Mininet, Floodlight та Open vSwitch.
6. Провести моделювання мережевих атак (DoS, ARP spoofing) та дослідити їх вплив на функціонування SDN-інфраструктури.
7. Розробити та апробувати механізми виявлення і протидії мережевим атакам у SDN-середовищі.
8. Оцінити ефективність впровадженого SDN-рішення з точки зору продуктивності, гнучкості та рівня безпеки.

Методи дослідження:

1. Аналіз сучасних підходів до побудови комп’ютерних мереж та SDN-технологій.
2. Системний аналіз архітектури програмно-визначених мереж.
3. Моделювання мережевої інфраструктури.
4. Експериментальне тестування роботи мережі.
5. Порівняльний аналіз ефективності традиційних і SDN-рішень.

Програмно-технічне рішення забезпечує реалізацію централізованого керування мережею з використанням технології програмно-визначених мереж (SDN). Запропонована система дозволяє здійснювати гнучке управління

мережевими ресурсами, контролювати потоки трафіку та оперативно реагувати на зміни стану мережі. Використання централізованого контролера забезпечує підвищення ефективності адміністрування та спрощує керування мультивендорною мережевою інфраструктурою.

Застосування протоколу OpenFlow у поєднанні з програмними засобами, такими як Mininet, Floodlight та Open vSwitch, дозволяє створити масштабоване та адаптивне середовище для дослідження та тестування мережевих рішень. Розроблена модель мережі забезпечує можливість моделювання різних сценаріїв функціонування, включаючи нормальний режим роботи та вплив мережевих атак.

У ході роботи було реалізовано механізми генерації та аналізу мережевого трафіку, що дало змогу дослідити поведінку системи під впливом DoS-атак і атак типу ARP spoofing. Запропоновано підхід до автоматичного виявлення аномальної активності та блокування зловмисного трафіку із використанням інструментів моніторингу та взаємодії з контролером через API.

Отримані результати можуть бути використані для підвищення ефективності керування сучасними мережевими інфраструктурами, а також для вдосконалення механізмів захисту SDN-мереж. Запропоноване рішення демонструє доцільність використання SDN-технологій як основи для побудови гнучких, масштабованих і захищених мереж.

1 ТЕОРЕТИЧНІ ОСНОВИ ПРОГРАМНО-ВИЗНАЧЕНИХ МЕРЕЖ

1.1 Сутність та принципи побудови SDN-мереж

Сучасні комп'ютерні мережі відіграють ключову роль у функціонуванні інформаційно-комунікаційних систем, забезпечуючи обмін даними між користувачами, серверами, хмарними платформами та різноманітними цифровими сервісами. З розвитком інформаційних технологій мережі суттєво ускладнилися: якщо раніше вони складалися з обмеженої кількості пристроїв і виконували переважно базові функції передачі даних, то сьогодні включають тисячі елементів, серед яких маршрутизатори, комутатори, сервери, системи зберігання даних, віртуальні машини та IoT-пристрої. Крім того, активне впровадження хмарних технологій, мікросервісної архітектури та мобільних застосунків призвело до динамічного характеру мережевої інфраструктури, яка постійно змінюється та потребує швидкої адаптації [1].

Зростання кількості мережевих пристроїв і сервісів безпосередньо вплинуло на складність управління мережею. У традиційних мережах кожен пристрій має власну площину керування (control plane) та площину передачі даних (data plane), що функціонують незалежно. Це означає, що адміністратор змушений налаштовувати кожен комутатор або маршрутизатор окремо, використовуючи специфічні інтерфейси керування, команди або протоколи, які можуть відрізнятися залежно від виробника обладнання. Такий підхід є трудомістким, потребує високого рівня кваліфікації та значно ускладнює централізоване управління мережею.

Особливо проблема проявляється в умовах мультивендорного середовища, де використовується обладнання різних виробників. У таких випадках виникає необхідність враховувати відмінності в реалізації протоколів, синтаксисі команд, механізмах конфігурації та політиках безпеки. Це призводить до збільшення часу на адміністрування, ускладнює інтеграцію нових рішень та підвищує ймовірність помилок під час налаштування. Навіть незначна помилка в конфігурації може

спричинити порушення роботи всієї мережі або створити вразливості для кіберзагроз. Низький рівень гнучкості традиційних мереж. Будь-які зміни в топології або політиках маршрутизації потребують ручного втручання та внесення змін у конфігурації багатьох пристроїв. Це значно уповільнює впровадження нових сервісів, масштабування інфраструктури та адаптацію до змінних умов роботи. У сучасних умовах, коли бізнес-процеси потребують швидкого реагування та високої доступності сервісів, такі обмеження стають критичними.

Ускладнення мережевих інфраструктур, зростання кількості пристроїв, необхідність підтримки різнорідного обладнання та недоліки ручного адміністрування зумовили потребу в новому підході до побудови та управління мережами. Відповіддю на ці виклики стала концепція програмно-визначених мереж (Software-Defined Networking, SDN), яка передбачає централізацію керування мережею, відокремлення площини керування від площини передачі даних та використання програмних інструментів для автоматизації мережевих процесів. Такий підхід дозволяє значно спростити адміністрування, підвищити гнучкість мережі та забезпечити ефективніше управління ресурсами.

Програмно-визначені мережі (Software-Defined Networking, SDN) — це сучасна концепція побудови та управління комп’ютерними мережами, яка передбачає перенесення функцій керування мережею у програмний рівень та централізацію прийняття рішень (рис. 1.1). На відміну від традиційних мереж, де кожен мережевий пристрій самостійно виконує як функції обробки трафіку, так і функції керування, SDN розділяє ці процеси, що дозволяє значно спростити адміністрування та підвищити ефективність роботи мережевої інфраструктури. У рамках цього підходу мережа розглядається як програмно керована система, що може гнучко налаштовуватися відповідно до потреб користувачів і вимог прикладних сервісів [2].

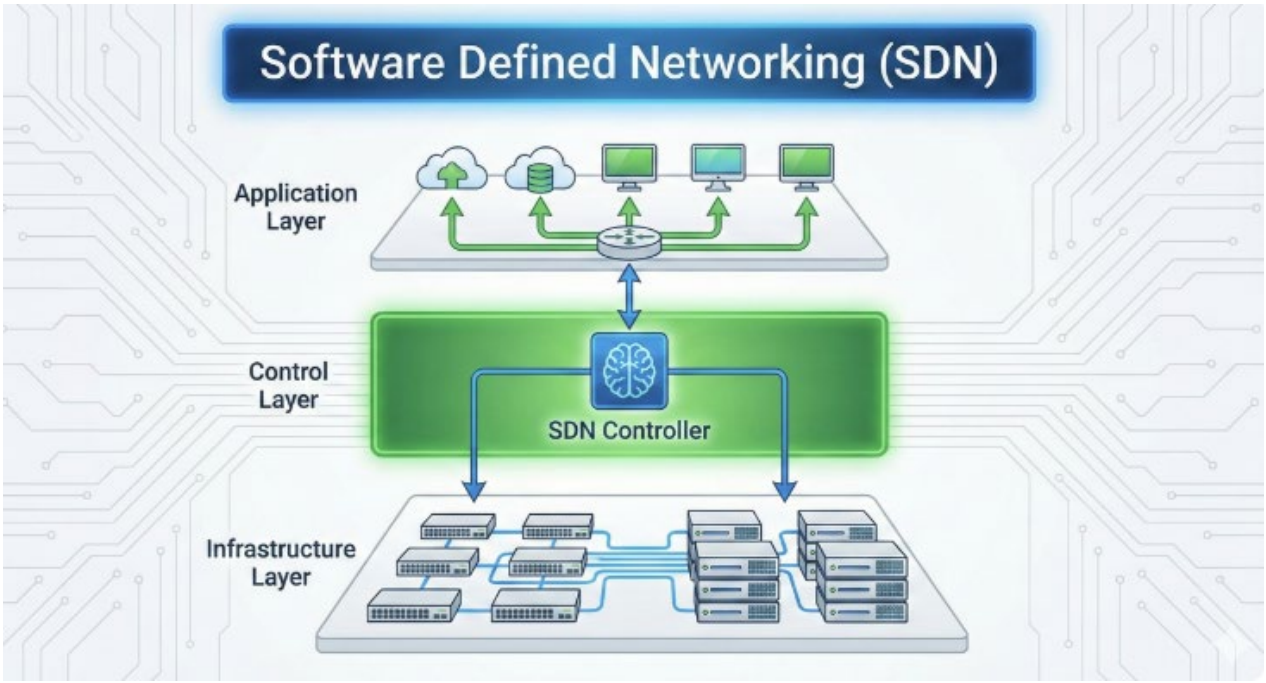

Рис. 1.1 Модель взаємодії рівнів у програмно-визначеній мережі

детальніше розглянути функціональне призначення її основних рівнів, кожен із яких виконує окрему роль у процесі управління та обробки мережевого трафіку.

Рівень застосунків (Application layer) є верхнім рівнем архітектури SDN і включає різноманітні програмні сервіси та додатки, що взаємодіють із мережею. До них належать системи моніторингу, балансування навантаження, забезпечення безпеки, управління політиками доступу, а також аналітичні інструменти. Саме на цьому рівні формуються вимоги до функціонування мережі, які передаються до контролера через відповідні програмні інтерфейси (API). Application layer не взаємодіє безпосередньо з мережевим обладнанням, а працює через логіку, реалізовану на рівні керування.

Рівень керування (Control layer) представлений централізованим елементом — SDN-контролером, який є ключовим компонентом усієї архітектури. Контролер відповідає за прийняття рішень щодо маршрутизації трафіку, розподілу ресурсів, реалізації політик безпеки та загального управління мережею. Він отримує інформацію про стан мережі від інфраструктурного рівня, аналізує її та формує відповідні команди для мережевих пристроїв. Завдяки централізованому підходу контролер забезпечує узгодженість дій усіх елементів мережі та дозволяє швидко

адаптувати її до змінних умов.

Інфраструктурний рівень (Infrastructure layer) є нижнім рівнем архітектури та включає фізичні та віртуальні мережеві пристрої, такі як комутатори та маршрутизатори. Основною функцією цього рівня є обробка та передача пакетів даних відповідно до правил, які надходять від контролера. На відміну від традиційних мереж, пристрої на цьому рівні не приймають самостійних рішень, а виконують інструкції, отримані з Control layer. Це дозволяє спростити їхню логіку роботи та підвищити ефективність функціонування всієї мережі.

Ідеєю SDN є відокремлення площини керування (control plane) від площини передачі даних (data plane). Площина керування відповідає за прийняття рішень щодо маршрутизації трафіку, визначення політик доступу та управління мережевими ресурсами. У традиційних мережах ці функції реалізуються безпосередньо на мережевих пристроях, що призводить до їх розподіленого характеру. Натомість у SDN control plane виноситься в окремий логічний елемент — контролер, який централізовано управляє всією мережею. Це дозволяє отримати повне уявлення про стан мережі та забезпечити узгоджене прийняття рішень [3].

Площина передачі даних (data plane), у свою чергу, відповідає за безпосередню обробку та пересилання пакетів між вузлами мережі. У SDN ця функція залишається на мережевих пристроях (комутаторах, маршрутизаторах), однак вони виконують лише інструкції, отримані від контролера. Таким чином, мережеві пристрої стають більш простими з точки зору логіки роботи, оскільки не приймають самостійних рішень, а діють відповідно до заданих правил.

Перед розглядом принципів функціонування програмно-визначених мереж доцільно узагальнити їх у структурованому вигляді. Це дозволяє наочно представити ключові особливості SDN та краще зрозуміти їх роль у забезпеченні ефективного управління мережею.

Таблиця 1.1

Основні принципи програмно-визначених мереж (SDN)

Принцип	**Характеристика**
Централізація керування	Передбачає винесення логіки керування мережею в централізований контролер, який приймає рішення щодо маршрутизації, політик доступу та управління трафіком.
Програмованість	Забезпечує можливість керування мережею за допомогою програмного коду та API, що дозволяє швидко змінювати конфігурації та впроваджувати нові функції.
Абстракція мережі	Полягає у відокремленні логічного рівня управління від фізичної інфраструктури, що спрощує взаємодію з мережею та її масштабування.
Автоматизація	Дозволяє автоматизувати процеси налаштування, моніторингу та реагування на події в мережі, зменшуючи вплив людського фактора.

Кожен із наведених принципів відіграє важливу роль у формуванні сучасного підходу до управління мережею. Їх поєднання забезпечує гнучкість, масштабованість та ефективність функціонування SDN, що вигідно відрізняє цей підхід від традиційних мережевих рішень.

Програмно-визначені мережі є підходом до побудови мережевої інфраструктури, що ґрунтується на відокремленні функцій керування та передачі даних. Така архітектура забезпечує централізацію управління мережею, що значно спрощує процес адміністрування та підвищує узгодженість роботи її елементів. Використання програмних механізмів керування дозволяє оперативно змінювати конфігурацію мережі, адаптувати її до нових вимог та ефективно управляти ресурсами [4].

SDN забезпечує високий рівень гнучкості, оскільки дозволяє швидко впроваджувати нові сервіси, масштабувати мережу та автоматизувати ключові процеси її функціонування. Завдяки централізованому підходу та використанню програмних інтерфейсів, мережа стає більш керованою та адаптивною до змінних умов сучасного цифрового середовища. Таким чином, SDN відкриває нові можливості для побудови ефективних, масштабованих і гнучких мережевих рішень.

1.2 Архітектура та компоненти технології OpenFlow

Одним із елементів реалізації концепції програмно-визначених мереж є протокол OpenFlow, який забезпечує взаємодію між централізованим контролером та мережевими пристроями. OpenFlow виступає як стандартний інтерфейс обміну даними між площиною керування (control plane) та площиною передачі даних (data plane), що дозволяє контролеру передавати інструкції комутаторам щодо обробки мережевого трафіку.

У межах SDN-архітектури OpenFlow виконує роль механізму, через який контролер отримує інформацію про стан мережі, а також формує правила для обробки пакетів. Комутатори, у свою чергу, виступають як виконавчі елементи, які реалізують ці правила, не приймаючи самостійних рішень щодо маршрутизації. Такий підхід дозволяє централізувати управління мережею та забезпечити узгоджене функціонування всіх її компонентів.

Протокол OpenFlow був розроблений та стандартизований організацією Open Networking Foundation (ONF), яка займається розвитком відкритих стандартів у сфері мережевих технологій. Завдяки цьому OpenFlow є універсальним рішенням, що підтримується різними виробниками мережевого обладнання, що особливо важливо для мультивендорних середовищ [5].

Функціонування протоколу OpenFlow базується на використанні спеціальних таблиць потоків (flow tables), які розміщуються на мережевих пристроях (комутаторах). Ці таблиці містять набір правил, що визначають, яким чином необхідно обробляти мережеві пакети. Кожне правило складається з умов співставлення (match) та відповідних дій (action), які мають бути виконані при виконанні цих умов.

При надходженні пакета до комутатора відбувається його аналіз за визначеними параметрами, такими як IP-адреса, MAC-адреса, номер порту, тип протоколу тощо. Далі комутатор намагається знайти у flow table правило, яке відповідає характеристикам пакета. Якщо відповідне правило знайдено,

виконується дія, зазначена у цьому правилі. У разі відсутності відповідного запису пакет може бути переданий до контролера для подальшого аналізу та прийняття рішення.

Основна логіка обробки пакетів у OpenFlow може бути представлена у вигляді моделі «match → action», де спочатку здійснюється співставлення параметрів пакета з умовами правила, а потім виконується відповідна дія. До типових дій належать пересилання пакета на певний порт, відкидання пакета, зміна його заголовків або передача інформації до контролера.

Для кращого розуміння принципу роботи OpenFlow розглянемо приклад. Припустимо, що комутатор отримує пакет із IP-адресою джерела 10.0.0.1 та IP-адресою призначення 10.0.0.2. У flow table може міститися правило: якщо пакет відповідає заданим IP-адресам (match), то його необхідно переслати на порт 2 (action). У цьому випадку комутатор, виявивши відповідність умовам, одразу виконує дію та пересилає пакет без залучення контролера.

Якщо ж відповідного правила не знайдено, комутатор надсилає пакет або його заголовок контролеру (повідомлення Packet-In). Контролер аналізує ситуацію, формує нове правило (Flow-Mod) та надсилає його назад до комутатора. Після цього всі подальші пакети аналогічного типу обробляються відповідно до встановленого правила.

Як показано на рисунку 1.2, при надходженні пакета комутатор спочатку виконує перевірку відповідності його параметрів записам у таблиці потоків. У разі знаходження відповідного правила виконується визначена дія, наприклад пересилання пакета на відповідний порт. Якщо ж відповідність не знайдена, пакет передається до контролера для подальшого аналізу та формування нового правила обробки.

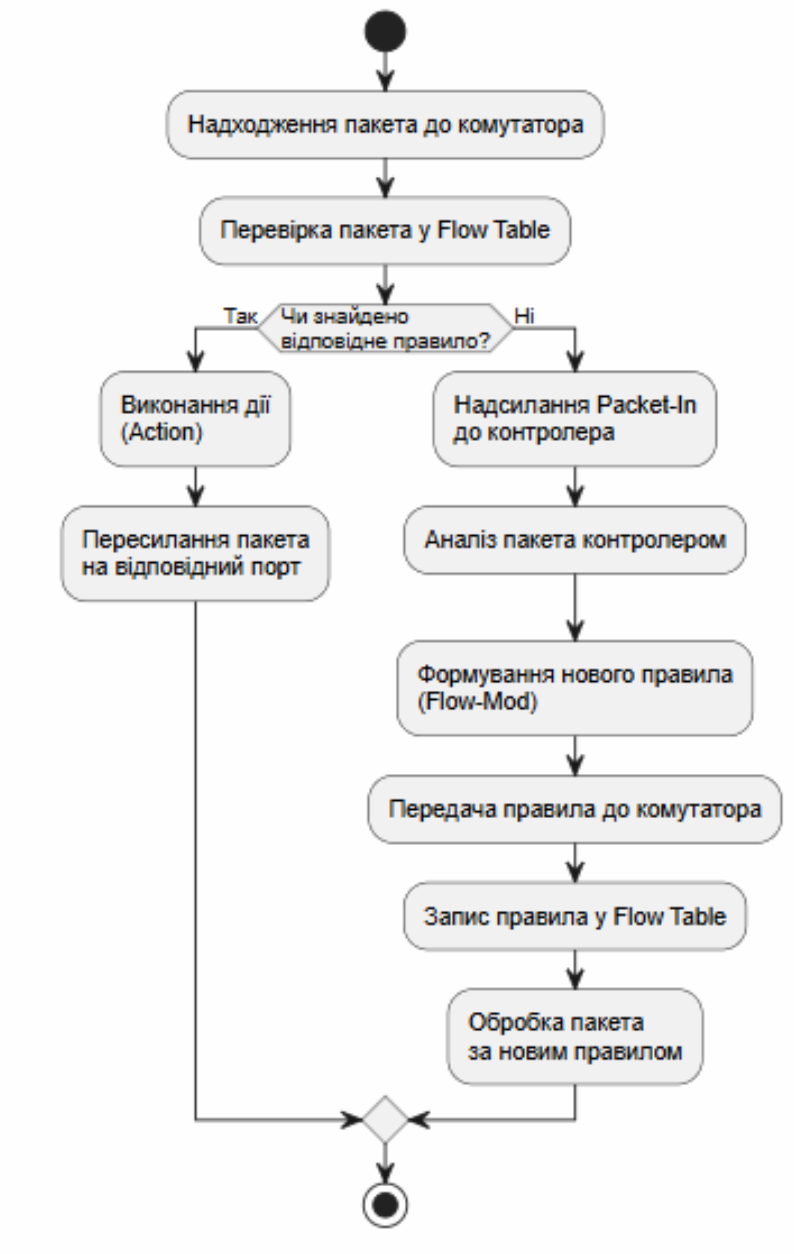

Рис. 1.2 Принцип обробки пакетів у OpenFlow (модель match–action)

Архітектура OpenFlow базується на взаємодії трьох ключових компонентів, які забезпечують централізоване керування мережею та ефективну обробку трафіку. До них належать контролер, комутатор та канали зв’язку між ними. Саме їх узгоджена робота дозволяє реалізувати принципи SDN та забезпечити гнучкість мережевої інфраструктури (табл. 1.2).

Таблиця 1.2

Ключові елементи архітектури OpenFlow

Компонент	**Опис**
Контролер	Центральний елемент керування мережею, який приймає рішення щодо маршрутизації трафіку, формує правила та передає їх до комутаторів. Забезпечує глобальне бачення мережі та централізовану логіку управління.
Комутатор	Мережевий пристрій, який виконує пересилання пакетів відповідно до правил, отриманих від контролера. Містить таблиці потоків (flow tables) та виконує обробку трафіку на основі моделі match–action.
Канали зв’язку	Канали взаємодії між контролером і комутаторами, через які передаються керуючі повідомлення (наприклад, Packet-In, Flow-Mod). Забезпечують обмін інформацією та синхронізацію стану мережі.

Узгоджена взаємодія зазначених компонентів дозволяє реалізувати централізовану модель управління мережею, де контролер виступає «інтелектуальним ядром», а комутатори виконують роль виконавчих елементів. Канали зв’язку забезпечують оперативний обмін даними, що дозволяє швидко реагувати на зміни мережевого трафіку та забезпечувати ефективне функціонування SDN-інфраструктури.

Для узагальнення ключових аспектів функціонування OpenFlow доцільно коротко розглянути типи службових повідомлень, а також основні переваги та недоліки цього підходу. Це дозволяє комплексно оцінити ефективність використання OpenFlow у сучасних мережах (табл 1.3).

Таблиця 1.3

Основні повідомлення, переваги та недоліки OpenFlow

Категорія	Елемент	Опис
Типи повідомлень	Packet-In	Надсилається комутатором до контролера у випадку відсутності відповідного правила для пакета.
	Flow-Mod	Використовується контролером для додавання, зміни або видалення правил у таблиці потоків комутатора.
	Packet-Out	Дозволяє контролеру безпосередньо вказати комутатору, як обробити конкретний пакет.
Переваги	Централізація	Забезпечує єдину точку керування мережею та глобальне бачення її стану.
	Гнучкість	Дає можливість швидко змінювати правила обробки трафіку відповідно до потреб.
	Незалежність від вендора	Дозволяє використовувати обладнання різних виробників у межах єдиної мережі.
Недоліки	Затримки	Можливі додаткові затримки при обробці пакетів через взаємодію з контролером.
	Залежність від контролера	Відмова контролера може призвести до порушення роботи всієї мережі.
	Безпека	Централізована архітектура створює нові вектори атак (на контролер або канал зв’язку).

OpenFlow забезпечує значні переваги у керуванні мережею, зокрема централізацію та гнучкість, однак водночас має певні обмеження, пов’язані із затримками, залежністю від контролера та питаннями безпеки. Це підкреслює необхідність впровадження додаткових механізмів захисту та оптимізації при використанні SDN-технологій.

1.3 Аналіз традиційних мереж та проблем мультивендорного середовища

Традиційні комп’ютерні мережі функціонують на основі розподіленої моделі керування (distributed control), відповідно до якої кожен мережевий пристрій виконує як функції обробки трафіку, так і прийняття рішень щодо його маршрутизації. У межах такої архітектури кожен комутатор або маршрутизатор

має власну площину керування (control plane) та площину передачі даних (data plane), які функціонують локально та незалежно від інших елементів мережі. Взаємодія між пристроями здійснюється за допомогою протоколів маршрутизації та обміну службовою інформацією, що забезпечує узгодження маршрутів і підтримку цілісності мережевої топології [6].

Попри ефективність такої моделі в умовах відносно стабільних мереж, сучасні вимоги до гнучкості, масштабованості та швидкості змін виявляють низку її обмежень. Однією з ключових проблем є висока складність конфігурації мережевих пристроїв. Налаштування виконується окремо для кожного елемента інфраструктури, що у великих мережах призводить до значних витрат часу та підвищує ризик виникнення помилок. Відсутність єдиного центру керування ускладнює координацію змін та контроль за їх впровадженням.

Суттєвим недоліком також є переважно ручний характер управління мережею. Використання командних інтерфейсів або окремих засобів адміністрування не забезпечує достатнього рівня автоматизації, що негативно впливає на оперативність реагування на зміни та інциденти. У цьому контексті особливої ваги набуває проблема людського фактора, оскільки навіть незначні помилки адміністратора можуть призвести до порушення роботи окремих сегментів або всієї мережі.

Окремої уваги потребує питання масштабованості, зі зростанням кількості мережевих пристроїв значно ускладнюється процес управління, оскільки будь-які зміни потребують оновлення конфігурацій на великій кількості вузлів. Це обмежує можливості швидкого розгортання нових сервісів та адаптації мережі до змінних умов експлуатації.

Традиційні мережі мають обмежені можливості щодо забезпечення інформаційної безпеки. Відсутність централізованого управління політиками безпеки ускладнює реалізацію єдиних підходів до контролю доступу, моніторингу трафіку та виявлення загроз. Це створює додаткові ризики, особливо в умовах зростання кількості кіберзагроз.

Зазначені проблеми набувають ще більшої актуальності у мультивендорному

середовищі, де використовуються мережеві пристрої різних виробників. Відмінності у підходах до конфігурації, підтримуваних протоколах та інтерфейсах управління ускладнюють інтеграцію таких пристроїв у єдину систему, що призводить до підвищення складності адміністрування та зниження загальної ефективності мережевої інфраструктури. Таким чином, обмеження традиційних мережевих підходів обумовлюють необхідність впровадження нових концепцій, зокрема Software-Defined Networking, які дозволяють подолати зазначені недоліки [7].

З характерних особливостей мультивендорного середовища є різноманітність інтерфейсів керування. Кожен виробник пропонує власний синтаксис командного рядка (CLI), програмні інтерфейси (API) та інструменти адміністрування, що ускладнює уніфікацію процесів конфігурації. Внаслідок цього мережеві адміністратори змушені володіти знаннями щодо різних платформ, що підвищує складність експлуатації мережі.

Проблеми мультивендорного середовища:

– складність інтеграції обладнання різних виробників у єдину мережеву інфраструктуру;

– відсутність уніфікованих підходів до конфігурації та управління;

– використання різних CLI та API, що ускладнює адміністрування;

– обмежена сумісність між пристроями через відмінності у реалізації протоколів;

– відсутність централізованого управління мережею;

– підвищене навантаження на мережевих адміністраторів;

– складність автоматизації процесів управління;

– ризик виникнення помилок через людський фактор;

– залежність від окремих постачальників (vendor lock-in);

– ускладнене впровадження нових технологій та оновлень.

Таблиця 1.4

Порівняльна характеристика традиційних мереж та SDN

Критерій	**Традиційна мережа**	**SDN**
Модель керування	Розподілене керування (distributed control)	Централізоване керування
Розділення площин	Control plane і data plane поєднані	Чітке розділення control plane і data plane
Гнучкість	Обмежена, залежить від обладнання	Висока, програмно керована
Налаштування	Ручне, на кожному пристрої окремо	Централізоване через контролер
Автоматизація	Обмежена	Високий рівень автоматизації
Масштабованість	Ускладнюється зі зростанням мережі	Легко масштабується
Управління мережею	Децентралізоване	Централізоване
Сумісність (мультивендорність)	Обмежена, залежить від виробника	Вища завдяки стандартам
Безпека	Складна реалізація політик	Централізоване управління безпекою
Впровадження змін	Повільне	Швидке та гнучке

Традиційні комп’ютерні мережі, побудовані на принципах розподіленого керування, характеризуються високою складністю адміністрування, обмеженими можливостями автоматизації та труднощами масштабування. Особливо ці проблеми загострюються в умовах мультивендорного середовища, де відсутність уніфікованих підходів до управління та сумісності ускладнює інтеграцію мережевих пристроїв і підвищує ризик помилок.

У свою чергу, концепція Software-Defined Networking пропонує ефективне вирішення зазначених проблем за рахунок централізації керування, програмованості та відокремлення площин управління і передачі даних. Це забезпечує підвищення гнучкості, спрощення адміністрування та створює передумови для побудови більш ефективних і керованих мережевих інфраструктур.

2 ПРОЄКТУВАННЯ СИСТЕМИ ЦЕНТРАЛІЗОВАНОГО КЕРУВАННЯ МЕРЕЖЕЮ

2.1. Аналіз вимог до системи централізованого керування мережею

У порівнянні з традиційними мережевими підходами, концепція SDN має низку суттєвих переваг, які безпосередньо впливають на підвищення рівня захищеності мережевої інфраструктури [17]. Зокрема, централізація управління, гнучкість налаштувань та можливість оперативного реагування на інциденти створюють більш контрольоване та безпечне середовище функціонування мережі. Основні характеристики, що визначають ці переваги, узагальнено та наведено в таблиці 2.1.

Таблиця 2.1

Ключові переваги архітектури SDN та їх практичне значення

Характеристика	Результат впровадження	Практичне використання
Глобальна видимість мережевої інфраструктури	Централізоване управління всіма мережевими елементами Агрегація та аналіз трафіку в реальному часі	Побудова систем виявлення вторгнень на рівні всієї мережі Виявлення аномальної або шкідливої активності мережевих пристроїв Проведення аналізу та розслідування інцидентів
Механізми автоматичного відновлення	Автоматичне застосування політик та правил Безперервний моніторинг мережевого трафіку	Динамічне блокування небажаного трафіку Автоматичне перенаправлення пакетів відповідно до політик
Моніторинг та контроль стану мережі	Організація мережі на основі потокових правил	Реалізація контролю доступу та управління політиками безпеки

Подальший розгляд передбачає детальний аналіз кожної з наведених характеристик.

1. Однією з ключових переваг архітектури SDN є можливість контролера в будь-який момент часу отримувати повне уявлення про стан усієї мережі. Саме ця особливість значною мірою визначає підвищений рівень безпеки у порівнянні з традиційними мережевими підходами. Такий ефект досягається за рахунок централізованої моделі управління, а також завдяки постійному збору статистичних даних з усіх мережевих пристроїв, які передаються до контролера.

На відміну від цього, у класичних мережах відсутній єдиний центр прийняття рішень, що ускладнює формування повної картини функціонування мережі. Для отримання навіть часткової інформації необхідний обмін значними обсягами даних між пристроями, а також певний час на узгодження їх станів. Крім того, ведення журналів подій і облік мережевого трафіку часто здійснюється лише окремими вузлами, що обмежує можливості аналізу.

Наявність глобального огляду мережі відкриває додаткові можливості для підвищення рівня її захищеності, серед яких варто виділити такі.

1) Комплексне виявлення вторгнень у масштабі всієї мережі.

Завдяки централізованій природі SDN контролер може інтегрувати механізми системи виявлення вторгнень (IDS), які функціонують не на рівні окремого вузла, а охоплюють всю мережеву інфраструктуру. Така система здійснює аналіз як статистичних даних трафіку, отриманих від комутаторів, так і журналів подій, що дозволяє своєчасно ідентифікувати підозрілу активність.

У традиційних мережах IDS, як правило, розгортається локально - на окремих пристроях або сегментах, що значно обмежує її ефективність через відсутність повної інформації про мережу.

Після отримання необхідних даних IDS у середовищі SDN може функціонувати за двома основними підходами:

– ***сигнатурний аналіз**:* система використовує базу відомих шаблонів атак і постійно порівнює поточний стан мережі з цими сигнатурами. У разі збігу фіксується факт вторгнення;

– ***поведінковий (аномалійний) аналіз***: на основі нормального функціонування мережі формується базовий профіль трафіку. Якщо в подальшому виявляються суттєві відхилення від цього профілю, система сигналізує про можливу загрозу.

Обидва підходи до виявлення вторгнень мають свої сильні та слабкі сторони. Зокрема, метод виявлення аномалій дозволяє ідентифікувати раніше невідомі типи атак і не потребує попередньо сформованої бази сигнатур зловмисної активності. Водночас його використання часто супроводжується підвищеною кількістю хибних спрацювань, оскільки будь-яке відхилення від типової поведінки мережі не обов'язково є ознакою атаки. На відміну від цього, сигнатурний підхід забезпечує більш точне виявлення відомих загроз, проте є обмеженим у випадку появи нових або модифікованих типів атак. У зв'язку з цим сучасні дослідження у сфері SDN спрямовані на вдосконалення IDS-рішень, зокрема на зниження обчислювального навантаження, підвищення точності аналізу та впровадження механізмів адаптивного навчання, які дозволяють системі автоматично оновлювати моделі як легітимної, так і зловмисної поведінки [17, 18].

2) Виявлення некоректної або зловмисної поведінки мережевих комутаторів. Наявність централізованого контролера з повним доступом до інформації про стан мережі дозволяє не лише аналізувати трафік, але й оцінювати поведінку самих мережевих пристроїв. Це створює додатковий рівень контролю, недоступний у традиційних архітектурах. Одним із характерних прикладів є ситуація, коли комутатор навмисно або через компрометацію починає відкидати частину або весь вхідний трафік, утворюючи так звану «чорну діру» (black hole). У класичних мережах виявлення такого пристрою є складним завданням, оскільки використовувані інструменти, зокрема traceroute, можуть бути неефективними у випадку вибіркового блокування трафіку. У середовищі SDN ця проблема вирішується значно ефективніше. Кожен комутатор регулярно передає контролеру статистику щодо кількості отриманих, переданих та відкинутих пакетів. Порівняння цих даних між сусідніми вузлами дозволяє виявити невідповідності та локалізувати потенційно скомпрометовані пристрої. Навіть у випадку, якщо

зловмисний комутатор намагається приховати свою поведінку шляхом підміни статистичних даних, аналіз інформації від суміжних пристроїв дає змогу встановити реальний стан обробки трафіку. Разом із тим, залишається відкритою проблема точного розмежування пакетів, втрачених унаслідок технічних збоїв каналу, та тих, що були відкинуті в результаті зловмисних дій, що потребує подальших досліджень і вдосконалення методів аналізу.

3) Однією з важливих переваг SDN є можливість централізованого збереження історичних даних про стан мережі. Контролер регулярно накопичує журнали подій і статистику трафіку, що значно спрощує проведення аналізу інцидентів. Завдяки цьому стає можливим відтворення мережевої активності за попередні періоди з метою встановлення механізмів реалізації атак, які не були своєчасно виявлені. Такий підхід дозволяє не лише зрозуміти природу інциденту, але й удосконалити засоби захисту для запобігання подібним загрозам у майбутньому. Крім того, аналіз історичних даних сприяє виявленню скомпрометованих вузлів, їх ізоляції, а також встановленню джерела атак, включаючи потенційно причетні особи або організації.

2. Механізми самовідновлення мережі.

Ще однією суттєвою особливістю SDN є наявність механізмів автоматичної реакції на події, які забезпечують здатність мережі до самовідновлення. Одним із прикладів таких механізмів є умовні правила, що встановлюються контролером і активуються при виконанні визначених умов.

Зазвичай ці умови базуються на аналізі статистичних показників, наприклад, коли обсяг трафіку певного типу перевищує встановлений поріг за визначений проміжок часу. У разі спрацювання умови комутатор автоматично виконує задану дію відповідно до визначеної політики.

До типових реакцій належать блокування підозрілого трафіку або його перенаправлення альтернативними маршрутами з метою зменшення навантаження на критичні сегменти мережі. Такий підхід дозволяє ефективно протидіяти атакам типу відмови в обслуговуванні (DoS), спрямованим як на окремі вузли, так і на мережеву інфраструктуру загалом.

Окремо варто відзначити можливість перенаправлення підозрілих пакетів до спеціалізованих ізольованих середовищ - honeypot. Такі середовища використовуються для детального аналізу шкідливої активності та збору додаткової інформації про методи атак. Важливо, що подібне перенаправлення здійснюється непомітно для джерела трафіку, що підвищує ефективність виявлення складних загроз, зокрема ботнетів.

3. Розширені можливості контролю мережевого трафіку.

У порівнянні з традиційними мережами, SDN забезпечує значно ширший рівень керування трафіком. Це пояснюється тим, що рішення щодо маршрутизації пакетів приймаються не лише на основі адреси призначення, а з урахуванням багатьох параметрів, що містяться у заголовках пакетів.

Контролер може визначати політики обробки трафіку залежно від різних характеристик, таких як адреса джерела, тип протоколу або інші атрибути. Наприклад, можливо обмежити передачу лише певного типу пакетів або дозволити доступ до мережі виключно визначеним вузлам.

Це дозволяє реалізувати більш ефективні механізми контролю доступу та фільтрації трафіку безпосередньо на рівні мережі. У традиційних мережах подібні функції здебільшого виконуються на кінцевих вузлах або міжмережевих екранах, що призводить до додаткового навантаження та затримок під час обробки пакетів.

2.2. Розробка архітектури SDN-рішення на основі OpenFlow

Рівень даних.

Однією з ключових проблем функціонування SDN на рівні даних є обмежені ресурси мережевих комутаторів, зокрема обсяг пам’яті для зберігання правил обробки трафіку. Оскільки контролер формує правила для різних типів мережевих потоків, їх повне збереження на комутаторі є неможливим. У зв’язку з цим у більшості реалізацій SDN застосовується механізм реактивного кешування.

Суть цього підходу полягає в тому, що при надходженні пакета, для якого відсутнє відповідне правило в таблиці потоків, комутатор тимчасово зберігає його

у буфері та надсилає запит до контролера у вигляді повідомлення типу PACKET-IN. Контролер, отримавши цей запит, аналізує параметри пакета та формує відповідне правило обробки, яке повертається до комутатора. Після цього пакет обробляється згідно з отриманою інструкцією, а саме правило зберігається у таблиці потоків для подальшого використання.

Попри свою ефективність, такий підхід створює певні вразливості. Зокрема, комутатори можуть стати об’єктом атак типу відмови в обслуговуванні (DoS). Зловмисник може генерувати значну кількість пакетів, що належать до різних потоків, для яких відсутні відповідні правила. Це призводить до масового формування PACKET-IN запитів і перевантаження як буфера комутатора, так і каналу взаємодії з контролером.

У разі використання пакетів із великим обсягом корисного навантаження ситуація ускладнюється ще більше, оскільки буфер комутатора швидко заповнюється. Як наслідок, легітимні пакети нових потоків можуть відкидатися через нестачу ресурсів, що негативно впливає на доступність мережі.

Для протидії подібним атакам запропоновано ряд підходів. Одним із найбільш ефективних є проактивне кешування, яке передбачає попереднє встановлення правил у комутаторах без необхідності очікування відповідних запитів. У цьому випадку контролер заздалегідь формує набір правил для найбільш імовірних сценаріїв трафіку.

Додатково застосовується використання узагальнених (агрегованих) правил потоків, які описують обробку одразу для групи пакетів із подібними характеристиками, а не для кожного окремого випадку. Це дозволяє суттєво зменшити кількість необхідних записів у таблицях потоків. Також важливу роль відіграє підвищення пропускної здатності каналів зв’язку між комутатором і контролером, що зменшує затримки обробки запитів і ускладнює реалізацію атак.

Обхід політик безпеки за допомогою шифрування трафіку.

Модель обробки трафіку в SDN базується на аналізі заголовків пакетів і дозволяє реалізувати гнучкі механізми контролю доступу. Контролер може

визначати, які типи пакетів дозволено передавати мережею, а які підлягають блокуванню або додатковій перевірці.

Однак використання шифрування значно ускладнює реалізацію таких механізмів. У випадку застосування зашифрованих тунелів значна частина інформації про пакет, включаючи його заголовки та корисне навантаження, стає недоступною для мережевих пристроїв. Інкапсульовані пакети передаються через мережу у зашифрованому вигляді, а їх обробка здійснюється лише після розшифрування на кінцевому вузлі.

Це створює можливості для обходу політик безпеки, оскільки шкідливий трафік може маскуватися під легітимний та проходити через мережу без належної перевірки. Таким чином, зловмисники можуть використовувати зашифровані канали для прихованого доступу або поширення атак.

Для протидії таким загрозам застосовуються методи непрямого аналізу зашифрованого трафіку. Зокрема, використовуються характеристики, які не залежать від вмісту пакетів: їх розмір, частота надходження, часові інтервали між передачами та інші статистичні параметри. Аналіз таких ознак дозволяє з певною точністю ідентифікувати підозрілу активність навіть у зашифрованих потоках.

Для кращого розуміння принципів функціонування запропонованого рішення доцільно розглянути узагальнену архітектуру SDN, побудовану на основі протоколу OpenFlow. Вона відображає взаємодію між рівнем керування та рівнем даних, а також демонструє механізм обробки мережевого трафіку.

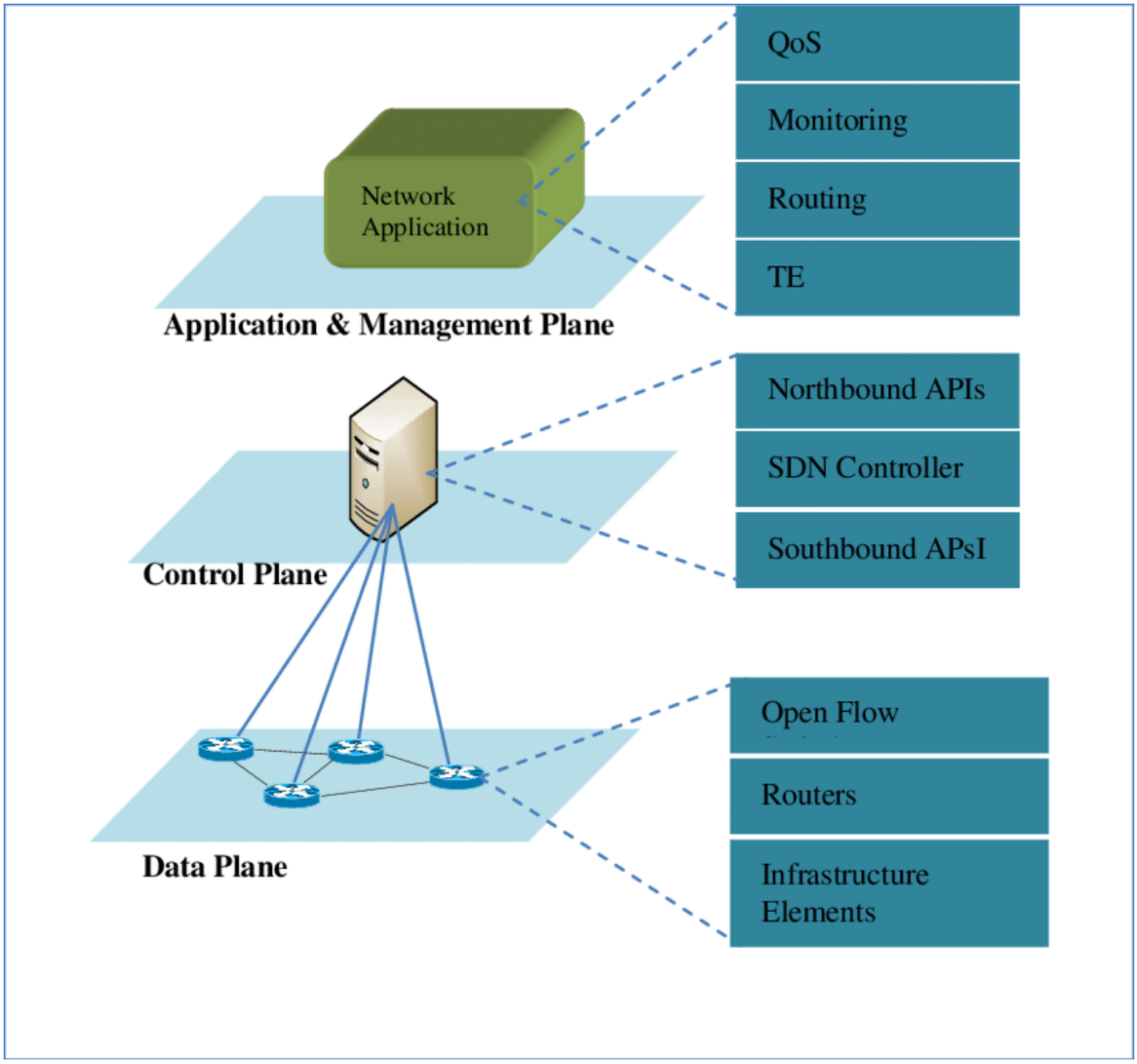

Рис. 2.1 Архітектура SDN на основі OpenFlow

Як показано на рисунку 2.1, архітектура SDN складається з двох основних рівнів: площини керування (Control Plane), представленої централізованим контролером, та площини даних (Data Plane), яка включає мережеві комутатори. Взаємодія між цими рівнями здійснюється за допомогою протоколу OpenFlow.

Контролер відповідає за прийняття рішень щодо обробки трафіку та формування правил маршрутизації, тоді як комутатори виконують ці правила, здійснюючи пересилання пакетів відповідно до таблиць потоків. У випадку відсутності відповідного правила комутатор надсилає запит до контролера (PACKET-IN), після чого отримує інструкцію для подальшої обробки трафіку.

Рівень керування.

Однією з критичних вразливостей площини керування в SDN є схильність до атак типу розподіленої відмови в обслуговуванні (DDoS). У такому сценарії значна кількість скомпрометованих вузлів генерує інтенсивний потік пакетів, які надходять до мережевих комутаторів. Частина цих пакетів не відповідає жодному правилу в таблицях потоків, що призводить до масового формування повідомлень типу PACKET-IN і їх передачі до контролера.

Надмірна кількість таких запитів створює значне навантаження на контролер, що може призвести до затримок обробки або навіть відмови у виконанні легітимних запитів. У результаті знижується доступність мережі та її загальна ефективність.

Одним із підходів до підвищення стійкості до подібних атак є використання архітектури з кількома контролерами. У такому випадку управління мережею розподіляється між кількома вузлами, що дозволяє балансувати навантаження. При цьому зберігається логічна централізація - мережа функціонує як єдина система, незважаючи на фізичну розподіленість контролерів.

Для забезпечення коректної роботи такої архітектури необхідна організація захищених каналів зв’язку між контролерами та узгодження правил обробки трафіку. Комутатори можуть бути підключені одночасно до декількох контролерів, а вибір активного контролера здійснюється з урахуванням затримок зв’язку та рівня навантаження. Такий підхід дозволяє підвищити відмовостійкість системи, однак ускладнює процес проєктування та оптимального розміщення контролерів у мережі.

Компрометація контролера.

Однією з найбільш небезпечних загроз є отримання зловмисником контролю над контролером. У цьому випадку атакуючий фактично отримує повний доступ до всієї мережі, оскільки контролер визначає правила обробки трафіку для всіх комутаторів.

Скомпрометований контролер може, наприклад, блокувати весь мережевий трафік або навпаки - перенаправляти його до певної цілі з метою здійснення атаки. Це

створює ризик повної втрати працездатності мережі або використання її як інструменту для атак на інші системи.

Одним із підходів до мінімізації ризиків є використання кількох контролерів. Проте якщо всі вони базуються на однаковому програмному забезпеченні або платформі, існує ймовірність їх одночасної компрометації через спільні вразливості. Тому доцільно застосовувати гетерогенний підхід - використання контролерів, побудованих на різних технологіях і програмних платформах.

Підміна контролера.

Ще одним можливим сценарієм атаки є підключення до мережі несанкціонованого контролера, який видає себе за легітимний. Це може бути реалізовано шляхом експлуатації вразливостей протоколів взаємодії, зокрема OpenFlow.

У такому випадку комутатори можуть почати виконувати команди зловмисного контролера, що створює серйозну загрозу безпеці мережі. Для протидії цьому рекомендується впровадження механізмів автентифікації контролерів, а також використання систем довіри або рейтингу, які дозволяють оцінювати надійність керуючих вузлів.

Атаки повторного відтворення (Replay-атаки).

Навіть при використанні захищених каналів зв’язку між контролером і комутаторами існує ризик перехоплення зашифрованих керуючих повідомлень. Зловмисник може зберегти ці повідомлення та повторно використати їх у майбутньому з метою повернення мережі до попереднього стану або виконання небажаних дій.

Для запобігання таким атакам доцільно застосовувати механізми контролю цілісності та актуальності повідомлень, зокрема використання часових міток або одноразових ідентифікаторів (nonce), що дозволяє виявляти повторне використання пакетів [19, 20].

2.3. Моделювання взаємодії мережевих пристроїв у мультивендорному середовищі

Недостатній рівень підтримки TLS

Однією з важливих проблем при побудові SDN-рішень у мультивендорному середовищі є нерівномірна підтримка захищених каналів зв'язку між контролером і мережевими пристроями. У початковій версії специфікації OpenFlow (v1.0) використання протоколу TLS для захисту каналу керування було обов'язковим. Проте у наступних версіях ця вимога була змінена на рекомендацію, що суттєво вплинуло на практику впровадження.

Основною причиною відмови від обов'язкового використання TLS стала складність його налаштування. Для коректної організації захищеного каналу необхідно виконати низку додаткових дій, зокрема: створення інфраструктури сертифікації, генерацію сертифікатів для контролера та комутаторів, їх підписання, а також встановлення відповідних ключів на кожному пристрої. У порівнянні з цим, організація незахищеного з'єднання потребує лише вказання адреси контролера, що значно спрощує процес розгортання мережі.

Унаслідок цього багато адміністраторів відмовляються від використання TLS, що створює додаткові ризики безпеки. Відкриті канали керування можуть бути перехоплені або модифіковані, що відкриває можливості для атак типу «людина посередині» (MITM), підміни керуючих повідомлень або несанкціонованого доступу до мережі [21-23].

Ситуація ускладнюється також тим, що через швидкий розвиток SDN-технологій різні виробники реалізовували підтримку OpenFlow з певними відхиленнями від специфікації. Це призвело до того, що не всі комутатори та контролери однаково підтримують TLS або взагалі його не реалізують.

Результати аналізу підтримки TLS у популярних рішеннях наведено в таблицях 2.2 та 2.3.

Таблиця 2.2

Аналіз реалізації TLS у мережевих комутаторах SDN

Виробник комутаторів	Підтримка TLS
HP	Відсутня [21]
Brocade	Відсутня
Dell	Відсутня
NEC	Обмежена (для окремих моделей)
Indigo	Відсутня
Pica8	Відсутня
OpenWRT	Підтримується [22]
Open vSwitch	Підтримується [23]

Таблиця 2.3

Порівняльний аналіз підтримки TLS у контролерах SDN

Контролер OpenFlow	Підтримка TLS
NOX	Часткова (автентифікація лише контролера)
POX	Підтримується [24]
Beacon	Відсутня [25]
Floodlight	Відсутня [26]
MuL	Відсутня [27]
FlowVisor	Підтримується [28]
Big Network Controller	Відсутня
Open vSwitch Controller	Підтримується [25]

Спостерігається загальна тенденція до недостатньої реалізації підтримки TLS як з боку комутаторів, так і з боку контролерів. У результаті обидві сторони фактично уникають додаткових витрат і складнощів, пов’язаних із впровадженням захищених каналів. Показовим є висловлювання одного з розробників контролера Floodlight, який зазначив, що додавання TLS не становить значної складності, однак відсутність попиту з боку користувачів не стимулює його впровадження. Це, у свою чергу, пояснюється обмеженою підтримкою TLS на рівні комутаторів [23].

Недостатня увага до захисту каналів керування створює сприятливі умови для реалізації атак у мережах OpenFlow. Відсутність шифрування та автентифікації дозволяє зловмисникам здійснювати несанкціонований доступ і залишатися малопомітними протягом тривалого часу. Варто зазначити, що рівень ризику залежить від середовища розгортання. У добре захищених центрах обробки даних фізичний доступ до обладнання обмежений, що значно знижує ймовірність атак. Натомість у кампусних мережах або віддалених офісах, де мережеві пристрої можуть розміщуватися у менш захищених місцях, загрози є значно більш реалістичними.

У разі, якщо зловмиснику вдається інтегрувати свій пристрій у канал взаємодії між контролером і комутатором, він отримує можливість перехоплювати керуючі повідомлення OpenFlow. Це відкриває доступ до широкого спектра атак: від встановлення додаткових правил обробки трафіку до модифікації або перехоплення конфіденційної інформації, а також до порушення роботи мережевих сервісів.

Хоча подібні атаки теоретично можливі і в традиційних мережах, їх реалізація значно ускладнена через різноманітність протоколів керування та специфічні формати повідомлень у різних виробників. У випадку SDN ситуація відрізняється, оскільки використання OpenFlow стандартизує механізми керування та збору статистики, що знижує технічний поріг для атакуючого.

Особливо небезпечними є атаки типу «людина посередині» (man-in-the-middle) у випадку використання in-band архітектури SDN. У такому середовищі зловмисник може не лише перехоплювати трафік, але й оперативно змінювати конфігурацію мережі, впливаючи на всі підконтрольні комутатори. На відміну від традиційних мереж, де для цього необхідно отримати облікові дані доступу до кожного пристрою окремо, у SDN достатньо отримати контроль над каналом взаємодії з контролером.

Постійний обмін повідомленнями між контролером і комутаторами у поєднанні з відсутністю належної автентифікації значно спрощує реалізацію атак.

Зловмисник може швидко встановити правила, які дозволяють непомітно перехоплювати або дублювати трафік, що ускладнює його виявлення.

Наприклад, у представленій архітектурі (рисунок 2.2) атакуючий може забезпечити прозору передачу керуючих повідомлень між контролером і комутатором, одночасно додаючи власні правила. Це дозволяє дублювати окремі потоки трафіку, наприклад ті, що пов’язані з обміном даними між клієнтами та базами даних, і перенаправляти їх на сторонній вузол для подальшого аналізу або використання.

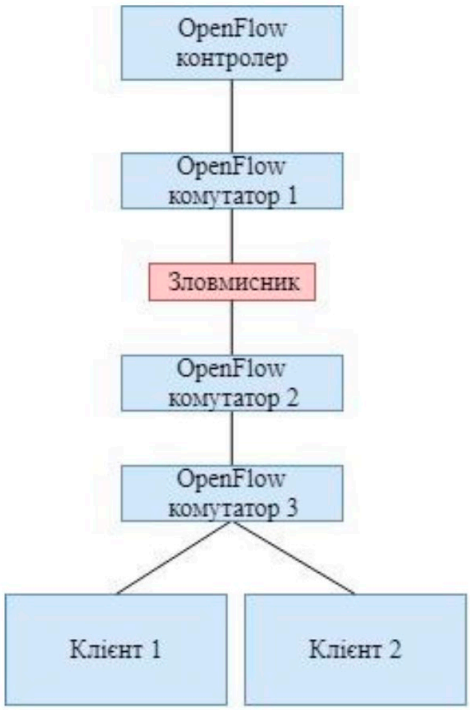

Рис. 2.2 Схема реалізації атаки типу «людина посередині» в SDN-мережі з in-band керуванням

У традиційних мережах організація перехоплення або дублювання окремих потоків трафіку між двома вузлами з подальшою передачею їх на віддалений сервер є доволі складним завданням. Така операція зазвичай потребує ручного налаштування кожного мережевого пристрою, отримання облікових даних для доступу, а також використання додаткових функцій, які підтримуються не всіма комутаторами, наприклад, інкапсуляції GRE або аналізу транспортних заголовків.

Хоча подібні атаки були можливі і в традиційних мережах, їх реалізація вимагала значних технічних зусиль. У середовищі SDN, зокрема при використанні OpenFlow, складність виконання таких дій суттєво знижується. Це пов’язано зі стандартизацією механізмів керування мережею, а також із можливістю централізованого встановлення правил обробки трафіку. У результаті подібні атаки можуть бути автоматизовані та навіть інтегровані у шкідливе програмне забезпечення.

Додатковим фактором ризику є усунення проблеми неоднорідності конфігурацій різних мережевих пристроїв, що раніше ускладнювало реалізацію атак. Крім того, відсутня необхідність очікування перехоплення даних для автентифікації, що ще більше спрощує доступ до керування мережею.

Рівень загрози значно зростає у випадку, якщо комутатор працює в режимі прослуховування («listener mode»). У такій конфігурації зловмиснику не потрібно здійснювати складну атаку типу man-in-the-middle. Достатньо виявити відповідний пристрій за допомогою засобів мережевого сканування, наприклад із використанням інструментів на кшталт nmap [24].

Отримавши доступ до такого комутатора, атакуючий може збирати інформацію про активні мережеві потоки, проводити розвідку та надалі встановлювати правила для перехоплення або модифікації трафіку. Це створює передумови для реалізації складніших атак і подальшого використання мережі як інструменту для компрометації інших систем.

У більшості поширених SDN-контролерів, таких як OpenDaylight, Floodlight, NOX, POX, Beacon, Ryu та Cisco Open SDN Controller, механізм виявлення топології мережі (OFDP - OpenFlow Discovery Protocol) базується на використанні протоколу LLDP. При цьому передача службових повідомлень LLDP, як правило, здійснюється без застосування шифрування, що створює додаткові вразливості.

Використання незахищених LLDP-пакетів відкриває можливості для реалізації ряду атак, зокрема підміни мережевих пристроїв (switch spoofing). Як показано на рисунку 2.2, LLDP-повідомлення містить службові поля, такі як версія протоколу, прапори, значення часу життя (TTL), а також набір структурованих

параметрів типу TLV (Type-Length-Value), які передають інформацію про мережевий пристрій.

Серед обов’язкових параметрів, що використовуються в OFDP, виділяють ChassisSubtype та PortSubtype. Перший з них містить ідентифікатор комутатора, який використовується для його однозначного визначення в мережі, тоді як другий описує конкретний порт або інтерфейс, з якого було надіслано LLDP-пакет.

Відсутність захисту цих повідомлень дозволяє зловмиснику формувати підроблені LLDP-пакети з довільними значеннями зазначених полів. У результаті контролер може отримувати некоректну інформацію про топологію мережі, що призводить до помилкових рішень щодо маршрутизації трафіку або навіть до перенаправлення потоків через зловмисні вузли.

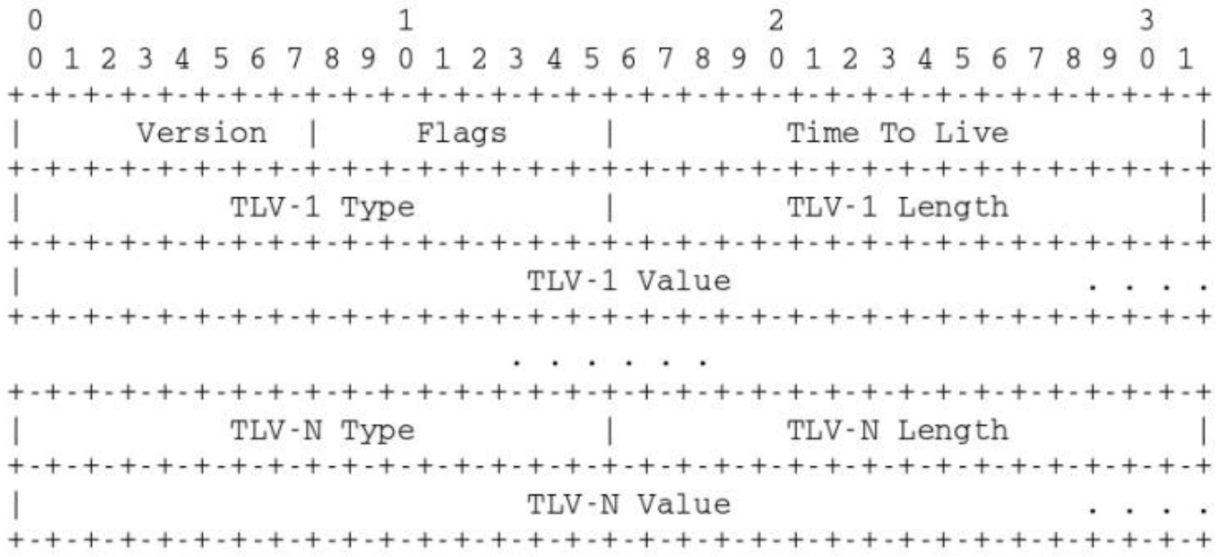

Рис. 2.3 Структура LLDP-повідомлення та склад його основних полів

Проблема полягає в тому, що більшість зазначених SDN-контролерів використовують у полі ChassisSubtype MAC-адресу локального інтерфейсу комутатора як його ідентифікатор. Такий підхід спрощує процес ідентифікації пристроїв, однак одночасно створює серйозні ризики безпеки, оскільки MAC-адреса може бути легко підроблена.

Оскільки LLDP-повідомлення передаються у відкритому вигляді, зловмисник має можливість перехоплювати їх та отримувати MAC-адреси

мережевих пристроїв. Використовуючи ці дані, атакуючий може імітувати легітимний комутатор, підставляючи відповідні значення у власні LLDP-пакети. У результаті контролер сприймає зловмисний пристрій як довірений елемент мережі.

Це дозволяє формувати викривлене уявлення про топологію мережі. Зокрема, контролер може отримувати інформацію про неіснуючі з'єднання або помилково змінювати маршрутизацію трафіку. Така підміна особливо небезпечна, оскільки вона відбувається на рівні керування і може впливати на роботу всієї мережі.

Як ілюструє приклад на рисунку 2.4, зловмисний комутатор може перехопити LLDP-пакети від легітимного пристрою, зчитати MAC-адресу його інтерфейсу та використати її для власної ідентифікації. Після повторного підключення до контролера він фактично «видає себе» за інший комутатор. Унаслідок цього контролер формує некоректну топологію, наприклад, фіксуючи зв'язки між вузлами, які в реальності відсутні.

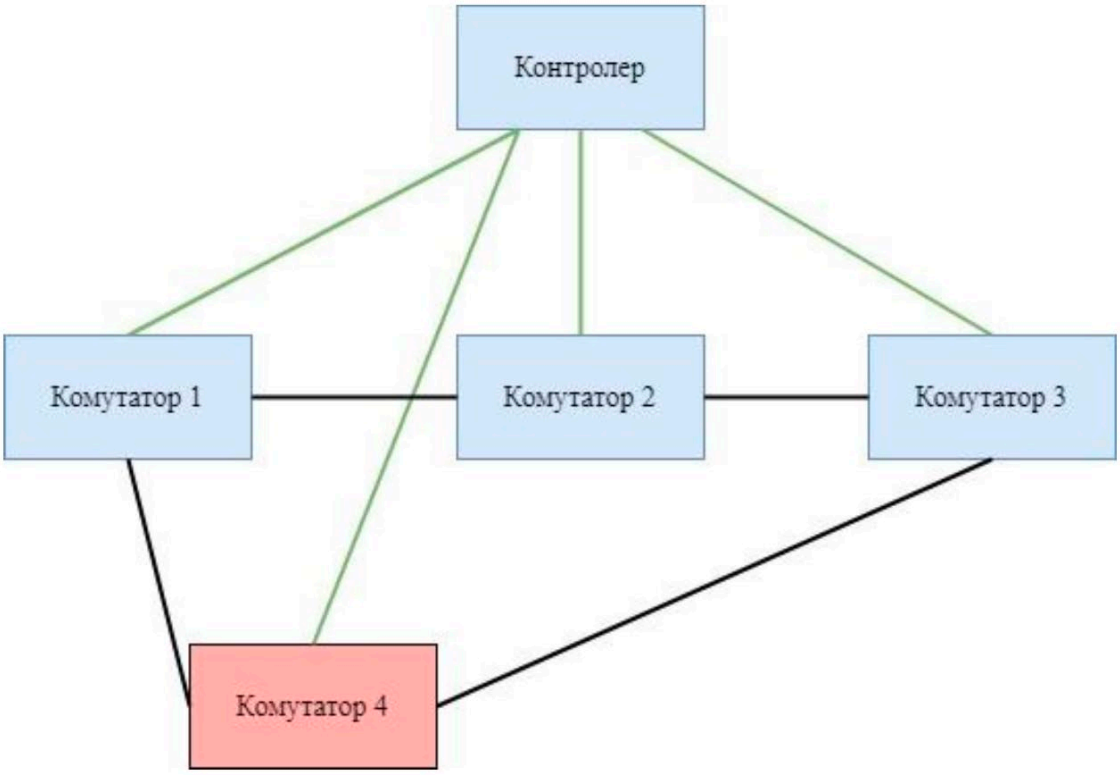

Рис. 2.4 Сценарій підміни мережевого комутатора в SDN-середовищі (Switch Spoofing)

Link fabrication (фальсифікація мережевих з'єднань).

Ще одним типом атаки на механізм виявлення топології в SDN є створення неіснуючих зв'язків між мережевими пристроями. Така атака базується на маніпуляції LLDP-повідомленнями, які використовуються контролером для формування уявлення про структуру мережі.

Як показано на рисунку 2.5, зловмисник, який отримав контроль над двома вузлами мережі (К1 і К2), підключеними відповідно до різних комутаторів, може організувати передачу LLDP-пакетів між цими вузлами. Зокрема, вузол К1 перехоплює LLDP-повідомлення від одного комутатора та передає їх до вузла К2, який, у свою чергу, пересилає ці пакети до іншого комутатора.

У результаті контролер отримує спотворену інформацію та помилково визначає наявність прямого з'єднання між комутаторами, які фактично не мають фізичного каналу. Це призводить до побудови некоректної топології мережі та прийняття неправильних рішень щодо маршрутизації трафіку.

Навіть у випадку, коли зловмисник має доступ лише до одного вузла, атака залишається можливою. Якщо відомий ідентифікатор іншого комутатора (DPID), атакуючий може формувати підроблені LLDP-пакети з відповідними параметрами та надсилати їх до мережі. У такому випадку контролер може зафіксувати односторонній (псевдо)зв'язок між комутаторами, що також призводить до викривлення топології.

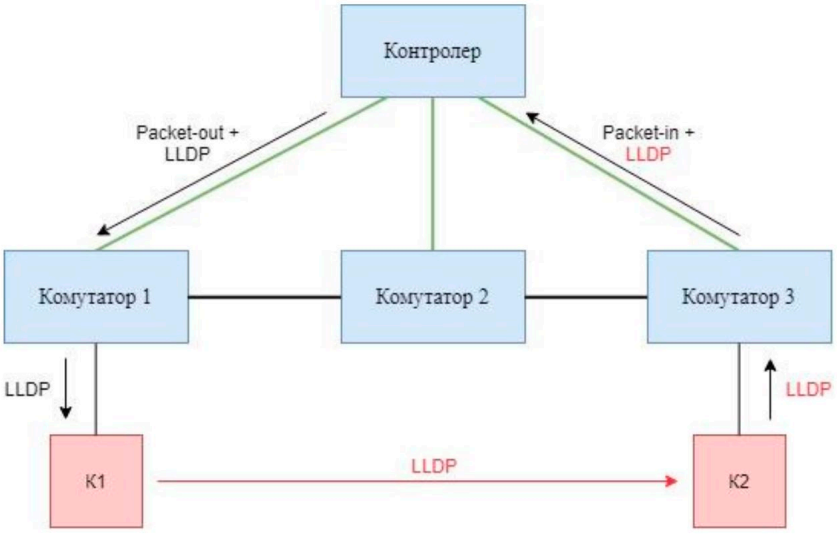

Рис. 2.5 Схема підробки зв'язків між комутаторами в SDN

LLDP-ін’єкція як різновид фабрикації каналів.

Ще одним способом створення фіктивних зв’язків у мережі є ін’єкція LLDP-повідомлень. У цьому випадку зловмисник спочатку здійснює пасивне прослуховування мережевого трафіку з метою отримання службових LLDP-пакетів, які розповсюджуються контролером для побудови топології.

Після аналізу отриманих даних атакуючий може генерувати власні LLDP-повідомлення з аналогічними параметрами та розсилати їх у мережі. У результаті контролер отримує некоректну інформацію та формує уявлення про неіснуючі з’єднання — як між комутаторами, так і між комутаторами та скомпрометованими вузлами. Це призводить до викривлення топології та потенційного перенаправлення трафіку через зловмисні елементи.

Controller Fingerprinting (ідентифікація контролера). Окремий тип атак пов’язаний із визначенням типу SDN-контролера на основі аналізу службових повідомлень. Як зазначається у [31], структура та вміст LLDP-пакетів можуть відрізнятися залежно від конкретної реалізації контролера.

Зловмисник, перехоплюючи LLDP-трафік, може порівнювати отримані дані з попередньо сформованою базою сигнатур різних контролерів. Таким чином стає можливим встановити, яке саме програмне забезпечення використовується для керування мережею.

Отримана інформація є цінною з точки зору підготовки подальших атак, оскільки дозволяє зосередитися на специфічних вразливостях конкретного контролера. Це значно підвищує ефективність цільових атак та спрощує процес їх реалізації.

Приклади перехоплених LLDP-повідомлень для різних контролерів, зокрема POX та Floodlight, наведені на рисунках 2.6 та 2.7, що демонструє відмінності у їх структурі та можливість ідентифікації.

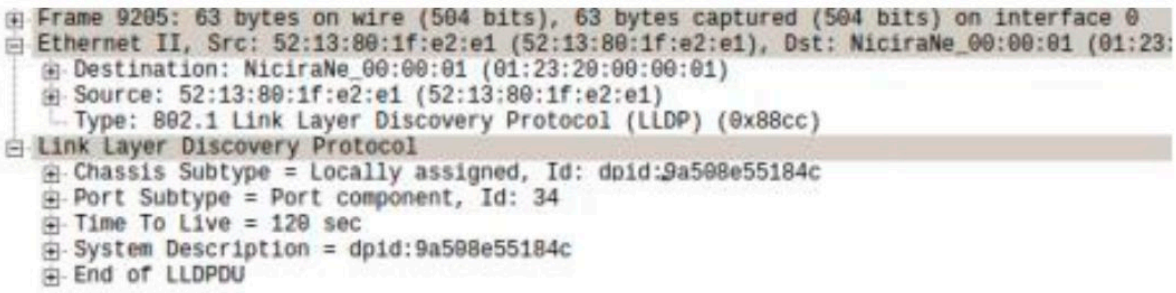

Рис. 2.6 Формування LLDP-повідомлення у середовищі POX-контролера

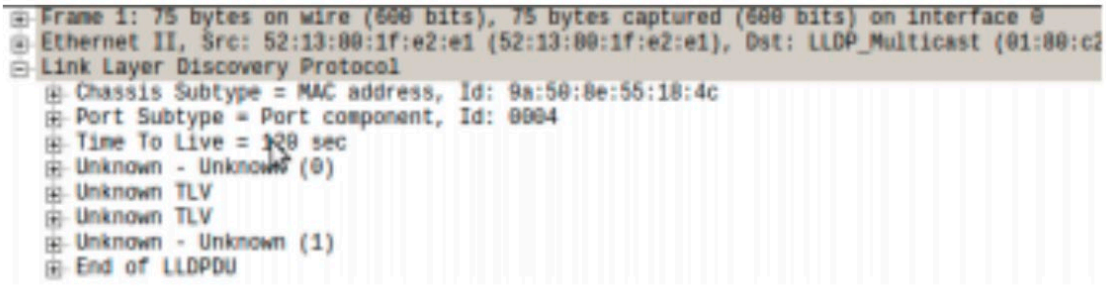

Рис. 2.7 Формування LLDP-повідомлення у середовищі Floodlight-контролера

LLDP Flood (перевантаження LLDP-повідомленнями).

Ще одним можливим сценарієм атаки є генерація великої кількості LLDP-пакетів скомпрометованим вузлом мережі з метою перевантаження контролера. У цьому випадку атакуючий створює надлишковий службовий трафік, який обробляється системою виявлення топології, що призводить до виснаження обчислювальних і мережевих ресурсів контролера.

Наслідком такої атаки може бути зниження продуктивності або часткова недоступність функцій керування мережею. При цьому стандартні механізми захисту, такі як блокування окремих портів комутаторів або фільтрація трафіку, не завжди є ефективними. Це особливо актуально для динамічних середовищ, зокрема хмарних інфраструктур, де топологія мережі може швидко змінюватися, а обмеження LLDP-трафіку може призвести до втрати легітимних службових повідомлень.

Модель загроз для SDN-мереж представлена на рисунку 2.8. Вона відображає основні вектори атак, спрямовані на ключові компоненти SDN-архітектури.

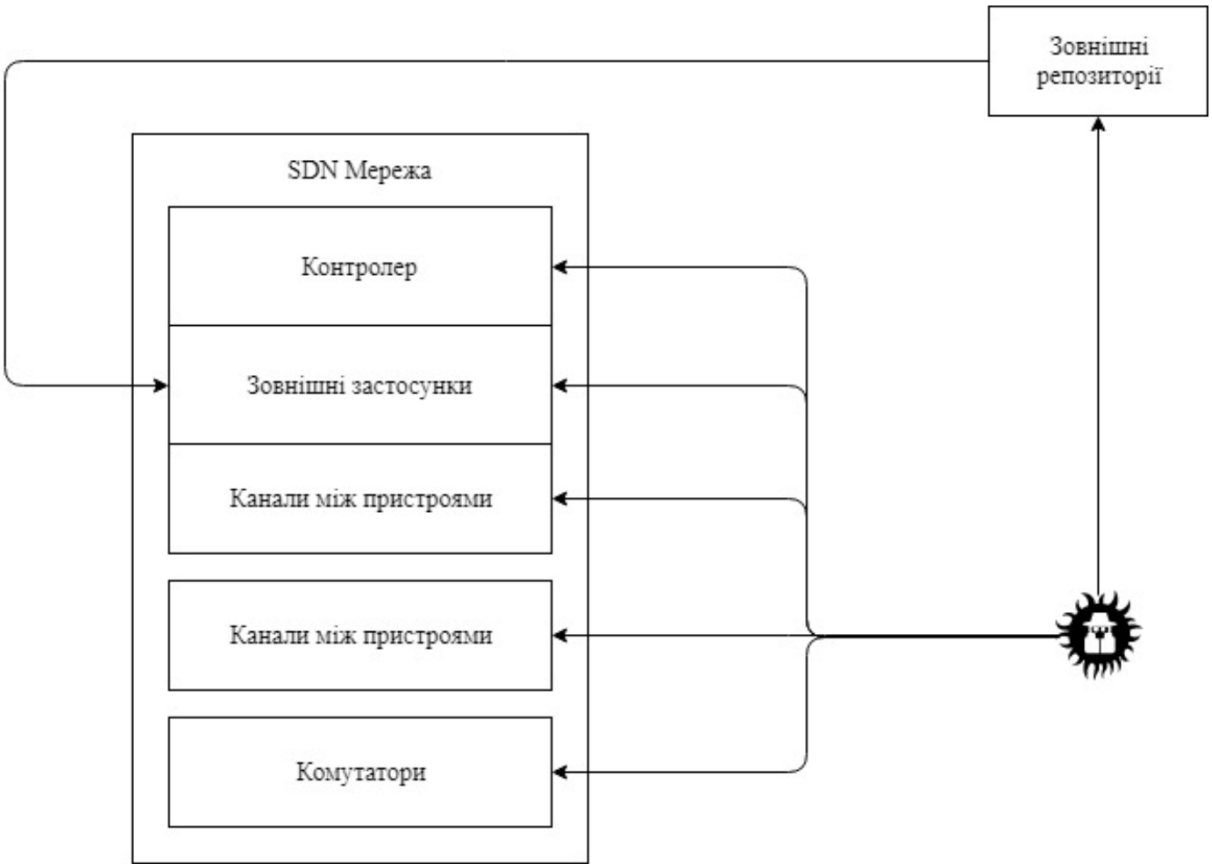

Рис. 2.8 Структурна модель атак на компоненти SDN

Деталізований аналіз наведеної моделі подано у таблиці 2.4, де систематизовано потенційні загрози, об’єкти їх реалізації, цілі атакуючої сторони та відповідні механізми протидії. Такий підхід дозволяє комплексно оцінити ризики та визначити ефективні засоби захисту мережі.

Таблиця 2.4

Класифікація загроз та заходів захисту в SDN-мережах

№	Об’єкт впливу	Мета атаки	Методи протидії
1	SDN-контролер	Порушення роботи мережі або отримання повного контролю над нею	Резервування контролера; оптимальне розміщення та балансування навантаження
2	Зовнішні застосунки	Впровадження шкідливого коду з метою компрометації мережі	Використання систем сканування вразливостей; аудит та перевірка залежностей
3	Протокол OpenFlow	Несанкціонований доступ до конфіденційних даних; DoS-атаки; ідентифікація контролера	Використання захищених каналів зв’язку (TLS); шифрування службового трафіку
4	Комутатори	Реалізація DoS-атак та перевантаження мережевих пристроїв	Проактивне встановлення правил; застосування комбінованих політик фільтрації
5	Канали між пристроями	Порушення стабільності роботи мережі та втручання у передачу даних	Використання захищених каналів зв’язку;

3 РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ SDN-РІШЕННЯ

3.1 Реалізація централізованого керування мережею з використанням OpenFlow

Для моделювання, дослідження та тестування програмно-керованої мережі було використано набір спеціалізованих інструментів, які дозволяють відтворити роботу SDN-інфраструктури в лабораторних умовах.

Зокрема, для побудови віртуального мережевого середовища застосовано емулятор Mininet [26], який базується на використанні віртуалізації процесів. Даний інструмент забезпечує можливість створення мереж із довільною топологією, включаючи комутатори, вузли та канали зв’язку, що дозволяє ефективно тестувати алгоритми керування та мережеві сценарії.

Для моніторингу стану мережі та аналізу трафіку було використано систему sFlow-rt [27], яка забезпечує збір і візуалізацію статистичних даних з мережевих пристроїв. Крім того, даний інструмент дозволяє формувати власні правила обробки трафіку та інтегруватися з контролером через API.

У ролі центрального елемента керування мережею обрано SDN-контролер Floodlight [28], реалізований мовою програмування Java. Контролер відповідає за обробку подій у мережі, формування правил маршрутизації та взаємодію з комутаторами через протокол OpenFlow.

Для генерації мережевого трафіку та моделювання атак було використано утиліту Hping3 [28], яка дозволяє створювати довільні пакети з різними параметрами. Це дає змогу перевіряти стійкість мережі до різних типів впливів, зокрема DoS-атак. У якості програмного комутатора застосовано Open vSwitch [29], який підтримує багаторівневу обробку пакетів і інтеграцію з протоколом OpenFlow. Даний компонент використовується для реалізації функцій комутації та взаємодії з контролером у віртуальному середовищі.

Моделювання DoS-атаки

З метою дослідження впливу DoS-атак на функціонування програмно-керованої мережі було здійснено моделювання відповідного сценарію в середовищі Mininet. У рамках експерименту сформовано SDN-мережу, що включає два програмні комутатори, до кожного з яких підключено по три вузли (хости).

Така конфігурація дозволяє відтворити базову мережеву топологію та проаналізувати поведінку системи в умовах перевантаження. Загальна структура змодельованої мережі наведена на рисунку 3.1.

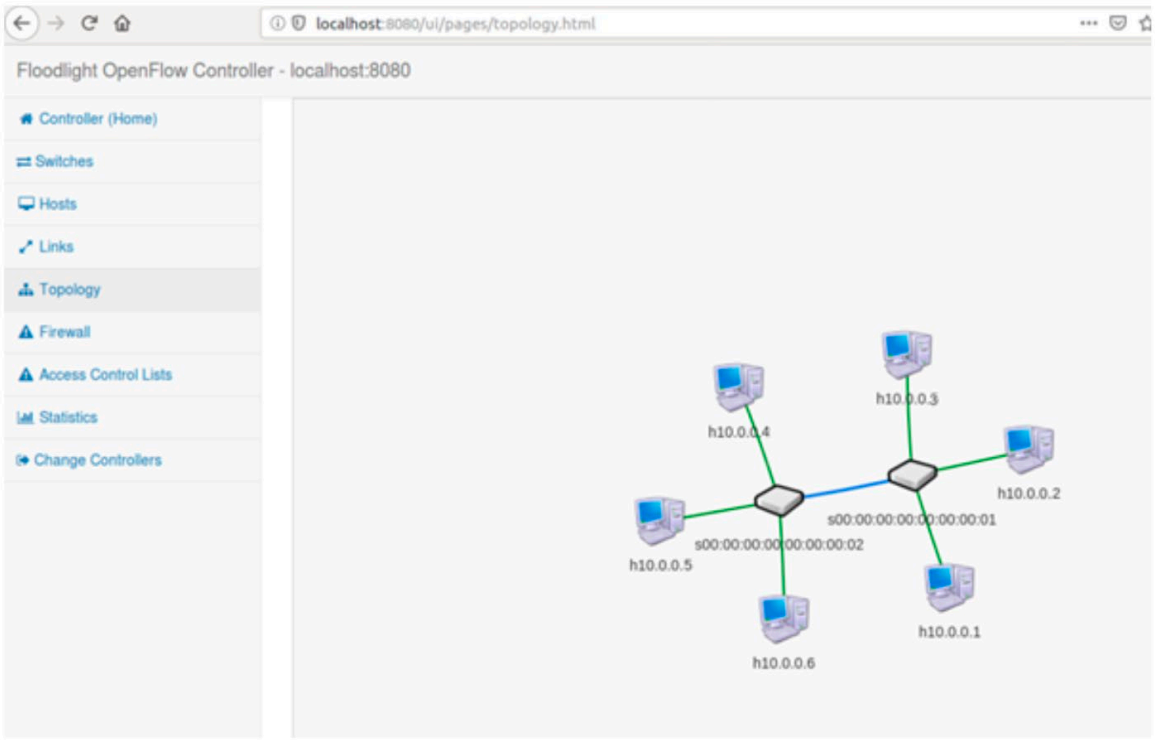

Рис. 3.1 Конфігурація тестової SDN-мережі

Для моделювання нормального мережевого навантаження на одному з вузлів мережі (хост 10.0.0.2) було розгорнуто простий HTTP-сервер із використанням вбудованих засобів мови програмування Python [27].

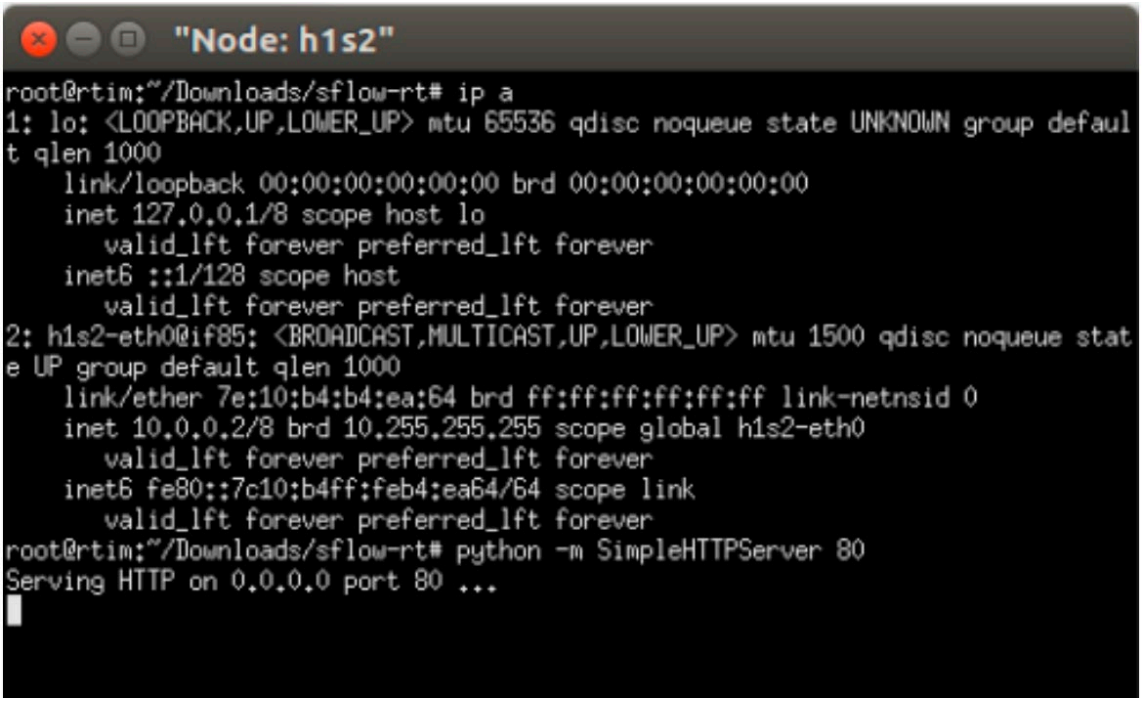

Рис. 3.2 Розгортання HTTP-сервера на хості 10.0.0.2

З метою формування регулярного мережевого навантаження на іншому вузлі мережі (хост 10.0.0.3) було запущено генератор трафіку з періодичністю відправки запитів кожні 3 секунди. Такий підхід дозволив імітувати стабільний клієнтський трафік та створити контрольоване середовище для подальшого аналізу впливу DoS-атаки на роботу мережі. Результати роботи генератора трафіку наведено на рисунку 3.3.

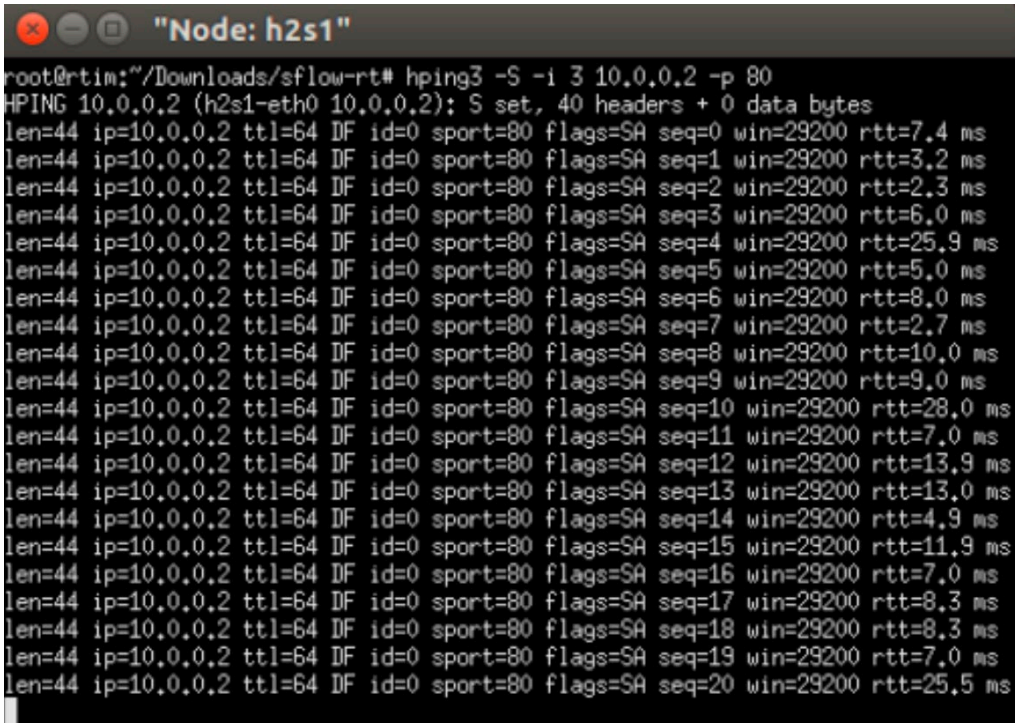

Рис. 3.3 Формування регулярного мережевого трафіку

Перед початком реалізації атаки за допомогою інструменту sFlow-rt було виконано моніторинг мережевої активності з метою фіксації її базового стану. Отримані графічні показники відображають рівень використання мережевих ресурсів за відсутності аномального навантаження. Результати моніторингу наведено на рисунку 3.4.

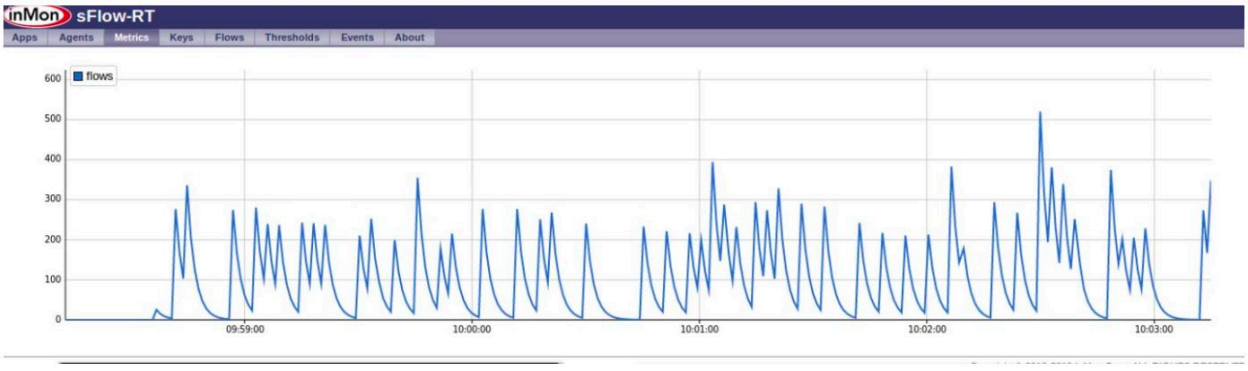

Рис. 3.4 Графіки мережевого навантаження у штатному режимі

Далі, відповідно до сценарію експерименту, на хості 10.0.0.1 було ініційовано генерацію зловмисного трафіку із використанням утиліти hping3 у режимі TCP SYN flood. У зазначеному режимі інструмент здійснює інтенсивну відправку TCP

SYN-пакетів до цільового вузла без очікування відповідей, що дозволяє максимально збільшити швидкість генерації трафіку та створити перевантаження мережевих ресурсів.

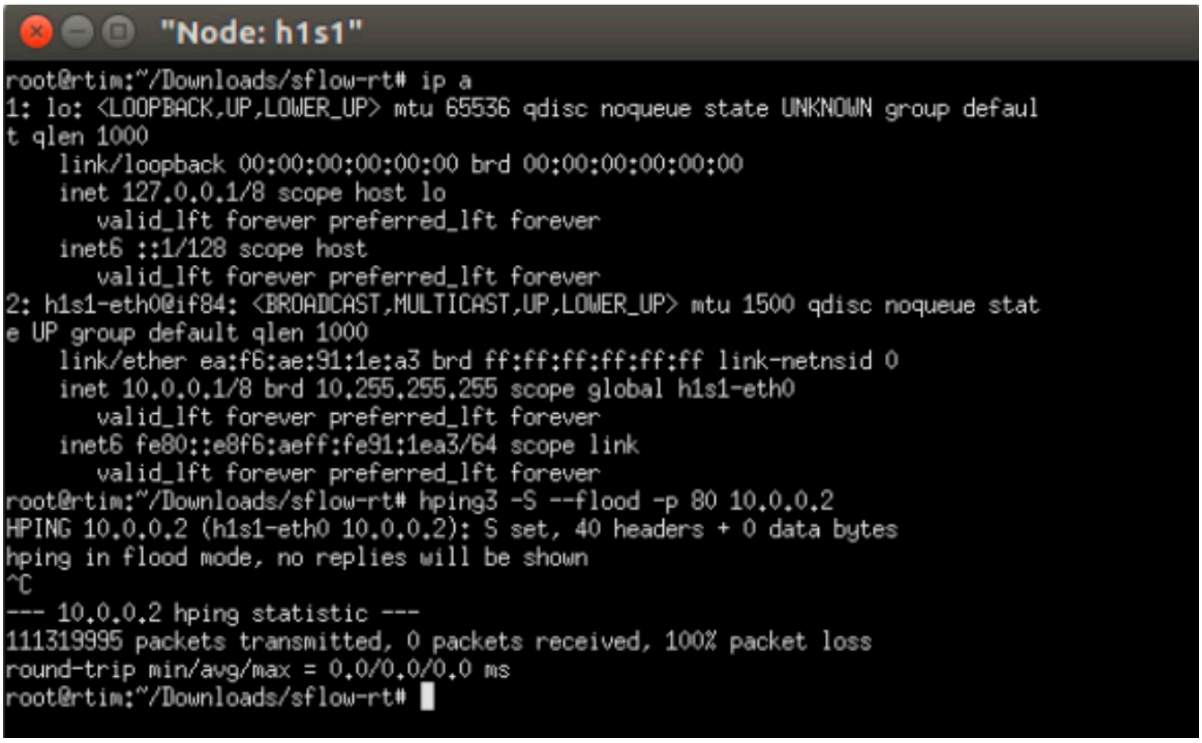

Рис. 3.5 Процес створення зловмисного трафіку у мережі

Після початку генерації зловмисного трафіку спостерігається різке зростання рівня використання мережевих ресурсів, що підтверджується відповідними графічними показниками (рис. 3.6).

Одночасно з цим легітимний трафік перестає оброблятися цільовим вузлом (рис. 3.7), оскільки обчислювальні та мережеві ресурси системи повністю зайняті обробкою великої кількості зловмисних запитів. У результаті відбувається фактична відмова в обслуговуванні, що є характерною ознакою DoS-атаки.

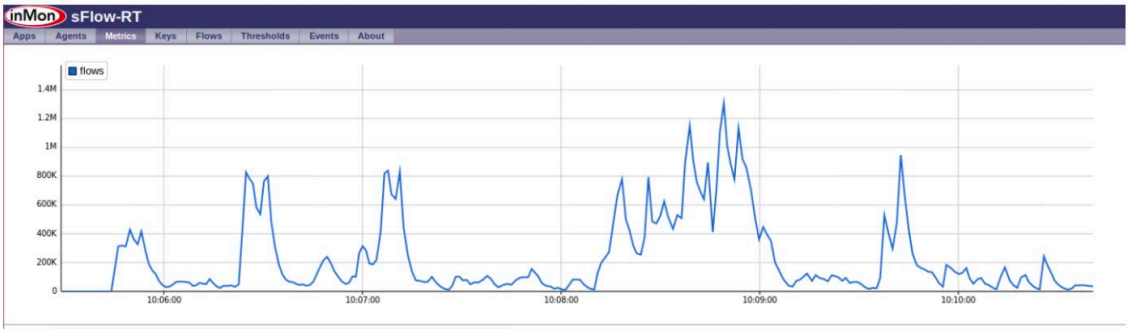

Рис. 3.6 Зростання мережевого навантаження під час DoS-атаки

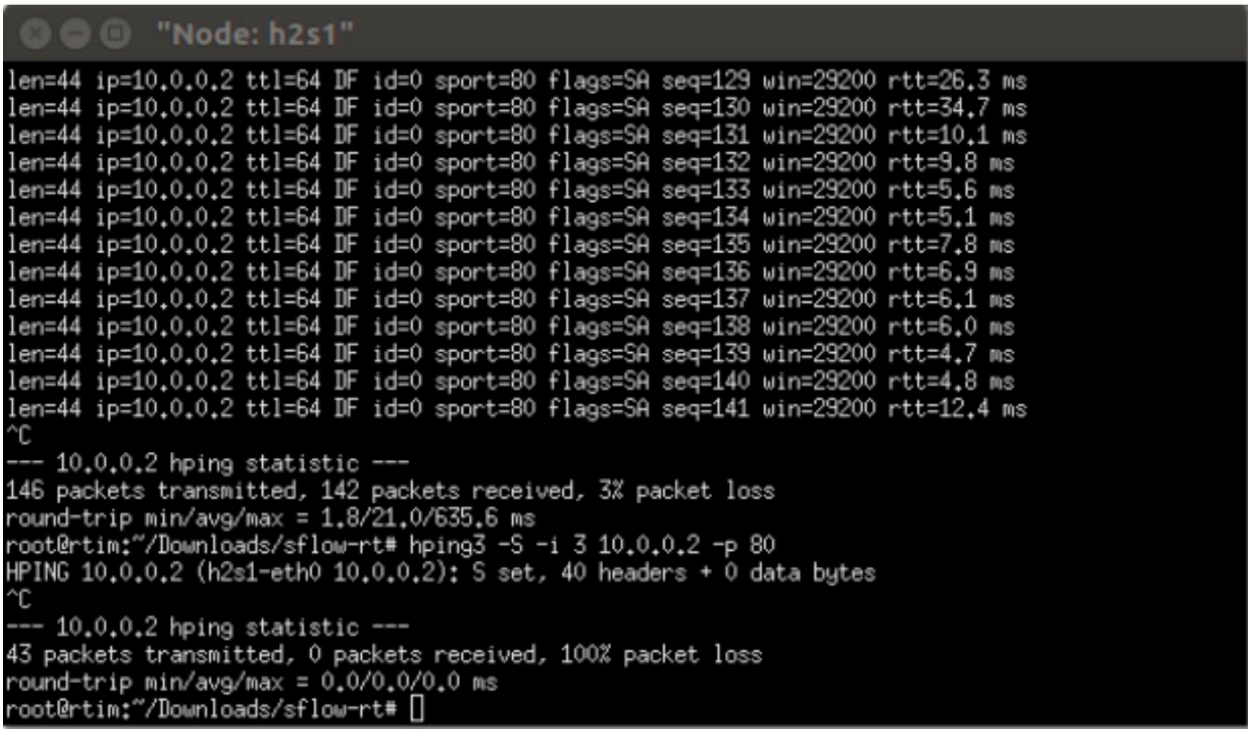

Рис. 3.7 Втрата доступності сервісу під впливом атаки

3.2 Тестування та забезпечення захисту від DoS атак

З метою автоматизованого виявлення та нейтралізації DoS-атак було розроблено програмний скрипт мовою JavaScript, який інтегрується із застосунком sFlow-rt.

Основним принципом роботи скрипта є встановлення граничного значення (порогу) кількості пакетів, що передаються протягом заданого інтервалу часу. У випадку перевищення визначеного порогу протягом певного періоду система ідентифікує IP-адреси джерела та призначення підозрілого трафіку.

Після цього за допомогою REST API SDN-контролера автоматично формується та застосовується правило фільтрації, яке блокує обмін трафіком між відповідними вузлами мережі. Такий підхід дозволяє оперативно обмежити поширення зловмисного навантаження та стабілізувати роботу мережі.

Після завершення заданого інтервалу часу (таймауту) правило блокування автоматично деактивується, що забезпечує відновлення нормальної взаємодії між вузлами без необхідності ручного втручання.

Ілюстрацію процесу функціонування розробленого механізму захисту наведено на рисунку 3.7.

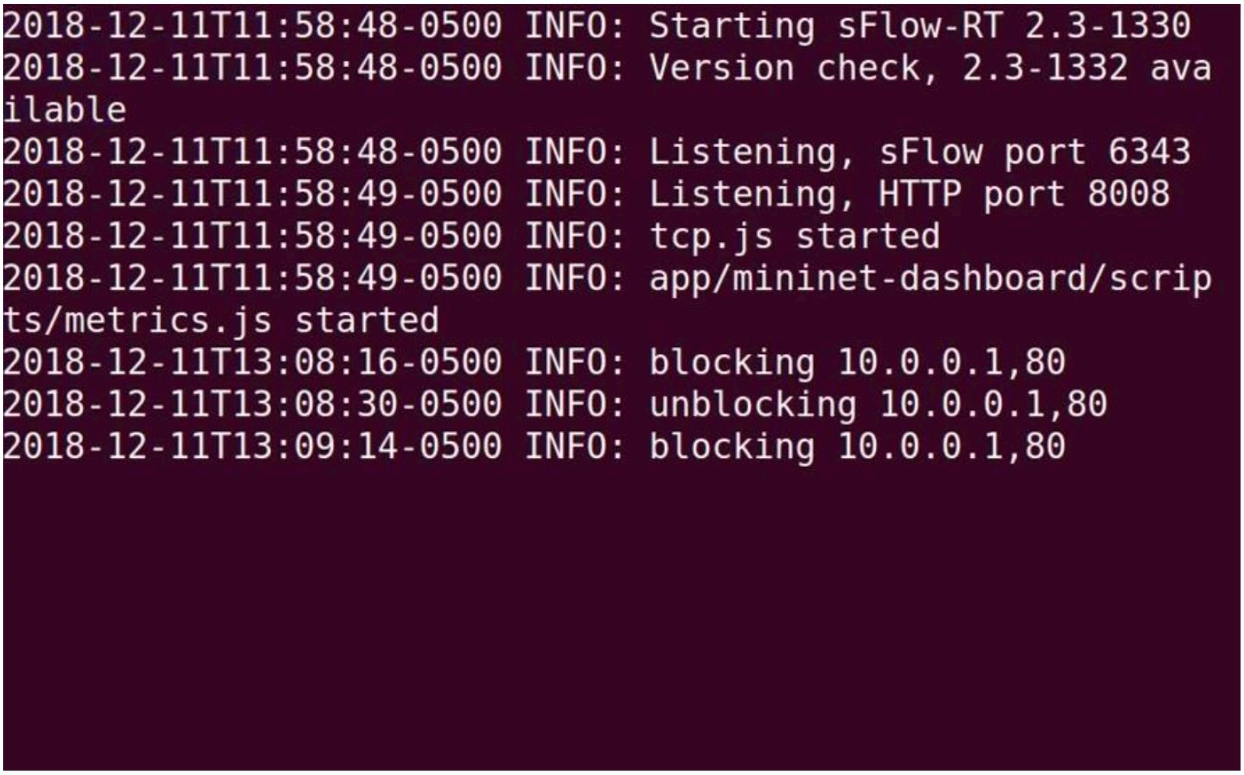

Рис. 3.7 Робота механізму автоматичного виявлення та блокування DoS-атаки

3.3 Відтворення атаки ARP Spoofing та оцінка ефективності впровадження SDN у мультивендорному середовищі

Отримавши контроль над одним із вузлів мережі, зловмисник може реалізувати атаку типу ARP spoofing, яка дозволяє підміняти мережеві відповідності між IP- та MAC-адресами.

Такий тип атаки може бути використаний для досягнення різних цілей, зокрема:

- реалізації атаки «людина посередині» (MITM);
- створення умов для відмови в обслуговуванні (DoS);

– спотворення мережевої топології та порушення коректної маршрутизації.

Відповідно до експериментального сценарію, зловмисник отримує доступ до хоста з IP-адресою 10.0.0.1. Після цього для проведення атаки використовується набір утиліт dsniff, призначений для аналізу та перехоплення мережевого трафіку.

Застосування зазначеного інструментарію дозволяє здійснювати підміну ARP-відповідей, перенаправляючи трафік через скомпрометований вузол. У результаті зловмисник отримує можливість перехоплювати, змінювати або блокувати передані дані між легітимними учасниками мережі.

Ілюстрація процесу реалізації атаки наведена на рисунку 3.8.

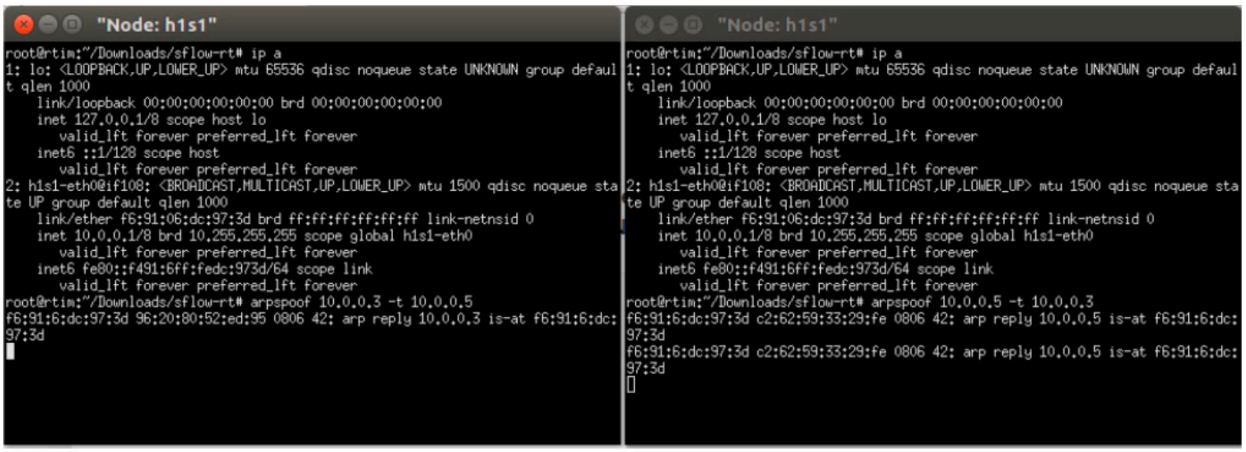

Рис. 3.8 Реалізація атаки ARP Spoofing із використанням dsniff

У результаті реалізації атаки мережевий трафік, що передається між вузлами 10.0.0.3 та 10.0.0.5, починає проходити через скомпрометований хост 10.0.0.1. Це свідчить про успішну реалізацію сценарію перехоплення даних у межах атаки типу «людина посередині» [28-30].

Крім того, спостерігається викривлення уявлення про топологію мережі з боку контролера, що проявляється у її нестабільності та постійній зміні. Така поведінка обумовлена підміною мережевих відповідностей і призводить до формування некоректної карти з’єднань.

Відповідні зміни топології мережі наведено на рисунку 3.9.

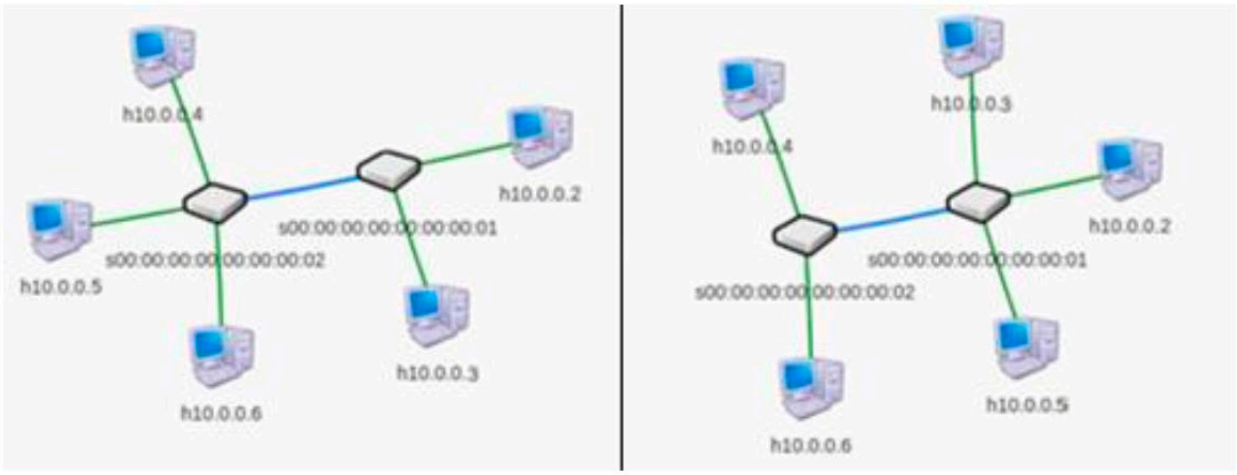

Рис. 3.9 Викривлення топології мережі внаслідок ARP spoofing-атаки

Проведене моделювання атак дозволяє оцінити ефективність використання SDN-підходу в умовах мультивендорного середовища, де мережеве обладнання може мати різні реалізації та рівень підтримки стандартів.

Однією з ключових переваг SDN є централізоване управління мережею, що забезпечує можливість оперативного виявлення аномальної активності та швидкого реагування на інциденти безпеки. Завдяки наявності єдиного контролера стає можливим отримання повної картини стану мережі, включаючи інформацію про трафік, топологію та поведінку окремих пристроїв.

Крім того, використання відкритих протоколів, таких як OpenFlow, дозволяє уніфікувати процес керування мережевими пристроями різних виробників. Це значно спрощує інтеграцію обладнання у єдину інфраструктуру та зменшує залежність від конкретного постачальника.

Разом з тим, результати експериментів показали, що централізація управління створює і додаткові ризики. Зокрема, компрометація контролера або каналів зв’язку між контролером і комутаторами може призвести до повного порушення роботи мережі. Також виявлено, що відсутність належного захисту службових протоколів (наприклад, LLDP або OpenFlow без TLS) відкриває можливості для реалізації атак, пов’язаних із підміною топології або перехопленням трафіку.

Таким чином, впровадження SDN у мультивендорному середовищі є ефективним інструментом підвищення гнучкості та керованості мережі, однак потребує обов’язкового застосування додаткових механізмів захисту. До таких механізмів належать використання захищених каналів зв’язку, автентифікація пристроїв, контроль доступу до контролера та впровадження систем виявлення вторгнень.

Узагальнюючи результати дослідження, можна зробити висновок, що SDN-технології забезпечують значні переваги у побудові сучасних мереж, проте їх ефективність безпосередньо залежить від рівня реалізації заходів інформаційної безпеки.

ВИСНОВКИ

Досліджено сучасні підходи до побудови та управління комп'ютерними мережами, зокрема проаналізовано особливості традиційних мережевих архітектур та визначено їх основні недоліки. Встановлено, що розподілена модель керування ускладнює адміністрування, знижує гнучкість мережі та створює труднощі в умовах масштабування і мультивендорного середовища. Особливу увагу приділено проблемам інтеграції обладнання різних виробників, що негативно впливає на ефективність функціонування мережевої інфраструктури.

Розглянуто концепцію Software-Defined Networking як перспективний підхід до вирішення зазначених проблем. Визначено, що ключовими перевагами SDN є централізація керування, програмованість та можливість автоматизації мережевих процесів. Проаналізовано роль протоколу OpenFlow як основного механізму взаємодії між контролером і мережевими пристроями, що забезпечує гнучке керування потоками трафіку та підвищує ефективність використання ресурсів мережі.

Розроблено підхід до автоматичного виявлення та блокування DoS-атак із використанням можливостей контролера та аналітичних інструментів. Запропонований механізм дозволяє оперативно ідентифікувати джерело зловмисного трафіку та застосовувати правила обмеження доступу, що забезпечує підвищення рівня захищеності мережі. Також було досліджено можливість реалізації атак типу ARP spoofing та оцінено їх вплив на топологію мережі та процес передачі даних.

У результаті виконаної роботи підтверджено, що впровадження SDN-технологій дозволяє значно підвищити ефективність управління мережею, спростити її адміністрування та забезпечити гнучкість у налаштуванні політик. Запропоновані підходи до виявлення та протидії атакам можуть бути використані у реальних мережевих інфраструктурах, зокрема в умовах мультивендорного середовища, що підкреслює практичну цінність отриманих результатів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Open Networking Foundation. Software-Defined Networking (SDN) Definition [Електронний ресурс]. – Режим доступу: https://opennetworking.org/sdn-definition/ (дата звернення: 04.04.2026).

2. Open Networking Foundation. OpenFlow Switch Specification Version 1.5.1 [Електронний ресурс]. – Режим доступу: https://opennetworking.org/sdn-resources/openflow-switch-specification/ (дата звернення: 04.04.2026).

3. Kreutz D., Ramos F.M.V., Verissimo P.E. et al. Software-Defined Networking: A Comprehensive Survey // Proceedings of the IEEE. – 2015. – Vol. 103, No. 1. – P. 14–76.

4. McKeown N., Anderson T., Balakrishnan H. et al. OpenFlow: Enabling Innovation in Campus Networks // ACM SIGCOMM Computer Communication Review. – 2008. – Vol. 38, No. 2. – P. 69–74.

5. Nadeau T., Gray K. SDN: Software Defined Networks. – Sebastopol: O'Reilly Media, 2013. – 384 p.

6. Feamster N., Rexford J., Zegura E. The Road to SDN: An Intellectual History of Programmable Networks // ACM SIGCOMM Computer Communication Review. – 2014. – Vol. 44, No. 2. – P. 87–98.

7. Benton K., Camp L.J., Small C. OpenFlow Vulnerability Assessment // Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. – 2013. – P. 151–152.

8. Scott-Hayward S., Natarajan S., Sezer S. A Survey of Security in Software Defined Networks // IEEE Communications Surveys & Tutorials. – 2016. – Vol. 18, No. 1. – P. 623–654.

9. Mininet Project. Mininet: An Instant Virtual Network on your Laptop [Електронний ресурс]. – Режим доступу: http://mininet.org/ (дата звернення: 04.04.2026).

10. Floodlight Project. Floodlight OpenFlow Controller [Електронний ресурс]. – Режим доступу: https://floodlight.atlassian.net/wiki/ (дата звернення: 04.04.2026).

11. McKeown N., Anderson T., Balakrishnan H. et al. OpenFlow: Enabling Innovation in Campus Networks [Електронний ресурс]. – Режим доступу: https://dl.acm.org/doi/10.1145/1355734.1355746 (дата звернення: 02.04.2026).

12. Kreutz D., Ramos F.M.V., Verissimo P.E. et al. Software-Defined Networking: A Comprehensive Survey [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/6994333 (дата звернення: 02.04.2026).

13. Open Networking Foundation. OpenFlow Switch Specification Version 1.5.1 [Електронний ресурс]. – Режим доступу: https://opennetworking.org (дата звернення: 02.04.2026).

14. Scott-Hayward S., O'Callaghan G., Sezer S. SDN Security: A Survey [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/7084321 (дата звернення: 02.04.2026).

15. Benton K., Camp L.J., Small C. OpenFlow Vulnerability Assessment [Електронний ресурс]. – Режим доступу: https://dl.acm.org/doi/10.1145/2491185.2491200 (дата звернення: 02.04.2026).

16. Nunes B.A.A., Mendonca M., Nguyen X.N. et al. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/6686025 (дата звернення: 02.04.2026).

17. Porras P., Shin S., Yegneswaran V. et al. A Security Enforcement Kernel for OpenFlow Networks [Електронний ресурс]. – Режим доступу: https://dl.acm.org/doi/10.1145/2382196.2382204 (дата звернення: 02.04.2026).

18. Shin S., Gu G. Attacking Software-Defined Networks: A First Feasibility Study [Електронний ресурс]. – Режим доступу: https://dl.acm.org/doi/10.1145/2636951.2636958 (дата звернення: 02.04.2026).

19. Al-Shaer E., Al-Haj S. FlowChecker: Configuration Analysis and Verification of Federated OpenFlow Infrastructures [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/6133290 (дата звернення: 02.04.2026).

20. Mininet Project. Mininet: An Instant Virtual Network on your Laptop [Електронний ресурс]. – Режим доступу: http://mininet.org (дата звернення:

02.04.2026).

21. InMon Corp. sFlow-RT Analytics Engine [Електронний ресурс]. – Режим доступу: https://sflow-rt.com (дата звернення: 02.04.2026).

22. Floodlight Project. Floodlight Open SDN Controller [Електронний ресурс]. – Режим доступу: https://floodlight.atlassian.net (дата звернення: 02.04.2026).

23. Open vSwitch Project. Open vSwitch Documentation [Електронний ресурс]. – Режим доступу: https://www.openvswitch.org (дата звернення: 02.04.2026).

24. Salvatore Sanfilippo. hping3: Network Tool for Packet Generation and Analysis [Електронний ресурс]. – Режим доступу: http://www.hping.org (дата звернення: 02.04.2026).

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ