ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка рекомендацій для підвищення ефективності роботи програмно-визначної системи SDN»

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 Комп'ютерні науки
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні науки
(назва)

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

_____ Тыльненко Ростислав
(підпис) *(Ім'я, ПРІЗВИЩЕ здобувача)*

Виконав:
здобувач вищої освіти Тільненко Ростислав
група КНД-42

Керівник: Юлія Березовська
науковий ступінь, вчене звання к.т.н., доцент

Рецензент:
науковий ступінь, вчене звання _____
(Ім'я, ПРІЗВИЩЕ)

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерних наук

Ступінь вищої освіти Бакалавр

Спеціальність Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедрою Комп'ютерних наук

_____ Віктор ВИШНІВСЬКИЙ
«_____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Тільненку Ростиславу Олександровичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Розробка рекомендацій для підвищення ефективності роботи програмно-визначної мережі SDN
керівник кваліфікаційної роботи Юлія Березовська, к.т.н., доцент.
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» 02.2024р. №36
2. Строк подання кваліфікаційної роботи «31» травня 2024р.
3. Вихідні дані до кваліфікаційної роботи: технологія програмно-визначної мережі SDN, протоколи Open Flow та STP, Науково технічна література з питань, пов'язаних з дослідженням програмно –визначних мереж.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   Програмно-визначні мережі для майбутніх мереж і послуг: основні технічні проблеми та наслідки для бізнесу
   Дослідження мережі SDN для покращення управління програмновизначеними мережами, а також балансування мережевого трафіку
   Архітектура на основі SDN для забезпечення якості обслуговування високопродуктивних розділених додатків
5. Перелік графічного матеріалу: *презентація*
   1. Мета та актуальність роботи
   2. Об'єкт, предмет та методи дослідження

3. Програмно визначні мережі SDN
4. Архітектура SDN
5. Переваги використання SDN
6. Сучасні проблеми застосування SDN мереж
7. Протоколи OpenFlow
8. Технічні проблеми розробки та розгортання SDN та способи вирішення
9. Архітектурна модель SDN
10. Протокол Spanning Tree
11. Оцінка та оптимізація існуючих модулів SDN
12. Qos та йогоможливості
13. Загальні фази надання QoS
14. Схема архітектури надання QoS на основі архітектури SDN
15. Тестування та оцінка запропонованої архітектури
16. Висновок

6. Дата видачі завдання «23» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	01.03-14.03.24	вик.
2	Пошук основних проблем програмно визначеної мережі SDN	15.03-28.03.24	вик.
3	Застосування протоколу OpenFlow під час побудови мережі SDN	29.03-11.04.24	вик.
4	Дослідження ефективності застосування Spanning Tree(STP) для уникнення вузлів в мережі	12.04-20.04.24	вик.
5	Дослідження ефективності застосування QoS з метою зменшення витрат та прискорення передачі даних	21.04-02.05.24	вик.
6	Вступ, висновки, реферат	03.05-07.05.24	вик.

Здобувач вищої освіти _____ Ростислав Тільненко
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Юлія Березовська
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Реферат

Текстова частина бакалаврської роботи 73 с., 37 рис., 1 табл., 31 джерело. SDN мережі, протокол OpenFlow, протокол Spanning Tree, Механізм QoS.

Об`єкт дослідження – програмно-визначені мережі SDN .

Предмет дослідження – способи, методи та технології підвищення ефективності програмно-визначеної мережі SDN.

Мета роботи – пошук методів підвищення ефективності роботи SDN аналіз результативності їхнього впливу на якість обслуговування.

Методи дослідження – аналіз, систематизація, експеримент, узагальнення.

Оскільки інформаційні технології постійно розвиваються і з’являються новітні технології (такі як хмарні обчислення та Big Data), вимоги до комп'ютерних мереж зростають, а їх застосування ускладнюється. Вони потребують більшої швидкості передачі даних та вдосконалення інструментів, що використовуються для мережевого управління і моніторингу. Це призводить до появи нових функціональних і технологічних мереж, з складнішою інфраструктурою. Старі методи моніторингу і управління не відповідають новим вимогам

SDN (software-defined networking, програмно-визначені мережі), мають переваги у гнучкості та можливостях масштабування, у порівнянні зі звичайними мережами. Проте ці мережі ще розвиваються і не є повністю досконалими. У зв’язку з цим в роботі були проведені дослідження з метою пошуку методів підвищення їх ефективності, а також проведений аналіз результативності їх впливу на якість обслуговуваня.

ЗМІСТ

ВСТУП

Тема програмно-визначених мереж (SDN) є надзвичайно **актуальною** в сучасному світі, де інформаційні технології проникли у всі сфери діяльності людини. Здатність мереж швидко адаптуватися до змін у навантаженні, забезпечувати високу швидкість маршрутизації, легкість конфігурації вузлів та ефективну обробку великих обсягів даних є критично важливою для підтримки конкурентоспроможності бізнесів та якості обслуговування користувачів. SDN мережі можуть значно підвищити ефективність роботи телекомунікаційних систем та забезпечити швидке впровадження інноваційних рішень.

Технологія SDN активно досліджується протягом останніх років, зокрема в контексті підвищення її ефективності та зменшення експлуатаційних витрат. Існує багато наукових праць та досліджень, які присвячені оптимізації SDN, їх безпеці, інтеграції з хмарними сервісами та віртуалізації мережевих функцій (NFV). Проте, питання підвищення ефективності роботи SDN в умовах зростаючих вимог до мережевої інфраструктури все ще потребує додаткового вивчення та практичних рішень.

Основними джерелами інформації для даного дослідження є наукові статті, монографії, патенти, технічна документація та звіти провідних компаній в галузі інформаційних технологій і телекомунікацій. Також використовуються матеріали конференцій, семінарів та симпозіумів, присвячених програмно-визначеним мережам, що дозволяє отримати актуальну та різнопланову інформацію з даної тематики.

Об'єктом дослідження є програмно-визначені мережі (SDN) та їх архітектура.

Предметом дослідження є методи та технології підвищення ефективності роботи програмно-визначених мереж (SDN).

Метою роботи є аналіз існуючих проблем та розробка методів підвищення ефективності роботи програмно-визначених мереж (SDN). Для досягнення поставленої мети необхідно вирішити такі **завдання**:

1. Провести огляд сучасного стану та тенденцій розвитку програмно-визначених мереж.
2. Дослідити основні проблеми та обмеження існуючих SDN рішень.
3. Розробити методи підвищення ефективності роботи SDN.
4. Оцінити ефективність запропонованих методів на основі експериментальних досліджень.

Для досягнення мети та вирішення завдань роботи використовуються такі **методи** дослідження:

1. Аналіз літературних джерел та огляд сучасних технологій в галузі SDN.
2. Математичне моделювання та комп'ютерна симуляція для оцінки ефективності запропонованих методів.
3. Експериментальні дослідження на базі реальних SDN мереж.

Наукова новизна роботи полягає у розробці нових методів та підходів для підвищення ефективності роботи програмно-визначених мереж. Це включає оптимізацію процесів маршрутизації, вдосконалення алгоритмів конфігурації мережевих елементів та зниження експлуатаційних витрат.

Практична значущість результатів дослідження полягає у можливості їх застосування для покращення роботи існуючих телекомунікаційних систем. Запропоновані методи можуть бути впроваджені у практику мережевих операторів та постачальників послуг, що дозволить підвищити якість обслуговування користувачів, знизити витрати на утримання мереж та прискорити впровадження нових сервісів.

1 МАЙБУТНІ МЕРЕЖІ І ПОСЛУГИ: ОСНОВНІ ТЕХНІЧНІ ПРОБЛЕМИ ТА ВПЛИВ НА БІЗНЕС

1.1 Програмно-визначені мережі

Програмно-визначені мережі, відомі як SDN (Software-Defined Networking), є інноваційним підходом до мережевих комунікацій, який передбачає централізоване управління мережею з використанням програмних засобів. Основна концепція SDN полягає у відокремленні контрольного рівня (control plane) від рівня передачі даних (data plane), що дозволяє досягти високого рівня гнучкості та оптимізації мережевих ресурсів.

Основні компоненти SDN включають контролер, мережеві пристрої та інтерфейси для взаємодії між ними:

1. Контролер SDN – центральний елемент, який здійснює управління мережею, збирає інформацію про стан мережі, приймає рішення про маршрутизацію і передає ці рішення до мережевих пристроїв.
2. Південно-західний інтерфейс (Southbound Interface) – інтерфейс між контролером та мережевими пристроями (комутаторами, маршрутизаторами), що забезпечує взаємодію з фізичною інфраструктурою.
3. Північно-східний інтерфейс (Northbound Interface) – інтерфейс між контролером та додатками, що працюють на рівні мережевого управління, надаючи більш високий рівень функціональності для автоматизації та оптимізації мережевих процесів.

Ключові переваги SDN:

1. Гнучкість і програмована конфігурація - забезпечення швидкої адаптації мережі до змін у навантаженні та вимогах користувачів.
2. Централізоване управління - спрощення процесів управління мережею через централізацію контрольних функцій.

3. Підвищення продуктивності- оптимізація маршрутів передачі даних, зниження затримок та збільшення пропускної здатності мережі.
4. Зменшення витрат - зниження операційних та капітальних витрат завдяки автоматизації та ефективному використанню ресурсів.

Основні технічні проблеми SDN:

1. Безпека - відокремлення контрольного та передавального рівнів створює нові можливості для кібератак, тому забезпечення безпеки є важливим аспектом.
2. Масштабованість - питання ефективного масштабування мережі, особливо у великих інфраструктурах.
3. Сумісність - інтеграція SDN з існуючими мережевими технологіями та протоколами може бути викликом.
4. Відмовостійкість - забезпечення безперебійної роботи мережі при відмові контролера або інших ключових компонентів.

Наслідки для бізнесу:

1. Швидке впровадження нових послуг- зниження часу на запуск нових сервісів дозволяє компаніям швидше реагувати на ринкові потреби.
2. Оптимізація витрат - зменшення витрат на обслуговування мережі завдяки автоматизації та зниженню необхідності в ручному втручанні.
3. Підвищення ефективності - покращення продуктивності мережі та якості обслуговування клієнтів.
4. Гнучкість бізнес-моделей - можливість швидко адаптуватися до нових ринкових умов та вимог клієнтів (таб. 1.1)

Таблиця 1.1 - Порівняння традиційних мереж та SDN

Параметр	Традиційні мережі	SDN
Управління	Розподілене	Централізоване
Налаштування	Ручне	Автоматизоване
Гнучкість	Обмежена	Висока
Продуктивність	Залежить від апаратного забезпечення	Оптимізована за допомогою програмних рішень
Масштабованість	Вимагає значних ресурсів	Легко масштабована
Витрати	Високі операційні витрати (OPEX)	Знижені операційні витрати (OPEX)
Безпека	Залежить від апаратних засобів	Забезпечується програмними рішеннями
Інновації	Тривалий цикл впровадження нових рішень	Швидке впровадження нових функцій та сервісів

Для досягнення мети та вирішення завдань роботи використовуються такі методи дослідження:

1. Аналіз літературних джерел. Огляд сучасних технологій в галузі SDN, дослідження актуальних проблем та тенденцій.
2. Математичне моделювання та комп'ютерна симуляція. Оцінка ефективності запропонованих методів на основі симуляційних моделей.
3. Експериментальні дослідження. Проведення експериментів на реальних SDN мережах для перевірки теоретичних результатів та оцінки їх практичної значущості (рис. 1.1).

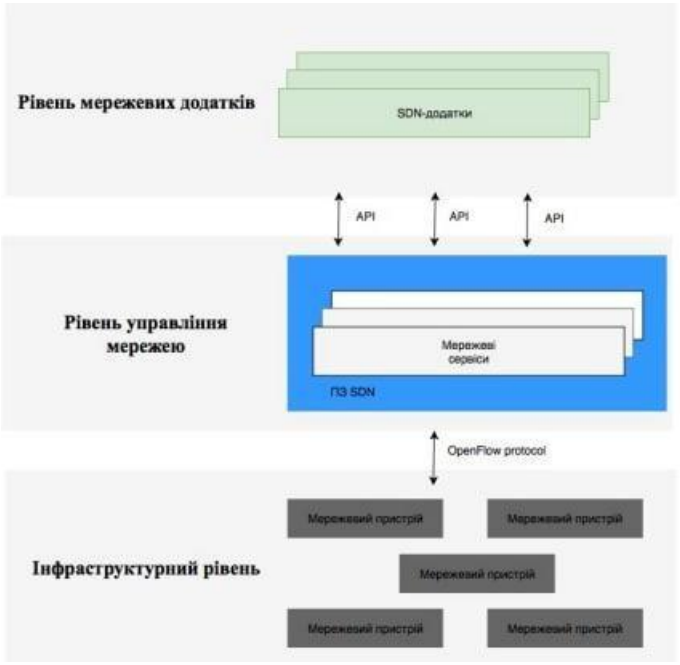


Рисунок 1.1 – Засади реалізації ідеї SDN

1.2 Структура SDN

Програмно-визначені мережі (SDN) є революційною концепцією у сфері телекомунікацій, яка дозволяє значно підвищити ефективність, гнучкість та керованість мережевих інфраструктур. В умовах зростання обсягів даних та збільшення складності мереж, традиційні мережеві архітектури стають все менш ефективними. Основні проблеми включають надмірну складність налаштування, обмежену гнучкість та високу вартість експлуатації.

Традиційна архітектура з трьома рівнями (доступ, агрегація, ядро) вимагає великої кількості операцій для обробки трафіку на кожному вузлі. Це зайва складність для високоефективної інфраструктури, необхідної для взаємодії між багатьма серверами та великими центрами обробки даних (ЦОД).

Архітектура програмно-конфігуровної мережі (ПКМ, SDN) базується на чотирьох ключових принципах, які дозволяють подолати ці обмеження та забезпечити більш ефективне управління мережевими ресурсами.

Традиційна архітектура з трьома рівнями (доступ, агрегація, ядро) та потреба у виконанні багатьох операцій для обробки трафіку в кожному вузлі стають зайвими для інфраструктури з високою ефективністю, необхідною для організації взаємодії між великою кількістю серверів та великими центрами обробки даних (ЦОД).

Архітектура програмно-конфігурованих мереж (ПКМ, SDN) ґрунтується на чотирьох принципах. Основні функції керування та передачі даних розділені. Мережевий пристрій виконує лише функції передачі даних без управлінських функцій. Рішення щодо передачі даних приймаються на основі потоків, а не за адресою призначення. Потік визначається як поєднання полів заголовка пакету, які діють як фільтр для співставлення мережевого трафіку з набором інструкцій по його обробці. У контексті SDN абстракція потоку забезпечує однакову обробку і передачу кожного пакета в послідовності від джерела до призначення.

Контролер SDN, або мережева операційна система, є зовнішнім об'єктом для передавальних пристроїв. Це програмна служба, що надає ресурси та інтерфейс для розробки програмних розширень, які використовують переваги централізованого управління мережею та обмежені лише набором критеріїв і дій передавальних пристроїв. Програмне забезпечення, що запускається на мережевому контролері, взаємодіє з передавальними пристроями та забезпечує можливість програмування функцій мережевої інфраструктури. Це є ключовою перевагою архітектури SDN.

Відповідно до визначення архітектури SDN, мережева інфраструктура поділяється на такі рівні:

- рівень передачі даних;
- рівень управління, який є результатом спрощення передавальних пристроїв і перенесення функцій управління на централізовану програмну платформу.

- рівень додатків є абстракцією, тісно пов'язаною з можливістю програмування мережевого контролера і його здатністю взаємодіяти з іншими додатками.

В результаті архітектура SDN має наступну структуру (рис. 1.2) [2].

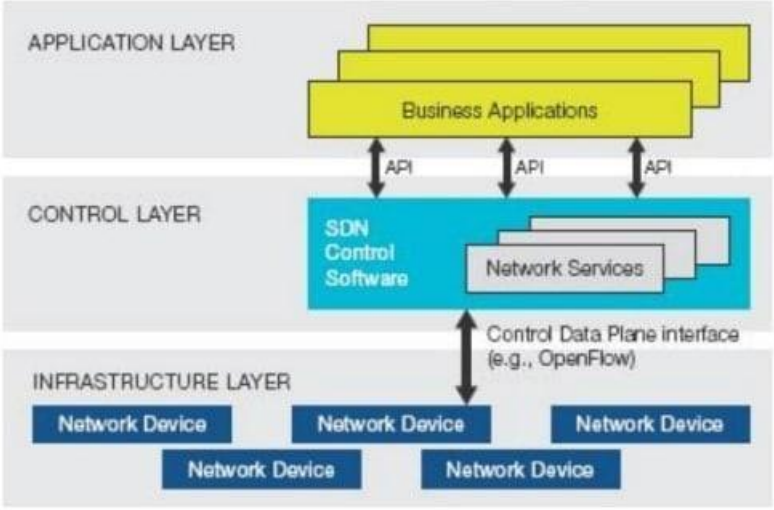


Рисунок 1.2 – Схематичне зображення структури SDN

Рівень передачі даних. На цьому рівні передавальні пристрої, як фізичні, так і віртуальні комутатори, доступні через уніфікований інтерфейс, виконують інструкції таблиць потоків для передачі мережевого трафіку.

Рівень управління. Він включає мережевий контролер або набір мережевих контролерів, що забезпечують функції управління мережею. Рівень передачі даних взаємодіє з пристроями через уніфікований програмний інтерфейс. На цьому рівні є три інтерфейси для взаємодії з іншими елементами мережевої інфраструктури:

- **південний інтерфейс** (для взаємодії з рівнем передачі даних);
- **північний інтерфейс** (для взаємодії з рівнем додатків);
- **східний та західний інтерфейси** (для взаємодії між елементами рівня управління).

Рівень додатків. Складається з високорівневих додатків, що взаємодіють з мережевим контролером або набором контролерів для реалізації конкретних функцій, що відповідають вимогам мережевої інфраструктури.

Взаємодія між мережевим контролером і передавальними пристроями здійснюється через програмний інтерфейс, який використовується для прямого

управління групами пристроїв. Найбільш розвиненим програмним інтерфейсом на цей час є OpenFlow.

Архітектура комутатора OpenFlow базується на одній або декількох таблицях правил, що визначають, як обробляти потоки мережевого трафіку. Кожне правило в таблиці є записом, що відповідає певному потоку трафіку. Відповідно до результату зіставлення, до пакетів з цього потоку застосовуються певні дії, такі як блокування, передача або модифікація.

Залежно від правил, встановлених мережевим контролером, комутатор OpenFlow може виконувати функції маршрутизатора, комутатора, брандмауера та інших елементів традиційної мережевої інфраструктури. Використання OpenFlow забезпечує гнучкість і централізоване управління мережею, що спрощує створення і модифікацію мережевої конфігурації. Основною перевагою архітектури SDN є можливість розширення функцій мережі шляхом інтеграції різноманітних програмних модулів, включаючи користувацькі, в мережевий контролер.

Архітектура SDN відкриває широкі можливості для інновацій у сфері керування процесом передачі даних і забезпечення інформаційної безпеки. Поєднання програмованості мережі і централізованого управління дозволяє покращити сучасні засоби та системи захисту даних, розширити їх функціональні можливості і підвищити продуктивність.

Механізми захоплення трафіку і збору статистики, активовані на мережевих пристроях, дозволяють формувати дані, що регулярно передаються на мережевий контролер. Додатки, що працюють на контролері, можуть обробляти і аналізувати дані з усієї мережі. На основі отриманих результатів аналізу нові чи модифіковані політики застосовуються на мережевих пристроях. Цей підхід значно підвищує ефективність контролю і протидії загрозам безпеки мережі.

1.3 Переваги використання технології SDN

Програмно-визначені мережі (SDN) пропонують низку переваг, які роблять їх привабливим рішенням для сучасних мережевих інфраструктур. Ці переваги охоплюють різні аспекти, включаючи гнучкість, ефективність, безпеку та економічність.

Однією з основних переваг SDN є можливість динамічної конфігурації мережевих пристроїв через програмні інтерфейси. Це забезпечує високу гнучкість в управлінні мережею, дозволяючи швидко адаптувати її до змін у навантаженні та вимогах користувачів. Завдяки централізованому контролеру можна швидко вносити зміни в конфігурацію мережі без необхідності фізичного доступу до кожного пристрою.

Централізація управління мережею через контролер SDN спрощує процеси моніторингу, управління та конфігурації. Це дозволяє мережевим адміністраторам мати повний контроль над усіма аспектами мережі з єдиного центру, що значно знижує складність управління великими мережами.

SDN дозволяє динамічно змінювати маршрути передачі даних, оптимізуючи трафік для зменшення затримок і підвищення пропускної здатності. Контролер може аналізувати поточні умови мережі та приймати рішення про найефективніші маршрути для передачі даних, що забезпечує більш ефективне використання мережевих ресурсів. Автоматизація управління мережею через програмні засоби зменшує необхідність ручного втручання, що знижує ймовірність помилок та зменшує витрати на обслуговування мережі. Це також дозволяє мережевим адміністраторам зосередитися на більш стратегічних завданнях, таких як планування розвитку мережі та впровадження нових сервісів (таб. 1.4)

SDN сприяє зниженню капітальних витрат завдяки використанню стандартного обладнання з програмною конфігурацією, що дозволяє уникнути

дорогих спеціалізованих апаратних рішень. Гнучкість та масштабованість SDN дозволяють організаціям поступово розширювати мережу без значних витрат на оновлення інфраструктури. Централізоване управління та програмована конфігурація мережі дозволяють швидко виявляти та реагувати на загрози безпеці. Контролер SDN може застосовувати політики безпеки на всіх пристроях мережі в режимі реального часу, забезпечуючи захист від кібератак та несанкціонованого доступу. Крім того, можливість динамічного налаштування маршрутизації дозволяє ізолювати підозрілий трафік і запобігати поширенню загроз.

SDN дозволяє швидко впроваджувати нові сервіси та функції завдяки програмній конфігурації мережевих пристроїв. Це забезпечує швидке реагування на змінні вимоги ринку та користувачів, підвищуючи конкурентоспроможність організації.

Таблиця 1.4 - Переваги використання технології SDN

Перевага	Опис
Гнучкість і програмована конфігурація	Динамічна конфігурація мережевих пристроїв через програмні інтерфейси.
Централізоване управління мережею	Спрощення процесів моніторингу, управління та конфігурації з єдиного центру.
Оптимізація маршрутизації	Динамічна зміна маршрутів для зменшення затримок і підвищення пропускної здатності.
Автоматизація та зменшення операційних витрат (OPEX)	Зниження необхідності ручного втручання, зменшення витрат на обслуговування мережі.
Зниження капітальних витрат (CAPEX)	Використання стандартного обладнання з програмною конфігурацією.
Підвищена безпека	Швидке виявлення та реагування на загрози безпеці через централізоване управління.
Спрощення впровадження нових сервісів	Швидке впровадження нових сервісів завдяки програмній конфігурації мережевих пристроїв.

Таким чином, програмно-визначені мережі забезпечують значні переваги для сучасних телекомунікаційних інфраструктур, сприяючи підвищенню їх ефективності, гнучкості та безпеки, а також зниженню витрат. Ці переваги роблять SDN ключовим елементом у розвитку майбутніх мереж і послуг.

1.4 Сучасні проблеми застосування програмно-визначених мереж

Програмно-визначені мережі (SDN) обіцяють значні переваги, проте їх впровадження та використання супроводжуються певними проблемами та викликами. Ці проблеми охоплюють різні аспекти, включаючи безпеку, масштабованість, сумісність, відмовостійкість та інші технічні та організаційні аспекти.

Один із основних викликів для SDN – це забезпечення безпеки мережі. Відокремлення контрольного рівня від рівня передачі даних створює нові можливості для кібератак. Зловмисники можуть спробувати отримати доступ до контролера, який є центральним елементом управління мережею, що може призвести до компрометації всієї мережі. Важливими аспектами безпеки є захист контролера, забезпечення безпеки комунікацій між контролером і мережевими пристроями, а також виявлення та реагування на аномалії в трафіку.

Забезпечення ефективного управління великими мережами є серйозним викликом для SDN. Контролери повинні мати здатність обробляти великий обсяг інформації про стан мережі в реальному часі. Це вимагає високопродуктивного апаратного забезпечення та оптимізованих алгоритмів управління трафіком. Масштабування контролера та забезпечення ефективної взаємодії між контролерами в розподілених середовищах є ключовими аспектами для вирішення цієї проблеми.

Інтеграція SDN з існуючими мережевими технологіями та протоколами може бути складною задачею. Багато організацій мають великий обсяг старого обладнання, яке не підтримує SDN, що ускладнює перехід до нової архітектури. Забезпечення сумісності та інтеграції з існуючими мережами є важливим аспектом успішного впровадження SDN.

Відмовостійкість є критично важливим аспектом для будь-якої мережевої інфраструктури. У SDN контролер є центральним елементом, і його відмова може призвести до значних перебоїв у роботі мережі. Необхідно розробити механізми для забезпечення безперебійної роботи контролера, включаючи резервування, відновлення після відмови та розподіл навантаження між декількома контролерами.

Затримки в обробці даних та продуктивність мережі є важливими аспектами для SDN. Оскільки контролер здійснює централізоване управління мережею, обробка запитів на рівні контролера може призвести до додаткових затримок у передачі даних. Оптимізація процесів обробки та прийняття рішень на рівні контролера є необхідною для мінімізації затримок і забезпечення високої продуктивності мережі. Хоча існують деякі стандарти, такі як OpenFlow, для взаємодії між контролером та мережевими пристроями, повна стандартизація SDN все ще є проблемою. Відсутність єдиних стандартів може призвести до несумісності між різними рішеннями та ускладнити впровадження SDN в масштабах всієї мережі (таб.1.5)

Таблиця 1.5 - Сучасні проблеми застосування програмно-визначених мереж

Проблема	Опис
Безпека	Захист контролера та комунікацій, виявлення та реагування на аномалії в трафіку.
Масштабованість	Ефективне управління великими мережами, оптимізація алгоритмів та апаратних засобів.
Сумісність	Інтеграція SDN з існуючими мережевими технологіями та протоколами.
Відмовостійкість	Забезпечення безперебійної роботи контролера, резервування та відновлення.
Затримки та продуктивність	Мінімізація затримок у передачі даних, оптимізація процесів обробки.
Відсутність стандартів	Відсутність єдиних стандартів для забезпечення сумісності між різними рішеннями.

Незважаючи на численні переваги програмно-визначених мереж, існує ряд проблем, які потребують вирішення для успішного впровадження та експлуатації SDN. Вирішення цих проблем вимагатиме спільних зусиль від розробників, дослідників та користувачів, а також подальших досліджень та розробок у сфері SDN. Це дозволить досягти повного потенціалу технології SDN та забезпечити ефективне, безпечне та надійне функціонування мережевих інфраструктур.

1.4.1 Огляд протоколу OpenFlow

В основі концепції SDN лежить відкритий стандарт OpenFlow, визначений громадською організацією Open Networking Foundation (ONF) у 2011 році. ONF був створений альянсом таких компаній, як Microsoft, Facebook, Google, Deutsche Telekom, Verizon та Yahoo. Метою цієї організації є просування і стандартизація SDN. Про його важливість свідчить той факт що склад ONF вже налічує більше

40 потужних компаній світу. Серед них: Brocade, Cisco, Citrix, Oracle, Dell, Ericsson, HP, IBM, Juniper, Marvell, NEC, Netgear, NTT, Riverbed та багато інших.

OpenFlow спочатку був розроблений для мереж Ethernet, але його можна адаптувати для використання в інших типах мереж. Мережеві пристрої можуть підтримувати як OpenFlow, так і традиційні методи передачі даних, що значно спрощує інтеграцію OpenFlow в корпоративні системи. Протокол пройшов кілька етапів розвитку, від версії 1.0 до 1.4.

Більшість виробників мережевого обладнання спочатку реалізували підтримку версії 1.0, яка мала численні обмеження і проблеми з масштабованістю, що вказувало на неповноцінність продукту, запропонованого ONF. Версії 1.1 і 1.2 вважаються перехідними, і їх підтримка майже не реалізується виробниками. Найбільш перспективною є версія 1.3, яка включає підтримку MPLS міток, лічильників per-flow, Provider Backbone Bridging (PBB) та інших корисних функцій.

Хоча OpenFlow здається повністю "відкритим" протоколом, його розробка контролюється закритими групами компаній, таких як Microsoft, Facebook, Google, Deutsche Telekom, Verizon та Yahoo, яких нараховується близько 150. Робота проводиться в закритому режимі, і її результати стають відомими лише після публікації нової версії як стандарту (рис. 1.3)

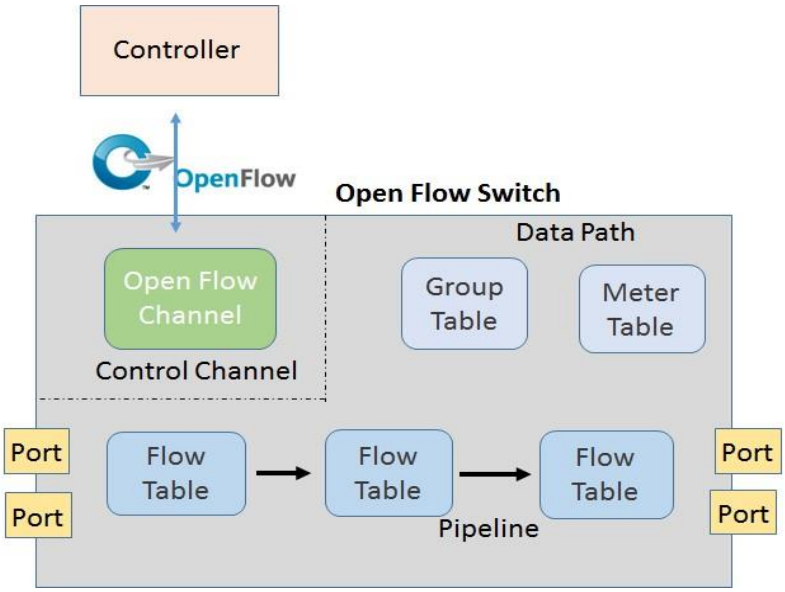


Рисунок 1.3 Складові OpenFlow комутатора.

1.4.2 Особливості протоколу OpenFlow

Протокол OpenFlow є одним з ключових елементів програмно-визначених мереж (SDN). Він забезпечує зв'язок між контролером SDN та мережевими пристроями, такими як комутатори і маршрутизатори, дозволяючи контролеру керувати поведінкою цих пристроїв. OpenFlow є першим стандартом, який реалізує концепцію SDN, і він відіграє важливу роль у впровадженні цієї технології.

Основні принципи та функціональні можливості OpenFlow

Протокол OpenFlow базується на кількох основних принципах, що забезпечують його ефективність і гнучкість в управлінні мережевими потоками:

1. Розділення рівнів управління та передачі даних: OpenFlow дозволяє відокремити контрольний рівень (control plane) від рівня передачі даних
2. (data plane). Це означає, що рішення щодо маршрутизації та обробки трафіку приймаються контролером, а комутатори виконують ці рішення.
3. Використання потоків для управління трафіком: У OpenFlow всі рішення про передачу даних базуються на потоках. Потік визначається поєднанням полів заголовка пакету, які діють як фільтр, що співставляє мережевий трафік з набором інструкцій щодо його обробки. Це дозволяє

централізовано контролювати маршрутизацію трафіку та забезпечувати гнучке управління мережею.

4. Таблиці потоків: Комутатори OpenFlow містять таблиці потоків, які визначають правила обробки пакетів. Кожен запис у таблиці містить умови для відповідності пакетів (полі заголовка) і дії, які повинні бути виконані з відповідними пакетами. Дії можуть включати пересилання пакета на певний порт, зміну полів заголовка, відкидання пакета тощо (таб. 1.6)
5. Прямий контроль над комутаторами: Контролер OpenFlow має прямий доступ до таблиць потоків комутаторів, що дозволяє динамічно оновлювати правила обробки трафіку в реальному часі. Це забезпечує високу гнучкість і адаптивність мережі до змін у навантаженні та вимогах користувачів.

Таблиця 1.6 - Основні функціональні можливості протоколу OpenFlow

Функціональна можливість	Опис
Розділення рівнів управління та передачі даних	Відокремлення контрольного рівня від рівня передачі даних для централізованого управління.
Використання потоків	Управління трафіком на основі потоків, визначених заголовками пакетів.
Таблиці потоків	Зберігання правил обробки пакетів у комутаторах для гнучкого управління мережею.
Прямий контроль над комутаторами	Динамічне оновлення правил обробки трафіку контролером у реальному часі.

Архітектура OpenFlow складається з кількох основних компонентів, які забезпечують його функціонування:

1. Контролер OpenFlow: Центральний елемент управління мережею, який здійснює моніторинг, прийняття рішень та управління комутаторами через

протокол OpenFlow. Контролер збирає інформацію про стан мережі та використовує її для прийняття рішень щодо маршрутизації та обробки трафіку.

2. Комутатори OpenFlow: Мережеві пристрої, що підтримують протокол OpenFlow і виконують команди, отримані від контролера. Вони містять таблиці потоків, що визначають правила обробки пакетів, і можуть динамічно оновлювати ці правила на основі команд від контролера.
3. Канали зв'язку OpenFlow: Взаємодія між контролером і комутаторами здійснюється через захищені канали зв'язку, що забезпечують передачу команд та інформації про стан мережі (таб.1.7)

Таблиця 1.7 - Компоненти архітектури OpenFlow

Компонент	Опис
Контролер OpenFlow	Центральний елемент управління мережею, що здійснює моніторинг та управління комутаторами.
Комутатори OpenFlow	Мережеві пристрої, що підтримують OpenFlow і виконують команди від контролера.
Канали зв'язку OpenFlow	Захищені канали для взаємодії між контролером і комутаторами.

Переваги:

1. Гнучке управління мережею. OpenFlow дозволяє динамічно змінювати правила обробки трафіку, що забезпечує високу гнучкість управління мережею.
2. Централізоване управління. Контролер забезпечує централізоване управління мережею, спрощуючи моніторинг, управління та конфігурацію.

3. Оптимізація трафіку. Використання потоків дозволяє оптимізувати маршрутизацію та забезпечити ефективне використання мережевих ресурсів.
4. Автоматизація процесів. Можливість автоматизації налаштувань і управління мережею знижує потребу в ручному втручанні.

Виклики:

1. Безпека. Забезпечення безпеки контролера та комунікацій між контролером і комутаторами є критичним аспектом (таб. 1.8)
2. Масштабованість. Ефективне управління великими мережами вимагає оптимізації алгоритмів та високопродуктивного обладнання.
3. Сумісність. Інтеграція з існуючими мережевими інфраструктурами може бути складною.

Таблиця 1.8 - Переваги та виклики використання OpenFlow

Аспект	Опис
Переваги	Гнучке управління мережею, централізоване управління, оптимізація трафіку, автоматизація процесів.
Виклики	Забезпечення безпеки, масштабованість, сумісність з існуючими інфраструктурами.

Таким чином, протокол OpenFlow є важливим елементом архітектури програмно-визначених мереж, який забезпечує гнучке та централізоване управління мережею. Незважаючи на численні переваги, впровадження OpenFlow супроводжується певними викликами, які потребують вирішення для досягнення повного потенціалу цієї технології.

1.4.3 Вразливості протоколу OpenFlow

Позитивні та негативні аспекти архітектури SDN для безпеки мережевої інфраструктури добре відомі. Оцінка можливих вразливостей архітектури повинна базуватися не лише на теоретичних міркуваннях, але й на результатах експериментів із використання протоколу OpenFlow в промислових мережах.

Сьогодні відомі такі типи загроз для мереж, що використовують OpenFlow. Зловмисники можуть легко ідентифікувати реактивний чи проактивний режими роботи контролера без застосування специфічних підходів і спеціального програмного забезпечення. Ідентифікація заснована на затримці першого пакету для нового потоку трафіку і може бути доступна для кожного користувача, який підключений до мережі або використовує сервіси цієї інфраструктури через зовнішні мережі. В результаті, деякі атаки можуть використовувати певний режим роботи контролера [31].

Наприклад, несанкціонована установка правил обробки на комутаторах, що призводить до зниження ефективності або порушення роботи мережі, простіше реалізується в реактивному режимі через особливості підходу контролера до управління таблицями потоків у цьому режимі. Вплив цієї атаки на проактивний контролер можливий, але більш складний, оскільки потрібна комплексна атака, і шанс швидкого виявлення факту атаки значно вищий.

Загроза, що стосується безпеки, є досить актуальною для більшості інформаційних систем. Такі загрози, як сканування портів та визначення мережевих служб, є досить критичними щодо архітектури SDN. Це спричинено вразливістю каналу OpenFlow та великою кількістю трафіку управління, який передається між мережевими контролерами та комутаторами.

Архітектура SDN є досить вразливою до DoS-атак, що є одним з найбільш небезпечних видів атак стосовно архітектури з централізованою точкою управління. Оскільки пристрої, що передають інформацію, не мають функцій

керування, вони обов’язково повинні мати стабільне під’єднання до контролера мережі для забезпечення безперебійної передачі даних.

Якщо контролер працює в реактивному режимі, навіть короткий період недоступності OpenFlow-контролера може спричинити проблеми в мережевій інфраструктурі. У випадку довгострокового блокування мережевого контролера, це може призвести до повної зупинки обробки мережевого трафіку або ще складнішої атаки. Наприклад, помилковий мережевий контролер може замінити заблокований, що поставить під загрозу всю інфраструктуру мережі.

Ще однією проблемою є вразливість програмної інфраструктури OpenFlow мережі. Здатність програмувати мережі та існування відкритих програмних інтерфейсів, які використовуються для інтеграції з мережевим контролером, спричиняють появу вразливостей ПЗ. Хоча контролери OpenFlow, розроблені професіоналами і підтримувані великими компаніями, можуть бути надійними, програмні модулі для контролера, розроблені інженерами для конкретних потреб інфраструктури, можуть створювати неочікувані вразливості, що вплинуть на безпеку всієї мережі. Ця проблема безпеки виникає через низький рівень стандартизації рівнів управління і додатків в архітектурі SDN.

Ще однією загрозою, яка часто виникає в традиційних мережах, є атака з підміною. У мережах SDN ця загроза стає ще більш потенційною порівняно з традиційними мережами. Оскільки вся мережа керується централізованим контролером, ризик того, що керування трафіком в мережі OpenFlow буде підмінено, залишається досить високим.

Підміною може бути здійснений несанкціонований доступ до пристроїв мережі, неправильне зібрання статистики або даних про стан мережі контролером. Це може негативно вплинути на роботу всієї мережі.

Оскільки стандарт OpenFlow досить гнучкий, це може призвести до вразливості мереж OpenFlow до атак підміни. Стандарт дозволяє взаємодію між

контролером мережі та комутаторами за допомогою протоколу TCP без шифрування, а підтримка протоколу TLS є необов'язковою для реалізації.

Усі розглянуті загрози безпеки пов’язані не з конкретними версіями протоколу OpenFlow. Вони актуальні для всіх версій протоколу, що були випущені, і, можливо, їх не вдасться усунути застосовуючи фундаментальні особливості архітектури SDN.

Можна зауважити, що завдання захисту ресурсів програмно-керованих мереж на сьогоднішній день залишаються невирішеними.

1.5 Основні технічні проблеми, які виникають під час розробки та впровадження SDN, включають такі аспекти:

1. **Управління та оркестрування:** Ця проблема стосується ефективного керування і координації ресурсами мережі, а також їх автоматизації.
2. **Сумісність та об'єднання:** Необхідність забезпечення сумісності різних компонентів мережі та їх інтеграції в єдину систему.
3. **Операції з мережевими та обчислювальними послугами:** Розробка механізмів для ефективного взаємодії між мережевими та обчислювальними ресурсами для надання комплексних послуг.
4. **Архітектурно-функціональні моделі:** Необхідність розробки ефективних моделей архітектури та функціонування SDN, що відповідали б вимогам сучасних мереж.

При роботі SDN з використанням протоколу OpenFlow також виникає проблема паралельного з'єднання в мережах з кількома комутаторами та утворення петель під час передачі даних, що може спричинити застій у мережі.

Окрім цього, існує потреба у підвищенні якості обслуговування. Тому велика увага приділяється створенню та розвитку програмно-конфігурованих мереж та контролю якості надання цих послуг.

Виявлення та вирішення цих проблем допоможе підвищити ефективність роботи SDN з використанням протоколу OpenFlow.

1.4.4 Управління та оркестровка

Фундаментальна особливість SDN полягає у відокремленні площини керування мережею від площини пересилання (даних). Цей принцип, добре відомий у телекомунікаційних мережах, визначає функції та структуру площини керування, які часто стандартизовані Форумом TeleManagement та ITU-T. Ці функції включають управління помилками, конфігурацією, обліком, продуктивністю та безпекою.

У виробничих мережах особливо важлива роль управління, включаючи роботу з проблемами, конфігурацією, обліком, продуктивністю та безпекою. У таких мережах оператори зазвичай взаємодіють з мережею через систему управління. Передбачається, що для впорядкування майбутніх потреб та зростаючої кількості керованих пристроїв, управління мережею повинно бути автоматизовано.

Однак, управління SDN залишається недостатньо розвинутим. Це пояснюється експериментальним статусом розгорнутих мереж. Нинішні підходи до SDN часто передбачають використання існуючих систем керування мережею IP у поєднанні з OpenFlow.

Відкритість OpenFlow дозволяє реалізувати деякі функції керування мережею, проте відсутність стандартизації інтерфейсу керування ускладнює використання сторонніх рішень управління, аналогічно до застарілих систем.

Мережі на основі SDN мають свої властивості, включаючи можливість власника або оператора визначати їх функціональні можливості. Існують також проблеми керування, які є характерними саме для SDN.

Більшість цих аспектів пов’язані з важливим завданням контролера, який має виконувати більшість своїх операцій у режимі реального часу. Це означає, що продуктивність контролера має бути відповідно контрольована, а його процеси повинні бути оброблені з урахуванням їхніх вимог у реальному часі. У випадку відмови контролера важлива можливість гарячої заміни. Програмування контролера вимагає особливої уваги. Для повного використання такої можливості оператор мережі повинен мати змогу додавати нові функції контролера та оновлювати наявні віддалено. Ця функціональність передбачає перепрограмування контролера під час роботи. Операції керування повинні мати можливість запускати, зупиняти та контролювати кожен програмний модуль.

Крім того, важливо відстежувати аномальну поведінку таких модулів і вживати заходів для їх виправлення. І, нарешті, безпека платформи контролера є критичною і потребує належного уваги. Це також охоплює аутентифікацію комутаторів, що щойно були додані. Важливо визначити межу між мережевими операціями SDN (які виконуються в контролері) та програмами на основі SDN, що розгорнуті на серверах додатків. Такий розподіл може сприяти визначенню ролі системи управління. Для впорядкування масштабованості та складності настійно рекомендується використовувати окремі платформи для контролю та керування мережами SDN. Необхідний належний інформаційний обмін між цими платформами, який може забезпечуватися програмованими інтерфейсами, залишається відкритим питанням, чи буде управління повністю централізованим, чи буде реалізовано за допомогою розподілених функцій. Однак, безсумнівно, додаткова складність не повинна значно збільшувати вартість вузлів SDN, і розподіл функціональних можливостей все одно має давати централізоване уявлення про мережу.

1. Адаптивні та автономні методи та системи: Цей дослідницький виклик полягає в розробці критичних методів, засобів і систем для автономних

функцій управління, що застосовуються не лише до фізичних, а й до віртуальних ресурсів всередині мережі. Хоча вимога уніфікації всіх автономних функцій може бути бажаною, досягнути її буде складно через велику кількість функцій. Тому слід розробити нові підходи, які дозволять автономним менеджерам координувати та вирішувати конфлікти з мінімальними порушеннями. Наприклад, функції управління та контролю можуть бути розподілені або розміщені поруч з керованою мережею, дозволяючи економію капітальних та операційних витрат.

2. Оптимізований розподіл та оркестровка ресурсів: Ця проблема полягає в динамічному створенні, оркеструванні та міграції віртуальних машин через мережі та інфраструктури послуг. Оркестровка також означає управління життєвим циклом ресурсів. Спеціальні методи оптимізації потрібні для розміщення та маршрутизації трафіку між віртуальними машинами. У випадку бездротової інфраструктури потрібно враховувати обмежену пропускну здатність та енергетичні можливості.

3. Енергетичний менеджмент: Важливо розробити механізми для оптимального розподілу віртуальних машин та стабілізації локальних мереж у відповідь на змінні вимоги попиту на електроенергію.

Охорона та безпека: Необхідно гарантувати безпеку віртуального середовища на рівні, що не нижче від сучасних мереж, з урахуванням нових можливостей віртуалізації та моделей тарифікації. Системи SDN потребують надійного резервування та заходів безпеки для контролерів та ресурсів.

1.4.5 Сумісність та об’єднання

• Рішення SDN вже використовують численні відкриті та комерційні програмні та апаратні комутатори та контролери OpenFlow. Однак значна різноманітність версій комутаторів, контролерів та протоколів OpenFlow ставить під сумнів їх сумісність. Розвиток протоколу OpenFlow та використання інших протоколів, крім OpenFlow, потребуватиме гарантії сумісності.

• Загалом, сумісність між різними OpenFlow та різними доменами з одним і тим же OpenFlow надзвичайно важлива. Цей виклик включатиме наступне:

- Розробка інтерфейсів, які дозволять об’єднати віртуальну функцію обслуговування/мережі. Завдяки цим інтерфейсам, мережева функція повинна здати співпрацювати для забезпечення міждоменного зв’язку.
- Реалізація аутентифікації для різних OpenFlow та різних доменів для підтвердження ідентичності різних користувачів певної послуги.
- Розробка механізмів зв'язку та програмування сервісних модулів, розгорнутих різними операторами для одного сервісу.

• Впровадження механізмів наскрізного управління послугами, моніторингу та обліку.

1.4.6 Операції з мережевими та обчислювальними послугами

• Service Operation відповідає за безперебійне функціонування інфраструктури та послуг, які вона надає. Це означає постійний моніторинг середовища виконання послуг для виявлення проблем, забезпечення доступності послуг та програмованості.

1. Забезпечення ефективного та безпечного середовища в мережі:
    - Створення єдиної платформи віртуального хостингу та віртуальних машин у мережі.
    - Встановлення єдиного рівня керування для всіх віртуальних середовищ, що дозволяє управляти ресурсами на рівні вузлів.
    - Розподіл ресурсів між віртуальними середовищами для забезпечення безперебійної роботи та обліку витрат.
2. Програмованість у майбутніх мережах і сервісах:

- Розробка швидких та гнучких рішень для розгортання нових мережевих послуг за допомогою програмованих активаторів і примітивів на всіх рівнях SDN.
- Впровадження динамічної програмованості мережі та її пристроїв для легкого розгортання нових мережевих послуг.
- Забезпечення безперебійного функціонування мережі та захисту від загроз для ефективного використання програмованих ресурсів.

• Ці завдання спрямовані на покращення якості обслуговування та гнучкості мережі, що є критичними для операторів та кінцевих користувачів.

1.4.7 Архітектурно-функціональні моделі

Поточні дослідження в галузі SDN і NFV створюють нові напрямки, пов'язані з тим, як забезпечити ефективне та постійне оновлення та модифікацію мережевих функцій, не переінвентарюючи архітектуру мережі або мережеві шари кожного разу.

Це передбачає використання програмного забезпечення для динамічного програмування окремих мережевих пристроїв, систем та служб, тим самим контролюючи, керуючи та програмованою поведінкою мережі в цілому.

Ключові програмні характеристики майбутніх мереж і послуг вже визначені і включають в себе різноманітність послуг, функціональну гнучкість, програмованість, простоту впровадження нових послуг, віртуалізацію ресурсів, споживання енергії, універсальність послуг, керування мережею, мобільність, оптимізацію, ідентифікацію, надійність та безпеку, які будуть реалізовані як частина майбутніх мережевих послуг і постійно оновлюватимуться.

Майбутні мережі та послуги мають перейти від простого визначення програмним забезпеченням до програмованого програмним забезпеченням та мати можливість підтримувати різні постачальники послуг, які використовують середовище, де послуги динамічно розгортаються та швидко адаптуються до

гетерогенного фізичного та бездротового зв'язку та інфраструктури інтелектуальних об'єктів.

Програмованість є ключовою властивістю, яку надає структура SDN, але це не робить її "простішою", оскільки належні абстракції та набір шарів ще не повністю визначені.

1. Архітектурні моделі

Існують різні абстракції, нашарування, концептуальні моделі та архітектурні підходи, які були запропоновані в науковій літературі. Деякі з них будуть розглянуті надалі.

Програмованість мережевих елементів (комутаторів, маршрутизаторів і т. д.) була представлена приблизно десять років тому і заклала основу для швидкого розгортання та налаштування нових служб, як показано на рис. 1.6

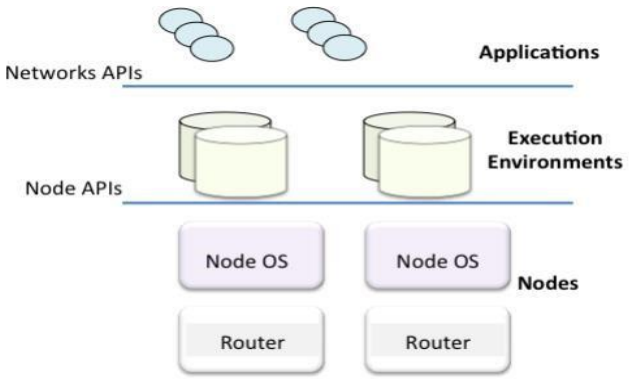


Рисунок 1.6 – Приклад моделі мережі програмованого маршрутизатора

Прогрес у розробці програмованих та віртуальних мереж спричинений упровадженням OpenFlow, що викликало вдосконалення архітектурної моделі на високому рівні, як на рис. 1.7.

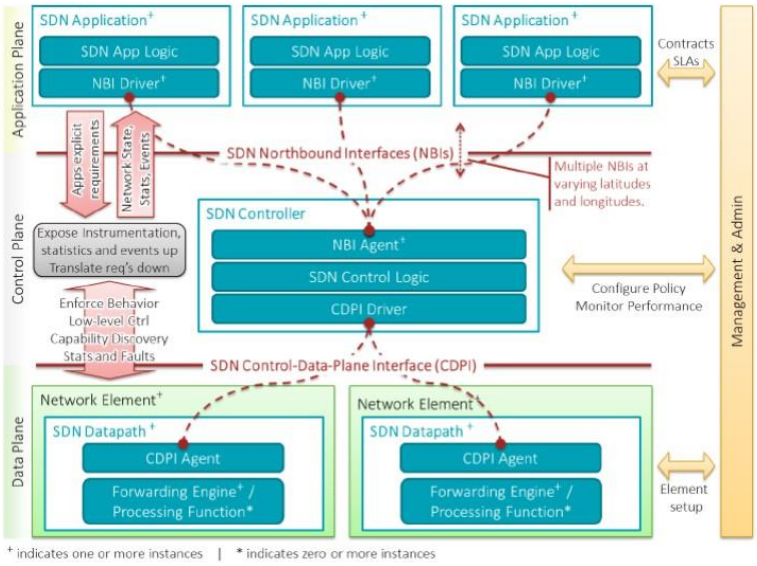


Рисунок 1.7 – Архітектурна модель SDN

Наразі спостерігається зростання зацікавленості, що переходить від централізованого управління та "монолітних" підходів, де системи вертикально інтегровані, до підходу на основі компонентів. У цьому новому підході системи складаються з кількох компонентів від різних виробників, які взаємодіють один з одним через відкриті інтерфейси для формування нових послуг. Очікувані результати покликані забезпечити рівні умови гри, коли різні зацікавлені сторони, такі як постачальники інфраструктури та послуг, будуть конкурувати один з одним, а користувачі матимуть можливість вибирати та налаштовувати послуги відповідно до своїх потреб. Фундаментальною характеристикою архітектурної моделі є еволюція до єдиного середовища, що інтегрує підключення, обчислювальну потужність і зберігання. Це передбачає належне покращення поточних планів контролю та управління та впровадження можливостей оркестрації (рис. 1.8).

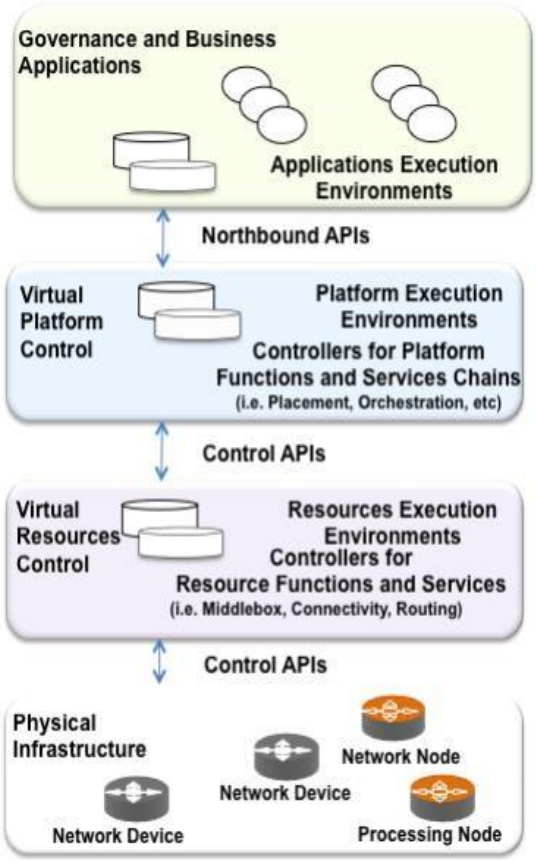


Рисунок 1.8 – Приклад моделі інтеграції єдиного середовища підключення, обчислювальна потужність і зберігання.

Перед розглядом цього, важливо визначити значущу функціональну модель, яка зможе абстрагувати та представляти всі можливі мережеві функції та можливості, що виникають у зв'язку з SDN та NFV, включаючи обробку та зберігання даних. Ця модель має бути реалізована та адаптована для використання в різних областях, таких як хмарні середовища, мережі доступу радіо, а також в мобільних та провідних мережах (див. рисунок 1.9).

Одна з ключових проблем попередніх моделей полягає в тому, що різноманітність та об'єм додатків і технологій роблять прості моделі непридатними для всіх сценаріїв (наприклад, простий API, орієнтований на північ, може бути недостатнім). Це вимагає застосування рекурсивного підходу, при якому розробляється новий набір абстракцій в залежності від конкретної області або домену, а також необхідного набору шарів.

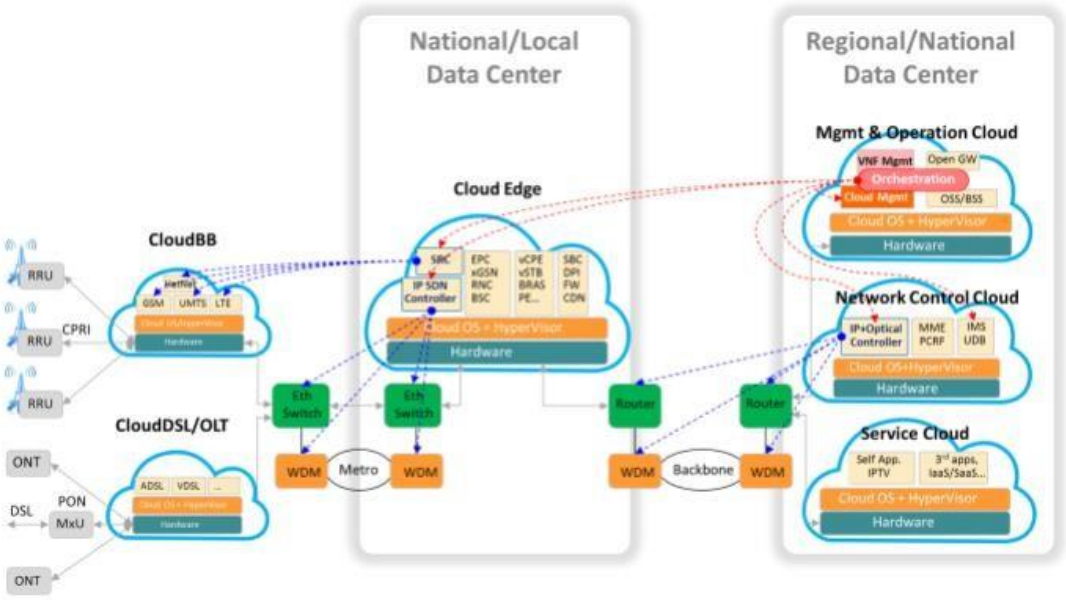


Рисунок 1.9 – Приклад екземплярів SDN у хмарних, RAN і стільникових/дротових доменах (семінар SDN4FNS)

Додатково, з урахуванням тенденції користувачів до переходу на мобільні середовища та пристрої, стає критично важливим створення більш щільних мереж та ефективне використання бездротових ресурсів. Це свідчить про зростаючий попит і надмірне використання ресурсів. Припускається, що узагальнення набору абстракцій для мобільних мереж та використання структури SDN можуть забезпечити додаткові переваги управління, конвергентного контролю ресурсів і гнучкого розгортання в мережах з ростучою щільністю.

У таких сценаріях виникає необхідність в ефективному контролі та управлінні ресурсами через різні бездротові протоколи, можливість поєднувати та координувати ці можливості з іншими функціями основної мережі, а також створення загальної та програмованої площини даних та операційної системи бездротової мережі.

Це можливо лише за умови перегляду архітектури 3GPP, що починається з визначення доменів логічної мережі, концепцій та архітектури управління сесіями, мобільністю та ідентифікацією, а також протоколів доступу та без

доступу Stratum. Наприклад, важливо визначити, як мобільність буде оброблятися в SDN без використання тунелів, і чи будуть протоколи, що реалізують послугу-носій (рівень 3GPP), наближатися до протоколів на рівні транспортної мережі, які розвивалися незалежно.

Підсумовуючи, мережам постачальників послуг потрібен більш складний набір елементів керування та багатошарових абстракцій, які можна буде реалізувати ітераційно. Багато деталей ще потребують розробки, але ключовою є функція Проксі координації/співробітництва. Цей проксіма може отримувати директиви політики "згори", наприклад, щодо бізнесу, технологій та якості, і перетворювати їх на відповідні застосунки, що є ключовим аспектом для досягнення глобальної мети.

Загальна архітектурна модель повинна охоплювати спільні віртуалізовані ресурси та всі відповідні абстракції, включаючи провідні, бездротові та мобільні пристрої, а також розумні об'єкти з обмеженими ресурсами. Розробка такої моделі є необхідною для полегшення інтеграції та надання різноманітних ІКТ-послуг, обчислювальних та мережевих хмар, а також для поліпшення інтеграції ключових технологій: програмування, мереж, віртуалізації мереж та віртуалізації мережевих функцій, а також самокерування.

Щодо функціонального моделювання, починаючи з OSI Layering, інтеграція інфраструктури програмного забезпечення та традиційних комунікаційних/телекомунікаційних технологій завжди була складною задачею для операторів мереж та послуг у контексті розгортання та керування послугами. Це переважно пов'язано з тим, що рівні OSI були спроектовані для вирішення парадигми доставки IP-пакетів на початкових етапах Інтернету. Цікаво, що для подолання цих проблем ITU-T розробив функціональний формалізм, здатний моделювати транспортні мережі технологічно незалежним способом. Рекомендація ITU-T G.805 описує моделювання транспортних мереж, орієнтованих на з'єднання (точка-точка); потім G.809 був створений для

моделювання мережі без з’єднання, і, нарешті, G.800 повідомляє про модель уніфікованої архітектури, орієнтованої на з'єднання/без з'єднання (рис. 1.10).

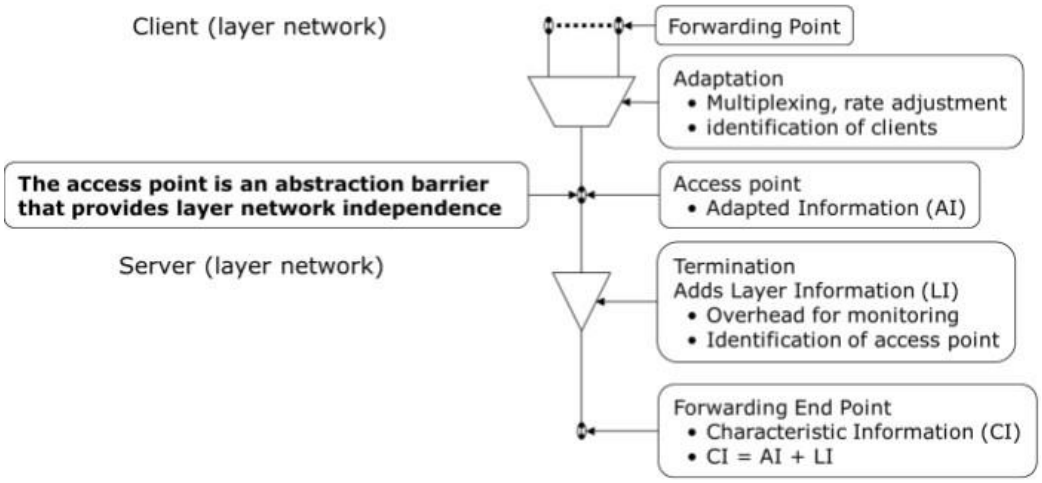


Рисунок 1.10 – Компоненти G.805 для функціонального моделювання

Основними атрибутами цього функціонального моделювання є його здатність рекурсивно описувати різноманітні транспортні мережі, зосереджуючись на їх можливостях передачі інформації, незалежно від конкретних технологій та за допомогою обмеженого набору компонентів. Крім того, ця модель може ефективно поєднувати концепції обладнання, логічних ресурсів та управління.

Цей підхід виявився дуже результативним у проектуванні та управлінні мережами. Заявляється, що аналогічний підхід, з відповідними удосконаленнями, має бути застосований у функціональному моделюванні SDN. Ключовим в цьому контексті є визначення обмеженого набору функціональних компонентів, які здатні рекурсивно моделювати SDN у всіх його аспектах, включаючи функції обробки та зберігання.

2 ДОСЛІДЖЕННЯ МЕРЕЖІ SDN ДЛЯ ПОКРАЩЕННЯ УПРАВЛІННЯ ПРОГРАМНО-ВИЗНАЧЕНИМИ МЕРЕЖАМИ, ТА БАЛАНСУВАННЯ МЕРЕЖЕВОГО ТРАФІКУ

2.1 Протокол Spanning Tree

Протокол Spanning Tree (STP), розроблений IEEE під стандартом 802.1D, є ключовим протоколом у традиційних Ethernet-мережах, який використовується для запобігання утворенню петльових структур. Петлі в мережі можуть призвести до серйозних проблем, таких як множинні копії пакетів, неправильне маршрутизування та збільшення навантаження на мережеві пристрої.

Протокол STP вирішує ці проблеми шляхом визначення логічного дерева мінімального покриття (spanning tree), яке включає всі вузли мережі, але не містить жодних петльових структур. Вузли, що не входять до цього дерева, автоматично блокуються для передачі трафіку, що забезпечує уникнення петель.

Основні компоненти та функції STP

1. Кореневий міст (Root Bridge): Центральний вузол дерева, який обирається на основі найнижчого пріоритету та MAC-адреси. Всі інші мости обчислюють свої шляхи до кореневого мосту, що визначає структуру дерева.
2. Порт кореневого моста (Root Port): Порт, через який міст досягає кореневого моста найкоротшим шляхом. Кожен не-кореневий міст має один порт кореневого моста.
3. Дизайгнований порт (Designated Port): Порт на кожному сегменті мережі, який має найкоротший шлях до кореневого моста. Він відповідає за передачу трафіку на цьому сегменті.
4. Блокований порт (Blocked Port): Порт, який не використовується для передачі трафіку, оскільки він не є кореневим або дизайгнованим портом. Блокування запобігає утворенню петель у мережі.

Алгоритм Spanning Tree

Протокол STP використовує алгоритм, який працює у кілька етапів:

1. Ініціалізація: Кожен міст вважає себе кореневим мостом та обчислює свої шляхи.
2. Вибір кореневого моста: Мости обмінюються Bridge Protocol Data Units (BPDU), які містять інформацію про пріоритет і MAC-адресу. Міст з найнижчим пріоритетом і MAC-адресою стає кореневим мостом.
3. Визначення портів: Кожен міст обчислює найкоротший шлях до кореневого моста та призначає свої порти як кореневий, дизайгнований або блокований.
4. Підтримка топології: Мости періодично обмінюються BPDU для підтримки актуальної топології мережі. Якщо топологія змінюється (наприклад, через відмову порту), мости оновлюють свої обчислення.

Роль STP у SDN-мережах (таб. 2.1)

У традиційних мережах протокол STP є основним засобом запобігання утворенню петель. Однак у контексті програмно-визначених мереж (SDN) цей протокол може бути замінений або доповнений іншими методами, що забезпечують більш гнучке та ефективне управління мережевою топологією.

В SDN-мережах контролер має глобальний огляд всієї мережі та може динамічно керувати маршрутизацією трафіку, що дозволяє ефективніше уникати петель та оптимізувати використання ресурсів. Однак, у випадках, коли SDN впроваджується у існуючі мережі з використанням традиційного обладнання, STP все ще може відігравати важливу роль у забезпеченні стабільності та надійності мережі.

Таблиця 2.1 - Порівняння використання STP у традиційних та SDN-мережах

Аспект	Традиційні мережі	SDN-мережі
Управління петлями	Використання STP для запобігання петлям	Використання контролера для динамічного управління
Оптимізація трафіку	Обмежена	Висока
Гнучкість	Низька	Висока
Обмін інформацією	Локальний обмін BPDU	Глобальний огляд мережі контролером

Проблеми та виклики використання STP

1. Час конвергенції: Протокол STP може вимагати значного часу для досягнення стабільної топології після змін, що може призводити до тимчасових перебоїв у роботі мережі.
2. Оптимізація використання ресурсів: STP блокує деякі порти для уникнення петель, що може призводити до неефективного використання доступної пропускної здатності мережі.
3. Складність налаштування: Налаштування пріоритетів та інших параметрів протоколу може бути складним завданням, особливо в великих мережах.

Незважаючи на ці виклики, протокол Spanning Tree залишається важливим інструментом для забезпечення стабільності та надійності традиційних мереж. У SDN-мережах його функції можуть бути доповнені або замінені іншими методами управління топологією, що забезпечує більш гнучке та ефективне управління мережевою інфраструктурою.

2.2 Вибір контролера та тестової платформи

Для проведення дослідження та тестування програмно-визначених мереж (SDN) було вирішено використати контролер OpenDaylight та тестову платформу Mininet. Цей вибір зумовлений широким функціональним спектром, високою продуктивністю та гнучкістю обох рішень, що забезпечує оптимальні умови для моделювання та тестування SDN-мереж.

Вибір контролера: OpenDaylight

Контролер OpenDaylight є одним із найпопулярніших контролерів SDN з відкритим кодом, який підтримує широкий спектр мережевих протоколів і функцій. Він розроблений під егідою Linux Foundation і має активну спільноту розробників, що забезпечує постійне вдосконалення та підтримку.

Основні характеристики OpenDaylight:

1. Функціональні можливості

   OpenDaylight підтримує широкий спектр мережевих функцій, включаючи маршрутизацію, балансування навантаження, забезпечення якості обслуговування (QoS), управління безпекою тощо. Це робить його універсальним інструментом для управління різноманітними мережевими конфігураціями.

2. Продуктивність

   OpenDaylight забезпечує високу продуктивність завдяки ефективному використанню апаратних ресурсів та оптимізованим алгоритмам управління трафіком. Це дозволяє обробляти великі обсяги трафіку в реальному часі.

3. Масштабованість

   Контролер підтримує високу масштабованість, що дозволяє розширювати мережу без значних витрат на модернізацію інфраструктури. OpenDaylight може керувати як малими мережами, так і великими інфраструктурами з численними пристроями.

4. Сумісність

OpenDaylight підтримує стандартний протокол OpenFlow, а також інші мережеві протоколи, що забезпечує сумісність з широким спектром мережевих пристроїв. Це дозволяє інтегрувати його з існуючими мережевими інфраструктурами.

5. Безпека

   Контролер забезпечує високий рівень безпеки, включаючи захист від кібератак та забезпечення конфіденційності даних. OpenDaylight має вбудовані механізми для виявлення та реагування на загрози безпеці.

6. Гнучкість та розширюваність

   OpenDaylight підтримує можливість додавання нових функцій та адаптації до змінних вимог мережі. Це дозволяє розробникам створювати власні додатки та розширення для покращення функціональності контролера (таб. 2.2).

Таблиця 2.2 - Характеристики контролера OpenDaylight

Характеристика	Опис
Функціональні можливості	Маршрутизація, балансування навантаження, QoS, безпека
Продуктивність	Висока продуктивність, оптимізація алгоритмів
Масштабованість	Підтримка малих та великих мереж
Сумісність	Підтримка OpenFlow та інших мережевих протоколів
Безпека	Високий рівень безпеки, вбудовані механізми захисту
Гнучкість	Підтримка додавання нових функцій, розширення

Вибір тестової платформи: Mininet

Mininet є популярною платформою з відкритим кодом, яка дозволяє створювати віртуальні мережі на одному комп'ютері. Вона забезпечує просте та ефективне середовище для моделювання та тестування SDN-мереж.

Основні характеристики Mininet:

1. Віртуалізація мереж

   Mininet дозволяє створювати віртуальні мережі, що включають віртуальні комутатори, маршрутизатори та хости. Це дозволяє моделювати складні мережеві топології без необхідності використання фізичного обладнання.

2. Підтримка OpenFlow

   Mininet повністю підтримує протокол OpenFlow, що дозволяє тестувати та оцінювати роботу контролерів SDN, таких як OpenDaylight, у реальних умовах.

3. Легкість використання

   Mininet має простий у використанні інтерфейс командного рядка, що дозволяє швидко створювати та налаштовувати мережеві топології. Це забезпечує зручність і швидкість роботи з платформою.

4. Масштабованість

   Mininet дозволяє створювати мережі будь-якого розміру, від малих до великих топологій. Це дозволяє моделювати різні сценарії та оцінювати ефективність різних мережевих конфігурацій.

5. Гнучкість

   Mininet забезпечує гнучкість у налаштуванні мережевих топологій та тестуванні різних рішень. Це дозволяє адаптувати мережу до змінних вимог та умов.

6. Економічність

   Використання Mininet дозволяє значно знизити витрати на тестування, оскільки не потребує придбання фізичного обладнання. Це робить його ідеальним інструментом для досліджень та навчання (таб. 2.3).

Таблиця 2.3- Характеристики тестової платформи Mininet

Характеристика	Опис
Віртуалізація мереж	Створення віртуальних мереж, включаючи комутатори, маршрутизатори та хости
Підтримка OpenFlow	Повна підтримка протоколу OpenFlow
Легкість використання	Простий інтерфейс командного рядка, швидке створення та налаштування топологій
Масштабованість	Підтримка мереж будь-якого розміру
Гнучкість	Гнучке налаштування мережевих топологій та тестування різних рішень
Економічність	Зниження витрат на тестування, відсутність необхідності у фізичному обладнанні

Процес налаштування OpenDaylight та Mininet

Для налаштування та використання контролера OpenDaylight та тестової платформи Mininet необхідно виконати кілька кроків:

1. Встановлення OpenDaylight. Завантажити останню версію OpenDaylight з офіційного веб-сайту та встановити її на сервері або віртуальній машині. Виконати початкову конфігурацію контролера, включаючи налаштування мережевих протоколів та безпеки.
2. Встановлення Mininet. Завантажити та встановити Mininet на локальному комп'ютері або віртуальній машині. Mininet може бути встановлений на будь-якій сучасній операційній системі, що підтримує віртуалізацію.
3. Створення віртуальної мережі в Mininet. Використовуючи інтерфейс командного рядка Mininet, створити віртуальну мережу, що включає

необхідну кількість комутаторів, маршрутизаторів та хостів. Налаштувати параметри мережевих пристроїв для забезпечення сумісності з контролером OpenDaylight.

4. Підключення OpenDaylight до Mininet. Налаштувати Mininet для взаємодії з контролером OpenDaylight через протокол OpenFlow. Переконатися, що контролер успішно підключений до віртуальних комутаторів та може керувати ними.
5. Тестування та оцінка. Виконати тестування створеної мережі, оцінюючи ефективність управління трафіком, балансування навантаження, забезпечення якості обслуговування та безпеку. Аналізувати результати та вносити необхідні корективи в налаштування контролера та мережевих пристроїв.

Вибір контролера OpenDaylight та тестової платформи Mininet забезпечує оптимальні умови для дослідження та тестування програмно-визначених мереж. Висока продуктивність, гнучкість та сумісність цих рішень дозволяють створювати складні мережеві конфігурації та оцінювати ефективність різних підходів до управління мережею. Це сприятиме досягненню поставлених цілей дослідження та впровадженню SDN у реальних мережах.

2.3 Оптимізація існуючих модулів конролера SDN

У цій версії алгоритму spanning tree для визначення зв'язуючого дерева застосовується метод обходу в ширину (Breadth First Traversal). Суть цього алгоритму полягає в обробці вершин графа на рівні і перед переходом до рівня і + 1.

Приклад роботи алгоритму Breadth First Traversal представлено на рисунках 2.6, 2.7 та 2.8.

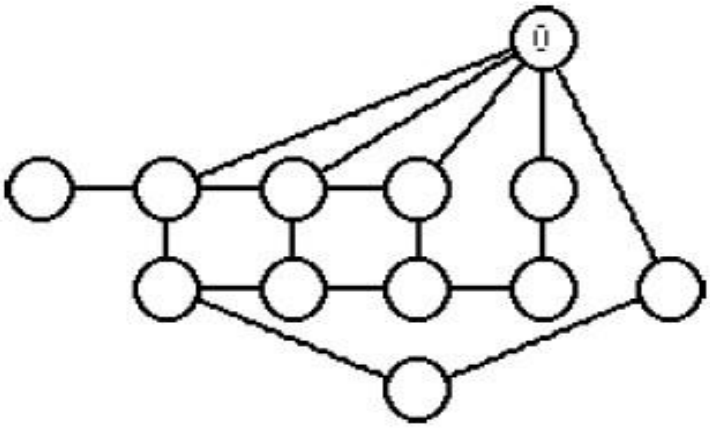


Рисунок 2.6 - Приклад розрахунку алгоритму Breadth First Traversal, початковий стан.

На рисунку 2.6 показано стан розрахунку дерева в мережі, в якому корінь дерева вже обраний. У імлементації Openflow це стан можна порівняти з кроками 1, 2, і 3. рисунки 2.7 і 2.8 відповідають крокам 4 - 7 вищеописаної імплементації.

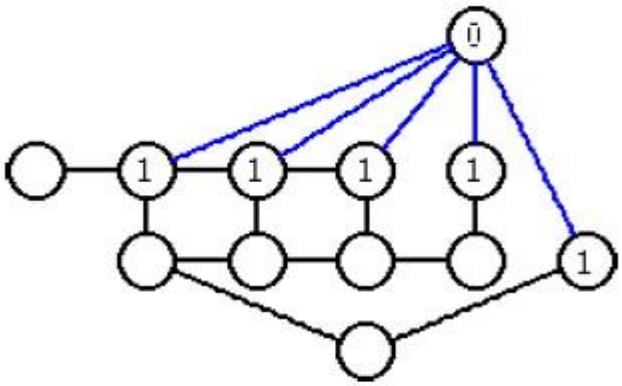


Рисунок 2.7 - Приклад розрахунку алгоритму Breadth First Traversal, стан після першого кроку.

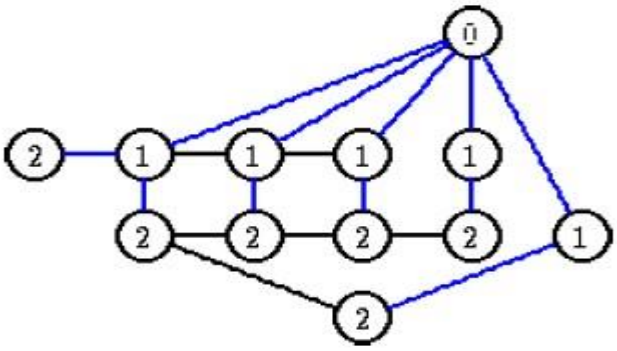


Рисунок 2.8 - Приклад розрахунку алгоритму Breadth First Traversal, результат

Блок схема роботи модуля Spanning tree відображена на рисунку 2.9.

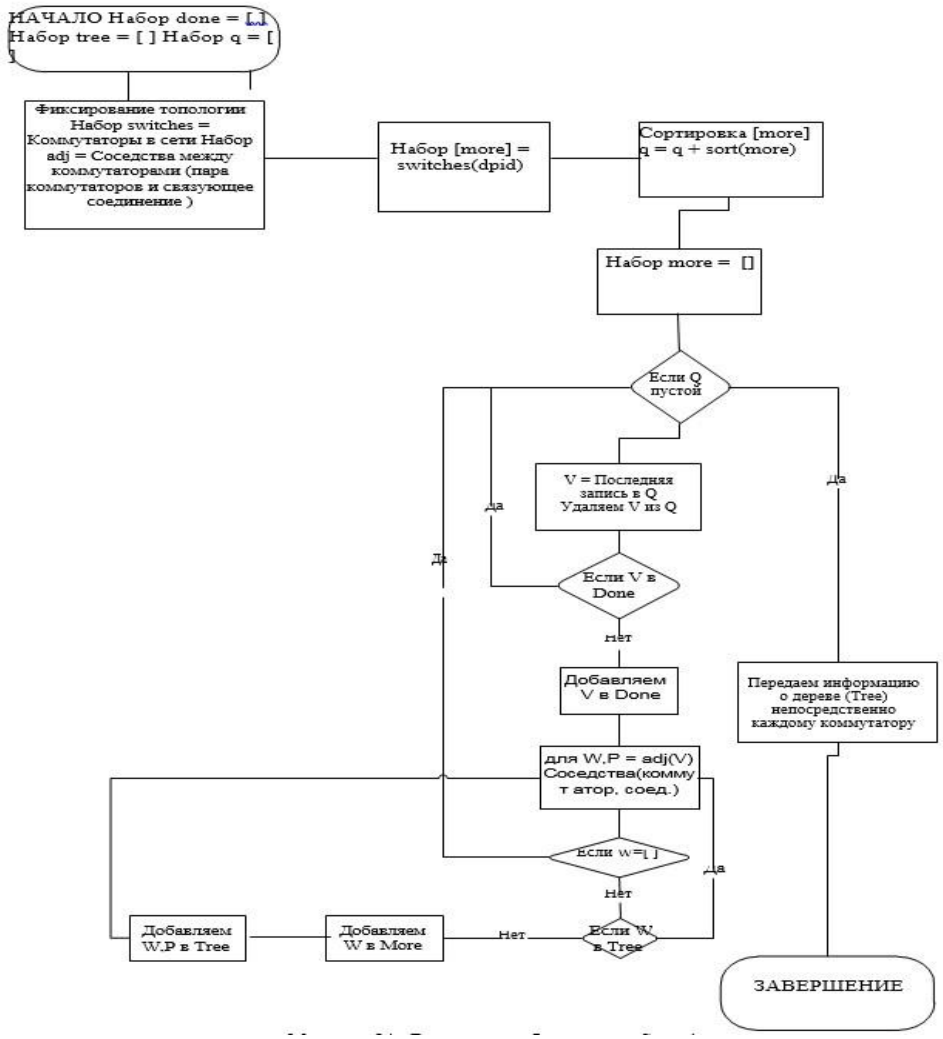


Рисунок 2.9 - Блок схема роботи модуля Spanning tree.

Модуль spanning-tree виконує функцію, яка не має прямого впливу на роботу комутаторів. Цей модуль служить додатковим елементом до модулів, які безпосередньо вирішують, куди направити пакети в мережі. Один з таких модулів - forwarding.l2_learning, який ми розглядали раніше. Модуль forwarding.l2_learning використовує модуль spanning_tree для того, щоб керувати швидкісним розподілом пакетів широкомовного трафіку лише через ті порти, які належать до вирахованого дерева.

Цей механізм спрямований лише на уникнення виникнення петель у мережі і не передбачає можливості використання додаткових резервних маршрутів для

розподілу трафіку. Щоб вирішити це обмеження, були розроблені додаткові механізми, які враховують наявність додаткових маршрутів для розподілу трафіку. Вони базуються на ідеї розрахунку не одного, а кількох дерев "spanning-tree" у мережі. В кожному з цих дерев корінь вибирається з різних комутаторів, а не лише одного, наприклад, комутатора з найменшим ідентифікатором dpid. Таким чином, розраховується не одне дерево, а кілька (N, де N - кількість комутаторів у мережі), кожне з яких має свій власний корінь і спрямоване на запобігання петель. Ці версії дерев розраховуються при підключенні контролера до мережі і оновлюються при будь-яких змінах в мережі.

Кількість дерев розповсюдження в з'єднаному графі, позначена як t(G), є важливим інваріантом, який добре досліджено. Наприклад, у випадку, коли G є деревом, t(G) = 1, а в циклічному графі Cn з n вершинами, t(G) = n. Згідно з формулою Кейлі, для повного графа з Kn вершинами виконується рівність:

$$t(K_n) = nn - 2. \quad (2.1)$$

У нашому випадку, коли застосовується алгоритм Breadth First Traversal, кількість використовуваних дерев не зростає експоненційно з кількістю комутаторів у мережі, а збігається з нею. Тобто:

$$T(G)\ n. \quad (2.2)$$

Після обчислення дерев дані передаються до модуля l2_learning, який також був модифікований для роботи з новим алгоритмом. У цій модифікації, замість комутації пакетів на основі MAC-адрес, контролер збирає інформацію про MAC-адреси, IP-адреси і порти TCP та UDP джерела та призначення, а комутатор пересилає пакети відповідно до цих правил. Кожен потік класифікується на основі цих параметрів. Наприклад, якщо два потоки мають різні MAC-адреси джерела

або різні порти TCP/UDP призначення, вони розглядаються як різні потоки. Блок-схема оптимізованого модуля L2 learning показана на рисунку 2.10.

Балансування здійснюється на основі класифікації потоків. Кожен раз, коли комутатор отримує пакет з параметрами, що не вказані в таблиці комутації, для цього пакета визначається дерево spanning-tree, відповідно до якого відбувається передача пакетів у цьому потоці. Таким чином, оскільки дерева використовують різні шляхи для досягнення від одного вузла до іншого (з урахуванням наявності резервних шляхів), трафік розподіляється по всій мережі з певним рівнем однорідності.

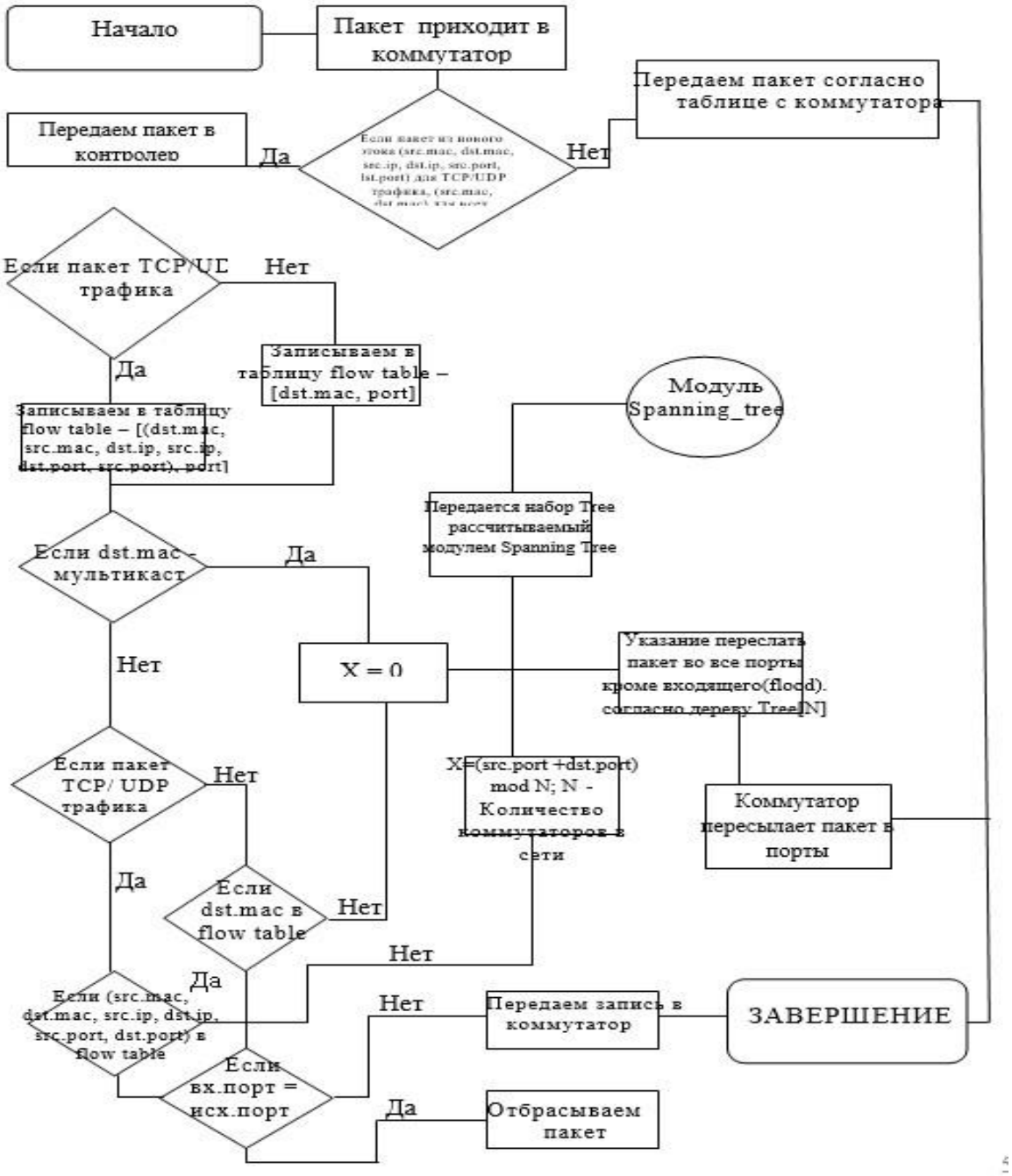


Рисунок 2.10 - Оптимізований модуль l2_learning

Блок схема роботи оптимізованого модуля Spanning tree на рисунку 2.11.

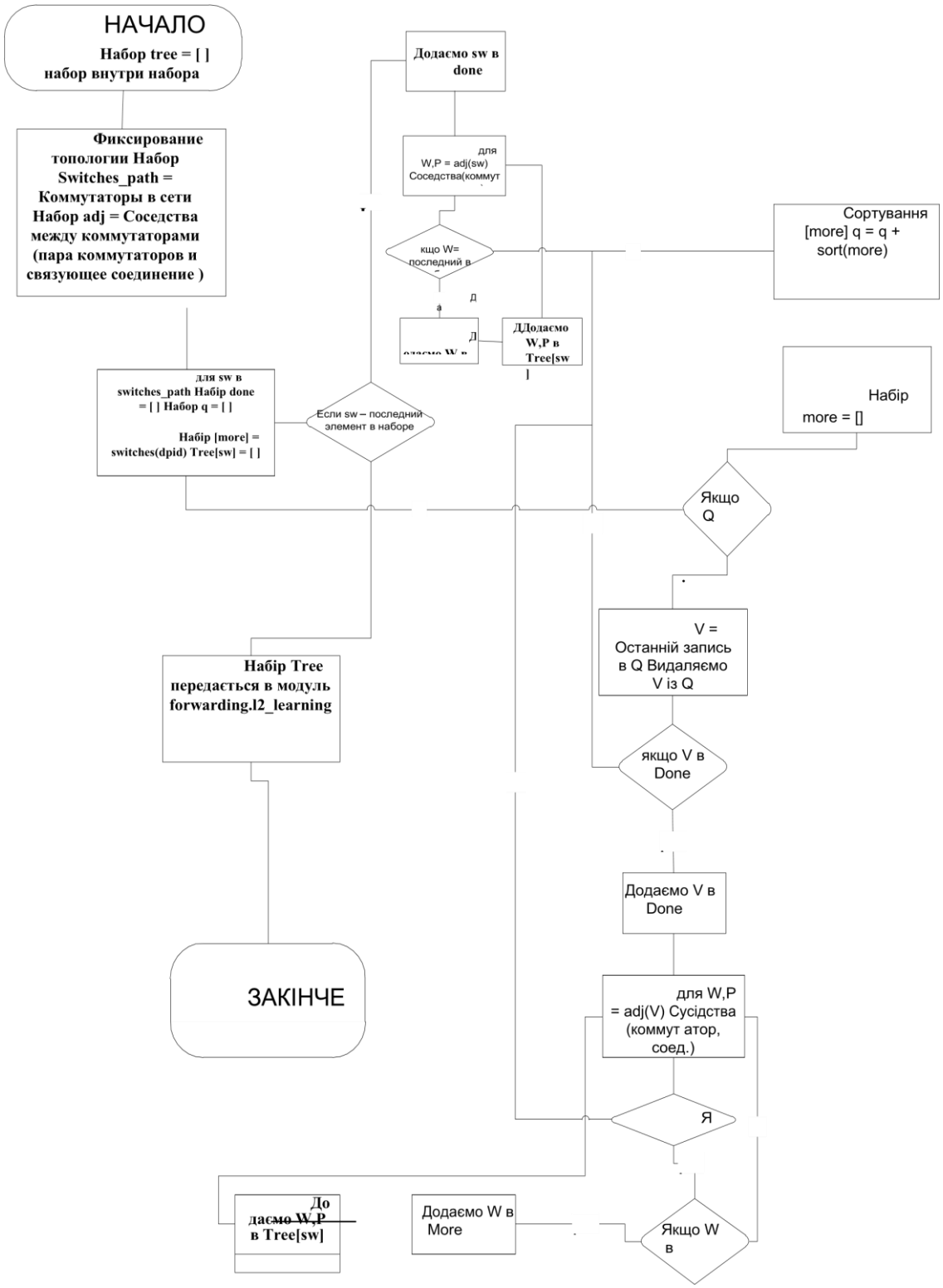


Рисунок 2.11 - Оптимізований модуль Spanning tree

Ефективність оптимізованих модулів залежить від наявності додаткових шляхів між точками маршрутизації трафіку. Для повно-мережі G з N комутаторів кількість шляхів P між будь-якою парою точок обчислюється за формулою:

$$(GN) = N - 1. \qquad (2.3)$$

Ця залежність може бути легко доведена, оскільки між будь-якою парою точок A і B існує пряме з'єднання, а також по одному резервному шляху для кожної з точок C, Року та Польщі через A-C-B, де ці шляхи не перетинаються. Кількість таких точок в графі становить N-2. Отже, за умови прямого з'єднання, кількість шляхів між двома точками дорівнює N-1. Оптимізований модуль, описаний у даній роботі, використовує всі ці шляхи, оскільки для повно-мережі шлях між точками A та B, при обраному корені C, пройде через A-C-B.

Отже, з урахуванням вищезазначених розрахунків, для повної мережі з N комутаторів, передбачуване збільшення корисного трафіку I в оптимізованих модулях, порівняно з існуючим модулем, буде:

$$I(G_N) = ((N - 1) - 1) \cdot 100\% = (N - 2) \cdot 100\%. \qquad (2.4)$$

2.4 Оцінка існуючих модулів конролера SDN

Контролер OpenDaylight є багатофункціональною платформою, що підтримує різноманітні модулі для управління мережею. Ці модулі забезпечують виконання різних завдань, таких як маршрутизація, балансування навантаження, забезпечення якості обслуговування (QoS), безпека тощо. Оцінка існуючих модулів контролера дозволяє визначити їх функціональні можливості, продуктивність та доцільність використання в конкретних мережевих сценаріях.

Основні модулі контролера OpenDaylight

1. L2-Switch. Модуль, що забезпечує функціональність комутатора другого рівня (L2). Він підтримує основні функції комутації кадрів Ethernet, такі як

навчання MAC-адрес, створення та підтримка таблиць MAC-адрес, а також пересилання кадрів на відповідні порти.

2. ODL L3. Модуль для маршрутизації на третьому рівні (L3), який забезпечує функції маршрутизації IP-пакетів між різними підмережами. Він підтримує динамічну маршрутизацію, використовуючи протоколи OSPF і BGP, а також статичну маршрутизацію.
3. OpenFlow Plugin. Модуль, який забезпечує підтримку протоколу OpenFlow, що є основним стандартом для взаємодії між контролером SDN та мережевими пристроями. Цей модуль дозволяє контролеру управляти комутаторами та маршрутизаторами, що підтримують OpenFlow.
4. Group Based Policy (GBP). Модуль, що забезпечує управління політиками на основі груп. Це дозволяє адміністратору мережі визначати та застосовувати політики безпеки, QoS та інші політики на основі груп користувачів або пристроїв.
5. DLUX. Веб-інтерфейс для візуалізації та управління мережею через контролер OpenDaylight. Він забезпечує зручний інтерфейс для налаштування та моніторингу мережевих ресурсів та політик.
6. YANG Tools. Модуль, що забезпечує підтримку моделювання даних за допомогою мови YANG. Це дозволяє створювати та використовувати моделі даних для управління конфігурацією мережевих пристроїв та сервісів.
7. MD-SAL (Model-Driven Service Abstraction Layer). Абстракційний шар сервісів, що забезпечує інтерфейси для взаємодії між різними компонентами та сервісами контролера OpenDaylight. Це дозволяє розробникам створювати нові сервіси, використовуючи модель керованого підходу (таб. 2.4).

Таблиця 2.4 - Основні модулі контролера OpenDaylight

Модуль	Опис
L2-Switch	Комутація другого рівня, навчання MAC-адрес, створення та підтримка таблиць MAC-адрес.
ODL L3	Маршрутизація третього рівня, підтримка OSPF, BGP та статичної маршрутизації.
OpenFlow Plugin	Підтримка протоколу OpenFlow, управління комутаторами та маршрутизаторами.
Group Based Policy (GBP)	Управління політиками на основі груп користувачів або пристроїв.
DLUX	Веб-інтерфейс для візуалізації та управління мережею через контролер OpenDaylight.
YANG Tools	Підтримка моделювання даних за допомогою мови YANG.
MD-SAL	Абстракційний шар сервісів для взаємодії між компонентами та сервісами контролера.

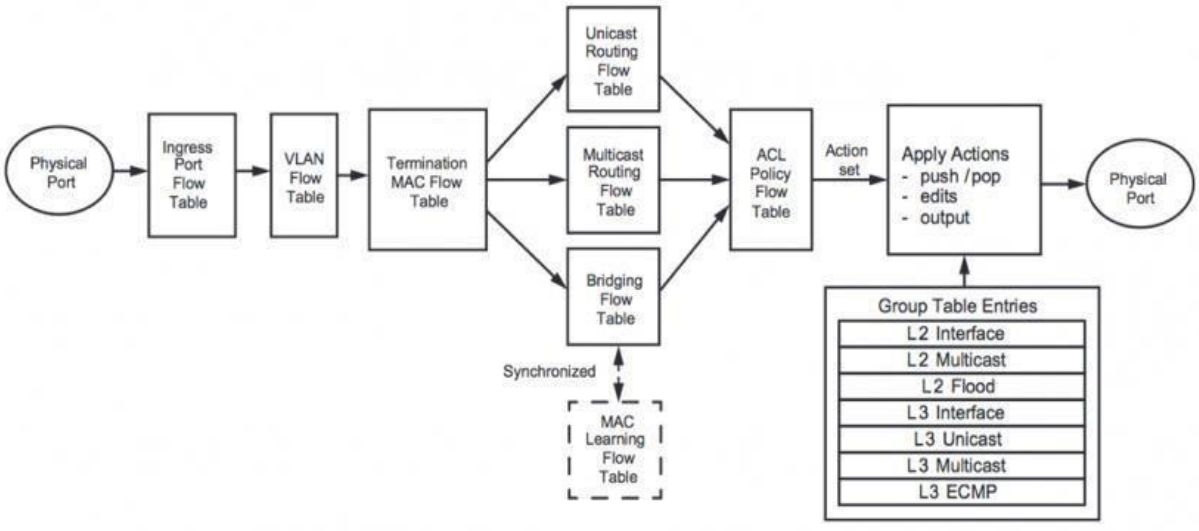


Рисунок 2.12. Пайплайн для реалізації маршрутизації та мережевого мосту.

Оцінка функціональних можливостей модулів

1. L2-Switch

Модуль L2-Switch забезпечує базові функції комутації кадрів Ethernet. Він дозволяє ефективно управляти трафіком на рівні другого рівня мережевої моделі OSI, що є критично важливим для локальних мереж (LAN). Модуль підтримує динамічне навчання MAC-адрес, що забезпечує оптимальне пересилання кадрів у мережі.

2. ODL L3

Модуль ODL L3 забезпечує функції маршрутизації на третьому рівні. Він підтримує динамічну маршрутизацію, використовуючи протоколи OSPF і BGP, що дозволяє автоматично адаптуватися до змін у мережевій топології. Крім того, модуль підтримує статичну маршрутизацію, що дозволяє налаштовувати фіксовані маршрути для специфічних сценаріїв.

3. OpenFlow Plugin

Модуль OpenFlow Plugin є ключовим компонентом для підтримки SDN. Він забезпечує взаємодію між контролером і мережевими пристроями, що підтримують OpenFlow. Це дозволяє контролеру динамічно управляти правилами пересилання трафіку, що забезпечує високу гнучкість та адаптивність мережі.

4. Group Based Policy (GBP)

Модуль GBP дозволяє адміністратору мережі визначати та застосовувати політики на основі груп. Це забезпечує можливість централізованого управління безпекою та якістю обслуговування (QoS) у мережі. Політики можуть бути налаштовані для різних груп користувачів або пристроїв, що підвищує рівень безпеки та ефективність використання мережевих ресурсів.

5. DLUX

Модуль DLUX надає зручний веб-інтерфейс для управління мережею. Він дозволяє візуалізувати мережеві топології, моніторити стан мережевих пристроїв та застосовувати налаштування. Це спрощує процес управління мережею та забезпечує швидкий доступ до необхідної інформації.

6. YANG Tools

Модуль YANG Tools забезпечує підтримку моделювання даних за допомогою мови YANG. Це дозволяє створювати та використовувати моделі даних для управління конфігурацією мережевих пристроїв та сервісів. Підтримка YANG є важливою для забезпечення сумісності з різними мережевими пристроями та стандартами.

7. MD-SAL

Модуль MD-SAL забезпечує абстракційний шар сервісів для взаємодії між різними компонентами та сервісами контролера OpenDaylight. Це дозволяє розробникам створювати нові сервіси, використовуючи модель керованого підходу, що спрощує процес розробки та інтеграції нових функцій (таб. 2.5).

Таблиця 2.5 - Оцінка функціональних можливостей модулів OpenDaylight

Модуль	Функціональні можливості
L2-Switch	Комутація другого рівня, навчання MAC-адрес, пересилання кадрів
ODL L3	Маршрутизація третього рівня, підтримка OSPF, BGP, статична маршрутизація
OpenFlow Plugin	Взаємодія між контролером і пристроями OpenFlow, управління правилами пересилання
Group Based Policy (GBP)	Управління політиками безпеки та QoS на основі груп
DLUX	Веб-інтерфейс для візуалізації та управління мережею
YANG Tools	Моделювання даних за допомогою мови YANG
MD-SAL	Абстракційний шар сервісів для взаємодії між компонентами та сервісами

Контролер OpenDaylight забезпечує широкий спектр функціональних можливостей через свої модулі, що дозволяє ефективно управляти мережею та забезпечувати високу гнучкість та продуктивність. Модулі L2-Switch та ODL L3

забезпечують базові функції комутації та маршрутизації, OpenFlow Plugin забезпечує інтеграцію з мережевими пристроями, що підтримують OpenFlow, а модулі GBP, DLUX, YANG Tools та MD-SAL забезпечують додаткові можливості для управління політиками, візуалізації, моделювання даних та інтеграції нових сервісів. Це робить OpenDaylight потужним та універсальним інструментом для побудови та управління SDN-мережами.

2.5 Оцінка розроблених механізмів

Для перевірки ефективності нових алгоритмів були проведені наступні експерименти:

Експеримент 1: Створено мережу з трьох комутаторів, де кожен з них має два альтернативних шляхи для передачі пакетів. Схема мережі показана на рисунку 2.13. Для віртуальної пари машин h1-h2 запускається утиліта iperf для передачі TCP трафіку. Запускаються три окремі сесії TCP з використанням різних портів TCP джерела, при цьому IP/MAC адреси залишаються незмінними.

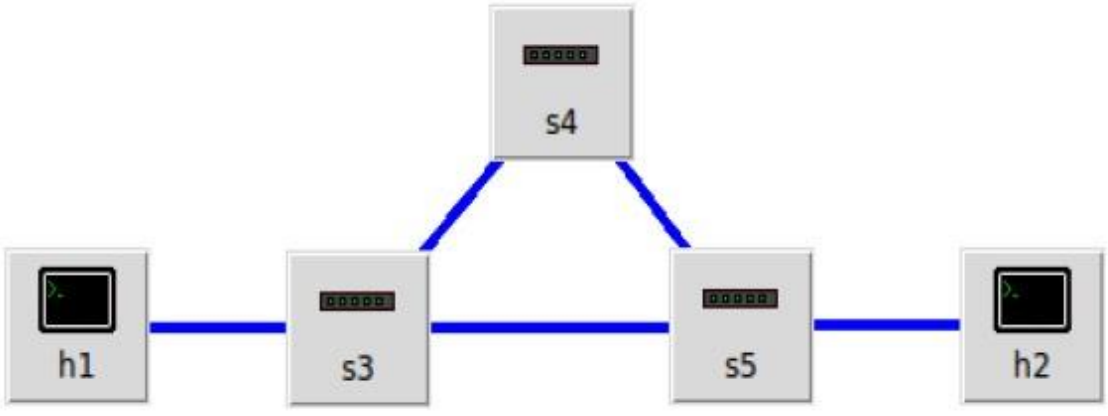


Рисунок 2.13 - Експериментальна мережа з 3-х комутаторів

Були протестовані алгоритми spanning-tree у вищезгаданій мережі до та після внесення додаткових змін. Загальний корисний трафік був обчислений для кожного варіанту. Для отримання результатів було проведено 30 вимірювань для

кожного випадку, при цьому кожна зміна застосовувалася протягом 10 секунд. Результати вимірювань наведені на рисунку 2.14.

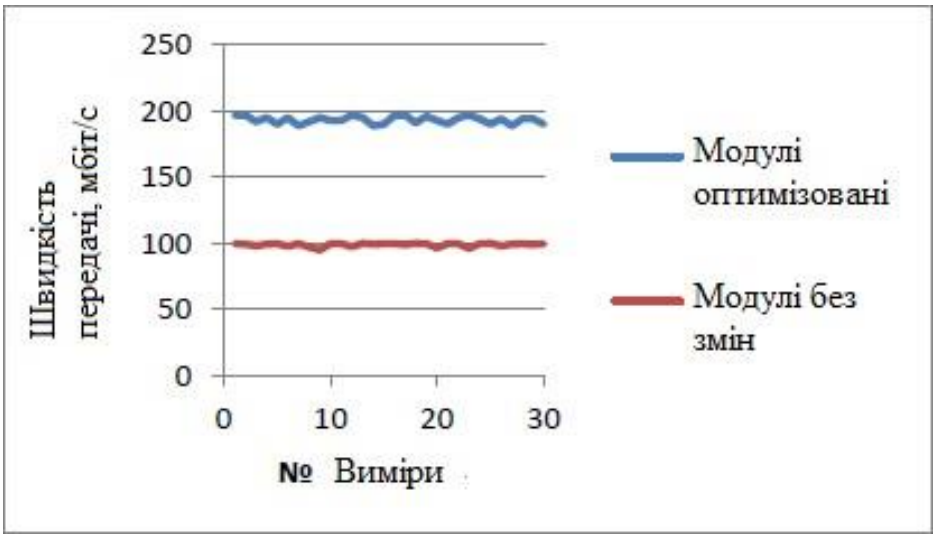


Рисунок 2.14 - Результати порівняння модулів spanning-tree в мережі з 3 комутаторів

Експеримент 2:

Аналогічний експеримент проведений на мережі з 4 комутаторами. На рисунку 2.15 відображена мережа, на якій проводився експеримент.

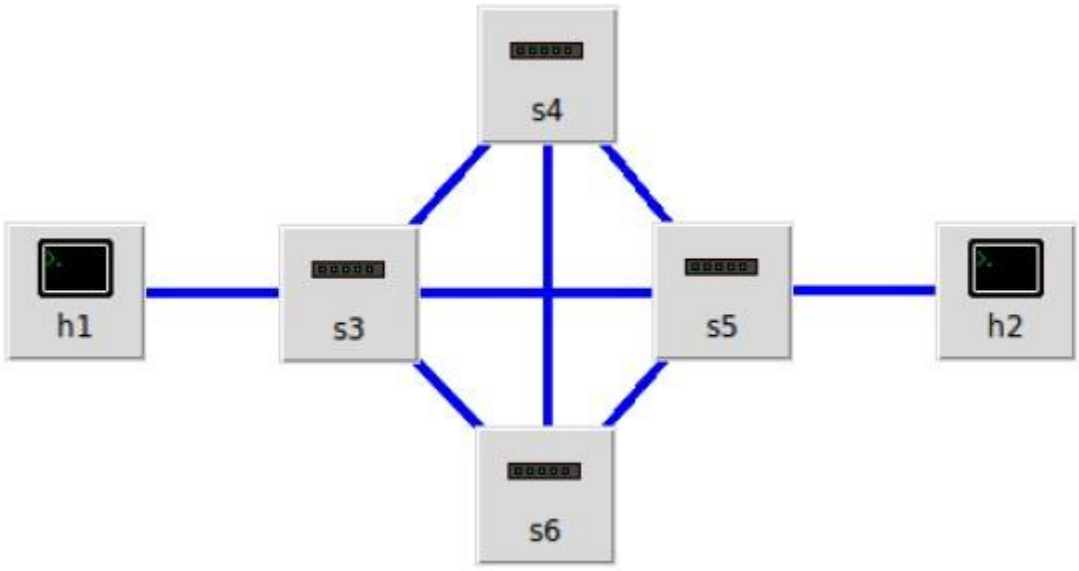


Рисунок 2.15 - Експериментальна мережа з 4-х комутаторів

Результати вимірювань наведені в рисунку 2.16.

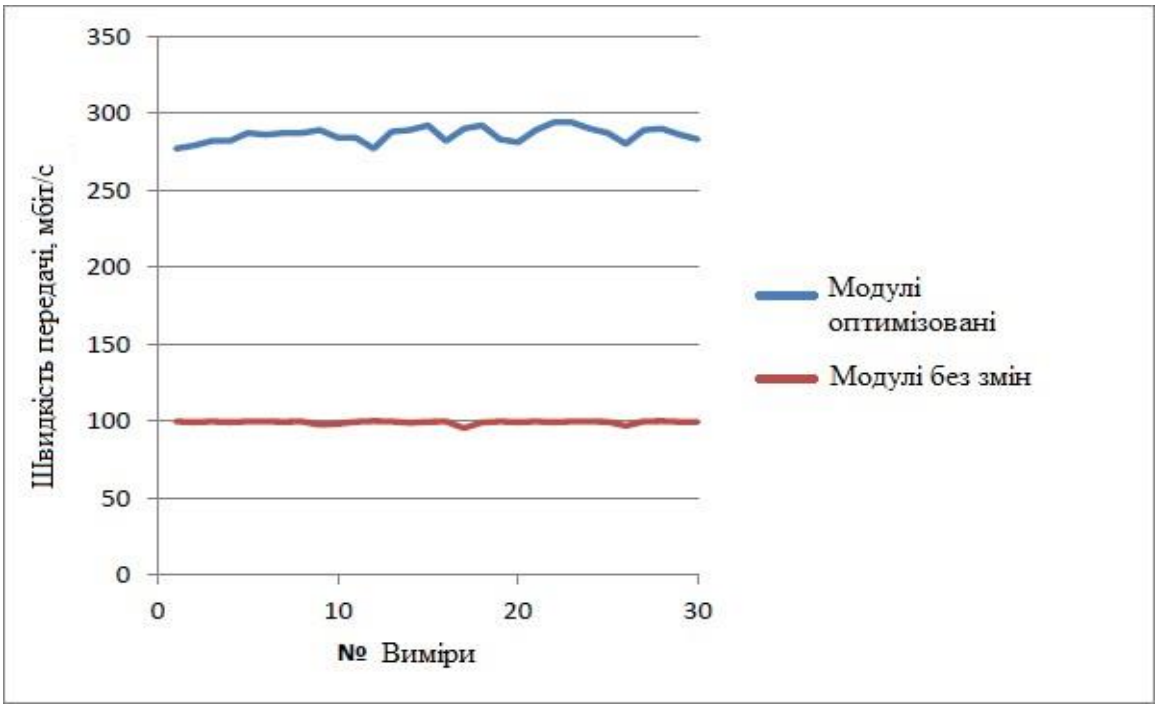


Рисунок 2.16 - Результати порівняння модулів spanning-tree в мережі з 4 комутаторів.

Експеримент 3:

Експеримент проведено на мережі з 6 комутаторами. На рисунку 2.17 відображена мережу, на якій проводився експеримент.

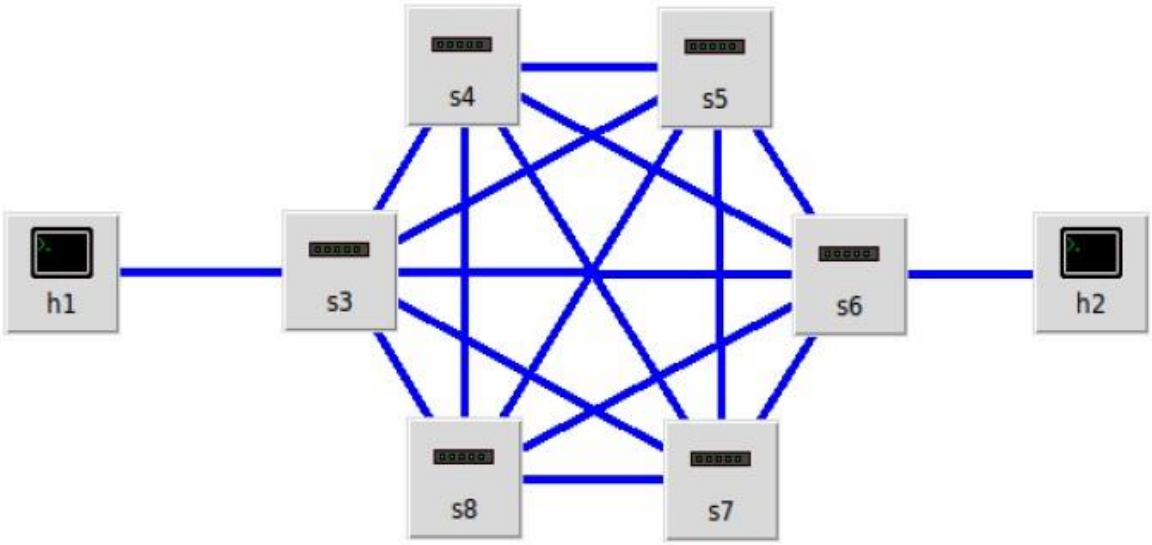


Рисунок 2.17 - Експериментальна мережу з 6 комутаторів

Результати вимірювань наведені в рисунку 2.18.

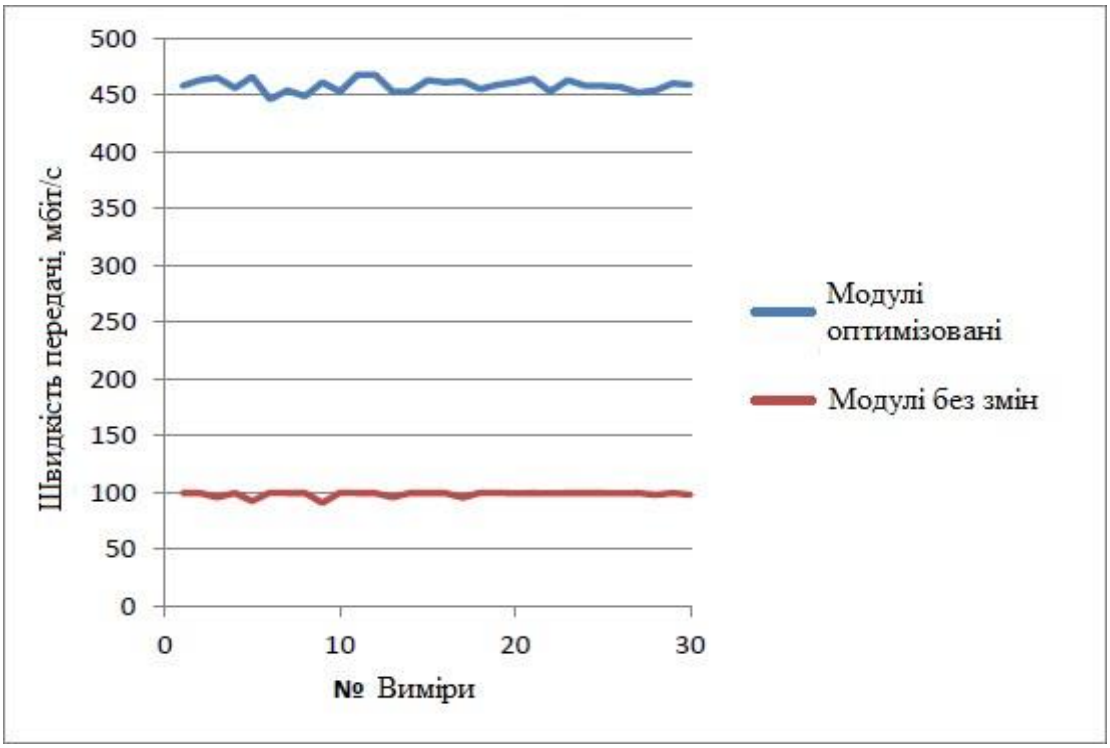


Рисунок 2.18 - Результати порівняння модулів spanning-tree в мережі з 6 комутаторів

2.6 Результати проведених експериментів

Для оцінки середнього значення пропускної здатності та відхилень у вимірюваннях, застосовувалися наступні формули:

$$m_x = \frac{\sum_{i=1}^{n} x_i}{n} \quad , \tag{2.5}$$

, (2.6)

де n - кількість проведених замірів, x_i- результат і-го виміру,

m_x - середнє значення отриманих результатів, ∂_n- середньоквадратичне відхилення.

Результати проведених експериментів показують, що в результаті балансування в мережі з 3 комутаторів, математичне очікування обсягу корисного трафіку становить: Для неоптимізованого модуля: використовуючи (3.5) і (3.6), отримуємо m_x,перв = 98,8Мбіт с при среднеквадратичном відхиленні ∂_n = 1,2 Мбіт/с;

Для оптимізованого модуля: m_x = 193,5 Мбіт/с при среднеквадратичном відхиленні ∂_n = 2,7 Мбіт/с;

Збільшення I математичного очікування обьема корисного трафіку становить 95,9%, при очікуваному I = (3-2) * 100% = 100%.

У мережі з 4 комутаторів: $m_{x,\text{перв}}$ = 98,9 Мбіт/с при среднеквадратичном відхиленні ∂_n = 1,2 Мбіт/с; $m_{x,\text{оптим}}$ = 286 Мбіт/с при среднеквадратичном відхиленні ∂_n = 4,6 Мбіт/с; Збільшення математичного очікування обьема корисного трафіку становить 189,1%, при очікуваному I = (4-2) * 100% = 200%.

У мережі з 6 комутаторів: $m_{x,\text{перв}}$ = 98,4 Мбіт/с при среднеквадратичном відхиленні ∂_{n} = 2,2 Мбіт/с; $m_{x,\text{оптим}}$ = 458 Мбіт/с при среднеквадратичном відхиленні ∂_n = 5,4 Мбіт/с; Збільшення математичного очікування обьема корисного трафіку становить 365,4%, при очікуваному I = (6-2) * 100% = 400%.

Отримані результати за проведеними експериментів показують, що фактичне збільшення пропускної здатності можна порівняти з прогнозованим.

Похибки, отримані в результататі, пояснюються обмеженням обробної здатності комутаторів і самих термінальних пристроїв, які фактично є віртуальними машинами, що використовують один процесорний ресурс - CPU локальної машини.

3 АРХІТЕКТУРА НА ОСНОВІ SDN ДЛЯ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ОБСЛУГОВУВАННЯ ВИСОКОПРОДУКТИВНИХ РОЗПОДІЛЕНИХ ДОДАТКІВ

Ми спостерігаємо стале зростання числа додатків, які мають високі вимоги до якості обслуговування (QoS). Онлайн системи охорони здоров'я, потокова передача мультимедіа, інтерактивні програми в реальному часі та паралельна обробка даних у центрах обробки даних є лише кількома прикладами платформ, які потребують гарантованої якості обслуговування. Забезпечення таких гарантій залишається складною задачею для сучасних мереж передачі даних. Більшість наявних мереж не мають достатньої гнучкості та ресурсів, що робить їх непридатними для високопродуктивних розподілених систем. Трансформація таких мереж в більш адаптивні структури є важливою для покращення ефективності цих додатків та забезпечення кращої якості користувацького досвіду (QoE).

У традиційних мережах можливості визначення та забезпечення якості обслуговування (QoS) обмежені через складність управління цими середовищами. Модифікація існуючої інфраструктури потребує значних адміністративних зусиль, оскільки кожен компонент сервісу вимагає власних ресурсів, керованих централізованою системою, щоб забезпечити необхідний рівень якості обслуговування для користувачів. У противагу цьому, програмно-визначені мережі (SDN) представляють нову парадигму, яка спрощує управління мережею та сприяє впровадженню нових абстракцій, що полегшують її еволюцію.

Однією з ключових особливостей SDN є розділення рівнів даних та управління в мережевих пристроях (комутаторах та маршрутизаторах). Мережевий інтелект переноситься до логічно централізованого контролера, який за допомогою безпечних комунікаційних протоколів, таких як OpenFlow, керує пристроями пересилання даних. Ця модель забезпечує глобальний огляд мережі та підвищений рівень програмованості. Завдяки цим можливостям контролер

може надавати прості, але ефективні механізми для забезпечення якості обслуговування. Наприклад, контролер може управляти трафіком, забезпечуючи для критичних додатків мінімальну затримку, низький джиттер або достатню пропускну здатність.

3.1 QoS та його можливості

Оцінка якості обслуговування (QoS) є необхідною складовою визначення ефективності впровадження нових рішень при розробці або покращенні мережевих інфраструктур.

Згідно з рекомендаціями ITU-T E.800, якість обслуговування (QoS) є концепцією, що визначає ступінь задоволення користувача наданою йому послугою зв’язку. Це поняття подальше уточнення в Рекомендації E.860: "Якість обслуговування - це ступінь відповідності обслуговування, наданого користувачеві постачальником, угоді між ними". Це надає додаткового значення угодам про рівень обслуговування (Service Level Agreement, SLA) між користувачами та постачальниками послуг. Компанією Cisco було введене таке визначення терміна якість обслуговування: "здатність мережі забезпечити необхідний рівень сервісу для заданого трафіку в рамках певних технологічних протоколів (Frame Relay, ATM, Ethernet і 802.1 мережі, SONET і IP-мережі)".

Відповідно до RFC 2475, під терміном "сервіс" мається на увазі набір характеристик передачі пакетів в одному напрямку по одному або декількох мережних маршрутах.

QoS має визначальне значення для управління трафіком у сучасних мережах, що базуються на пакетах, і включає такі можливості:

- Класифікація додатків з призначенням пріоритетів та диференціювання трафіку за протоколом, адресою та номером порту.
- Профілювання мережного трафіку.

- Обмеження (за необхідності) інтенсивності трафіку, який надходить від користувачів.
- Управління чергами з встановленням пріоритету обробки пакетів на мережних вузлах.
- Маршрутизація мережного трафіку.
- Читання та запис вимог щодо поведінки QoS в заголовку пакета.
- Контроль перевантаження, щоб пристрій надсилав трафік найвищого пріоритету на основі пріоритетів планувальника.

Контроль втрати пакетів за допомогою алгоритмів випадкового раннього виявлення (RED), щоб пристрій міг визначати пакети, які потрібно скинути або обробити.

3.2 Загальні фази надання QoS

Набір вимог до якості обслуговування (QoS) є змінною, яка варіюється залежно від конкретних сервісів і, як правило, є функцією типу програми.

Два основні принципи, пов'язані з наданням QoS, можуть бути виокремлені:

1. Специфікація вимог користувача.
2. Забезпечення механізмів, що відповідають вимогам користувачів.

Зазвичай вимоги визначаються з точки зору максимальної затримки, статистичних змін затримки (джиттера), пропускної здатності, частоти помилок і втрат. Забезпечення механізмів, які гарантують такі вимоги, означає, що зазначені параметри гарантуються наскрізь у додатках під час їх виконання.

На основі цих принципів можна визначити чотири загальні фази надання QoS:

1. Запит на послугу.
2. Укладання угод про надання послуг.
3. Підтримання угод про надання послуг.
4. Завершення угод про надання послуг.

Під час етапу запиту на обслуговування можуть використовуватися статичні механізми, які можуть створювати значне адміністративне навантаження, або динамічні, такі як протоколи сигналізації. Після того, як зроблений запит, укладається неявна угода між програмою та мережею. Потім мережа забезпечує транспортування потоків програми, доки програма генерує трафік відповідно до того, що було вказано на запит. Ця угода підтримується відповідними механізмами до моменту її завершення, яке також може відбуватися статичним або динамічним способом при звільненні зарезервованих ресурсів.

1. Враховуючи ці принципи та етапи, необхідні функціональні можливості для загальної системи надання QoS можуть бути наступними:

1. **Параметризація послуг** - визначення параметрів та категорій послуг.
2. **Механізми розподілу адрес, класифікації та планування розподілу ресурсів** - забезпечення механізмів для ефективного розподілу адрес, класифікації та планування використання ресурсів.
3. **Оркестровка ресурсів** - координація функцій узгодження, контролю доступу та налаштування ресурсів.
4. **Адаптація послуг** - зміна поведінки вищезгаданих механізмів і функцій з урахуванням змін в середовищі.

2. Конфігурація функцій та механізмів забезпечення QoS в кожному елементі мережі у звичайних мережах передачі даних призводить до значних адміністративних витрат. З іншого боку, SDN спрощує управління цими мережами. На перший погляд, SDN відноситься до мережевої архітектури, побудованої на чотирьох стовпах:

1. **Роз’єднання планів даних і управління**.
2. **Прийняття рішень на основі потоків**.
3. **Централізація логіки управління** у логічно централізованому зовнішньому контролері.
4. **Мережеве програмування** за допомогою програмних додатків, що працюють на верхній частині моделі.

3. На рисунку 3.1 зображена типова архітектура SDN.

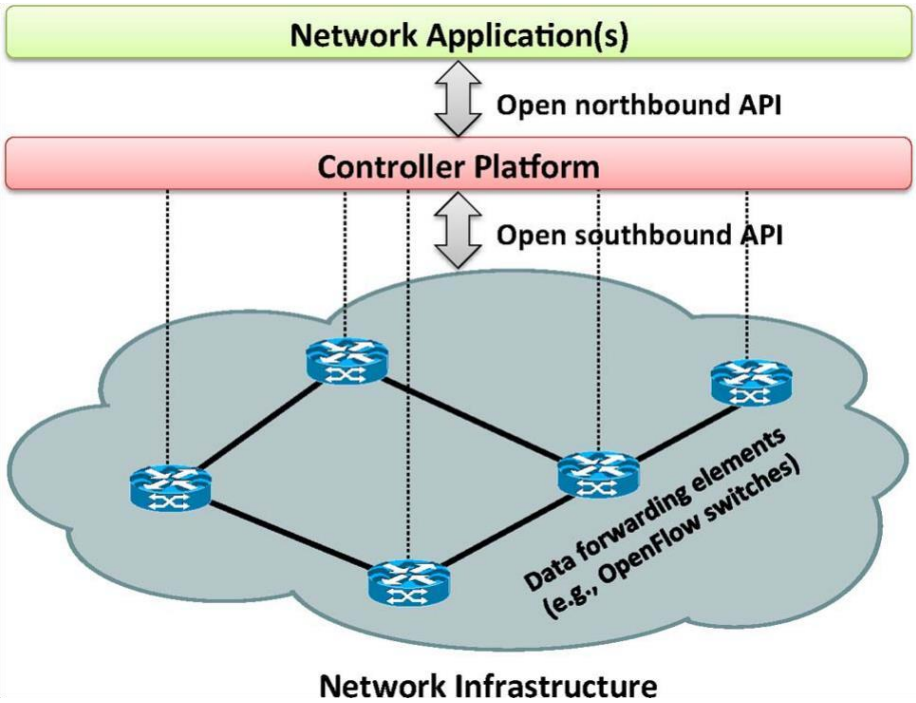


Рисунок 3.1– Спрощений вигляд архітектури SDN [25].

Інтерфейс між контролером SDN та пристроями мережевої інфраструктури (південний API) забезпечується за допомогою безпечних протоколів, таких як OpenFlow [3]. На відміну від північного API, який ще не має стандартів, OpenFlow вважається стандартом де-факто. Отже, в типовому середовищі SDN елементи пересилання даних (комутатори та маршрутизатори) активовані для роботи з OpenFlow, тобто їх можна керувати за допомогою контролера, сумісного з OpenFlow.

Починаючи з версії 1.0, OpenFlow API має вбудовані функції, які сприяють розробці механізмів надання QoS. Це дозволяє легко реалізувати розподіл пакетів, класифікацію та планування. Завдяки деталізації обробки окремих потоків і глобальному погляду на SDN, можна також надавати механізми узгодження QoS, контролю доступу та налаштування. Ці можливості використовуються в рішеннях QoS, які використовують перемаршрутизацію пакетів і балансування трафіку чутливих потоків. Однак, в багатьох випадках ці методи не реалізовані. Наприклад, політика маршрутизації, застосована до мережі, може перешкоджати

встановленню альтернативних шляхів. У інших випадках доступні маршрути можуть бути обмеженими або перевантаженими. У цьому відношенні методи формування трафіку через наскрізне резервування пропускної здатності та обмеження швидкості виявляються простими та ефективними рішеннями для забезпечення QoS, які можуть гарантувати вимоги QoS для програм.

3.3 Схема архітектури надання QoS на основі архітектури SDN

Запропонована архітектура враховує принципи, фази та функціональні можливості, що були зазначені раніше, і безпосередньо впливає на проектування компонентів та розробку прототипів, що використовуються при оцінці. Проте важливо зауважити, що розробка цієї архітектури обмежується забезпеченням рівнів якості в підсистемі мережі зв'язку, не враховуючи забезпечення QoS на кінцевих хостах, таких як буфери введення/виведення, процеси операційної системи, стек протоколів і т. д., як запропоновано. У високорівневому представленні запропонованої архітектури, яка зображена на Рис. 3.2, ми бачимо, що вона складається з різних модулів, що пов'язані з об'єктами зв'язку, такими як запитувач і постачальник, модулі обміну повідомленнями, контролер SDN (QoS) та мережа адміністрування, спільно зі структурою даних.

Модуль запитувача відповідає за повторний запит на послуги з розподіленої програми до модуля постачальника. Останній, крім зберігання запитуваних даних, перевіряє можливість надання послуги з необхідними гарантіями якості, використовуючи структуру даних, яка містить ідентифікацію категорії послуги та її параметри.

Модуль адміністрування мережі відповідає за управління мережею та створення бази даних класів обслуговування, а також за початкову конфігурацію мережевих політик QoS, оскільки OpenFlow не має цієї можливості.

Модулі обміну повідомленнями відповідають за процедури запиту та підтвердження резервування ресурсів, тоді як модуль контролера забезпечує

контроль доступу до резервування та налаштування в мережі для забезпечення QoS.

У цій архітектурі розподілені програми мінімально модифіковані, що дозволяє їм запитувати певний рівень QoS до мережі без значного втручання розробників. Процес запиту та підтвердження резервування базується на протоколі RSVP, як показано на Рис. 3.2.

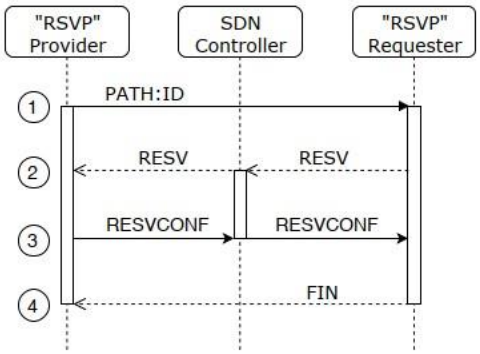


Рисунок 3.2 – Схема послідовності повідомлень про резервування.

Процедура контролюється контролером SDN, який переглядає пакети, що містять відповідні повідомлення, та впроваджує механізми надання QoS. Спочатку (i) процес на стороні постачальника надсилає повідомлення "PATH" процесу на стороні запитувача разом з числовою ідентифікацією, що відповідає класу послуги, яка буде надана. Під час обробки кожного пакета, що містить це повідомлення, підпрограми модуля контролера встановлюють правила пересилання на комутатори/маршрутизатори, а потім встановлюють наскрізний шлях.

Потім (ii) процес запитувача, отримавши повідомлення "PATH", розпізнає ідентифікатор класу та генерує повідомлення "RESV" (запит на резервування), вставляючи відносний номер до ідентифікатора в поле ToS заголовка пакета. Під час аналізу цього пакету в кожному елементі пересилання даних на маршруті між об'єктами, що спілкуються, підпрограми QoS модуля контролера виконують контроль допуску резервувань, обчислюючи в кожному порту каналу доступність

пропускної здатності, необхідної для цього класу послуги, шляхом вивчення поля ToS. Надходження повідомлення "RESV" на стороні постачальника вказує на те, що бронювання можливе на всьому маршруті.

Таким чином, (iii) процес постачальника генерує новий пакет із повідомленням підтвердження резервування, "RESV-CONF", з ідентично позначеним полем ToS. Коли модуль контролера аналізує такий пакет, він встановлює політики QoS на інтерфейсах пристроїв, через які мають передаватися конфіденційні потоки, а також налаштовує їх черги за правилами, що відповідають цим потокам (за допомогою директиви OpenFlow ENQUEUE). Це гарантує розподіл пакетів, класифікацію та планування.

Нарешті, (iv) надходження повідомлення "RESVCONF" до процесу запитувача підтверджує резервування ресурсу на шляху, що дозволяє надіслати повідомлення "FIN", закриваючи процедуру. З цього моменту провайдер починає надсилати потік даних, для якого буде гарантовано QoS.

У описаній вище логіці не враховано можливі винятки, такі як недоступність пропускної спроможності під час резервування. Випадки винятків обробляються через повідомлення про помилки та вичерпання резерву/шляху, як, наприклад, у протоколі RSVP. Крім того, не були враховані більш складні функції забезпечення QoS, такі як механізми налаштування та адаптація сервісу.

3.4 Доступ до мережі Інтернет

Доступ до Інтернету через центральний сайт є одним з основних сервісів (див. Рисунок 3.3). Суть цього сервісу полягає в тому, що певний трафік спочатку направляється на центральний сайт, де працює мережевий екран (Firewall), перед тим як його відправити до мережі Інтернет. Деякий інший трафік може бути направлений в мережу Інтернет напряму з локального сайту.

Це здійснюється таким чином:

- За допомогою протоколу NETCONF контролер налаштовує статичний маршрут за замовчуванням на маршрутизаторі центрального сайту, який направляє трафік на мережевий екран. На маршрутизаторах локальних сайтів контролер налаштовує два VRF: VRF1 та VRF2. VRF1 використовується для оверлейної мережі, а VRF2 – для андерлейної мережі Інтернет. Між цими VRF контролер налаштовує політику, що дозволяє перенаправляти трафік з одного VRF в інший.

- Маршрутизатор на центральному сайті оголошує цей статичний маршрут решті сайтів за допомогою протоколу динамічної маршрутизації через налаштовані тунелі.

Користувачі на сайтах отримують приватні IP-адреси та IP-адресу DNS-сервера від локальних маршрутизаторів. Трафік, який має бути переданий на центральний сайт, проходить через оверлейний тунель, а трафік, який має бути направлений до мережі Інтернет, фільтрується згідно з налаштованою політикою, переходить до VRF2 і передається через андерлейний тунель.

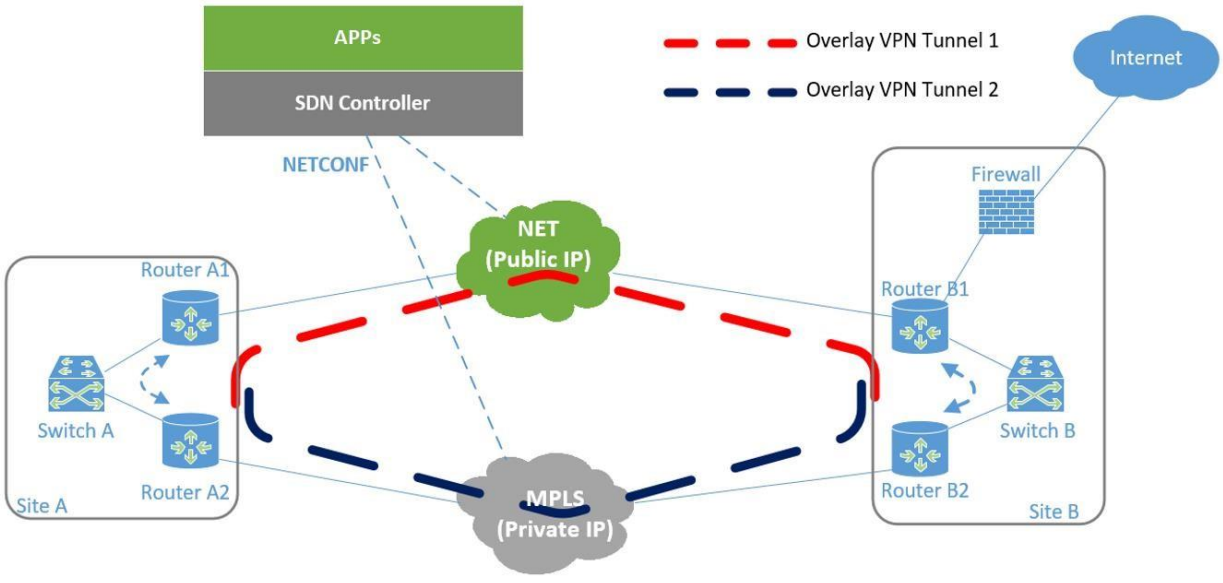


Рисунок 3.3 - Реалізація доступу до мережі Інтернет.

Архітектуру SDN можна інтегрувати в RAN-мережу оператора зв’язку (див. Рисунок 3.4). У такому випадку архітектура RAN буде містити наступні додаткові компоненти:

- API, які регулюють надання послуг і забезпечують взаємодію користувача з мережею;
- контролер, який забезпечує взаємодію з API за допомогою Restful/https/YANG інтерфейсів, а взаємодія з елементами мережі відбувається за допомогою протоколу NETCONF. Також можлива інтеграція існуючих систем моніторингу, тоді контролер може збирати інформацію про мережеві елементи за допомогою Restful протоколів з даних систем моніторингу;

NMS, яка збирає дані з елементів мережі за допомогою протоколу SNMP та може передавати їх контролеру за допомогою Restful протоколів.

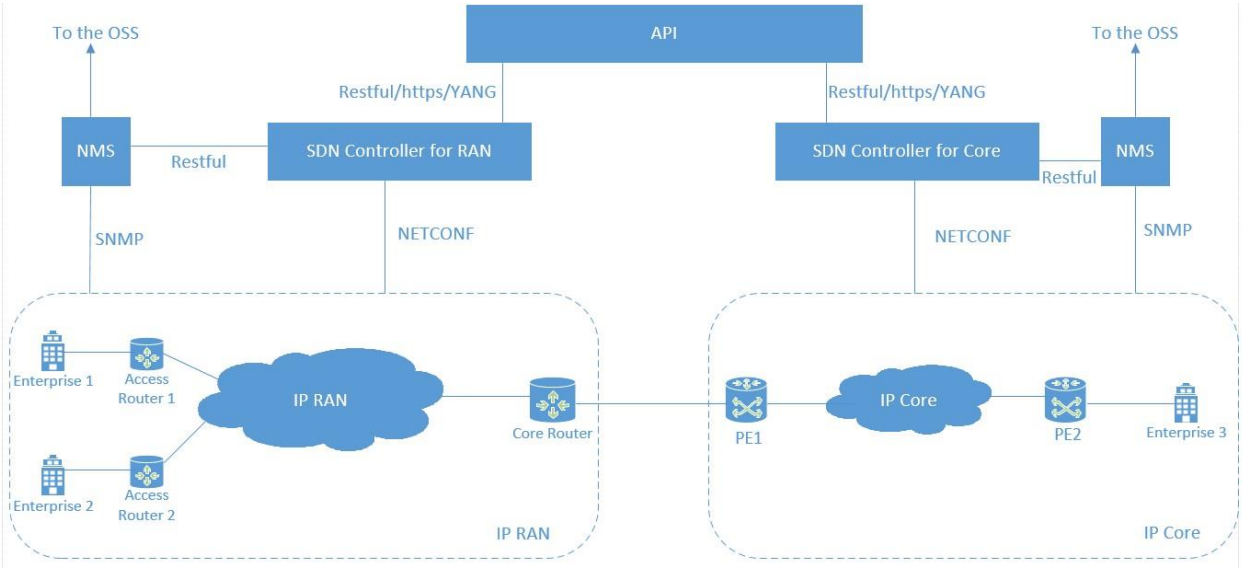


Рисунок 3.4 - Інтеграція доRAN-мережі оператора

Дана інтеграція не потребує кардинальних змін в структурі мережі та заміни існуючого мережевого обладнання.

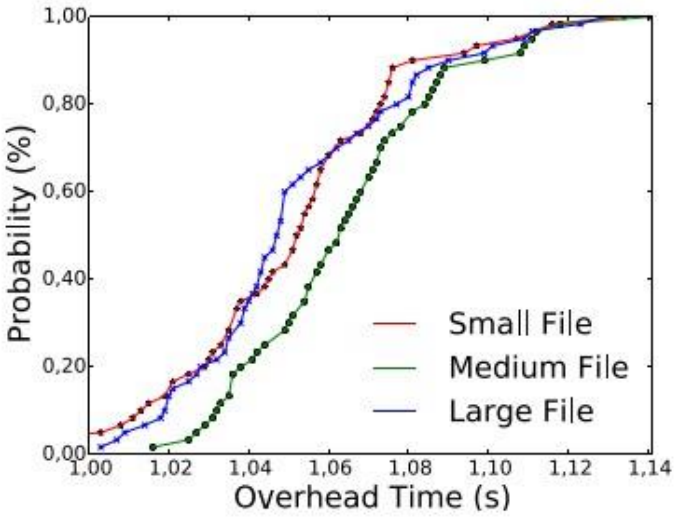
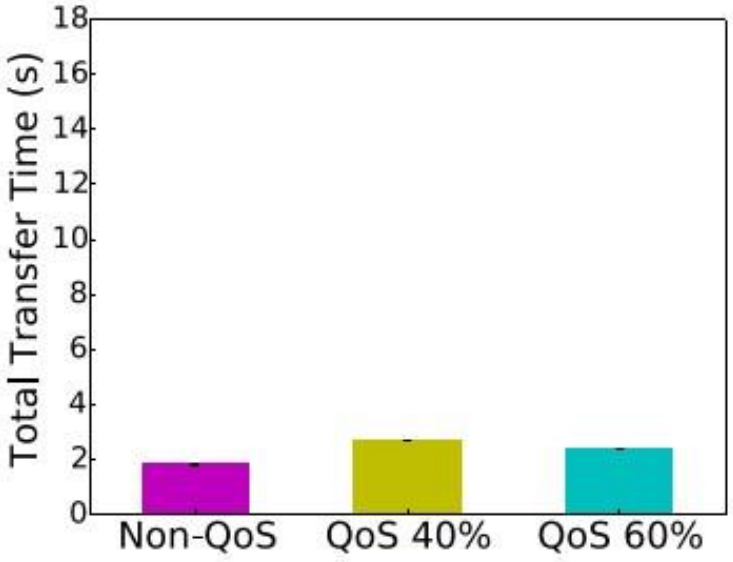


Рисунок 3.5 – Розподіл накладних витрат для кожного розміру файлу, що оцінюється.

Поміж самим процесом обміну повідомленнями на етапі запиту на резервування, цей захід також пов'язаний з кількістю мережевих переходів. Іншими словами, чим більше переходів, тим більші накладні витрати, оскільки процедури резервування, які контролює контролер, потребують налаштування кожного елемента мережі. Зрозуміло, що для конкретного сценарію моделювання ця середня величина має великий вплив, особливо на короткі передачі даних, які зазвичай мінімізуються в ситуаціях, коли передачі тривають довше. У цьому контексті можна припустити, що в середовищах з великими файлами або великим обсягом трафіку, наприклад, для потокового відео або передачі великих обсягів даних (які можуть тривати десятки хвилин), накладні витрати компенсуються гарантією QoS, забезпеченою для користувачів.

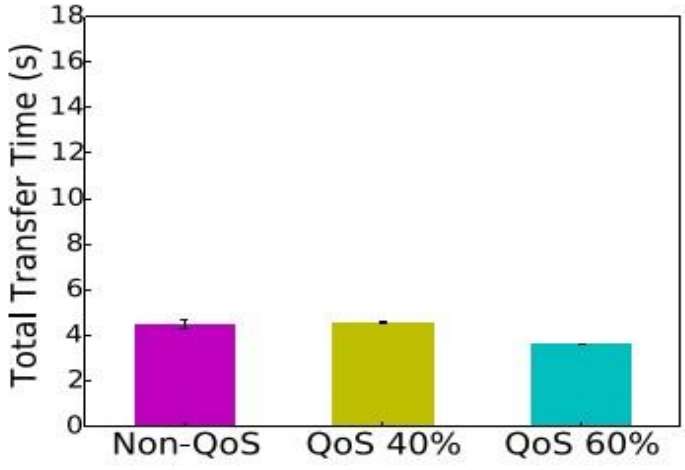
Як показано на рисунку 7, ми спостерігаємо значне скорочення часу передачі в залежності від параметрів та встановлених категорій послуг. Наприклад, для передачі невеликих файлів (див. рисунок 3.6)



(a) Small File

Рисунок 3.6 - Передача невеликого файлу.

Система без будь-якого механізму забезпечення якості обслуговування працює краще (1,82 с), ніж система, що використовує запропонований нами механізм (2,71 с і 2,40 с для резервування пропускної здатності на рівні 40% і 60% відповідно). Як видно на рис. 3.7, механізм забезпечення якості обслуговування перевершує традиційну систему (для резервування 60% відповідності), зменшуючи середній час передачі практично на 20%

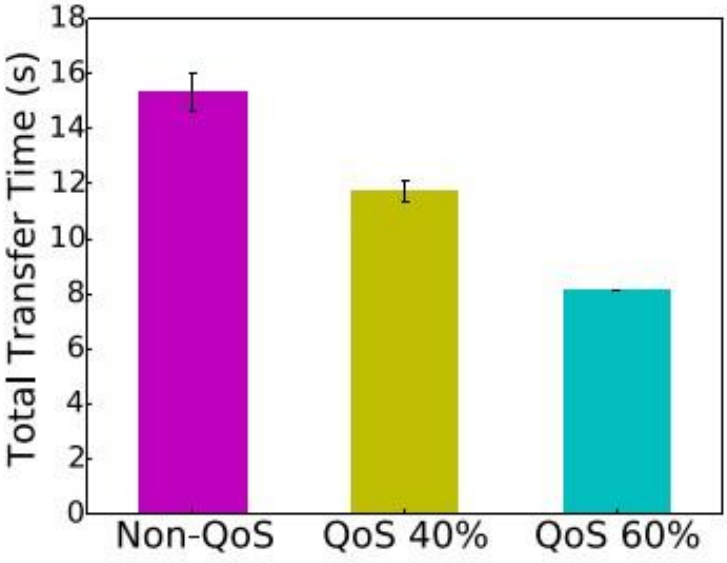


(b) Medium File

.

Рисунок 3.7 - Передача середнього розміру файлу.

Використання механізму забезпечення якості обслуговування (QoS) під час передачі великих файлів (див. рис. 3.8) виявляє найкращу продуктивність, при цьому час передачі збільшується приблизно на 23% і 47% для резервування пропускної здатності на рівні 40% і 60% відповідно. У цьому випадку, незважаючи на додаткові витрати, архітектура QoS продемонструвала значно кращу продуктивність, підкреслюючи її ефективність.



(c) Large File

Рисунок 3.8 - Передача великого обсягу даних.

Під час додаткової оцінки було проведено порівняльний аналіз за парними спостереженнями між виконанням додатків без забезпечення якості обслуговування (QoS) і виконанням з резервуванням ресурсів. Ми розрахували довірчі інтервали та середні значення для рівня довіри 95%, як показано в таблиці 3.1.

Таблиця 3.1 - Порівняльний аналіз за парними спостереженнями.

	Non-QoS/QoS 40%	Non-QoS/QoS 60%
Small File	(-0,92 , -0,87)	(-0,61 , -0,55)
Medium File	(-0,25 , 0,10)	(0,68 , 1,02)
Large File	(2,89 , 4,27)	(6,54 , 7,90)

Негативні показники довірчих інтервалів для порівняння "Non-QoS/QoS 40%" і "Non-QoS/QoS 60%" з використанням невеликого файлу свідчать про те, що традиційний підхід переважає над підходом з забезпеченням якості обслуговування (QoS). У сеансі моделювання з файлом невеликого обсягу, включення нульового значення в інтервал порівняння "Non-QoS/QoS 40%" показує, що для даної конфігурації обидва підходи є еквівалентними. Проте, при збільшенні резервування до 60%, використання архітектури з QoS стає вигідним, як підтверджує позитивний інтервал значень. Нарешті, порівняння вимірювань з великим файлом показують позитивні інтервали у обох випадках, що свідчить про перевагу підходу з надання QoS, особливо для більших файлів, де збільшення продуктивності компенсує витрати. Ця методологія оцінки підтверджує попередній аналіз. Зрозуміло, що конкретні переваги залежать від розгорнутих додатків і специфікацій QoS, що використовуються в мережі.

ВИСНОВОК

Розвиток програмно-визначених мереж (SDN) представляє собою значний крок вперед у сфері управління сучасними телекомунікаційними інфраструктурами. Ця технологія дозволяє значно підвищити ефективність, гнучкість та продуктивність мереж завдяки централізованому управлінню та програмованості мережевих ресурсів. Одним з ключових аспектів SDN є можливість забезпечення високої якості обслуговування (QoS) для різноманітних додатків, які мають суворі вимоги до мережевих ресурсів. Використання SDN дозволяє долати обмеження традиційних мереж, які часто є закостенілими та мають дефіцитні ресурси.

Архітектура SDN забезпечує розділення контрольного та передавального рівнів, що дозволяє централізовано управляти мережею за допомогою контролера. Це сприяє створенню нових абстракцій у мережі та спрощує її еволюцію. Контролер SDN має глобальний огляд мережі, що дозволяє динамічно адаптувати політики QoS у реальному часі відповідно до змін у навантаженні та вимогах користувачів. Завдяки централізованому моніторингу та аналізу, контролер може вчасно виявляти та усувати проблеми, що виникають, забезпечуючи стабільну та високу якість обслуговування.

Одним з ключових компонентів успішного впровадження SDN є вибір відповідного контролера та тестової платформи. Контролер OpenDaylight і тестова платформа Mininet забезпечують оптимальні умови для дослідження та тестування програмно-визначених мереж. OpenDaylight надає широкий спектр функціональних можливостей, високу продуктивність та масштабованість, тоді як Mininet дозволяє створювати віртуальні мережі та моделювати різні мережеві сценарії без необхідності фізичного обладнання.

Оцінка існуючих модулів контролера OpenDaylight показала, що ця платформа є потужним інструментом для управління мережею та забезпечення QoS. Модулі для комутації, маршрутизації, підтримки протоколу OpenFlow,

управління політиками та моніторингу трафіку дозволяють ефективно керувати мережею та адаптувати її до змінних вимог. Впровадження QoS за допомогою SDN дозволяє забезпечити стабільну затримку, мінімальний джиттер та необхідну пропускну здатність для критичних додатків, таких як відеоконференції, потокова передача мультимедіа та онлайн системи охорони здоров'я.

Загалом, програмно-визначені мережі (SDN) відкривають нові можливості для покращення якості обслуговування у високопродуктивних розподілених додатках. Вони дозволяють забезпечити гнучкість, масштабованість та ефективність мережевих ресурсів, що є критично важливим у сучасному світі, де зростає кількість додатків з високими вимогами до мережевої інфраструктури. Впровадження SDN сприяє підвищенню якості досвіду користувачів (QoE) та забезпечує більш ефективне використання мережевих ресурсів, що є ключовим фактором успішного функціонування сучасних телекомунікаційних систем.

ПЕРЕЛІК ПОСИЛАНЬ

1. SDN [Электронный ресурс]/ SCIENCE: Режим доступа:https://science.donntu.edu.ua/tks/volynskyi/diss/indexu.htm

2. SDN&NFV [Электронный ресурс]/ Bellintegrator: Режим доступа: http://www.bellintegrator.ru/services-sdn-nfv.html (10.02.2017) 88

3. Барсков А. SDN: кому и зачем это надо?/ Журнал сетевых решений/LAN.

2012. No 12. – [Электронный ресурс]. – Режим доступа: https://www.osp.ru/ lan/2012/12/13033012 (12.02.2017)

4. Смелянский Р. Л. Программно-конфигурируемые сети // Открытые системы. СУБД. 2012. No 9. С. 3843. - [Электронный ресурс]. – Режим доступа: http://www.osp.ru/os/2012/09/13032491 (26.02.2017)

5. Anders Nygren. OpenFlow Switch Specification / Anders Nygren, Ben Pfaff,

Bob Lantz [Електронний ресурс]. — Електронні текстові дані. — Режим доступу: https://www.opennetworking.org/images/stories/downloads/sdnresources/onfspecifications/openflow/openflow-switch-v1.5.1.pdf — Дата доступу: 5.03.2018. — Open Networking Foundation

6. Коломеец А. Е. Программно-конфигурируемые сети на базе протокола OpenFlow / А. Е. Коломеец, Л. В. Сурков // Инженерный вестник. – 2014.

– № 5. – С. 518–525.

7. IETF NETCONF Configuration protocol http://tools.ietf.org/html/rfc4741; 8. IETF SNMP - Simple Network Management Protocol http://tools.ietf.org/html/rfc5343:

9. J. Mueller, T. Magedanz, 'Towards a Generic Application Aware Network Resource Control Function for Next-Generation-Networks and Beyond', International Symposium on Communications and Information Technologies (ISCIT),

DOI:10.1109/ISCIT.2012.6381026, ISBN:978-1-4673-1156-4, Page(s): 877 - 882, Gold Coast, Australia, October 2–5, 2012, www.iscit2012.org;

10. D. Kreutz, F.M.V. Ramos, P. Verissimo, “Towards secure and dependable softwaredefined networks”- the second ACM SIGCOMM workshop on Hot topics in software defined networking, August 2013, http://conferences.sigcomm.org/sigcomm/2013/;

11. S. Shin et al. “FRESCO: Modular Composable Secu- rity Services for Software-

Defined Networks”. In: In- ternet Society NDSS 2013;

12. Y.3001 ITU-T recommendation – “Future networks: Objectives and design goals” –

2011 http://www.itu.int/rec/T-REC-Y.3001-201105-I;

13. D. Matsubara, T. Egawa, N. Nishinaga, M.-Ki, Shin, V. P. Kafle, A. Galis, -

“Toward Future Networks: A Viewpoint from ITU-T” - IEEE Communications Magazine, March 2013, Vol. 51, No. 3, pp: 112 – 118;

14. E. Haleplidis, S. Denazis, K. Pentikousis, J. Hadi Salim, D. Meyer, O. Koufopavlou,

SDN Layers and Architecture Terminology draft-haleplidis-sdnrg-layer-terminology-

03, December 5, 2013, http://tools.ietf.org/html/draft-haleplidis-sdnrg-layerhttp://tools.ietf.org/html/draft-haleplidis-sdnrg-layer-terminology-03terminology-03;

15. J. Rubio-Loyola, A. Galis, A. Astorga, J. Serrat, L. Lefevre, A. Fischer, A. Paler, H. de Meer, “Scalable Service Deployment on Software Defined Networks”– IEEE Communications Magazine/ Network and Service Management Series, ISSN: 01636804; December 2011 http://dl.comsoc.org/ci1/;

16. OpenStack - Open source software for building private and public clouds- http://www.openstack.org;

17. ETSI ‘Software-aware and Management-aware SDN” - 3rd ETSI Future Networks Workshop 9-11 April 2013 -

http://docbox.etsi.org/Workshop/2013/201304_FNTWORKSHOP/eproceedings_FNT_2013.pdf;

18. M. Banikazemi, D. Olshefski, A. Shaikh, J.Tracey, and G. Wang, “Meridian: An

SDN Platform for Cloud Network Services”, IEEE Communications Magazine • February 2013;

19. L. Erran Li, Z. Morley Mao J. Rexford, CellSDN: Software-Defined Cellular

Networks, Open Network Summit (Research Track). Santa Clara, CA, USA (April 2013);

20. A. Gudipati_, D. Perry_, L. Erran Li, S.Katti, SoftRAN: Software Defined Radio

Access Network, HotSDN 2013;

21. Open Systems Interconnection (OSI) model (1994)- www.ecmainternational.org/activities/Communications/TG11/s020269e.pdf;

22. D. Matsubara, T. Egawa, N. Nishinaga, M.-Ki, Shin, V. P. Kafle, A. Galis, - “Open the Way to Future Networks – a viewpoint framework from ITU-T” – invited paper “The Future Internet- Future Internet Assembly 2013: Validated Results and New

Horizons” Lecture Notes in Computer Science 7858, Springer, pp370, May 2013, ISBN 978-3-642-38081-5; http://www.springerlink.com/content/978-3-642-38081-5/;

23. A. Galis, J. Rubio-Loyola, S. Clayman, L. Mamatas, S. Kukliński, J. Serrat, T.

Zahariadis, “Software Enabled Future Internet” - 5th International Conference on Mobile Networks and Management (MONAMI 2013), 23-25 Sept 2013, Cork, Republic of Ireland, http://mon-ami.org/2013/show/home

24. T. Humernbrum, F. Glinka, and S. Gorlatch, “A northbound api for qos management in real-time interactive applications on software-defined networks,” Journal of Communications, vol. 9, no. 8, pp. 607–615, 2014.

25. D. Kreutz, F. Ramos, P. Ver´ıssimo, C. Rothenberg, S. Azodolmolky, and S. Uhlig,

"Software-defined networking: A comprehensive survey," Proceedings of the IEEE, vol. 103, no. 1, pp. 14–76, 2015.

26. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford,

S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008.

27. QOS [Электронный ресурс]/ ZNANIUS: Режим доступа: https://www.znanius.com/3814.html

28. ITU-T, "Definitions of terms related to quality of service," Recommendation E800,

September 2008. [Online]. Available: http://www.itu.int/rec/T-REC-E.800-200809-I/en

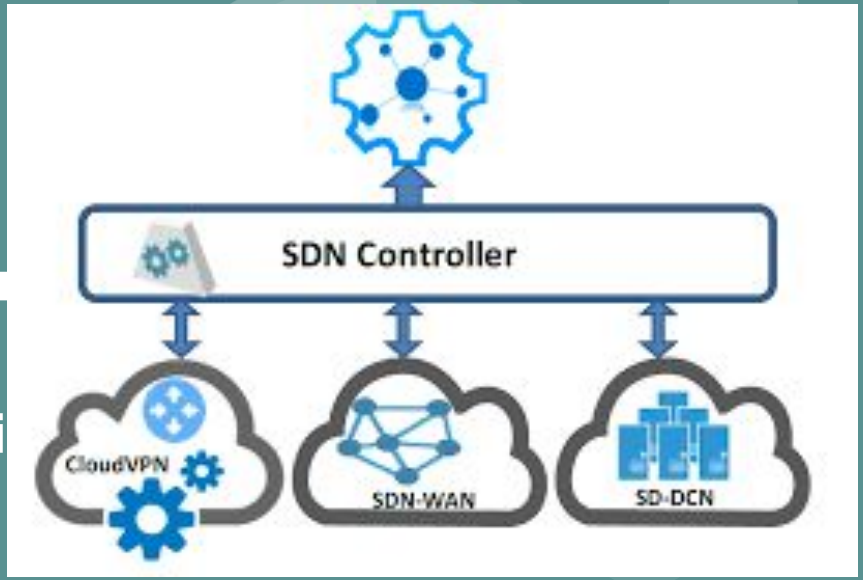
29. M. Moreno, C. Neto, A. Gomes, S. Colcher, and L. Soares, "Qosos: An adaptable architecture for qos provisioning in network operating systems," Journal of Communication and Information Systems, vol. 18, no. 2, pp. 118–131, 2003.

30. B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in Proc. of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, 2010.

31. Diego Kreutz, Fernando Ramos, and Paulo Verissimo. Towards

secure and dependable soft-ware-defined networks. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, p. 55–60. 2013.

Презентація: Розробка рекомендацій для підвищення ефективності роботи програмно-визначеної мережі (SDN)



Мета і задачі роботи

Метою роботи є аналіз існуючих проблем та розробка методів підвищення ефективності роботи програмно-визначених мереж (SDN). Для досягнення поставленої мети необхідно вирішити такі завдання:

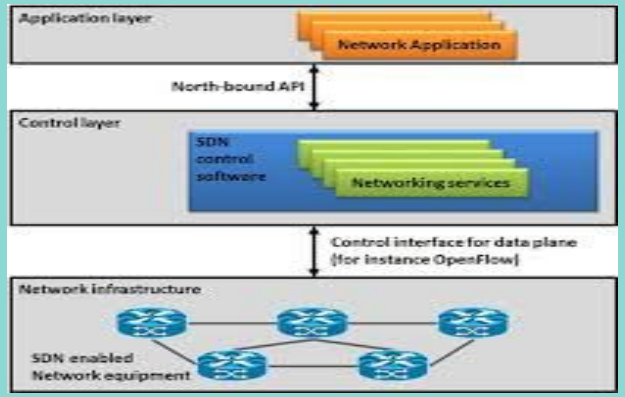
1. **Провести огляд сучасного стану та тенденцій розвитку програмно-визначених мереж.**

2. **Дослідити основні проблеми та обмеження існуючих SDN рішень.**

3. **Розробити методи підвищення ефективності роботи SDN.**

4. **Оцінити ефективність запропонованих методів на основі експериментальних досліджень.**

Основи SDN



Вступ

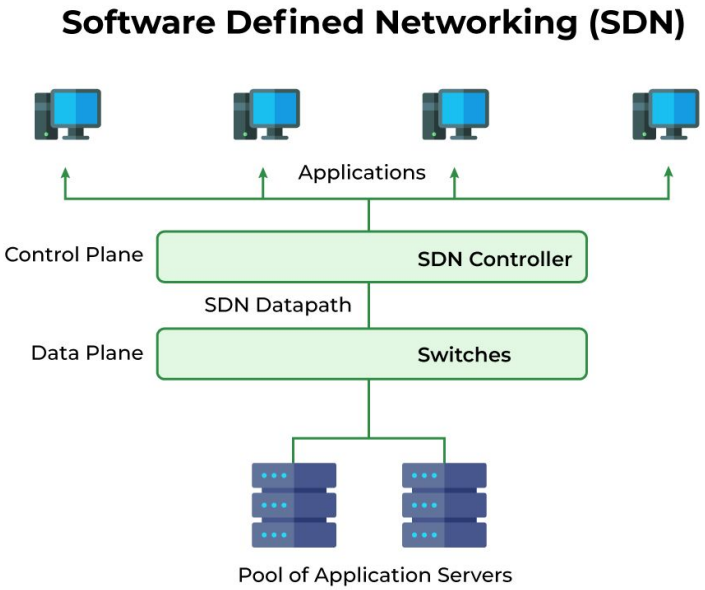
- **SDN (Software-Defined Networking)** - це підхід до комп'ютерних мереж, який дозволяє програмно керувати та централізовано контролювати мережеву інфраструктуру.
- **Мета SDN** - спростити управління мережами та зробити їх більш гнучкими та адаптивними до вимог користувачів.

Основні Компоненти SDN

1. **Контролер SDN**
   - Центральний мозок мережі.
   - Управляє трафіком через програмне забезпечення.
2. **Північний інтерфейс (Northbound API)**
   - Інтерфейс між контролером SDN та програмами.
   - Дозволяє додаткам керувати мережею.
3. **Південний інтерфейс (Southbound API)**
   - Інтерфейс між контролером SDN та мережевими пристроями.
   - Використовується для передачі інструкцій до мережевих пристроїв.

-

Переваги та недоліки SDN



Переваги SDN

- **Гнучкість**: Швидке налаштування та зміна мережевих параметрів.
- **Централізоване управління**: Єдина точка управління для всієї мережі.
- **Зниження витрат**: Оптимізація використання ресурсів та зменшення необхідності в спеціалізованому обладнанні.
- **Автоматизація**: Зменшення ручної праці завдяки автоматичним налаштуванням.

Недоліки SDN

- **Безпека**: Збільшення ризиків через централізовану точку управління.
- **Сумісність**: Проблеми з інтеграцією з існуючим обладнанням.
- **Складність впровадження**: Необхідність перепроектування існуючої мережі.

Технічні проблеми розробки та розгортання SDN та способи їх вирішення

Розробка та розгортання SDN (Software-Defined Networking) може зіткнутися з рядом технічних проблем. Ось деякі з них та можливі способи їх вирішення:

Сумісність з існуючими мережевими пристроями: Однією з основних проблем при розгортанні SDN є сумісність існуючих мережевих пристроїв з новими SDN-компонентами. Деякі пристрої можуть не підтримувати протоколи, які використовуються в SDN.

Рішення: Використання протоколів тунелювання, таких як VXLAN або GRE, може допомогти у створенні абстракційного шару, який дозволить існуючим пристроям працювати з контролером SDN.

Масштабування: При великому розмірі мережі може виникнути проблема масштабування контролера SDN та обробки великої кількості запитів на керування.

Рішення: Використання розподіленого контролера SDN, який може обробляти навантаження паралельно, або використання технологій, таких як мультикастинг, для розподілу стану мережі між кількома контролерами.

Безпека: Розгорнута SDN мережа може бути вразливою до різних типів кіберзагроз, таких як атаки внедрення відмов, атаки з перехопленням даних тощо.

Рішення: Використання механізмів аутентифікації, авторизації та шифрування для захисту комунікації між компонентами SDN, а також застосування систем моніторингу та виявлення вторгнень для вчасного виявлення та реагування на потенційні загрози.

Управління ресурсами: Ефективне управління ресурсами мережі, такими як пропускна здатність та потужність обробки, може бути складною задачею в SDN, особливо при динамічних змінах у трафіку.

Рішення: Використання алгоритмів управління ресурсами, таких як механізми QoS (Quality of Service), для призначення пріоритетів та резервування ресурсів для важливих видів трафіку.

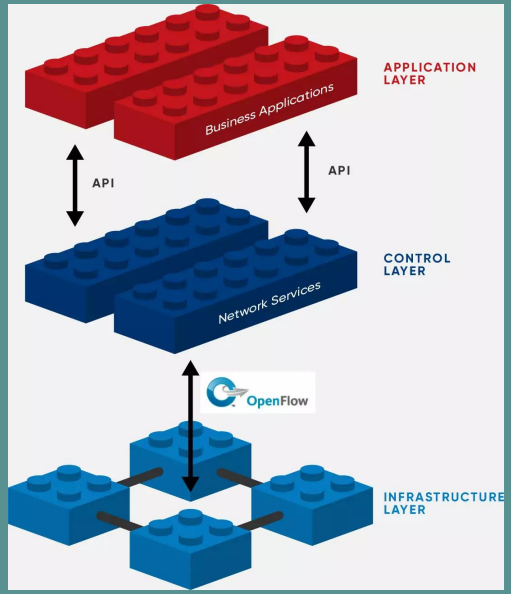
Огляд протоколу OpenFlow

Основні Компоненти OpenFlow

1. **Контролер**
   - Центральний компонент, який керує мережею.
   - Вирішує, як буде оброблятися трафік.
2. **Комутатор OpenFlow**
   - Мережевий пристрій, який виконує інструкції від контролера.
   - Має таблиці потоків для управління трафіком.
3. **Таблиця потоків**
   - Містить правила, які визначають, як обробляти пакети.
   - Складається з критеріїв матчу, дій та статистики.

Як Працює OpenFlow

1. **Ідентифікація Пакетів**
   - Комутатор OpenFlow порівнює кожен вхідний пакет з правилами в таблиці потоків.
2. **Прийняття Рішень**
   - Якщо знайдено відповідне правило, виконується визначена дія (пересилання, зміна, відкидання).
   - Якщо правило не знайдено, пакет передається до контролера для визначення дій.
3. **Виконання Дій**
   - Дії можуть включати пересилання пакета на певний порт, зміну заголовків пакетів, агрегацію статистики.



Переваги та недолік

Переваги Використання OpenFlow

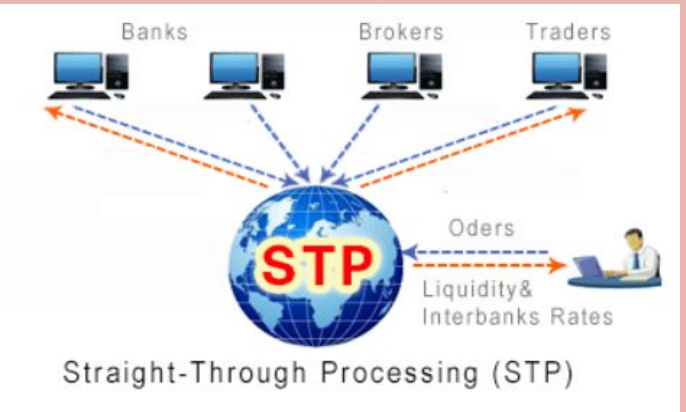
- **Гнучкість**: Легке оновлення та зміна правил обробки трафіку.
- **Централізоване управління**: Єдина точка управління для всієї мережі.
- **Прозорість**: Видимість і контроль над усіма потоками даних у мережі.

Недоліки OpenFlow

- **Сумісність**: Проблеми з підтримкою старих мережевих пристроїв.
- **Продуктивність**: Можливі затримки через необхідність передачі пакетів до контролера.
- **Складність управління**: Необхідність вміння налаштовувати і управляти OpenFlow.

Протокол Spanning Tree

Протокол Spanning Tree Protocol (STP) - це механізм, який дозволяє уникнути створення петель в мережах Ethernet, які можуть спричинити проблеми, такі як безкінечне відправлення пакетів або перенавантаження комутаторів. Основна мета STP - забезпечити стабільну топологію мережі, уникаючи цих петель.



Основні компоненти та функції протоколу Spanning Tree:

1. **Вибір кореневого моста (Root Bridge)**: Визначення основного комутатора, який стає кореневим мостом мережі. Він є центральною точкою, від якої відраховуються всі інші шляхи в мережі.
2. **Визначення найкоротшого шляху (Shortest Path)**: STP обчислює найкоротший шлях від кожного комутатора до кореневого моста, використовуючи поняття стоїмості шляху (Cost).
3. **Блокування портів (Port Blocking)**: Відбувається блокування непотрібних портів, щоб уникнути створення петель. При цьому залишається активним лише один шлях до кожного комутатора.
4. **Перевибір кореневого моста (Root Bridge Recalculation)**: У випадку відмови поточного кореневого моста, STP автоматично перевибирає новий кореневий міст.
5. **Перевибір шляху (Path Recalculation)**: При виявленні зміни топології мережі STP перераховує шляхи до кореневого моста для оптимізації трафіку.

Плюси протоколу Spanning Tree:

- **Запобігає петлям**: STP гарантує, що в мережі не буде петель, що забезпечує безперебійну роботу мережі.
- **Автоматичне управління**: Протокол автоматично адаптується до змін у топології мережі.
- **Відновлення після відмови**: У випадку відмови протокол швидко переключає шляхи для відновлення зв'язку.

Мінуси протоколу Spanning Tree:

- **Затримка відновлення**: Перевибір шляхів може зайняти певний час, що може вплинути на час відновлення мережі після відмови.
- **Потенційне перенавантаження шляхів**: У мережах з великою кількістю комутаторів може виникнути перенавантаження шляхів до кореневого моста.

Де використовується:

Протокол Spanning Tree використовується у всіх типах мереж, особливо у великих мережах Ethernet, де є багато комутаторів та можливість виникнення петель. Він є стандартом для побудови надійних мереж Layer 2.

Порівняння використання STP у традиційних та SDN-мережах

Порівняння використання протоколу Spanning Tree (STP) в традиційних та SDN-мережах може відрізнятися через особливості самого архітектурного підходу SDN.

Традиційні мережі:

1. **STP як основний механізм безпеки**: У традиційних мережах, де комутатори працюють автономно, STP використовується як основний механізм безпеки для уникнення петель. Це означає, що виключені порти можуть бути потенційно неактивними, але все одно резервуються для використання в разі відмови.
2. **Можливість маршрутизації за межами STP**: У традиційних мережах, які працюють на основі STP, маршрутизація може бути реалізована за допомогою інших протоколів, таких як OSPF або EIGRP, для використання шляхів більш відомих або забезпечення балансування навантаження.

SDN-мережі:

1. **STP в контексті програмованої мережі**: У SDN-мережах протоколи, такі як STP, можуть використовуватися на рівні фізичного шару (які зазвичай невидимі для контролера SDN), однак саме програмована логіка контролера може надавати більш гнучкий підхід до управління трафіком та уникнення петель.
2. **Автоматизація та оптимізація маршрутизації**: У SDN-мережах розуміння всієї топології мережі дає контролеру можливість автоматично оптимізувати маршрутизацію трафіку, уникати перенавантажень та використовувати резервні шляхи без залежності від традиційних механізмів, таких як STP.
3. **Динамічність та гнучкість**: У SDN-мережах можна реалізувати алгоритми роботи, які більш адаптивні до змін в топології мережі, що дозволяє більш гнучке управління мережею та оптимізацію шляхів.

Отже, хоча протокол STP може залишатися важливим елементом безпеки в традиційних мережах, у SDN-мережах його функції можуть бути частково заміщені більш гнучкими та автоматизованими методами управління мережею.

Оцінка існуючих модулів конролера SD

OpenDaylight (ODL):

- OpenDaylight є одним з найбільш відомих відкритих контролерів SDN.
- Має велику спільноту розробників та підтримується кількома відомими компаніями.
- Підтримує різні протоколи і стандарти, такі як OpenFlow, NETCONF, і SNMP.
- Має різноманітні додаткові модулі та інструменти для створення різноманітних сервісів.

Floodlight:

- Floodlight - це ще один відкритий контролер SDN, який є популярним в галузі досліджень та виробництва.
- Він має простий і легкий у використанні API та добре підходить для вивчення та експериментів у сфері SDN.
- Він має деякі базові функціональні можливості, такі як керування комутаторами через OpenFlow.

ONOS (Open Network Operating System):

- ONOS є ще одним відкритим контролером SDN, який розвивається спільнотою.
- Він має високу масштабованість і продуктивність, дозволяючи обробляти великі мережі.
- ONOS підтримує різні застосування та сервіси, такі як віртуалізація мережі, мобільність і IoT.

Cisco Application Centric Infrastructure (ACI):

- Це комерційний контролер SDN від Cisco, спеціально розроблений для їхньої SDN-платформи ACI.
- ACI забезпечує централізоване керування та автоматизацію мережевих операцій від краю до краю.
- Він має широкі можливості для створення політик мережі та розгортання додатків.

QoS та його можливості

Quality of Service (QoS) - це механізм, який дозволяє призначати пріоритети різним видам трафіку в мережі з метою забезпечення відповідних рівнів обслуговування. Основні можливості QoS включають:

1. **Приоритизація трафіку**: QoS дозволяє встановлювати пріоритети для різних типів трафіку. Наприклад, голосовий трафік VoIP може мати вищий пріоритет порівняно з трафіком веб-переглядача.
2. **Керування пропускною здатністю**: QoS дозволяє обмежувати або розподіляти пропускну здатність мережі між різними видами трафіку. Наприклад, можна встановити максимальну пропускну здатність для певних видів трафіку, щоб уникнути перенавантаження мережі.
3. **Керування затримками та втратами пакетів**: QoS дозволяє контролювати затримки та втрати пакетів, що є критичними для деяких додатків, таких як голосовий або відео трафік.
4. **Контроль і стабілізація мережевої продуктивності**: QoS може бути використаний для стабілізації продуктивності мережі, уникнення переповнення трафіку та покращення взаємодії з додатками.
5. **Управління смугою пропускання і підтримка SLA**: QoS дозволяє встановлювати обмеження на використання пропускної здатності мережі для різних користувачів або сервісів, що дозволяє виконувати Service Level Agreements (SLA) та гарантувати якість обслуговування.
6. **Мультимедійна передача трафіку**: QoS може бути використаний для забезпечення оптимального обслуговування мультимедійного трафіку, такого як аудіо та відео стрімінг, забезпечуючи низькі затримки та мінімальні втрати.

Загалом, QoS грає важливу роль у забезпеченні ефективної та надійної роботи мережі, дозволяючи пристроям та додаткам працювати ефективно в умовах обмежених ресурсів мережі.

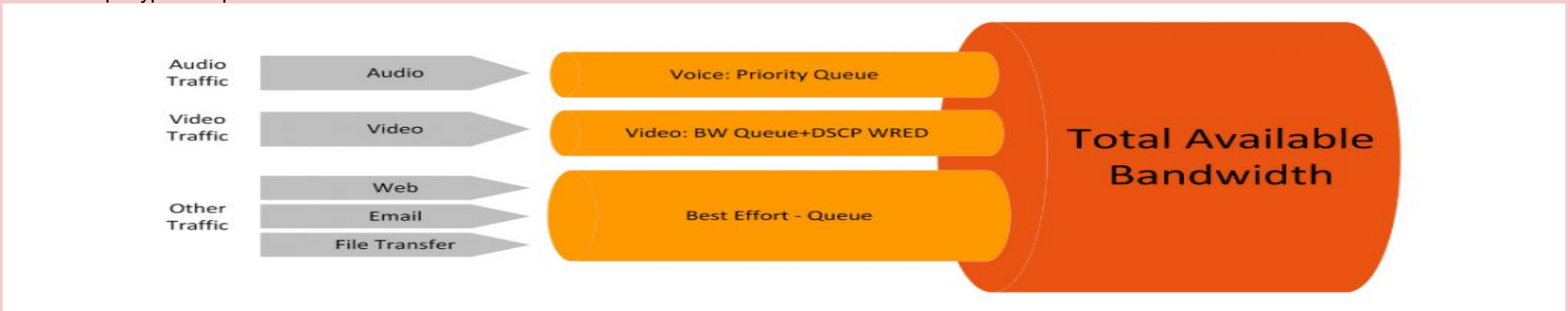
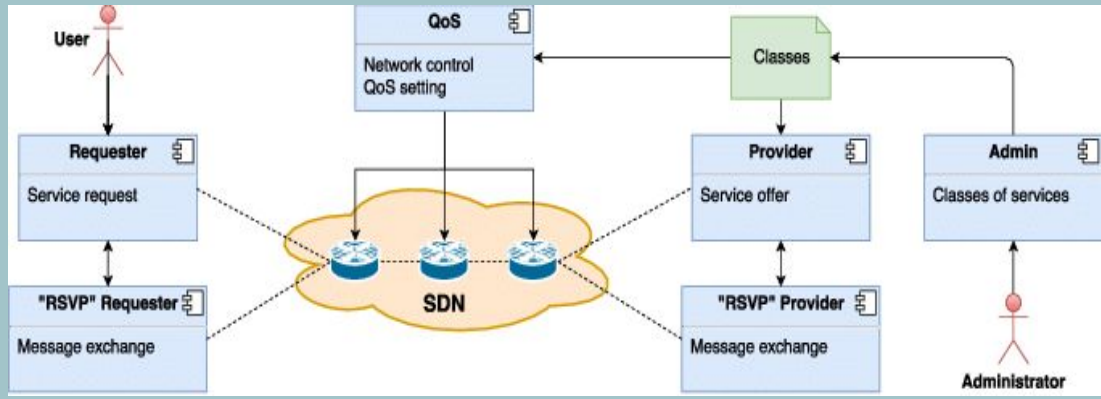


Схема архітектури надання QoS на основі архітектури

Схема архітектури надання QoS на основі архітектури SDN може включати наступні основні компоненти:

1. **Контролер SDN (SDN Controller)**:
   - Центральний елемент, який керує всією мережею SDN.
   - Він отримує дані зі всіх пристроїв мережі та приймає рішення щодо маршрутизації трафіку та надання QoS.
2. **Модуль QoS (QoS Module)**:
   - Цей модуль вбудований в контролер SDN та відповідає за реалізацію політик QoS.
   - Він аналізує поточний стан мережі та виробляє рішення щодо призначення пріоритетів та обмежень трафіку.
3. **Протоколи тунелювання (Tunneling Protocols)**:
   - Деякі архітектури SDN можуть використовувати протоколи тунелювання, такі як VXLAN або GRE, для реалізації QoS.
   - Ці протоколи можуть використовуватися для створення тунелів, через які буде передаватися трафік з певними характеристиками QoS.
4. **Пристрої мережі (Network Devices)**:
   - Програмовані комутатори та маршрутизатори, які здатні взаємодіяти з контролером SDN і реалізовувати політики QoS, які він надає.
5. **Політики QoS (QoS Policies)**:
   - Це правила та параметри, які визначають пріоритети, обмеження та поведінку трафіку в мережі з точки зору QoS.
   - Політики можуть бути налаштовані та змінюватися динамічно з контролеру SDN.



Ця архітектура дозволяє контролювати трафік в мережі та надавати різні рівні обслуговування в залежності від вимог додатків та користувачів, що дозволяє максимізувати продуктивність та якість обслуговування.