



Звіт про оригінальність

● Оцінка схожості

% 24

● Ризик плагіату

НАЙВИЩИЙ

👤 Olga Kagalo 🕒 2025-06-19 23:09

Посилання на звіт: 10mDJ / Посилання користувача: qEAc



Ось вона – Ваша звіт про оригінальність!

Ми раді повідомити, що перевірка вашого документа завершена, і результати вже готові! Наші алгоритми старанно працювали, щоб знайти збіги в наших базах даних.

На наступних сторінках ви знайдете результати перевірки:

Бали

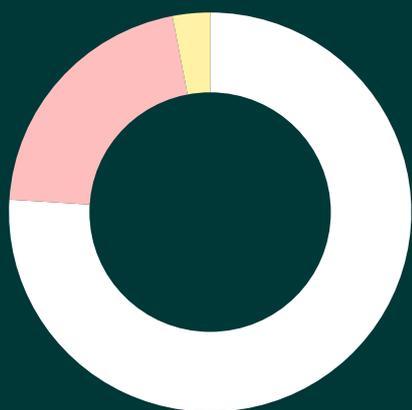
Збіги

Посилання

Ваш документ було перевірено за такими джерелами:

- База даних інтернет-джерел
- База даних наукових статей
- Глибока перевірка (наш вдосконалений алгоритм)

Бали



● Збіги тексту	21%
● Перефразування	3%
● Цитований текст	0%
● Неправильне цитування	0%
● Збігів не знайдено	76%

Ризик плагіату

НАЙВИЩИЙ

Ризик плагіату вказує, як збіги тексту розподілені по документу. Вищий ризик виникає, коли збіги з'являються близько один до одного, наприклад, у тому самому абзаці або розділі.

Оцінка схожості

Оцінка схожості показує, скільки слів або символів у вашому документі збігаються з текстами інших документів, включаючи перефразовані тексти або неправильні цитати.

% 24

Збіги

1 ДОСЛІДЖЕННЯ ПЕРСПЕКТИВ РОЗВИТКУ ТЕЛЕКОМУНІКАЦІЙНОГО ІНФРАСТРУКТУРНОГО ПРОСТОРУ

1.1 Стратегія створення мережі наступного покоління NGN

NGN, або мережа наступного покоління, є пакетно-орієнтованою інфраструктурою, придатною для надання різноманітних телекомунікаційних послуг та використання різноманітних технологій передачі даних широкого спектру [2]. Поява NGN була умовлена декількома факторами, такими як спеціалізація існуючих мереж, наявність багатьох окремих мереж та високі витрати на їх утримання. Концепція NGN ґрунтується на ідеї розробки універсальної мережі, яка здатна переносити 8 різні типи інформації, такі 8 як 8 голос, дані, відео, аудіо, графіка тощо, і надавати широкий спектр послуг у сфері інформаційно-комунікаційних технологій.

Принципи, що лежать в основі побудови мережі NGN, включають у себе такі аспекти:

- Забезпечення максимальної простоти та зручності підключення до мережі без потреби використання проміжних систем.
- Початково формується базова пакетна транспортна мережа, яка потім розвивається для надання різноманітних сервісів.

Рисунок 1.1– Прогрес від стандартів ISO/OSI до концепції NGN

NGN охоплює широкий спектр мереж, від прововодових до бездротових, та від телекомунікаційних до комп'ютерних. Вона забезпечує передачу різноманітних послуг через єдину та відкриту мережеву інфраструктуру. NGN складається з чотирьох рівнів:

Рисунок 1.2 – Структура NGN мережі

Отже, давайте розглянемо кожен з цих рівнів NGN більш детально [2]:

1. На вищому рівні NGN розглядається прикладний рівень. Основне завдання цього рівня - забезпечення всіх доступних послуг на мережах наступного покоління. Концепція NGN передбачає можливість надання абонентам послуг Triple-Play (тобто

передача голосу, даних і відео) за допомогою мультисервісних мереж, які формуються за допомогою модернізації існуючих мереж електрозв'язку. Перехід до NGN розкриває практично необмежені можливості для впровадження послуг і в корпоративному секторі. У звичайних мережах такі послуги зазвичай надаються місцевими операторами і часто вимагають значних тимчасових або фінансових вкладень.

2. Рівень управління послугами відповідає за надання інформаційних послуг кінцевому користувачеві, а подальший розвиток мережі залежить від того, наскільки ці послуги будуть привабливими для нього. Перехід до NGN **9** відкриває практично необмежені можливості для реалізації послуг як для індивідуальних, так і для корпоративних користувачів. У звичайних мережах подібні послуги часто надаються локальними операторами, і їх підключення може потребувати значних тимчасових або фінансових вкладень. З використанням єдиної IP-системи виникає однаковий набір послуг для всіх користувачів, а механізм їх підключення стає значно простішим - достатньо обрати потрібну послугу зі списку і надіслати відповідний запит. Сервери, які надають самі послуги, можуть бути розташовані як в межах мережі, так і поза нею (наприклад, веб-сервери, сервери, належні ASP-провайдерам). Важливим компонентом рівня управління послугами є інформаційні центри або центри управління послугами - це власні ресурси мережі, на основі яких здійснюється обслуговування користувачів. Ці центри можуть зберігати два типи інформації: інформацію користувача, таку як веб-портали з різноманітною інформацією та новинами, і допоміжну службову інформацію, що дозволяє надавати користувачам додаткові послуги.

3. Завдання транспортного рівня полягає у забезпеченні безперервної передачі інформації користувача шляхом комутації та маршрутизації.

1.2 Передумови виникнення стратегії SDN та її концепція

Технологія SDN (Software Defined Networking) перевертає звичайний підхід до створення та керування мережами. Вона відрізняється тим, що розділяє управлінський шар мережі (Control plane), що відповідає за маршрутизацію трафіку, від шару передачі даних (Data plane), який просто пересилає трафік відповідно до правил, наданих управлінським шаром. Крім того, SDN уніфікує управлінський шар, дозволяючи одному набору керуючих програм керувати багатьма мережевими пристроями на рівні передачі даних. Це досягається за допомогою стандартизованого інтерфейсу **10** програмного забезпечення API (Application Programming Interface), такого як OpenFlow. Для розбудови мережі SDN необхідно, щоб на мережевих пристроях, зокрема на комутаторах і маршрутизаторах, була підтримка OpenFlow. На цих пристроях зазвичай знаходиться таблиця або таблиці маршрутизації, кожне правило яких визначає, як пересилати пакети для конкретної сесії або потоку трафіку.

Ідея SDN є відносно новою, але її коріння можна відстежити вже протягом більш як

двадцяти років. Перші відзнаки цієї концепції можна виявити в розвитку 5 ранніх телефонних мереж на 5 основі комутації каналів, де 5 управління мережею (сигналізація) було відокремлене 5 від мережі 5 каналної комутації мовного трафіку. Це було зроблено з 5 метою спростити управління та впровадження 5 нових послуг. Концепція "Програмних комутаторів" (Softswitch) для телекомунікаційних мереж на базі комутації пакетів також має значні схожості з SDN щодо функцій та реалізації.

Центральною для концепції SDN є технологічний стандарт OpenFlow, розроблений Фондом відкритих мережових технологій (ONF). Цей стандарт визначає протокол зв'язку, який служить основою для програмно-керованих 5 мереж (SDN). Інтерфейс OpenFlow забезпечує взаємозв'язок між рівнями управління та інфраструктурою 5 архітектури SDN, як фізичної, так і віртуальної. Завдяки централізованому управлінню пристроями на рівні інфраструктури, OpenFlow спрощує адміністрування мережі та розширює можливості програмування, що відповідає основним принципам 5 SDN.

5 Концепція SDN передбачає:

5 – Відокремлення 5 управління мережовим обладнанням від передачі даних в маршрутизаторах. Управління виноситься на окремий комп'ютер, під контролем адміністратора мережі.

– Перехід від управління окремими пристроями 5 мережевого обладнання до управління мережею в цілому.

5 – Створення 5 інтелектуального програмно-керованого інтерфейсу 5 між мережним додатком і транспортним середовищем рис 1.3.

Рисунок 1.3 – Основні ідеї концепції SDN

У теорії концепція програмно-керованих мереж має численні переваги:

5 – підвищується продуктивність за рахунок прискорення переміщення трафіку;

5 – знижуються витрати на 5 побудову та підтримку мережі завдяки 5 віртуалізації управління мережею;

5 – покращується зручність управління, безпека та спрощується виконання різноманітних завдань;

– надаються необмежені можливості для розширення та масштабування залежно від потреб.

2 ОСНОВНІ СКЛАДОВІ І ПРИНЦИПИ ФУНКЦІОНУВАННЯ МЕРЕЖ SDN

2.1 Основні компоненти та організаційна схема мережі SDN

Один з ключових напрямів розвитку телекомунікаційних мереж – це використання концепції Software-defined networking (SDN). Суть SDN полягає в розділенні функцій управління та передачі трафіку, з усіма управлінськими функціями, що централізуються у мережі. Цей підхід сприяє ефективній обробці великих обсягів даних та спрощує контроль і налаштування мережевого обладнання.

Основна відмінність цього підходу від традиційних мережевих моделей полягає в тому, що управління мережею відділене від передачі даних. Багато функцій управління здійснюються за допомогою програмного забезпечення.

Архітектуру мережі SDN можна умовно розділити на три рівні рис. 2.1 [7]:

- рівень додатків;
- рівень управління;
- рівень мережі.

Рисунок 2.1 – Структура програмно-керованих мереж

Функціональний рівень програмно-керованих мереж включає в себе набір програмних рішень, спрямованих на розширення можливостей мережі. Ці рішення переважно є програмними додатками, які взаємодіють з централізованим контролером мережі. Вони базуються на відкритих інтерфейсах API, які забезпечують взаємодію між програмними модулями контролера та додатками SDN. Ці інтерфейси є доступними для розвитку та модифікації з боку клієнтів, партнерів і спільноти з відкритим вихідним кодом. Функціональний рівень програмно-керованих мереж включає різноманітні програмні додатки, які задовольняють різні потреби користувачів. Серед цих потреб - автоматизація мережі, гнучкість та програмованість. Додатки можуть забезпечувати інженерію трафіку, віртуалізацію мережі, моніторинг та аналіз мережі, дослідження мережевих сервісів, контроль доступу та інші функції. Логічне управління для кожного додатку може бути виконане як окремий процес на апаратному забезпеченні контролера в межах кожного домену. Це загальна концепція функціонального рівня програмно-керованих мереж.

Рівень управління забезпечує координацію та контроль роботи мережі.

Централізований контролер SDN приймає та обробляє запити від програмного рівня за допомогою чітко визначених API. Він керує та моніторить мережеве обладнання за допомогою стандартних протоколів. Кожен контрольний домен мережі має свій

власний контролер, який відповідає за збір інформації про стан мережі в межах своєї ділянки.

1 Рівень мережі складається з фізичного мережевого обладнання, такого 1 як 1 комутатори та маршрутизатори. Цей рівень забезпечує програмне забезпечення та апаратне забезпечення, відповідне галузевим стандартам. Фізична мережа включає апаратні пристрої 1 передачі даних, які зберігають таблиці для ефективної передачі пакетів та пов'язані метадані, такі як пакети, потоки та лічильники портів.

Мережі SDN відрізняються 1 від традиційних мереж тим, що вони пропонують більш гнучкий та швидкий підхід до управління мережами. У традиційних мережах управління розділене і децентралізоване, що часто призводить до складнощів у впровадженні нових технологій.

У мережах SDN введено централізований аналіз стану мережі та розділення процесів пересилання пакетів та формування маршрутів. Це дозволяє забезпечити більшу гнучкість та ефективність управління мережею. Основним елементом управління є контролер, який може легко модифікуватися шляхом зміни програмного забезпечення, що робить перехід на нові технології більш простим і швидким.

Порівняння між традиційними мережами та мережами SDN представлено на рис. 2.2 та рис. 2.3.

Рисунок 2.2 – Класична мережа

Рисунок 2.3 – Мережа з використанням технології SDN

На рис. 2.2 представлено конфігурацію традиційних мереж, де на кожному мережевому пристрої виконуються відповідні протоколи, які забезпечують функціонування сервісів обробки пакетів. У випадку мереж SDN рис. 2.3., завдання комутаторів та маршрутизаторів зводиться до передачі трафіку, оскільки всі керуючі функції перейняв контролер.

При створенні мережі з використанням технології SDN використовується обмежений набір елементів, в основному – комутатори та контролер.

2.2 Організація SDN контролера

Структура контролера SDN полягає у системі централізованого керування мережею, яка вирішує різноманітні завдання, пов'язані з управлінням трафіком даних. Це включає формування планів розподілу у вигляді маршрутних матриць, визначення пріоритетності обслуговування різних видів навантаження, моніторинг обсягу трафіку на різних елементах мережі і вжиття заходів для запобігання перевантаження, а також

забезпечення доставки повідомлень до заданих адрес з відповідною якістю обслуговування.

Структура контролера SDN включає в себе ряд компонентів, таких як пам'ять, процесор, таблицю маршрутизації, ASIC та порти. Це показано на рис. 2.4.

Рисунок 2.4 – Архітектура контролера SDN

Потік трафіку - це послідовність пакетів, що передають дані від джерела до призначення. Мережевий пристрій SDN може ідентифікувати пакети, які належать конкретному потоку даних, на основі адреси джерела, адреси призначення та іншої інформації в заголовках пакетів. Кожен запис в таблиці маршрутизації відповідає певному потоку трафіку.

Рисунок 2.5 – Співпраця мережевого пристрою з архітектурою SDN та її контролером.

Важливо підкреслити, що канал управління SDN призначений виключно для обміну ключовими управлінськими даними. Ці дані включають такі аспекти:

- Передача запитів від мережевого пристрою SDN до контролера, спрямованих на пошук маршрутів для пакетів, що потрапили в область, де відсутні відповідні записи у таблицях маршрутизації.
- Отримання відповідей від контролера SDN щодо необхідних змін у таблицях маршрутизації при появі нових потоків або змін у мережевій структурі. Ця інформація зазвичай передається через канал SDN з використанням протоколу OpenFlow.

Протокол SDN може забезпечувати зв'язок між мережевими пристроями SDN та контролером, використовуючи різноманітні мережеві технології, такі як LAN, WAN, провідні, оптичні або бездротові з'єднання.

Мережевий пристрій надсилає контролеру дані про свій поточний стан через канал SDN, використовуючи спеціальні повідомлення. Ці повідомлення містять інформаційний заголовок, який вказує на їх мету, а також корисну інформацію.

Корисні дані включають різноманітні поля, які описують стан системи та його параметри. Цей підхід дозволяє передавати різні типи інформації про стан системи, забезпечуючи гнучкість і універсальність у використанні.

2.3 Організація комутатора у програмно-визначених мережах

OpenFlow-коммутатор – це особливий мережевий пристрій, який використовує протокол OpenFlow для зв'язку з контролером. Він реалізує функції аналізу та пересилання пакетів відповідно до набору правил, що зберігаються в таблицях потоків

та таблиці груп. Взаємодія між OpenFlow-коммутатором і контролером відбувається через протокол OpenFlow, що забезпечує ефективне управління мережею. OpenFlow-коммутатори можуть бути розроблені на базі протоколу OpenFlow або мати його сумісність.

Для ефективної роботи комутатора OpenFlow необхідна співпраця трьох ключових складових: таблиць потоків, розташованих на комутаторах, контролера та протоколу OpenFlow для безпечного обміну даними між контролером та комутаторами. Таблиці потоків, які зберігаються на комутаторах, дозволяють визначати дії, які треба виконати з певними типами даних. Контролери взаємодіють з комутаторами через протокол OpenFlow, керуючи потоками даних. Крім того, контролер може оптимізувати маршрутизацію через мережу з урахуванням різних факторів, **1** таких як швидкість передачі, мінімальна кількість переходів або мінімальна затримка.

Рисунок 2.6 – Ключові елементи структури комутатора з протоколом OpenFlow

Структура комутатора OpenFlow включає таблиці потоків для обробки пакетів та безпечний канал, що з'єднує його з зовнішнім контролером рис. 2.6. **1** Контролер керує перемиканням даних **1** через цей канал за допомогою **1** протоколу OpenFlow. Використовуючи цей протокол, контролер може вносити, оновлювати та видаляти правила потоків у таблицях комутатора.

Таблиця потоків у комутаторі містить записи, які визначають, які дії застосовувати до пакетів з певними значеннями заголовків. Кожен запис включає лічильники активності та набір дій, які виконуються над пакетами. При отриманні пакету комутатор порівнює його з записами у таблиці **1** потоків. Якщо відповідний запис знайдено, виконуються відповідні дії (наприклад, пересилання пакета через певний порт). У випадку відсутності відповідних записів **1** пакет пересилається на контролер через безпечний канал. Контролер вирішує, як обробляти такі пакети та **1** керує таблицею потоків комутатора, додаючи або видаляючи записи потоків.

Записи потоку вказують, куди направити пакети. Ці напрямки можуть включати фізичні порти, логічні порти, інші комутатори або зарезервовані порти, визначені специфікацією. Крім того, **1** дії, пов'язані з записами потоку, можуть спрямовувати **1** пакети до груп для **1** додаткової обробки. Групи представляють **1** собою набори **1** дій для потокового оброблення та можуть мати складні семантики пересилання, такі як **1** багатопроменеве поширення, швидке перенаправлення та **1** агрегація каналів. Крім того, групи **1** дозволяють декільком записам потоку спрямовувати **1** пакети **1** на один ідентифікатор, що **1** спрощує зміну вихідних дій **1** в записах потоку.

1 2.4 Синхронізація елементів під час проходження пакетів через мережу

Контролер OpenFlow виконує функцію керівної одиниці, що відповідає за обчислення оптимальних маршрутів та передачу даних до комутаторів OpenFlow. Ці комутатори відповідають за перенаправлення пакетів, використовуючи отримані від контролера правила потоків [7].

Рисунок 2.7 – Ключові елементи структури комутатора з підтримкою OpenFlow

В структурі комутатора OpenFlow присутня таблиця потоків, що включає записи з інформацією про відповідність полів, лічильників та наборів інструкцій для обробки пакетів.

Комутатор OpenFlow може містити **2** кілька таблиць потоків, які опрацьовуються послідовно. Наприклад, коли пакет потрапляє на комутатор і відповідає запису потоку в першій таблиці, йому надається інструкція про продовження обробки в наступних таблицях. Процес обробки відбувається до тих пір, поки він не зупиниться на відповідній таблиці.

Також присутня таблиця груп, яка містить записи з пакетами дій або групами дій.

2 Пакети, що відповідають записам у таблиці груп, обробляються з урахуванням відповідних дій або груп дій.

OpenFlow комутатори мають підтримувати 3 типи портів відповідно до стандартів OpenFlow [7]:

– фізичні порти призначені для з'єднання комутаторів між собою та для з'єднання з зовнішньою мережею.

При передачі пакету до першої таблиці потоків вибирається **2** відповідність з найвищим пріоритетом, і набір інструкцій, що відповідає цій відповідності, направляє **2** пакет до наступної таблиці потоків рис 2.8. Якщо ж жодна з інструкцій **2** не спрямовує пакет до наступної таблиці, конвеєрна обробка припиняється, і застосовується відповідна дія до пакету, якщо така дія встановлена.

2 У випадку, коли пакет не відповідає жодному запису у **2** таблиці потоків, це вважається **2** "пропуском" таблиці. Якщо для такого запису зроблена спеціальна конфігурація, запис про пропуск таблиці може бути оброблений різними способами, **2** такими як відкидання пакету, прозоре пересилання його до контролера та інші.

Рисунок 2.8 – Процес пропуску пакетів у системі OpenFlow

2 Таблиця потоків є ключовим елементом протоколу OpenFlow. Кожен запис у таблиці потоків містить наступні складові рис. 2.9:

– поля відповідності (match fields), які включають вхідні порти та можуть також містити

додаткові відповідні поля або метадані, передані з попередньої таблиці:

– пріоритет (priority), що визначає пріоритетність запису у таблиці потоків;

– лічильники (counters), які вказують кількість пакетів, що відповідають даному запису, та оновлюються в режимі реального часу;

– інструкції (instructions), які визначають зміни в наборах дій або конвеєрній обробці;

– таймаут (timeouts), який вказує максимальний час життя потоку;

– куки (cookie), які використовуються контролером для фільтрації, модифікації або видалення потоку, хоча цей елемент не використовується під час обробки пакетів.

Рисунок 2.9 – Ключові складові запису потоку у таблиці потоків

Поля match fields та priority в таблиці потоків визначають умови відповідності та пріоритет для обробки пакетів.

2.5 Основні протоколи в архітектурі SDN

2.5.1 Протокол комутації OpenFlow

Наразі виникла неоднозначність стосовно вибору між протоколами OpenFlow і OF-CONFIG для різних мережних сценаріїв. Давайте розглянемо завдання, які вони вирішують і у яких контекстах кожен з них може бути вигідним.

Протокол OpenFlow представляє собою відкритий стандарт, введений ONF для стандартизації спілкування між контролером та мережевими пристроями у концепції SDN. У процесі розвитку мережових технологій OpenFlow інтегрується у програмне забезпечення та апаратне забезпечення Ethernet комутаторів, маршрутизаторів та бездротових точок доступу, розширюючи їх можливості як засіб реалізації SDN.

Передача інформації про стан системи від мережевого пристрою до контролера SDN через канал протоколу SDN, такого як OpenFlow, забезпечує швидку доставку повідомлень про зміни. Використання SDN-протоколу сприяє швидкому обміну інформацією про стан системи між пристроями та контролером.

Рисунок 2.10 – Основні елементи повідомлення про стан системи (1)

Рисунок 2.10 демонструє загальну структуру повідомлення про стан системи. Кожне повідомлення містить заголовок і корисне навантаження. У заголовку присутнє поле типу повідомлення, яке ідентифікує його як повідомлення про стан системи.

4 Рисунок 4 2.11 – Структура 4 повідомлення про стан системи (2)

На рис. 2.11 показана ще одна модифікована структура 4 повідомлення про стан системи. Ця структура аналогічна тій, яка була показана на рис. 2.10, але має додаткове поле - 4 поле довжини повідомлення та поле довжини 2 для повідомлення про стан 4 системи. Ці 4 поля допомагають контролеру визначити кінець 4 повідомлення про стан системи або 4 окремого 4 повідомлення про стан системи в межах цього повідомлення.

Рисунок 2.12 – Заголовок стану системи в протоколі OpenFlow

На рис. 2.12 наведено конкретний приклад заголовка 4 повідомлення про стан системи 2 для протоколу OpenFlow з більш детальним описом. У додаток до полів, які вже були описані на рис. 2.11, заголовок на рис. 2.12 включає 4 поле версії протоколу OpenFlow (OFP), яке вказує, що це повідомлення відповідає OFP, і 4 може також 4 вказувати версію OFP. Крім 4 того, він містить 4 поле ID транзакції, яке вказує номер транзакції для цього повідомлення. У повідомленні також міститься корисне навантаження, яке 4 включає щонайменше одне повідомлення про стан системи. Це повідомлення може мати структуру, яка показана на рис. 2.10 4 або на рис. 2.11.

Цей протокол призначений для вирішення більш високорівневих завдань 2 порівняно з OpenFlow. Він зазвичай використовується для побудови мережевого середовища 3 в цілому, конфігурації комутаторів та прийняття рішень, наприклад, щодо відкриття або закриття окремих портів.

2.5.2 OF-CONFIG протокол

Протокол OF-CONFIG (OpenFlow Management and Configuration Protocol) визначає наступні абстракції рис. 2.13:

– Віртуальний 3 комутатор OpenFlow – це абстракція вузла передачі даних OpenFlow. За допомогою протоколу OF-CONFIG можна налаштувати віртуальний комутатор OpenFlow, 3 щоб контролер OpenFlow міг з ним взаємодіяти та управляти ним за допомогою протоколу OpenFlow.

– Сумісний з OpenFlow коммутатор – це фізичний або віртуальний мережевий пристрій, 3 ресурси якого (порти, черги і т. д.) виділені одному або кільком логічним комутаторам OpenFlow. Протокол 3 OF-CONFIG дозволяє динамічно призначати ресурси OpenFlow-сумісного комутатора розміщеним на ньому віртуальним комутаторам OpenFlow.

– Конфігураційна точка OpenFlow – це 3 джерело повідомлень OF-CONFIG для OpenFlow-сумісних 3 комутаторів. 3 Взаємодія точок конфігурації OpenFlow з

контролерами OpenFlow наразі не регламентується специфікаціями ONF [7].

Рисунок 2.13 – Ключові аспекти протоколу OF-CONFIG

Очікується, що майбутні версії OF-CONFIG також включатимуть такі можливості, як виявлення комутаторів та топологій, конфігурація характеристик, обробка тригерів, ініціалізація мережі OpenFlow та підтримка більшої кількості конфігурованих тунелів.

Протокол OF-CONFIG визначає OpenFlow комутатор як абстракцію – логічний комутатор (Logical Switch). Одна фізична пристрій може включати кілька Logical Switch, кожен з яких відповідає за пересилку потоків даних різних елементів мережі - Logical Switch Capabilities. Загалом, взаємодія між рівнями управління і передачі даних здійснюється на основі двох протоколів:

– OF-CONFIG – цей протокол дозволяє конфігурувати окремі Logical Switch для створення надійного каналу передачі керуючої інформації.

– Протокол OpenFlow – він відповідає за керування переадресацією і модифікацією пакетів.

На рис. 2.14 наведена схема взаємодії різних компонентів системи конфігурації і управління перемикачем.

Рисунок 2.14 – Операційний контекст перемикача OpenFlow

Протокол OpenFlow встановлює вимоги до налаштувань комутатора, таких як IP-адреси контролерів OpenFlow, для забезпечення ефективного функціонування мережі. Він також надає можливість віддаленого конфігурування комутаторів OpenFlow через протокол OF-CONFIG. Особливість роботи полягає в тому, що сам протокол OpenFlow оперує на часових інтервалах потоку даних, тобто при додаванні або видаленні потоків. З іншого боку, OF-CONFIG працює на менш частому часовому інтервалі. Наприклад, його можна використовувати для побудови таблиць маршрутизації або при включенні/вимиканні портів, коли не потрібно реагувати на кожний окремий потік даних. Такий підхід дозволяє ефективно використовувати ресурси мережі та оптимізувати її роботу.

3 ОНОВЛЕНІ АРХІТЕКТУРИ МЕРЕЖ НА ОСНОВІ КОНЦЕПЦІЇ ПРОГРАМОВАНОЇ МЕРЕЖІ SDN

3.1 Впровадження концепції програмованої мережі з використанням інноваційних технологій та платформи SEBA

SEBA представляє собою інноваційну платформу для створення високошвидкісного доступу в мережі SDN.

Платформа SEBA забезпечує можливість підтримки як провідного, так і бездротового доступу, що дозволяє оптимізувати шлях трафіку безпосередньо до головної магістралі, уникнувши затримок, пов'язаних з обробкою віртуальних мережевих функцій на сервері[9].

Рисунок 3.1 – Інноваційні рішення для високошвидкісного доступу у мережах

Представлена на рис. 3.1 концепція SEBA є синтезом кількох проєктів, таких як VOLTNA, ONOS і TRELIS.

SEBA розроблена для забезпечення комплексного набору рішень, які включають у себе компоненти цих програмних продуктів. Основою SEBA є платформа на базі Kubernetes, де весь необхідний функціонал розгортається як контейнери на обчислювальних вузлах, що дозволяє використовувати оркестрацію Kubernetes для створення групи контейнерів (SEBA Pod).

Розглянемо кожен з проєктів – VOLTNA, ONOS і TRELIS – окремо.

ONOS виступає як контролер SDN, здатний керувати VOLTNA і Trellis. Призначений для вирішення потреб операторів, ONOS надає підтримку як для конфігураційного, так і для управлінського функціоналу в реальному часі, що усуває необхідність в запуску маршрутизації та комутації протоколів управління всередині мережевої інфраструктури.

Рисунок 3.2 – Структура та компоненти ONOS

Основні характеристики архітектури ONOS:

1. Включає в себе корисні додатки, такі як реактивне перенаправлення, ProхуARP, сегментна маршрутизація, SDN-IP та інші.
2. Забезпечує передачу мережевої інформації на рівень програмного забезпечення, а також надає інтерфейс для управління нижнім рівнем компонентів.
3. Має широкий спектр основних функцій та підтримує розподілену кластеризацію для забезпечення високої доступності і масштабованості.
4. Пропонує абстрактний інтерфейс для управління мережевою інфраструктурою.
5. Реалізує мережеві протоколи для керування мережевими пристроями, такі як OpenFlow, NetConf та інші.

VOLTNA представляє собою віртуальний абстрактний шар для реалізації ширококутового доступу в рамках житлової інфраструктури CORD.

Цей інструментарій надає абстракцію на рівні PON для підтримки Ethernet і керування цими ресурсами через контролер.

У візуалізації VOLTNA API відображаються на спільну модель основних даних через використання адаптерів на південь. У свою чергу, VOLTNA забезпечує зв'язок з пристроями PON за допомогою спеціальних постачальників або адаптерів OLT та ONU.

Ця спільна система контролю та управління дозволяє координувати роботу всіх OLT та ONU в мережі.

Рисунок 3.3 – Структура та функціональні компоненти VOLTNA

Архітектура VOLTNA полягає у взаємодії між різними модулями, зокрема адаптерами, які виконують функції управління пристроями PON.

У північному інтерфейсі VOLTNA відображає мережу PON як програмований Ethernet-перемикач на контролері SDN.

Адаптери VOLTNA на півдні взаємодіють з апаратними пристроями PON, використовуючи специфічні протоколи через адаптери OLT та ONU.

Trellis пропонує мережевим операторам інноваційний підхід до управління мережами, що дозволяє отримати переваги в порівнянні з традиційними методами мережевого керування.

Рисунок 3.4 – Структура та функціональні компоненти системи Trellis

Trellis використовує сучасний підхід до мережевого керування, заснований на контролері SDN (наприклад, ONOS), який оперує незалежно від апаратного обладнання. У цій архітектурі набір додатків, що функціонують на контролері SDN, реалізує всі необхідні можливості мережевої інфраструктури, такі як комутація Ethernet, IP-маршрутизація, багатоадресна передача і т.д.

Внутрішня архітектура Trellis використовує концепцію маршрутизації сегментів (SR), де **1** глобально значущі мітки MPLS надаються кожному листовому та хребтовому перемикачу. Це дозволяє мінімізувати **1** стан міток у мережі в **1** порівнянні **1** з традиційними мережами MPLS, де мітки, які мають локальне значення, потрібно міняти на кожному вузлі. У Trellis листові перемикачі надають ярлики MPLS, що вказують на призначення ToR для трафіку **1** IPv4 або IPv6.

1 SEBA об'єднує функціональність трьох проектів і використовує найкращі аспекти кожного з них, щоб створити модульну архітектуру, що **1** представлена на наступному рисунку.

1 Рисунок 3.5 – Схема концептуальної архітектури

Ця концептуальна схема 1 архітектури вимагає дотримання кількох ключових принципів та визначень:

– Інфраструктурний шар описується через фізичні компоненти, такі як 1 вузли доступу, перемикачі агрегації та обчислювальні 1 ресурси.

1 – 1 Сервісний шар визначає зв'язок 1 між компонентами та рівнем 1 інфраструктури для надання послуг.

1 – Контролер SDN забезпечує автономність 1 управління для кожного компонента інфраструктурного шару, який бере участь у наданні сервісів.

– Функціональний блок ASG відповідає за 1 агрегацію, перемикання та маршрутизацію в площині даних, а також управління трафіком у 1 площині управління в межах POD, а 1 також забезпечує 1 можливості Service Edge.

1 – Контролер управління платформою (VMC) представляє собою функціональний інтерфейс 1 до функцій управління обладнанням та є ключовим терміном для галузі.

SEBA Pod містить 1 обладнання OLT та ONU, яке керується VOLTNA для управління мережею PON. Крім того, в SEBA Pod присутній контролер SDN ONOS, що включає пару API та додатків, які розкривають можливості, які надаються мережею 1 PON. Наприклад, OLT з'єднується з вимикачами AGG, що є частиною тканини мережі або листовою частиною хребта, які також керуються контролером SDN. Після цього комутатори можуть бути підключені 1 до зовнішніх маршрутизаторів, BNG, локальних комп'ютерів тощо.

1 У SEBA і R-CORD існують відмінності в маршрутизації абонентського трафіку. У SEBA трафік 1 абонента просто проходить через апаратне забезпечення та виходить в Інтернет, надаючи абоненту "швидкий шлях" до Інтернету. Трафік лише для обчислювальних вузлів переходить до віртуальних середовищ, наприклад, коли абонентам 1 надаються сторонні послуги.

1 Крім того, варто розглянути R-CORD як альтернативний дизайн, який створений на основі концепції SEBA. Давайте детальніше розглянемо, що саме 1 включає в себе R-CORD.

R-CORD є відкритим вирішенням, побудованим на базі платформи CORD, яке спрямоване на надання широкосмугових житлових послуг. Ця платформа трансформує крайову мережу оператора в гнучку і ефективну 1 платформу надання послуг, що дозволяє операторам 1 забезпечувати 1 кращий досвід для кінцевих користувачів,

включаючи інноваційні послуги нового покоління.

В проєкті R-CORD особлива увага приділяється віртуалізації обладнання **1** для фізичного підключення абонентів, такого як GPON або DOCSIS. Наприклад, проєкт VOLTHA використовується для керування цим спеціалізованим обладнанням як керованим ресурсом OpenFlow.

Також, R-CORD включає **1** віртуальний шлюз абонентів (vSG) та **1** використовує **1** віртуальний маршрутизатор (vRouter) як основний мережевий сервіс. Перший **1** реалізований у вигляді контейнера, **1** який прив'язаний до кожного абонента, а останній керується **1** додатком ONOS.

1 Обладнання для споживчих приміщень (ОСП), відоме як "домашній маршрутизатор" або "житловий шлюз" в житловому середовищі, виконує набір основних та додаткових функцій, таких як брандмауер і батьківський контроль, в ім'я мешканців абонентів. Також він може включати більш складні підприємницькі функції, такі як WAN. Розширення можливостей ОСП у хмарному середовищі дозволяє забезпечити нові послуги з додатковою вартістю та полегшити обслуговування клієнтів.

Наша віртуальна версія обладнання для споживачів, яку ми називаємо віртуальним шлюзом абонентів (vSG), виконує набір обраних абонентами функцій, але це відбувається на центральному обладнанні, розташованому у центральному офісі, а не в приміщенні клієнта. На місці у клієнта залишається лише пристрій (який ми також називаємо CPE), проте він може бути зведений до простого металевого перемикача. Більшість функцій, які раніше працювали на CPE, тепер виконуються у віртуальному обчислювальному екземплярі, такому як віртуальна машина або контейнер, на комерційних серверах у центральному офісі.

Послуга vRouter функціонує як шлюз між інфраструктурою CORD та висхідною мережею, забезпечуючи доступ до Інтернету для абонентів та інших сервісів в межах CORD. Ця послуга є кінцевою стадією обслуговування в ланцюжку трафіку користувача перед виходом з системи CORD і фізично представляє собою інтерфейс між CORD і провайдером вищого рівня. vRouter надає Інтернет-послугу іншим службам в CO та реалізується як програма управління мережею, що працює на ONOS.

Послуга vRouter має дві ключові складові - площину управління та площину даних, які функціонують незалежно одна від одної.

У площині даних vRouter керує декількома пристроями, які виконують функції маршрутизації і виглядають як єдиний маршрутизатор зовнішнього світу. Ці пристрої підтримують різноманітні протоколи маршрутизації, що дозволяє vRouter працювати з різними мережевими пристроями без необхідності власної імплементації цих

протоколів.

Основний функціонал vRouter полягає в обміні маршрутною інформацією з іншими маршрутизаторами. Щоб уникнути необхідності реалізації протоколів маршрутизації в ONOS, ми використовуємо відкритий стек маршрутизації Quagga, який підтримує різноманітні протоколи маршрутизації.

Наразі vRouter працює в середовищах з невеликою кількістю маршрутів, що дозволяє уникнути проблем з продуктивністю, які можуть виникнути в масштабах Інтернету.

Quagga буде налаштований для взаємодії з маршрутизаторами на вищому рівні мережі - у нашій польовій пробній справі це буде використання OSPF та iBGP.

Рисунок 3.6 – Структура управління vRouter

Архітектура управління відображає ті ж принципи, які використовуються в ONOS SDN-IP, з окремим уточненням нашого випадку. Основна різниця полягає у **1** тому, що ми враховуємо не лише пирінг BGP, а й потребу підтримки IGP. Там, де SDN-IP використовує iBGP для зв'язку **1** між Quagga та ONOS, в нашому випадку ми користуємося інтерфейсом FPM з використанням vRouter.

Управління трафіком. Перш ніж Quagga зможе обмінюватися будь-якими маршрутами з маршрутизаторами вищого рівня, програма vRouter спочатку налаштовує площину даних для пропуску трафіку **1** між сервером Quagga та зовнішнім маршрутизатором. Для **1** цього **1** сервер Quagga підключений до порту на панелі даних vRouter, і **1** вхідні / **1** вихідні пакети маршрутизації спрямовуються на / з цього порту. Цей **1** процес обходить стандартну **1** функцію маршрутизації vRouter, оскільки стосується трафіку управління площиною, який призначений для самого маршрутизатора.

1 Рисунок 3.7 – Управління трафіком маршрутизації здійснюється через пересилання на сервер Quagga, який підключений до панелі даних

Заснована **1** на інфраструктурній платформі CORD, M-CORD **1** революціонізує мобільні мережі **1** шляхом дезагрегації та віртуалізації функцій стільникової мережі та послуг операторів. Це створює можливість для гнучкого розгортання служб, які відповідають конкретним потребам і можуть динамічно масштабуватися. M-CORD надає фундамент для розвитку мереж та сервісів 5G, підтримуючи дезагреговане та віртуалізоване ядро пакетної комутації, розділення з **1** кінця до кінця від RAN до EPC, мобільні обчислення на краєвому рівні та програмовану мережу радіодоступу.

1 Stratum – це відкрита операційна система для комутаторів, розроблена для програмно визначених мереж. Ця система спрямована на створення готового до

виробництва розподілу для білих коробкових комутаторів. Stratum надає широкий набір інтерфейсів нового покоління SDN, що дозволяє забезпечити взаємозамінність пристроїв переадресації та програмованість їхньої поведінки.

Ця система допомагає уникнути блокування, що часто відбувається з постачальниками, шляхом надання відкритих інтерфейсів та програмних API, що дозволяють легко інтегрувати пристрої в операторські мережі. Stratum представляє повноцінне рішення для перемикачів білого поля, реалізуючи концепцію SDN через програмне забезпечення.

Проект Stratum розширює можливості використання SDN, надаючи повний набір інтерфейсів для управління життєвим циклом, конфігурацією та операціями.

Ініціатива "Відкрита та роз'єднана оптична транспортна мережа" (ODTN) є стратегічним проектом, який створений операторами з метою розвитку інноваційних оптичних мереж. Основна мета полягає в тому, щоб сприяти розвитку оптичної мережі, яка буде гнучкою, відкритою та готовою до співпраці з різними постачальниками обладнання. ODTN спрямована на зниження витрат, роз'єднання та відкритість оптичних мереж за допомогою стандартизованих рішень та відкритого програмного забезпечення для управління та взаємодії з різними компонентами мережі.

ODTN створить оптимізовану екологічну систему "периферійних пристроїв", що дозволить поєднувати різні компоненти та інтегрувати їх у складні рішення. Ця ініціатива дозволить постачальникам фокусуватися на розробці конкретних компонентів (наприклад, транспондерів), не обмежуючись створенням повних рішень, що сприятиме швидшим інноваціям та зниженню витрат. Оператори отримають можливість вибирати компоненти найвищого класу та уникнуть залежності від одного постачальника, що забезпечить їм гнучкість при розширенні їх мережі.

Рисунок 3.8 – Архітектура ODTN

Вимоги до оптичних мереж, що виникають внаслідок аналогового характеру зв'язку на великі відстані, створили потребу в нових підходах на ринку. ODTN пропонує рішення, яке дозволяє використовувати збірну пару транспондерів від різних постачальників для кожного оптичного посилання. Це дозволяє мережі користуватися різними марками транспондерів для різних відрізків мережі, забезпечуючи при цьому відкриту лінійну систему.

ODTN використовує контролер SDN ONOS для автоматизації та управління всією транспортною мережею, надаючи можливість вибору між різними постачальниками. Проект спирається на відкриті стандарти галузі, **1** такі як TAPI (Transport API) і OpenConfig, щоб створити **1** нейтральне **1** для постачальників рішення. Реалізація

ODTN розпочнеться з простих "точка-точка" систем відкритої лінії та поступово розвиватиметься до більш складних мережевих сценаріїв, включаючи розчленоване оптичне обладнання.

Mininet – це інструмент для створення віртуальних прототипів і тестування програмно-визначених мереж (SDN). Він надає можливість швидкого прототипування та тестування SDN на ноутбуці чи ПК без необхідності підключення до фізичної мережі. Mininet дозволяє розробникам працювати над однією топологією одночасно, використовуючи розширюваний API Python для створення мереж та експериментів. Цей інструмент розповсюджується під ліцензією BSD Open Source та активно підтримується спільнотою ентузіастів мереж і SDN.

Мережа Mininet включає в себе наступні компоненти:

– Ізольовані середовища: Група процесів на рівні користувача розміщена в мережевому просторі імен, що забезпечує виключну власність інтерфейсів, портів та таблиць маршрутизації.

– Емульовані зв'язки: Linux Traffic Control (tc) встановлює швидкість передачі даних для кожного зв'язку, що дозволяє формувати трафік з відповідною швидкістю. Кожен емульований хост має свій віртуальний інтерфейс Ethernet.

– Емульовані комутатори: Linux Bridge або Open vSwitch, які працюють у ядрі або у просторі користувача, використовуються для перемикання пакетів між інтерфейсами.

3.2 Віртуалізація мереж та функцій SDN

Паралельно з терміном SDN існує також термін NFV, що означає віртуалізацію функцій мережі (Network Function Virtualization). Основна мета NFV полягає в трансформації традиційних, дорогих мережевих пристроїв у віртуальні функції, що запускаються на віртуалізованих серверах.

Віртуалізація мережевих функцій (NFV) – це концепція перетворення мережевої архітектури за допомогою технологій віртуалізації IT. Вона полягає в тому, щоб віртуалізувати функції мережевих вузлів у будівельні блоки, які можуть поєднуватися для створення різноманітних комунікаційних послуг. Наприклад, за допомогою NFV можна перемістити служби, такі як балансування навантаження і міжмережевий екран, з фізичного обладнання в віртуалізоване середовище. Рішення NFV часто розгортаються у центрах обробки даних для хмарних платформ, які використовуються як на підприємствах, так і у постачальників послуг.

1 Рисунок 1 3.9 – Спільна 1 архітектура SDN та NFV"

1 NFV та SDN – це дві технології, які часто використовуються разом і можуть доповнювати одна одну. Багато платформ NFV включають 1 контролери SDN. З одного боку, хоча NFV може функціонувати і без SDN, використання принципів SDN може покращити 1 продуктивність, спростити сумісність з існуючими системами 1 та полегшити експлуатацію та технічне 1 обслуговування. З іншого боку, NFV може служити 1 базою для розгортання 1 SDN, надаючи інфраструктуру для 1 запуску SDN-програмного забезпечення. 1 Крім того, NFV відповідає 1 цілям SDN щодо використання стандартних 1 серверів і комутаторів. Передбачається, 1 що ці технології зможуть інтегруватися 1 з системами оркестрації, такими як хмарні платформи управління або платформи оркестрації мережевих служб, для створення ще більш оптимізованої мережевої інфраструктури.

NFV впливає на еволюцію мережевих технологій, сприяючи переходу від традиційних апаратних засобів до програмованих, віртуалізованих середовищ. При комбінації з SDN, NFV відкриває широкі можливості для 1 автоматизації та програмування мережі. Ця технологія дозволяє операторам та постачальникам послуг створювати гнучкі мережеві інфраструктури, які можуть динамічно адаптуватися 1 до потреб та вимог клієнтів. Зокрема, великі оператори мереж виявляють інтерес до NFV через її підтримку програмованих відкритих стандартів, що дозволяє їм уникнути обмежень, пов'язаних з 1 власницькими мережевими платформами.

1 SDN є важливим інструментом для реалізації функцій, пов'язаних з NFV. Фактично, 1 багато випадків використання SDN можуть включати концепції, 1 що походять від 1 NFV. Наприклад, централізований контролер може 1 керувати 1 розподіленою функцією пересилання пакетів, яка також 1 може бути віртуалізованою на 1 пристрої обробки або маршрутизації.

1 Перехід на технології SDN і NFV в мережах операторів не можна очікувати, щоб пройти гладко і безболісно. Це схоже на перехід від аналогових до цифрових технологій у телефонних станціях, який 1 зайняв роки і навіть десятиліття. Багато з нас пам'ятають, як повільно розвивалася цифрова трансформація у мережах операторів. Так само і з переходом до NFV в операторських мережах: цей процес буде поступовим.

Поступово, з виробленням 1 терміну служби або моральним старінням апаратних платформ, таких як IMS/EPC, агрегаційних і доступових мереж, оператори будуть переходити 1 на "хмарні" платформи SDN/NFV для віртуалізації vIMS/vEPC. Однак основною проблемою тут буде доступність 1 власних дата-центрів у операторів або можливість використання дата-центрів у постачальників хмарних послуг 1 для аутсорсингу ресурсів стандартних серверів і систем зберігання даних.

Посилання

Це джерела виділених збігів у вашому документі. Кожен збіг позначено темно-зеленим числом, яке відповідає вказаному тут джерелу. Джерела впорядковані за схожістю — чим вищий бал, тим сильніше збіг.

#	Джерело	%
1	tk-its.kpi.ua	12.3%
2	tk-its.kpi.ua	2.1%
3	ela.kpi.ua	2.0%
4	conferenc.its.kpi.ua	1.8%
5	ela.kpi.ua	1.7%
6	journals.dut.edu.ua	0.3%
7	dspace.nau.edu.ua	0.3%
8	duikt.edu.ua	0.1%
9	studme.com.ua	0.1%
10	ela.kpi.ua	0.1%
11	itgip.org	0.1%



Дякуємо, що перевірили
свій документ за допомогою
Plag!