

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут телекомунікаційних систем

(повне найменування інституту, факультету)

Кафедра телекомунікацій

(повна назва кафедри)

До захисту допущено
В.о. завідувача кафедри

_____ Явіся В.С.
(підпис) (ініціали, прізвище)

“4” червня 2020 р.

Дипломна робота

на здобуття освітнього ступеня “бакалавр”
(назва ОС)

Спеціальність 172 Телекомунікації та радіотехніка

(код і назва)

на тему: Принципи побудови і методи управління мережами SDN

Виконала: студентка 4 курсу, групи ТЗ-61
(шифр групи)

_____ Кочура Марина Юріївна _____

(прізвище, ім'я, по батькові)

(підпис)

Керівник професор каф ТК, д.т.н., професор, Романов О.І.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Рецензент доцент каф ТС к.т.н., доцент Созонник Г. Д.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

(підпис)

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут телекомунікаційних систем

(повна назва)

Кафедра телекомунікацій

(повна назва)

Освітній ступінь бакалавр

Спеціальність 172 Телекомунікації та радіотехніка

(код і назва)

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Явіся В.С. _____

(прізвище ініціали) (підпис)

“22” січня 2020р.

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Кочури Марини Юріївни

(прізвище, ім'я, по батькові)

1. Тема роботи Принципи побудови і методи управління мережами SDN

Керівник роботи Романов Олександр Іванович, д.т.н. професор

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

затверджені наказом по університету від 30 березня 2020 р. №924-с

2. Термін подання студентом роботи 04.06.2020

3. Вихідні дані до роботи: мережі SDN, протоколи OpenFlow та OF-CONFIG,
віртуалізація NFV

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

4.1. Аналіз напрямків розвитку телекомунікаційної мережі

- 4.2. Концепція побудови мереж SDN.
- 4.3. Структура і основні елементи мереж SDN.
- 4.4. Принцип побудови контролера SDN.
- 4.5. Принцип побудови комутатора SDN.
- 4.6. Взаємодія елементів в процесі обслуговування пакетів.
- 4.7. Протоколи мережі SDN
- 4.8. Порівняльна характеристика протоколів.
- 4.9. Сучасні платформи побудови мереж SDN.
- 4.10. Платформа SEBA.
- 4.11. Побудова SDN з віртуалізацією мережевих елементів.

5. Перелік графічного (ілюстративного) матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): Презентація по темі “Принципи побудови і методи управління мережами SDN” в кількості 8 слайдів

- 5.1. Багаторівнева архітектура телекомунікаційних мереж
- 5.2. Архітектура мережі SDN.
- 5.3. Структура мережі SDN, перелік елементів та їх функції.
- 5.4. Структура протоколів мережі SDN.
- 5.5 Структурна схема елементів мережі SDN.
- 5.6. Взаємодія елементів мережі в процесі обслуговування навантаження.
- 5.7. Сучасні платформи мереж SDN.

Дата видачі завдання 22.01.2020

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Вступ	23.01.2020-05.02.2020	виконано
2.	Вивчення особливостей архітектури програмно-конфігуруючих мереж	06.02.2020-10.03.2020	виконано
3.	Пошук інформації про протоколи OpenFlow, OF-Config	11.03.2020-01.04.2020	виконано
4.	Пошук інформації про рівень мережевих сервісів	02.04.2020-27.04.2020	виконано
5.	Аналіз процесу обслуговування пакетів у мережі SDN на основі комутаторів OpenFlow	28.04.2020-20.05.2020	виконано
6.	Висновки	21.05.2020-03.06.2020	виконано

Студент

Керівник роботи

Кочура М.Ю.
(прізвище та ініціали)

Романов О. І.
(прізвище та ініціали)

РЕФЕРАТ

Дипломна робота містить: 60 сторінок, 25 рисунків, 20 посилань.

Високий темп розвитку сервісів, масштабів їх охоплення, а так само зростання кількості та варіативності контенту призвели до зміни концепції організації обчислень — місце застарілої клієнт-серверної архітектури зайняли центри обробки даних (ЦОД) і хмари [1].

З розвитком Інтернету моніторинг і управління мережею стали вузьким місцем в класичному мережевому підході. Це призвело до появи нової концепції побудови мереж на базі архітектури SDN (Software defined network).

Особливістю цієї архітектури є відділення рівня управління від рівня передачі даних. При цьому вся логіка управління мережею зосереджена в контролері, а всі інші пристрої мережі використовуються лише для передачі даних.

Застосування Software-Defined Networks повинно забезпечити наступні можливості [3]:

- гнучке управління мережами та мережевими ресурсами;
- швидке впровадження нових сервісів;
- незалежність від виробників мережевого обладнання;
- широкі можливості забезпечення мобільності користувачів;
- швидке впровадження нових технологій.

Це досягається за рахунок нового підходу до розподілу функцій передачі та контролю між пристроями мережі SDN.

Таким чином, технологія SDN дозволяє значно підвищити ефективність функціонування телекомунікаційних мереж, пропускну спроможність та якість обслуговування. Мережі SDN – це майбутнє телекомунікацій.

Ключевые слова: SDN, OpenFlow ,маршрутизатор.

ABSTRACT

The diploma work contains: 60 pages, 25 pictures, 20 links.

The high rate of development of services, the extent of their coverage, as well as the increase in the number and variability of content have led to a change in the concept of organizing computing – data centers and clouds have replaced the outdated client-server architecture.

With the development of the Internet, network monitoring and management has become a bottleneck in the classic network approach. This led to the emergence of a new concept for building networks based on the SDN (Software defined network) architecture.

A feature of this architecture is the separation of the control layer from the data transfer layer. In this case, the entire network management logic is concentrated in the controller, and all other network devices are used only for data transmission.

Application of Software-Defined Networks must provide the following possibilities:

- flexible management of networks and network resources;
- rapid implementation of new services;
- independence from manufacturers of network equipment; □ wide possibilities of ensuring mobility of users;
- rapid introduction of new technologies.

This is achieved through a new approach to the distribution of transfer and control functions between SDN devices.

Thus, SDN technology allows to significantly improve the efficiency of telecommunication networks, bandwidth and quality of service. SDNs are the future of telecommunications.

Key words: SDN, OpenFlow , router, network

Зміст

Список скорочень.....	8
1. Аналіз напрямків розвитку телекомунікаційної мережі	10
1.1 Концепція побудови мережі NGN	10
1.2 Історичні умови появи техно логії SDN. Концепція мереж SDN.	14
Висновки до 1-го розділу :	17
2. Структура і основні елементи мереж SDN.....	18
2.1 Структура мережі SDN.	18
2.2. Структур а контролер а SDN.....	21
2.3 Структура коммутатора SDN.....	24
2.4 Взаємодії елементів в процесі обслуговування пакетів.	26
Висновки до 2-го розділу:	30
3 Протоколи мережі SDN.....	31
3.1 Протокол OpenFlow	31
3.2 Протокол OF-CONFIG	33
3.3 Порівняльні характеристики протоколів.	35
Висновки до 3-го розділу:	36
4. Сучасні платформи побудови мереж SDN	37
4.1 Побудова мережі SDN з використання платформи SEBA	37
4.2 SDN та мережеві функції віртуалізації	53
Висновки до 4-го розділу:	55
Висновок	57
Список використаної літератури.....	58

					КПІ ім. Ігоря Сікорського № 1068-с 1.ТЗ-61.2020.ПЗ			
	Лист	№ докум.	Підпис	Дата				
Розроб.	Кочура М.Ю.				Принципи побудови і методи управління мережами SDN	Літ.	Арк.	Акрушів
Перевір.							6	66
Реценз.	Льченко М. Ю.							
Н. Контр.	Петрова В.М.							
Затверд.	Явіся В.С.							

Список скорочень

API - інтерфейс прикладного програмування

ONF (Open Networking Foundation)

SDN - Software Defined Networks/Networking

VNF (Virtual Network Functions)

VIM (Virtualized Infrastructure Manager)

vOLT(Virtual OLT)

vSG (Virtual Subscriber Gateway)

vRouter (Virtual Router)

NGN (Next Generation Network)

Вступ

Мережі потребують більшої швидкості передачі даних та вдосконалення інструментів, що використовуються для мережевого управління і моніторингу. Така ситуація призводить до появи нових функціональних і технологічних мереж, з ускладненою інфраструктурою. Старі методи моніторингу і управління не відповідають новим вимогам.

Тому останнім часом зростає популярність програмно-конфігурованих мереж SDN (Software-Defined Networks). Сучасні тенденції розвитку інформаційних технологій на очах змінюють корпоративні мережі. Йде нарощування обсягів мережевого трафіку, тому все частіше виникає необхідність конфігурації все більш складних мережевих структур.

Спростити цю задачу можуть технології програмно-конфігурованих мереж SDN (Software-Defined Networking) і функціональна віртуалізація мереж NFV (Network Function Virtualization), які дозволяють перевести мережеві елементи під контроль, зробити їх більш інтелектуальними і гнучкими, полегшити управління ними.

Ключові принципи програмно-конфігурованих мереж поділ процесів передачі та управління даними, централізація управління мережею за допомогою програмних засобів. Протокол OpenFlow, який реалізує незалежний від виробника інтерфейс між логічним контролером мережі і мережевим інфраструктурою, є однією з реалізацій концепції програмно-конфігурується мережі і вважається рушійною силою її поширення і популяризації. В результаті виходить гнучка, керована, адаптивна та економічна архітектура, яка здатна ефективно адаптуватися під передачу великих потоків різних видів трафіку.

1. Аналіз напрямків розвитку телекомунікаційної мережі

1.1 Концепція побудови мережі NGN

NGN(мережа наступного покоління) - мережа з комутацією пакетів, придатна для надання телекомунікаційних послуг та використання безлічі широкосмугових транспортних технологій [2].

Передумовами появи NGN були: вузька спеціалізація сучасних мереж, наявність великої кількості виділених мереж та витрати на утримання різноманітних мереж. В основу концепції побудови NGN покладено ідею про створення універсальної мережі, яка б дозволяла переносити будь-які види інформації, такі як мовлення, дані, відео, аудіо, графіку тощо, а також забезпечувати можливість надання самого широкого спектра інфокомунікаційних послуг.

Ідеологічними принципами побудови мережі NGN було наступне:

- Підключення до мережі повинно бути максимально простим і зручним без використання проміжних систем
- Спочатку будується базова пакетна транспортна мережа, а потім поверх цієї мережі будується потужний комплекс сервісів.

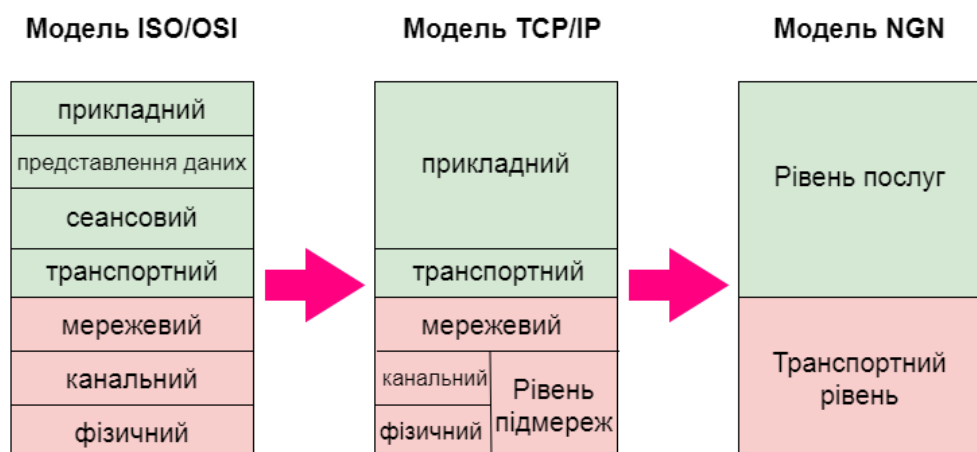
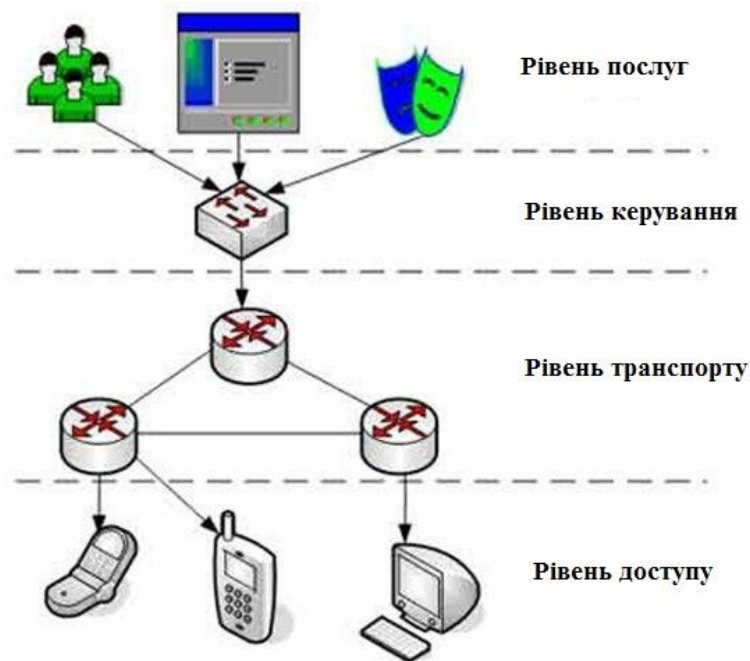


Рис.1.1 Еволюція від моделі ISO/OSI до NGN

NGN охоплює різноманітні типи мереж: від дротового до бездротового, і від телекомунікаційного до комп'ютерного, і вона передаватиме різноманітні різні послуги через загальну та відкриту мережеву інфраструктуру.

NGN включає в себе чотири рівні:



1.2 Архітектура мережі NGN

Отже, розглянемо безпосередньо кожен з цих рівнів детальніше [2]:

1. Останнім рівнем NGN прийнято вважати прикладний рівень. Його завдання - забезпечення всього спектра послуг, доступного на мережах наступного покоління. Ідеологія побудови NGN забезпечує можливість надання абонентам послуг Triple-Play (передача мови, даних і відео) на базі мультисервісних мереж, створюваних шляхом модернізації існуючих мереж електрозв'язку.

Перехід до NGN відкриває практично необмежені можливості по реалізації послуг і для корпоративного сектора. У традиційних мережах такі послуги надаються локальними операторами, і їх підключення нерідко вимагає великих тимчасових або фінансових витрат.

2. Рівень управління послугами відповідає за надання кінцевому користувачеві інформаційних послуг, і від того, наскільки ці послуги зацікавлять його, залежить подальший розвиток мережі. Перехід до NGN

відкриває практично необмежені можливості по реалізації послуг і для корпоративного сектора. У традиційних мережах такі послуги надаються локальними операторами, і їх підключення нерідко вимагає великих тимчасових або фінансових витрат. У разі використання однорідного IP-середовища існує єдиний набір послуг для всіх користувачів. Механізм їх підключення також помітно спрощується: достатньо вибрати цікаву послугу зі списку і послати відповідний запит. Вже сьогодні популярні нові широкосмугові послуги: «відео на вимогу», «розширене телебачення» (ТБ Інтернет), ТБ - комерція тощо.

Сервери, що забезпечують саме надання послуг, можуть перебувати як усередині, так і за межами самої мережі (веб-сервери, сервери, що належать ASP-провайдерам). Важливою складовою рівня управління послугами також є інформаційні центри або центри управління послугами (data centers, services control point) — це власні інформаційні ресурси мережі, на основі яких здійснюється обслуговування користувачів. У таких центрах може зберігатися інформація двох типів:

- інформація користувача, тобто ті дані, які безпосередньо цікавлять користувачів мережі;
- допоміжна службова інформація, яка дозволяє надавати користувачам додаткові послуги.

Прикладом інформаційних ресурсів першого типу можуть служити веб-портали, на яких розташована різноманітна довідкова інформація й новини, інформація електронних магазинів тощо. Раніше в телефонних мережах роль таких центрів відігравали служби екстреного виклику (наприклад, міліції, швидкої допомоги) і довідкові служби різних організацій і підприємств — вокзалів, аеропортів, магазинів тощо. У телевізійних мережах такими центрами були телестудії, які поставляли «живу» картинку або ж відтворювали раніше записані сюжети або фільми.

До ресурсів другого типу належать, наприклад, різні системи автентифікації й авторизації користувачів, за допомогою яких організація, що

володіє мережею, перевіряє права користувачів на одержання тих або інших послуг; системи білінгу, які в комерційних мережах підраховують плату за надані послуги; бази даних облікової інформації користувачів, які зберігають імена й паролі, а також переліки послуг, на які підписаний кожний користувач.

3. Завдання транспортного рівня — забезпечення прозорості передачі інформації користувача шляхом її комутації та маршрутизації.

Транспортний рівень NGN заснований на IP і може використовувати перевагу MPLS. Транспортний рівень формує ядро Мережі. В основному він складається з збірки маршрутизаторів з оптичною мережею, які відповідають за передачу трафіку, створюваного рівнем доступу. Оскільки одна і та ж базова мережа буде використовуватися для всіх типів абонентів, що користуються різними видами послуг в реальному часі і не в реальному часі, вона повинна мати можливість використовувати політики пропускну здатності і політики QoS.

Оператор повинен думати про керувану мережі для своїх абонентів. Базова транспортна інфраструктура пакетів і інфраструктура мультимедіа згруповані по транспортному рівню, який також взаємодіє з мережею з комутацією каналів через шлюзи мультимедіа, так що існуючі мережі можуть співіснувати і не повинні бути знищені.

Існують певні вимоги до можливостей транспортного рівня:

- підтримка з'єднань у реальному часі й з'єднань, нечутливих до затримок;
- підтримка різних моделей з'єднань: «точка — точка», «точка — багатоточка», «багатоточка — багатоточка», «багатоточка — точка»;
- гарантовані рівні продуктивності, надійності, доступності, масштабованості.

4. Рівень доступу. До рівня доступу належать [4,5]:

- шлюзи;
- вузли агрегування доступу;

- мережі доступу (МД), тобто мережі електрозв'язку, які забезпечують підключення термінальних пристроїв користувачів до приграничного вузла транспортної мережі.

Для організації рівня доступу можуть використовуватися різні середовища передачі. Це може бути мідна пара, коаксіальний кабель, волоконно-оптичний кабель, радіоканал, супутникові канали або будь-яка їхня комбінація .

Мережа доступу, як і NGN у цілому, може складатися з декількох рівнів. Комутатори, установлені у вузлах нижнього рівня, мультиплексують інформацію, що надходить по численних абонентських каналах (що часто називаються абонентськими закінченнями, local loop), і передають її комутаторам верхнього рівня, щоб ті, у свою чергу, передали її комутаторам транспортного рівня. Кількість рівнів мережі доступу залежить від її розміру; невелика мережа доступу може складатися з одного рівня, а велика — із двох-трьох. Наступні рівні здійснюють подальшу концентрацію трафіка, збираючи його й мультиплексуючи в більш швидкісні канали.

Головна архітектурна особливість NGN полягає в тому, що передача і маршрутизація пакетів і базові елементи транспортної інфраструктури (канали, маршрутизатори, комутатори, шлюзи) фізично і логічно відокремлені від пристроїв і механізмів управління викликами і доступом до послуг.

Мережі наступного покоління (NGN) представляють собою нову концепцію мережі, що комбінує в собі голосові функції, якість обслуговування (QoS) і комутовані мережі з перевагами і ефективністю пакетної мережі. Мережі NGN означають еволюцію існуючих телекомунікаційних мереж, що відображається в злитті мереж і технологій. Завдяки цьому забезпечуються широкий набір послуг, починаючи з класичних послуг телефонії та закінчуючи різними послугами передачі даних або їх комбінацією.

1.2 Історичні умови появи технології SDN. Концепція мереж SDN.

Програмно-конфігуровані мережі SDN (Software Defined Network) змінюють підхід до проектування та адміністрування мереж. По-перше, SDN відокремлює площину управління мережею (Control plane), яка займається маршрутизацією трафіку, від площини передачі даних (Data plane), яка передає трафік згідно з правилами, отриманим від площини управління. По-друге, SDN «консолідує» площину управління, при цьому один комплекс керуючих програм на сервері керує багатьма пристроями на площині даних. Для цього використовується стандартизований інтерфейс прикладного програмування API (Application Programming Interface). Це, наприклад, такий інтерфейс, як OpenFlow. Відповідно, для того, щоб побудувати мережу SDN, на мережевих елементах, перш за все, комутаторах і маршрутизаторах, повинна бути реалізована підтримка OpenFlow. При цьому на кожному з них є таблиця, або таблиці, правил маршрутизації. Кожне правило визначає, як маршрутизувати пакети певної сесії або потоку трафіку

Незважаючи на те, що концепція SDN стала поширеною в останні роки, сама ідея досить ненова, і еволюціонує вже більше 20 років. Її сліди можна простежити навіть у розвитку ранніх телефонних мереж на базі комутації каналів, коли управління мережею (сигналізація) було відокремлено від мережі каналної комутації мовного трафіку. І це було зроблено рівно з тією ж метою, що і в SDN - щоб спростити управління і введення нових послуг. Концепція «Програмних комутаторів» Softswitch для телекомунікаційних мереж на базі комутації пакетів також дуже близька до SDN за функціями і реалізації.

В основі концепції SDN розвивається відкритий стандарт OpenFlow - стандарт, визначений фондом ONF (Відкритий фонд мережевих технологій). Це відкритий стандартний протокол зв'язку, який лежить в основі програмно керованих мереж (SDN). Інтерфейс OpenFlow надає доступ і зв'язок між рівнями управління та інфраструктури архітектури SDN, як фізичної, так і віртуальної. Завдяки централізації управління пристроями рівня

інфраструктури OpenFlow спрощує управління мережею і розширює можливості програмування - саме те, що обіцяє концепція SDN.

Концепція SDN передбачає[6]:

- відокремити в маршрутизаторі управління мережевим обладнанням від управління передачею даних. Управління винести на окремий комп'ютер, який буде знаходитися під контролем адміністратора мережі;

- перейти від управління окремими екземпляром мережевого обладнання до управління мережею в цілому;

- створити інтелектуальний програмно-керований інтерфейс між мережним додатком і транспортної середовищем (Рисунок 1.3)

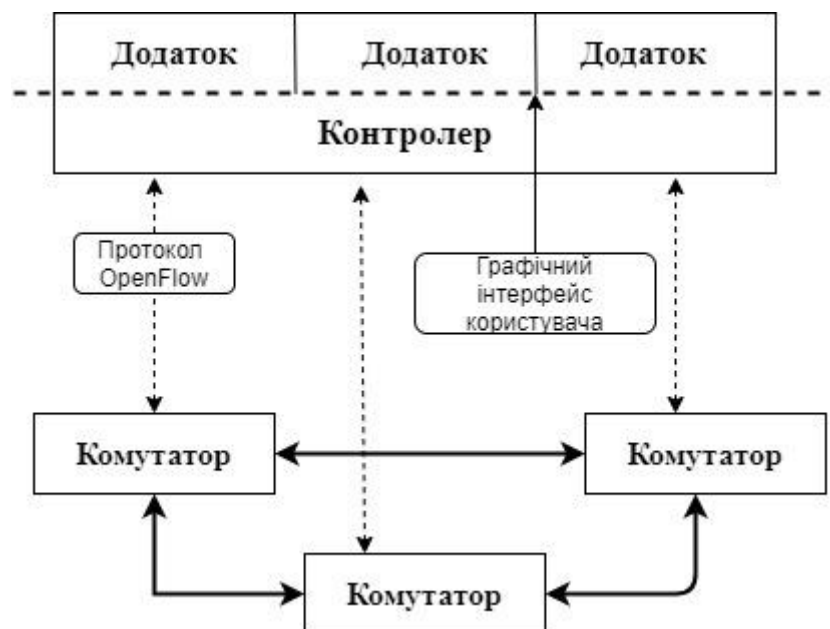


Рис. 1.3 Принцип реалізації концепції SDN

Таким чином, реалізація концепції SDN - поділ управління мережею (площини управління) і механізму просування даних (площини даних), перенесення функцій управління в окремі обчислювальні пристрої, які називаються SDN-контролерами, приводить до зміни традиційного розподіленої моделі маршрутизації централізованої моделлю, перетворюючи процес управління мережею, що включає створення маршрутів, в процес програмування мережі в цілому.

В теорії концепція програмно-конфігуруючих мереж має багато переваг:

- підвищується продуктивність (за рахунок прискорення переміщення трафіку);
- знижуються витрати на побудову і супровід мережі (за рахунок віртуалізації управління мережею);
- підвищується зручність управління, безпека і спрощується виконання ряду інших завдань (на централізованому контролері системний адміністратор може спостерігати всю мережу як єдине ціле);
- необмежені можливості до розширення і масштабованості в залежності від поставлених завдань і інше.

Висновки до 1-го розділу :

В цьому розділі було розглянуто концепцію побудови мережі NGN та частину недоліків чому дана концепція більше невикористовується.

Також концепція програмно-визначених мереж (SDN) ґрунтовно змінює принципи функціонування мереж і їх управління. У швидко мінливому сучасному світі саме мережі передачі даних, обмежують зростання продуктивності додатків по мірі зростання кількості мобільних користувачів, масштабування віртуальних середовищ.

2. Структура і основні елементи мереж SDN

2.1 Структура мережі SDN.

Одним з найбільш перспективних напрямків розвитку телекомунікаційних мереж є їх побудова на базі концепції мережі SDN (Software-defined networking). Суть концепції SDN полягає в поділі функцій управління і передачі трафіку. При цьому всі функції управління покладаються на централізований елемент мережі. Даний підхід допомагає ефективно обробляти великі потоки даних, а також спрощує контроль і налаштування великої кількості мережевого обладнання.

Основна відмінність такого підходу від принципу побудови існуючих мереж є те, що рівень управління мережею відділений від рівня передачі даних. При цьому, ряд багато рішень багатьох завдань управління покладається на програмні модулі.

Архітектуру мережі SDN логічно можна розділити на три рівні [7]: прикладний рівень, рівень управління та мережевий рівень (рис. 2.1)

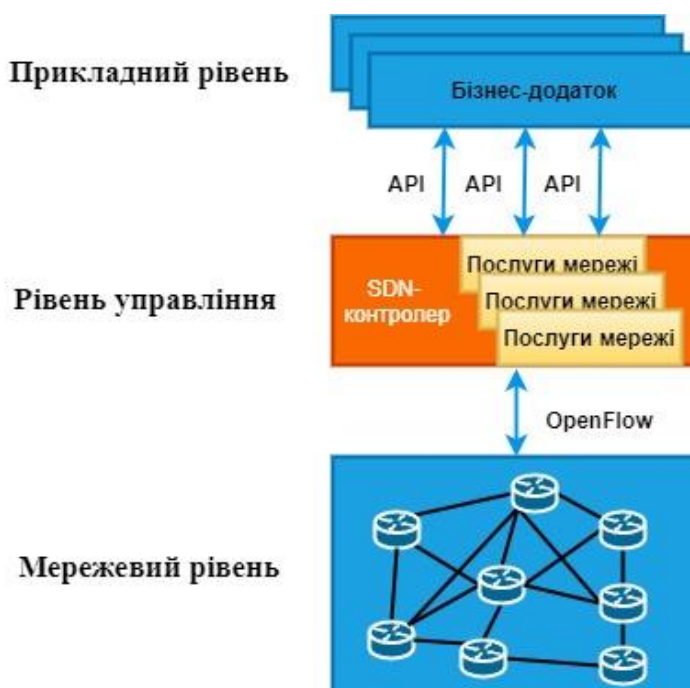


Рис. 2.1 Архітектура програмно-конфігуруючих мереж

Прикладний рівень охоплює рішення, орієнтовані на розширення мережевих служб. Ці рішення в основному є програмними додатками, які взаємодіють з контролером. Відкриті інтерфейси API посилаються на програмні інтерфейси між програмними модулями контролера та додатками SDN. Ці інтерфейси є відкритими для клієнтів, партнерів та спільноти з відкритим вихідним кодом для розвитку та модифікації. Прикладний рівень охоплює велику кількість додатків для задоволення різних потреб клієнтів, таких як автоматизація мережі, гнучкість та програмованість тощо. Деякі домени програм SDN включають інженерію трафіку, віртуалізацію мережі, моніторинг та аналіз мережі, дослідження мережевих служб, контроль доступу тощо. Логіка управління для кожного екземпляра програми може бути запущена як окремий процес безпосередньо на апаратному забезпеченні контролера в кожному домені. Загальна архітектура мережі SDN .

Рівень управління включає логічно централізований контролер SDN. Він приймає запити від прикладного рівня через чітко визначені API та здійснює управління та моніторинг мережевих пристроїв за допомогою стандартних протоколів. Він є ядром архітектури SDN і реалізується контролерами кожного домену, які збирають інформацію про фізичний стан ділянки мережі, яка належить кожному контрольному домену.

Рівень мережі складається з фізичного мережевого обладнання, включаючи комутатори та маршрутизатори. Даний рівень забезпечує програмоване і швидкісне обладнання та програмне забезпечення, яке відповідає галузевим стандартам. На нижньому рівні фізична мережа складається з апаратних пристроїв передачі даних, які зберігають таблиці для швидкої передачі пакетів (FIB), а також пов'язані метадані, включаючи пакети, потоки і лічильники портів.

Основна відмінність мереж SDN від традиційних мереж в тому, що архітектура традиційних мереж є децентралізованою і громіздкою в той час як сучасні мережі вимагають більшої гнучкості і швидкого впровадження нових

сучасних технологій. У мережах SDN перейшли до централізованого аналізу стану мережі, розділили процес пересилання пакетів (площину даних) і процес формування маршрутів за різними показниками якості обслуговування (площину управління). Великим досягненням мереж SDN є те, що в разі необхідності переходу на нову сучасну технологію, всі зміни, в основному, будуть відбуватися на рівні управління, основним елементом якого є контролер. При цьому ми можемо просто взяти контролер і переписати його програмне забезпечення.

Відмінності традиційних мереж від мереж SDN показані на рис.1.2 та рис.1.3:

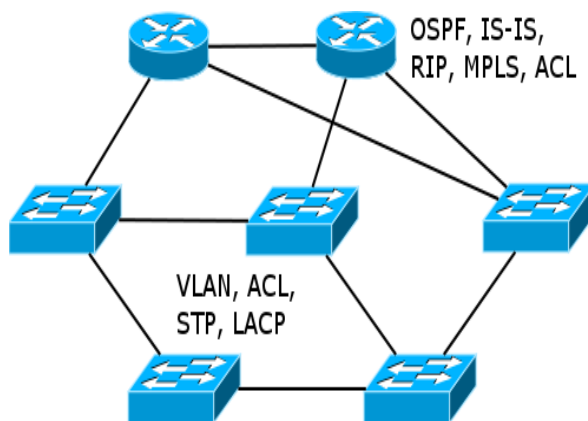


Рис.2.2 Традиційна мережа

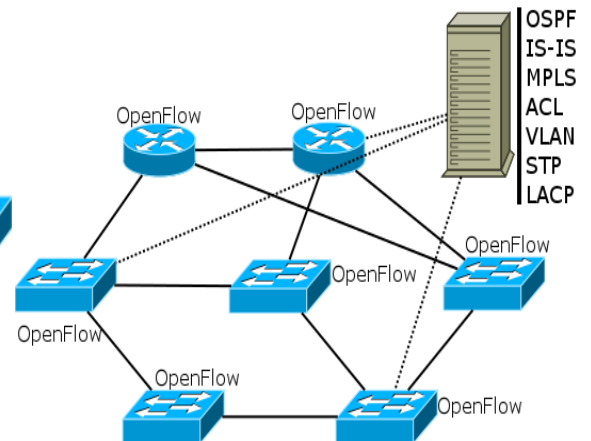


Рис.2.3 Мережа по використанню SDN

SDN

На рис2.2 показано, що в традиційних мережах на кожному мережевому пристрої працюють потрібні протоколи, які забезпечують функціонування сервісів обслуговування пакетів. У SDN мережах (рис2.3) завдання комутаторів і маршрутизаторів полягає в передачі трафіку, а всі керуючі функції взяв на себе контролер.

При побудові мережі SDN використовується обмежений набір елементів. Основними є комутатор і контролер.

2.2. Структура контролера SDN

Контролер SDN за своєю суттю є системою централізованого управління мережею. Він вирішує всі завдання, пов'язані з прийняттям рішень по управлінню трафіком даних. Це, в першу чергу, формування плану розподілу у вигляді маршрутних матриць. Визначення пріоритетності в обслуговуванні різних видів навантаження. Відстеження обсягу трафіку на різних елементах мережі і вжиття заходів для усунення можливості перевантаження. Забезпечення доставки повідомлення по заданих адресах з необхідною якістю обслуговування.

Контролер складається з пам'яті, процесора, таблиці маршрутизації, ASIC та портів на рис 2.4



Рис 2.4 Структурна схема контролера

Контролер SDN містить безліч портів (1, 2, 3, 4) зв'язку, які можуть бути, наприклад, портами Ethernet, оптичними портами, портами волоконного каналу та інші, крім того процесор і таблицю маршрутизації. Таблиця маршрутизації зберігається в енергозалежних пристроях зберігання даних або пам'яті. Пристрій SDN також включає в себе ASIC, призначену для обробки операції пересилання потоку на основі її вмісту таблиці маршрутизації [3].

Контролер SDN може дати команду щоб додати або виділити записи з таблиці пересилання мережевого пристрою SDN. Таблиця пересилання мережевого пристрою може бути заповнена на активній основі або пасивної основі або їх комбінації. При використанні мережевий пристрій або контролер буде попередньо заповнювати таблицю пересилання і заздалегідь вводити часто використовувані потоки, а в методі відповіді контролер буде додавати записи про потоках в міру необхідності.

На рис.2.5. представлений мережевий пристрій SDN і контролер SDN, які взаємодіють один з одним по спеціальному каналу управління(він може бути захищеним або відкритим). Роботою каналу управляє процесор мережевого пристрою SDN за допомогою функціонального блоку , який має назву менеджер каналу SDN .

Основними елементами мережевого пристрою є процесор і таблиця маршрутизації. МП-SDN приймає потоки трафіку від клієнтських пристроїв, або інших мережевих пристроїв і передає їх у відповідності до змісту своєї таблиці маршрутизації. Потік трафіку є потоком з одного або декількох пакетів, що переносять дані з джерела в пункт призначення. МП-SDN здатне ідентифікувати пакети конкретному потоку даних на основі адреси джерела, адреси призначення та іншої інформації в заголовку пакетів. Кожен запис в таблиці маршрутизації відповідає певному потоку трафіку.

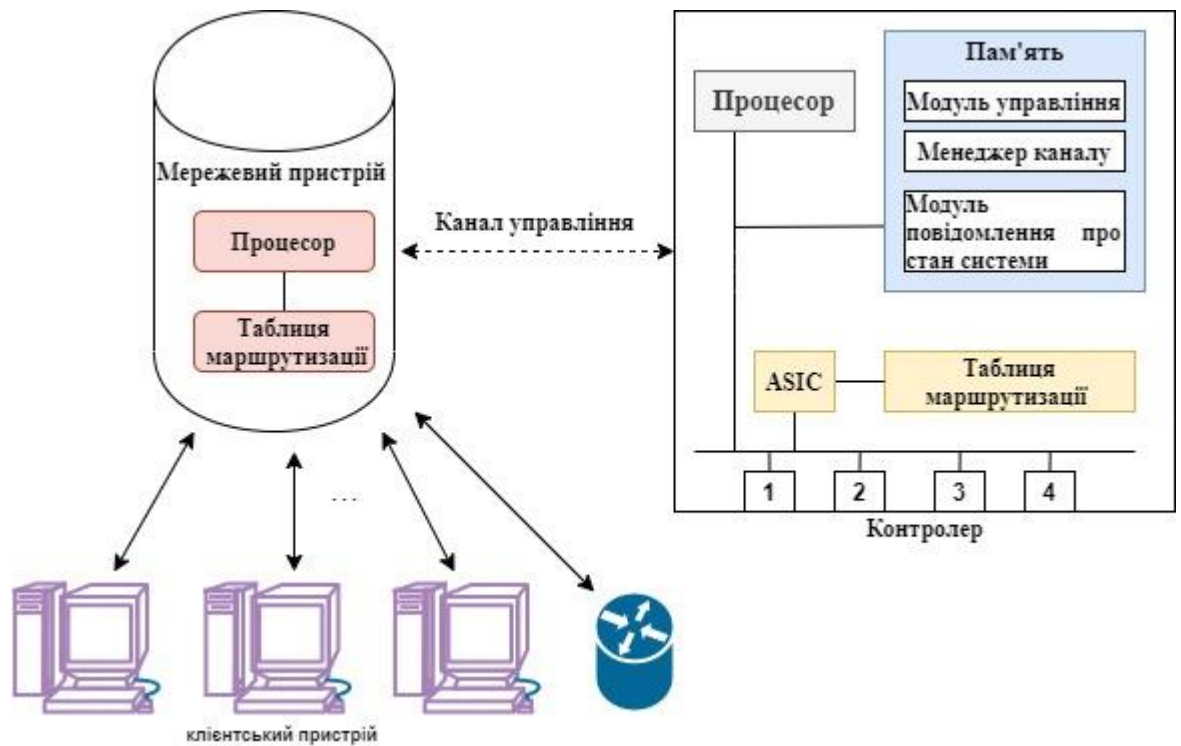


Рис. 2.5 Взаємодія мережевого пристрою SDN (МП-SDN) і контролера SDN (К-SDN).

Слід зазначити, що канал SDN використовується для передачі тільки керуючої інформації. Це може бути, наприклад [3]:

- заявка від МП-SDN до контролера на пошук маршруту для пакета, який немає необхідних даних в таблиці маршрутизації.
- відповідь контролера SDN про внесення необхідних змін до таблиці маршрутизації при появі нових потоків і зміни структури мережі. Зазвичай передача інформації по каналу SDN ведеться з використанням протоколу OpenFlow.

Канал протоколу SDN може працювати через пряме з'єднання між мережевим пристроєм SDN і контролером SDN або непряме з'єднання через безліч вузлів в мережі. Фізичні з'єднання, які підтримують канал SDN, можуть бути з'єднаннями LAN, WAN-з'єднаннями, проводовими, оптичними або безпроводовими з'єднаннями і т.д.

Мережевий пристрій передає інформацію про стан системи мережевого пристрою в контролер, посилаючи повідомлення стану системи по каналу протоколу певного програмного забезпечення (SDN). Повідомлення стану

системи може включати в себе заголовок, який вказує, що повідомлення відноситься до стану системи, і корисне навантаження.

Корисне навантаження може включати в себе поле типу стану системи, що вказує тип інформації стану системи, і поле значень, що надає інформацію стану системи.

Оскільки повідомлення стану системи включає в себе поле типу стану системи, а також поле значення стану системи, цей підхід є гнучким і не обмежується тільки одним типом стану системи.

2.3 Структура коммутатора SDN

Коммутатор OpenFlow - это коммутатор данных с поддержкой OpenFlow, который связывается по каналу OpenFlow с внешним контроллером. Он выполняет поиск и пересылку пакетов в соответствии с одной или несколькими таблицами потоков и таблицей групп. Коммутатор OpenFlow связывается с контроллером, и контроллер управляет коммутатором через протокол коммутатора OpenFlow. Они либо основаны на протоколе OpenFlow, либо совместимы с ним.

Комутатор OpenFlow може функціонувати тільки при спільній роботі трьох основних елементів: таблиць потоків, встановлених на комутаторах, контролера і протоколу OpenFlow для безпечної взаємодії контролера з комутаторами. Таблиці потоків встановлюються на перемикачах. Контролери спілкуються з комутаторами по протоколу OpenFlow і накладають на потоки. Контролер може встановлювати шляху через мережу, оптимізовані для конкретних характеристик, таких як швидкість, мінімальна кількість стрибків або зменшена затримка.

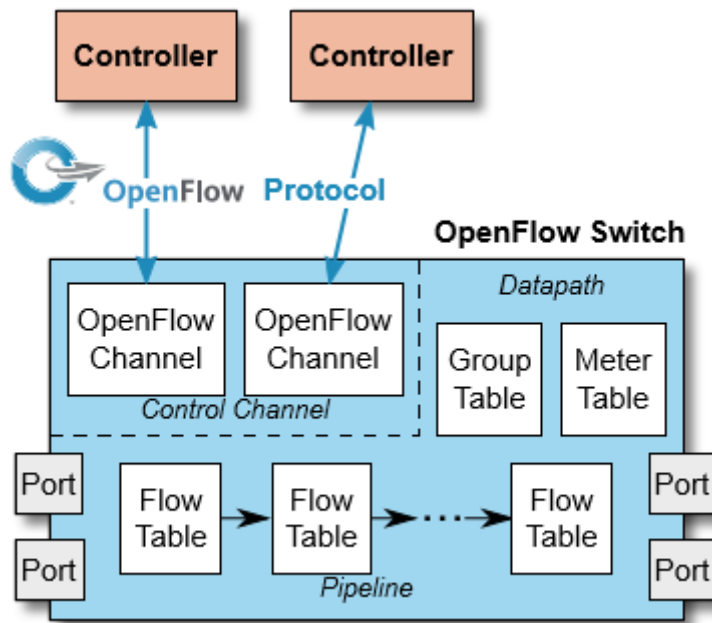


Рисунок 2.6. Основні компоненти комутатора OpenFlow.

Комутатор OpenFlow складається з таблиці потоків, яка виконує пошук і пересилку пакетів, і захищеного каналу до зовнішнього контролера (рисунок 2.6). Контролер керує перемиканням по безпечному каналу з використанням протоколу OpenFlow. Використовуючи протокол комутації OpenFlow, контролер може додавати, оновлювати і видаляти записи потоків в таблицях потоків.

Таблиця потоків містить набір записів потоку (значення заголовків, з якими зіставляються пакети), лічильники активності і набір з нуля або більше дій, що застосовуються до зіставляється пакетам. Всі пакети, оброблені комутатором, порівнюються з таблицею потоків. Якщо відповідний запис знайдено, будь-які дії для цього запису виконуються над пакетом (наприклад, дія може полягати в пересиланні пакета із зазначеного порту). Якщо збігів, не знайдено, пакет пересилається на контролер по безпечному каналу. Контролер відповідає за визначення того, як обробляти пакети без допустимих записів потоку, і він керує таблицею потоків комутатора, додаючи і видаляючи записи потоку.

Записи потоку можуть пересилати в порт. Зазвичай це фізичний порт, але він також може бути логічним портом, певним комутатором, або зарезервованим портом, визначеними цією специфікацією.

Дії, пов'язані з записами потоку, можуть також направляти пакети в групу, що вимагає додаткової обробки. Групи є наборами дій для потокового кодування, а також більш складну семантику пересилання (наприклад, багатопроточне поширення, швидке перенаправлення і агрегацію каналів). Як загальний рівень непрямого, групи також дозволяють декільком записам потоку пересилати на один ідентифікатор (наприклад, пересилання IP на загальний наступний перехід). Ця абстракція дозволяє ефективно змінювати загальні вихідні дії в записах потоку.

2.4 Взаємодії елементів в процесі обслуговування пакетів.

Контролер OpenFlow служить блоком управління для обчислення маршрутів та розподілу даних до OpenFlow комутаторів, які відповідають за переадресацію пакетів на основі отриманих записів потоків [7].

OpenFlow комутатор - це пристрій, який отримує команди або інформацію про таблиці потоків від контролера та відправляє на контролер інформацію про їх стан. На основі інформації про потоки, що генерується та надсилається з контролера, OpenFlow комутатор функціонує як пристрій для фізичного пересилання пакетів даних.

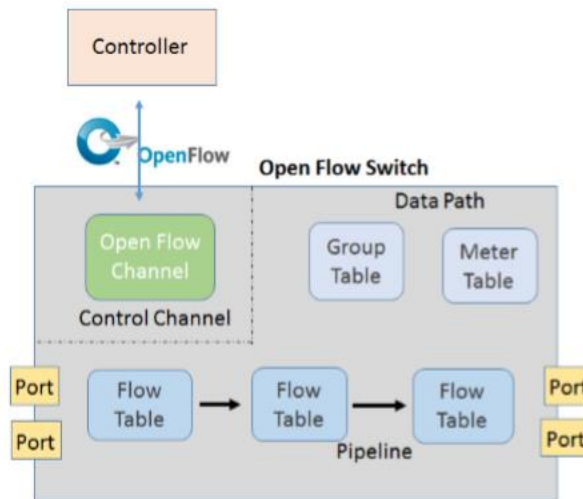


Рисунок 2.7 Складові OpenFlow комутатора.

Таблиця потоків в комутаторі OpenFlow складається із записів, які містять інформацію про поля відповідності, лічильники та набори інструкцій, які слід застосувати для відповідних пакетів.

У комутаторі OpenFlow можуть бути кілька таблиць потоків, які будуть опрацьовані за допомогою конвеєрної обробки. Наприклад, коли пакет приходить на комутатор і відповідає запису потоку в першій таблиці потоків, відповідна інструкція буде пов'язана з інструкцією конвеєрної обробки, яка дозволяє передавати пакети в наступні таблиці для подальшої обробки. Конвеєрна обробка таблиці припиняється, якщо інструкції, пов'язані з відповідним записом потоку, невизначають наступну таблицю потоків.

Таблиця груп містить список записів групи, і кожен запис групи асоціюється з пакетами дій або групами дій. Пакети, які визначені у групі та збігаються із записами групи, будуть опрацьовані відповідно до одного або кількох пакетів дій.

OpenFlow комутатор повинен підтримувати 3 типи стандартних портів OpenFlow [7]:

- фізичні порти, які слугують для з'єднання комутаторів між собою та для підключення до зовнішньої мережі;
- логічні порти, які є портами вищого рівня і не асоціюються напряму з фізичними портами, та можуть бути реалізовані за допомогою

таких технологій, як агрегація портів, тунельні інтерфейси чи loopback-інтерфейси. Комутатори OpenFlow підключаються один до одного логічно через порти OpenFlow. Пакети OpenFlow поступають на вхідний порт, далі пакети проходять через конвеєрну обробку, і можуть бути передані на вихідний порт.

➤ резервні порти є визначеними специфікацією OpenFlow портами, які використовуються для певних дій по переадресації після того, як було знайдено відповідність в записах потоків.

Таблиця потоків для комутатора OpenFlow послідовно пронумерована, починаючи з 0. Коли пакет передається в першу таблицю потоків, то обирається відповідність з найвищим пріоритетом, а набір інструкцій, який їй відповідає, спрямовуватиме пакет до наступної таблиці потоків, і процес обробки пакету повториться знову (Рисунок 2.8). Якщо інструкція не спрямовує пакет до наступної таблиці, конвеєрна обробка зупиняється і над пакетом буде виконана відповідна дія, якщо вона встановлена.

Якщо пакет невідповідає жодному запису потоку в таблиці потоків, то це називається пропуском таблиці. Якщо для запису пропуску таблиці зроблена особлива конфігурація, запис потоку пропуску таблиці можна обробляти різними наборами дій, такими як відкидання, прозоре пересилання їх до контролера тощо.

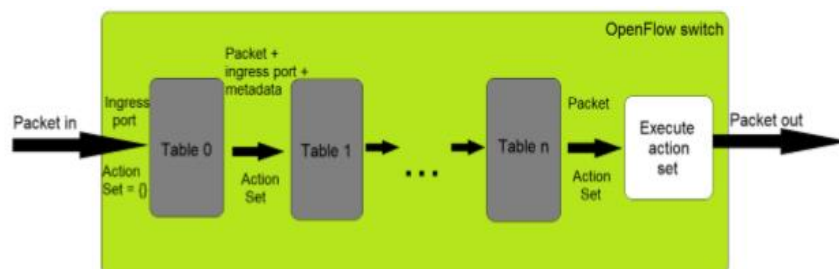


Рисунок 2.8 Потік пакетів через конвеєрну обробку OpenFlow.

Якщо є метадані, визначені для обробки наступною таблицею потоків, то пакет буде перенаправлений до наступної таблиці потоків, повторно виконуючи перевірку полів відповідності та перевірку інструкцій. Якщо

пакети вже були надіслані до останньої таблиці потоків з найбільшим порядковим номером, це означає, що більше немає таблиці потоків, яку потрібно перевірити, і в такому випадку буде виконано набір дій, визначений в останній таблиці потоків.

У випадку коли в таблиці потоків немає відповідних записів потоку, пакет буде відкинуто, якщо на комутаторі не буде налаштовано запис пропуску таблиці. Якщо використовується функція пропуску таблиці, то пакет буде оброблятися згідно інструкцій, визначених для випадку пропуску таблиці

Таблиця потоків є основним елементом протоколу OpenFlow. Кожен запис таблиці потоків містить наступні елементи (Рисунок 2.9):

- match fields, що складається з вхідних портів та необов'язкових відповідних полів або метаданих, зазначених у попередній таблиці;
- priority, яка визначає пріоритет запису таблиці потоків;
- counters, який показує кількість пакетів, що відповідають даному запису і оновлюється в режимі реального часу;
- instructions, в яких зазначено зміни наборів дій чи конвеєрної обробки;
- timeouts, який показує максимальний час життя потоку;
- cookie, який показує вибране контролером значення, яке, використовується ним для фільтрації потоку, модифікації потоку або видалення потоку.

Даний елемент невикористовується під час обробки пакетів.

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags
--------------	----------	----------	--------------	----------	--------	-------

Рисунок 2.10. Основні компоненти запису потоку в таблиці потоків.

Поля match fields та priority використовуються для ідентифікації відповідних записів таблиці потоків.

Висновки до 2-го розділу:

1. Архітектура програмно-визначених мереж складається з трьох основних рівнів: інфраструктури, управління і додатків. Рівень додатків керує рівнем інфраструктури через рівень управління – додатки SDN повідомляють про свої мережеві вимоги рівень управління, котрий їх переводить і потім здійснює низькорівневий контроль на елементами мережі (рівень інфраструктури).
2. Традиційні мережі незабаром стануть важко керованими і конфігуруються, так як досягнутий занадто великих розмірів. Всі ці проблеми пропонується вирішувати за допомогою використання технології Software-Defined Networks.
3. Контролер SDN управляє управлінням потоком на комутаторах / маршрутизаторах «нижче» (через південні API) і додатках і бізнес-логікою «вище» (через північні API) для розгортання інтелектуальних мереж. Вони об'єднують і здійснюють посередництво між різними доменами контролерів, використовуючи загальні інтерфейси додатків.
4. Завданням комутатора є забезпечення передачі трафіку даних на основі інформації, що управляє, яка надходить з рівня управління. Всю необхідну інформацію для обслуговування надходять пакетів він отримує від контролера.
5. Таблиця потоків в комутаторі OpenFlow складається із записів, які містять інформацію про поля відповідності, лічильники та набори інструкцій. OpenFlow комутатор повинен підтримувати 3 типи стандартних портів OpenFlow: фізичні , логічні та резервні порти .

3 Протоколи мережі SDN

На сьогоднішній день немає однозначного рішення про те, який протокол і де доцільно використовувати OpenFlow або OF-CONFIG? Розглянемо які завдання і яким чином ці протоколи вирішують.

3.1 Протокол OpenFlow

Протокол OpenFlow є відкритим стандартом. Він був запропонований ONF для стандартизації взаємодії контролера з мережевими пристроями в архітектурі ПКМ. Дотримуючись еволюційному шляху розвитку мереж зв'язку, протокол OpenFlow, на даному етапі розвитку і практичної реалізації концепції ПКМ, впроваджується у вигляді програмного модуля в апаратні реалізації Ethernet комутаторів, маршрутизаторів і бездротових вузлів доступів, як розширення можливостей і їх застосування в якості пристроїв ПКМ.

Оскільки мережевий пристрій передає інформацію про стан системи контролеру SDN по каналу протоколу SDN, воно може бути доставлено відносно швидко. Використання SDN-протоколу, такого як OpenFlow Protocol для передачі інформації про стан системи, дозволяє контролеру швидко отримувати повідомлення про зміни [3].

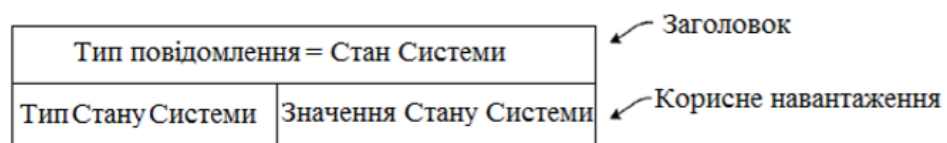


Рис. 3.1 Загальна структура повідомлення про стану системи (1)

На рис.3.1, показана загальна структура повідомлення про стану системи. Повідомлення включає в себе заголовок і корисне навантаження. Заголовок включає в себе поле типу повідомлення, яке вказує, що повідомлення відноситься до стану системи.

Тип повідомлення = Стан Системи		Довжина Повідомлення	
Тип Стану Системи	Довжина Оповіщення	Значення Стану Системи	

Рис.3.2 загальної структури для повідомлення про стану системи(2)

На рис. 3.2 показаний ще один приклад загальної структури для повідомлення про стану системи. Він аналогічний показаному на рис.3,1, але також включає в себе поле довжини повідомлення і поле довжини повідомлення про стан системи. Поле довжини допомагає контролеру дізнатися, коли закінчилося повідомлення стану системи або окреме повідомлення про стан системи в повідомленні стану системи.

Версія	Тип= Оповіщення про стан системи	Довжина
Ідентифікатор (ID) транзакції		
Корисне навантаження		

Рис. 3.3. Заголовок повідомлення про стан системи протоколу OpenFlow.

Рис.3.3 ілюструє конкретний приклад заголовку повідомлення про стан системи, використовуючи протокол OpenFlow більш докладно. Додатково до полів, вже описаних на рис.3.2, заголовок на рис.3.3 включає в себе поле версії протоколу OpenFlow (OFP), яке вказує, що повідомлення відповідає OFP, а також може вказувати версію OFP, використовувану, і поле ID транзакції, яке вказує номер транзакції для повідомлення. Повідомлення включає в себе корисне навантаження, що включає щонайменше одне

повідомлення про стан системи, яке може, наприклад, мати структуру, показану на рис.3.1, або на рис. 3.2.

Даний протокол розроблений для вирішення завдань більш високого, порівняно з протоколом OpenFlow, рівня. В основному це завдання з побудови мережевої середовища в цілому, конфігурації комутаторів і прийняття рішень, наприклад, про закриття або відкриття окремих портів.

3.2 Протокол OF-CONFIG

Протоколом OFCONFIG (OpenFlow Management and Configuration Protocol) передбачені наступні абстракції (Рис 3.4):

- логічний комутатор OpenFlow - абстракція вузла передачі даних OpenFlow. Протокол OF-CONFIG дозволяє здійснити конфігурацію логічного комутатора OpenFlow так, щоб контролер OpenFlow міг взаємодіяти з ним і керувати ним за протоколом OpenFlow;

- OpenFlow сумісний комутатор - фізичний або логічний мережевий елемент, ресурси якого (порти, черги тощо.) Виділені одним або декількома логічними комутаторами OpenFlow. Протокол OF-CONFIG дозволяє динамічно призначати ресурси OpenFlow-сумісного комутатора розміщеним на ньому логічним комутаторів OpenFlow;

- точка конфігурації OpenFlow - джерело повідомлень OF-CONFIG для OpenFlow-сумісних комутаторів. Сутність точки конфігурації OpenFlow і взаємодія точок конфігурації з контролерами OpenFlow зараз специфікаціями ONF нерегламентуються [7].

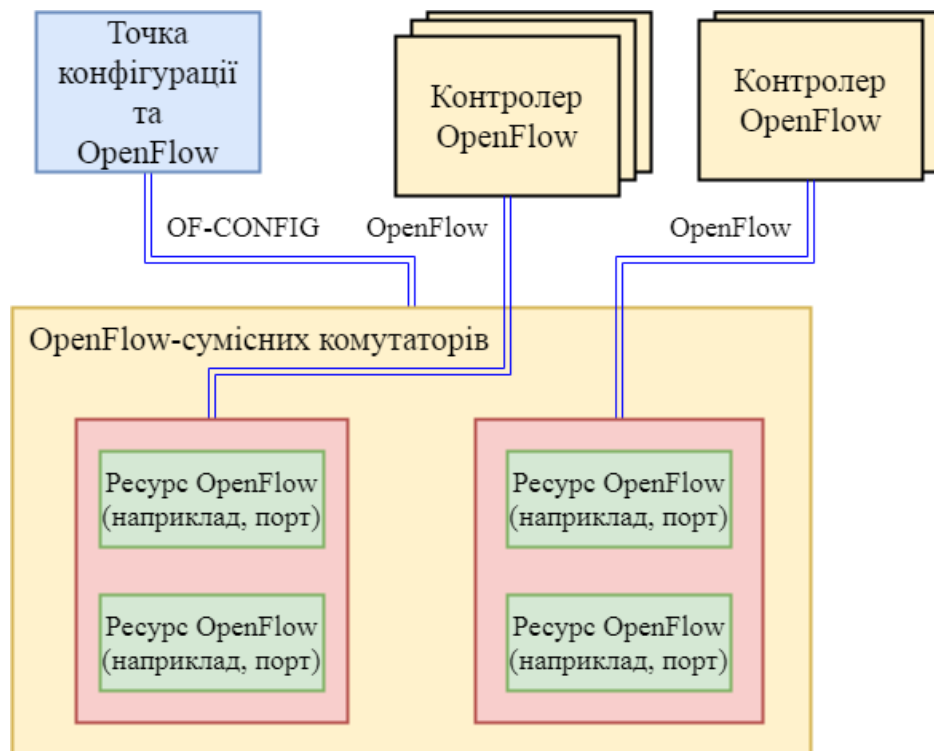


Рисунок 3.4 - Основні абстракції протоколу OF-CONFIG

Протокол OF-CONFIG використовується для вирішення наступних завдань:

- призначення одного або більше контролерів OpenFlow комутатора;
- конфігурація портів і черг;
- віддалене зміна властивостей портів;
- конфігурація сертифікатів для безпечної взаємодії логічних комутаторів OpenFlow і контролерів OpenFlow;
- запит можливостей логічних комутаторів OpenFlow;
- конфігурація обмеженого набору тунелів;
- ініціалізація логічних комутаторів OpenFlow;
- призначення ресурсів OpenFlow-сумісного комутатора одному і більше логічного комутатора OpenFlow;
- підтримка узгоджених моделей ділянки передачі.

Передбачається, що такі версії OF-CONFIG передбачатимуть також такі можливості, як виявлення комутаторів і топології, конфігурація

характеристик, обробка тригерів, пов'язаних з подіями, ініціалізація мережі OpenFlow, підтримка більшої кількості конфігурованих тунелів.

3.3 Порівняльна характеристика протоколів.

Протокол OF-CONFIG представляє OpenFlow комутатор як абстракцію - логічний комутатор (Logical Switch), при цьому одна фізична пристрій може містити кілька Logical Switch, які відповідають за пересилку потоку даних різних елементів мережі - Logical Switch Capabilities. У загальному випадку взаємодія між рівнем управління і рівнем передачі даних здійснюється на основі двох протоколів:

1. OF-CONFIG, який дозволяє конфігурувати окремі Logical Switch для створення якісного каналу передачі керуючої інформації;
2. Протокол OpenFlow, який дозволяє керувати переадресацією і модифікацією пакетів.

На рис.3.5 представлена схема взаємодії різних частин системи конфігурації і управління перемикачем.

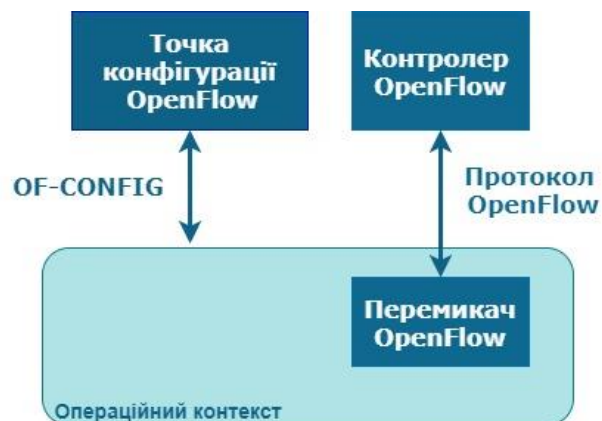


Рис.3.5 Точка конфігурації OpenFlow взаємодіє з операційним контекстом, який може підтримувати роботу перемикача OpenFlow, використовуючи конфігурацію і протокол управління (OF-CONFIG)

Протокол OpenFlow передбачає, що комутатор OpenFlow (наприклад, комутатор Ethernet, підтримує протокол OpenFlow) налаштований з різними параметрами, такими як IP-адреси контролерів OpenFlow.

Протокол конфігурації OpenFlow має можливість віддаленого налаштування комутаторів OpenFlow. При цьому, протокол OpenFlow зазвичай працює на інтервал часу потоку, тобто поки потоки додаються або видаляються. При цьому, OF - CONFIG працює на більш повільному інтервалі часу. Прикладом є побудова таблиць пересилання і прийняття рішень про дії пересилання, які виконуються по протоколу Openflow. Або під час включення / вимикання порту, коли не потрібно вирішувати задачу в інтервалі часу проходження потоку і, отже, може виконуватися по протоколу - OF - CONFIG.

Висновки до 3-го розділу:

В цьому розділі ми більш детально розібрали протоколи мережі SDN . Основним протоколом мережі SDN є OpenFlow. Він дозволяє вирішувати завдання управління переадресацією і модифікацією пакетів. Останнім часом з'явився протокол OF-CONFIG, який дозволяє конфігурувати окремі Logical Switch для створення якісного каналу передачі керуючої інформації.

4. Сучасні платформи побудови мереж SDN

4.1 Побудова мережі SDN з використання платформи SEBA

SEBA - це платформа широкосмугового доступу на мережі SDN.

SEBA підтримує як житловий доступ, так і бездротовий зворотний зв'язок і оптимізований таким чином, що трафік може проходити «швидкою доріжкою» прямо до магістралі, не вимагаючи обробки VNF на сервері[9].

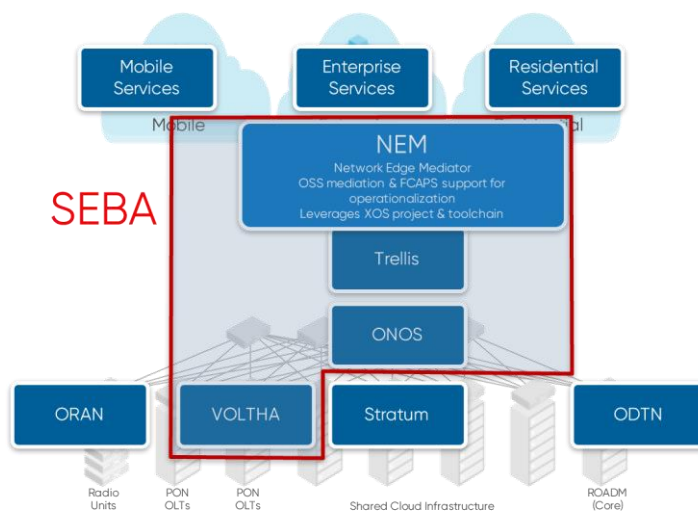


Рис. 4.1 програмне забезпечення SEBA

Як показано на рисунку 4.1, SEBA є об'єднанням проектів, таких як VOLTHA, ONOS і TRELIS.

SEBA повинна надавати інформацію про набір рішень для цих версій програмного забезпечення. SEBA - платформа на базі Kubernetes, де все програмне забезпечення управління розгортається як контейнери на обчислювальних вузлах, використовуючи оркестрацію Kubernetes, створюючи потім SEBA Pod (групу контейнерів).

Тепер розглянемо окремо три проекта VOLTHA, ONOS і TRELIS

ONOS - контролер SDN, який може управляти VOLTNA і Trellis .

ONOS був розроблений для задоволення потреб операторів, які бажають створити рішення для перевізників класу. ONOS підтримує як конфігурацію, так і управління в режимі реального часу, виключаючи необхідність запускати маршрутизацію та комутацію протоколів управління всередині тканини мережі.

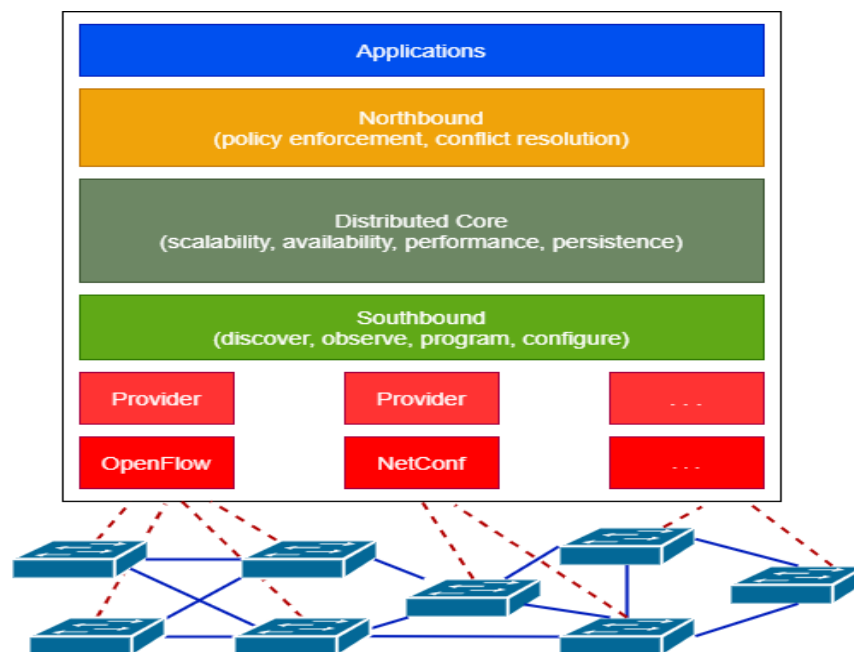


Рис. 4.2 Архітектура ONOS

1. Містить користувацькі додатки, наприклад, реактивну переадресацію, ProхуARP, сегментну маршрутизацію, SDN-IP і т. Д.
2. Передача інформації про мережу на рівень програми. Надання інтерфейсу управління для управління компонентом нижнього рівня.
3. Містить безліч основних функцій. Забезпечує розподілену кластеризацію. для підтримки НА і масштабованості

4. Надати абстрактний інтерфейс для управління мережевою інфраструктурою

5. Реалізація мережевого протоколу для управління мережевими елементами. Наприклад, OpenFlow, NetConf.

VOLTNA

VOLTNA - абстракція віртуального обладнання для широкосмугового доступу в межах житлової CORD

VOLTNA: абстрактний PON для підтримки Ethernet, який може контролюватися контролером .

На малюнку показано, як у VOLTNA, API відображаються на загальну модель основних даних за допомогою адаптерів на південь. На своїй південній стороні, VOLTNA спілкується з пристроями PON через спеціальні постачальники або адаптери OLT та ONU [10].

(спільна система контролю та управління, що ділиться всіма OLT та ONU)

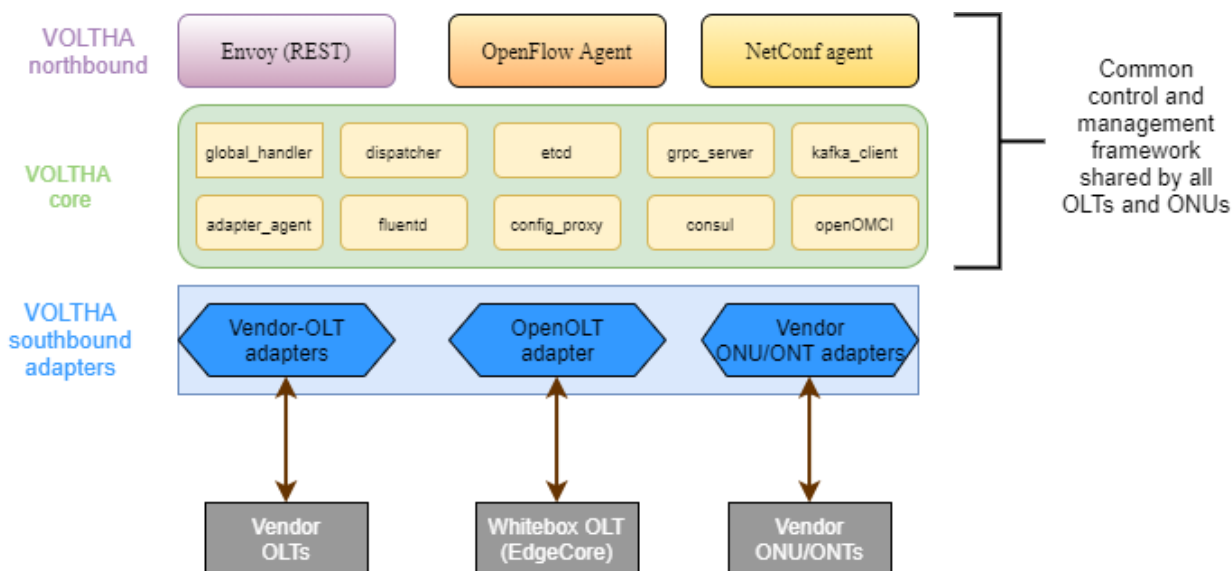


Рис 4.3 Архітектура VOLTNA

VOLTHA має адаптери, адаптери є досить складними, повноцінними модулями, що реалізують ядро управління пристроями PON.

У своєму північному інтерфейсі VOLTHA резюмує мережу PON, щоб вона відображалася як програмований перемикач Ethernet на контролер SDN.

VOLTHA northbound - У своєму північному інтерфейсі VOLTHA резюмує мережу PON, щоб вона відображалася як програмований перемикач Ethernet на контролер SDN.

VOLTHA southbound adapter - На своїй південній стороні VOLTHA спілкується з апаратними пристроями PON, використовуючи специфічні протоколи протоколів через адаптери OLT та ONU

Trellis

Trellis надає мережевим операторам ряд переваг SDN порівняно з традиційним мережевим керуванням.

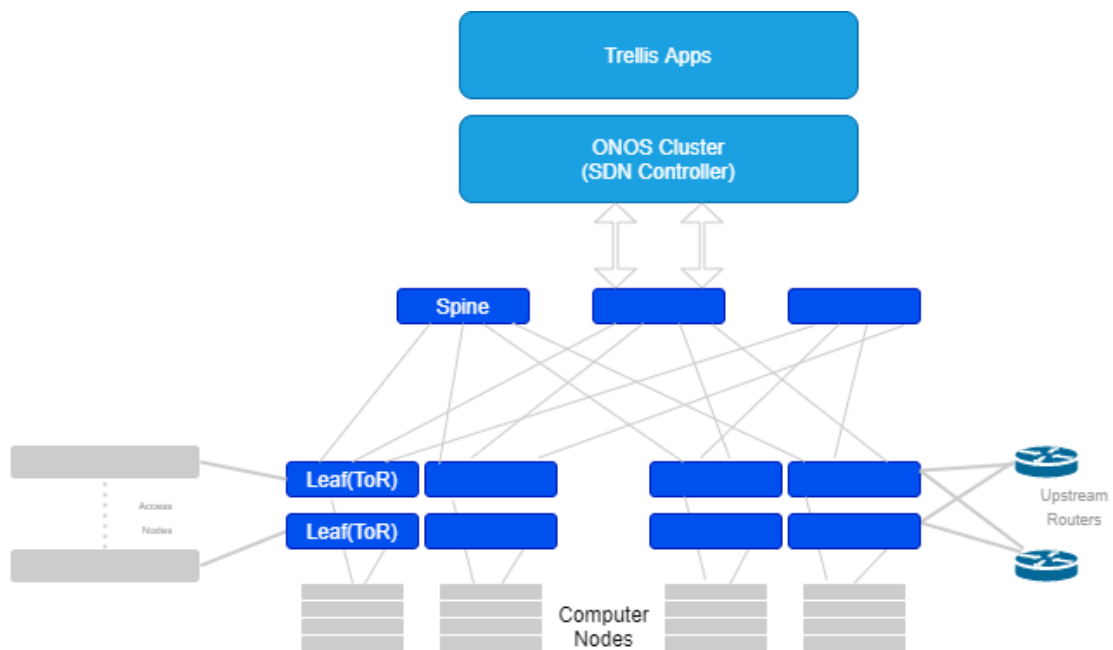


Рис 4.4 Архітектура Trellis

Trellis використовує контролер SDN (ONOS), від'єднаний від обладнання апаратної площини. У цьому дизайні набір додатків, що працюють на ONOS,

реалізує всі функціональні можливості та функції тканини, такі як комутація Ethernet, маршрутизація IP, багатоадресна передача, тощо.[12]

Архітектура Trellis внутрішньо використовує такі поняття маршрутизації сегмента (SR), як глобально значущі мітки MPLS (MPLS це метод маркування пакетів, який встановлює пріоритетність даних), які присвоюються кожному перемикачу листя та хребта. Це мінімізує стан міток у мережі порівняно з традиційними мережами MPLS, де мітки, які мають місцеве значення, потрібно міняти на кожному вузлі. У Trellis перемикачі аркушів висувують ярлик MPLS, що позначає пункт призначення ToR (leaf) на трафік IPv4 або IPv6.

Seba об'єднує всі ці три проекти і бере найкраще що було в цих проектах і в результаті модульну архітектуру можна уявити, як структуру яка представлена на наступному рисунку.

Як показано на рисунку 4.5, SEBA складається з декількох програмних модулів високого рівня, включаючи:

- Медіатор Edge Network (NEM)
- Управління SDN
- Керування програмами
- Драйвер доступу до вузла (AN)
- Драйвер агрегації та обслуговування (ASG)

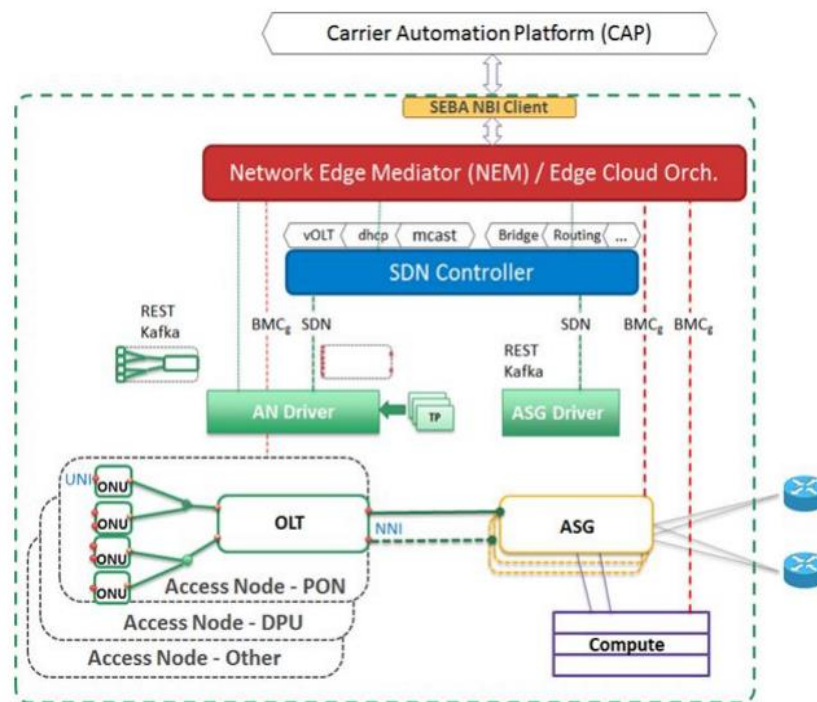


Рис.4.5 Діаграма цільової архітектури

вони зроблені у вигляді окремих контейнерів (віртуалізація) і забезпечують вирішення завдань які представлені на цьому малюнку.

Діаграма цільової архітектури вимагає дотримання деяких принципів та визначень:[11]

1. Інфраструктурний шар не позначається, але він включає фізичні компоненти - вузли доступу, перемикачі агрегації та обчислення.
2. Сервісний шар визначає прив'язку компонентів до рівня інфраструктури для надання послуги.
3. Контролер SDN підтримує самостійність структури управління для кожного компонента інфраструктурного шару, що бере участь у сервісі.
4. ASG - це лише функціональний блок, який підтримує агрегацію, перемикання та маршрутизацію площини даних, площину управління та трафік площини управління в межах POD і підтримує можливості Service Edge.

5. Контролер управління платою (BMC) - це функціональне визначення та галузевий термін для інтерфейсу до функцій управління обладнанням.

У SEBA Pod є обладнання OLT та ONU, кероване VOLTHA, яке керує мережею PON, і контролер SDOS ONOS, який включає в себе пару арі / додатків, які відкривають послуги, що надаються PON. Наприклад, OLT підключається до пари вимикачів AGG, частини тканини або листової частини хребта, якими також керуватиме контролер SDN. Потім обладнання комутаторів може підключатися до зовнішніх маршрутизаторів та / або BNG, до локальних комп'ютерів тощо.

Однією з особливостей, що робить SEBA відмінною від R-CORD, є те, що трафік плати даних для абонента просто проходить через апаратне забезпечення та виходить в Інтернет. Це створює абонентський трафік "швидким шляхом" до Інтернету. Шлях даних залишається апаратним і виходить лише для обчислення вузлів, коли це необхідно, наприклад, коли для певних абонентів надаються сторонні послуги. Для порівняння, у R-CORD трафік, що надходить від резидентного абонента, проходить через апаратне забезпечення, але потім він переходив до обчислення вузлів через віртуальний комутатор, як OVS, та відвідує контейнер VSG (Virtual Subscriber Gateway), а потім повертається назад в апаратне забезпечення та в Інтернет

Також можна сказати, що SEBA - це еталонний дизайн, заснований на варіанті R-CORD. Розглянемо, що ж себе приставляє R-CORD.

R-CORD

R-CORD - це рішення з відкритим кодом, засноване на платформі CORD для надання надширокосмугових житлових послуг. R-CORD перетворює край мережі оператора в спритну платформу надання послуг, що дозволяє оператору забезпечити кращий досвід для кінцевих користувачів разом з інноваційними послугами нового покоління. [18]

Хоча для фізичного підключення абонентів (через GPON, DOCSIS або подібне) необхідне спеціалізоване обладнання для доступу, проект VOLTHA відводить навіть це обладнання спеціального призначення, щоб зробити його керованим як керований ресурс OpenFlow.

R-CORD також включає в себе віртуальний шлюз абонентів (vSG) і використовує основний мережевий сервіс - віртуальний маршрутизатор (vRouter); перший реалізується контейнером, який прив'язаний до кожного абонента, а останній - керуючий додаток ONOS.

Розглянемо їх більш детально:

- Virtual Subscriber Gateway (vSG)
- Virtual Router (vRouter)

Шлюз віртуального абонента (vSG)

Обладнання для споживчих приміщень(ОСП) - часто називається "домашній маршрутизатор" або "житловий шлюз" в житловому середовищі - запускає сукупність основних функцій та додаткових послуг (наприклад, брандмауер, батьківський контроль) від імені резидентів абонентів. Більш складні функції підприємства також є поширеними (наприклад, WAN . Розширюючи можливості ОСП у хмарі, нові послуги з додатковою вартістю, а також можливості обслуговування клієнтів можна забезпечити легше.

Наша віртуалізована версія ОСП, що називається віртуальним шлюзом абонентів (vSG), виконує набір функцій, вибраних абонентами, але робить це на товарному обладнанні, розташованому в центральному офісі, а не в приміщенні клієнта. Вдома все ще є пристрій (який ми ще називаємо CPE), але його можна звести до голого металу перемикача, при цьому більшість функціональних можливостей, які працювали на вихідній CPE, переміщені в СО і працюють у віртуальному. обчислювальний екземпляр (наприклад, VM або контейнер) на товарних серверах.

Архітектура CORD дозволяє отримати широкий спектр варіантів реалізації для vSG. Є два основні проблеми дизайну: функціональність та ефективність.

На сьогоднішній день ми зосередилися на простому пакеті абонентів, реалізованому в Linux. CORD підтримує північний інтерфейс, який дозволяє абонентам та операторам вибирати та контролювати окремі функції (наприклад, встановлювати параметри батьківського контролю на конкретних пристроях).

Поточний дизайн реалізує пакет функцій абонента в Linux, який працює в контейнері, прив'язаному до цього абонента. Реалізація посилок підтримує базове підключення до Інтернету, а також набір додаткових функцій, включаючи:

- Призупинити / відновити: Оператори можуть призупинити та відновити підключення абонента. Призупинення реалізується шляхом налаштування контейнера не пересилати трафік з приміщення клієнта до Інтернету, але замовник все ще може дістатися до служб, що працюють у vSG (наприклад, DHCP, DNS).
- Обмежений доступ: Оператори можуть тимчасово обмежувати доступ передплатників, перенаправляючи весь трафік на вибраний веб-сайт (наприклад, веб-сайт для виставлення рахунків для абонентів із делінквентними обліковими записами, веб-сайт для навчання авторських прав для абонентів, які позначені як порушення авторських прав). Обмежений доступ реалізується за допомогою iptables для перенаправлення трафіку HTTP абонента на локальний веб-сервер, який надає проксі для віддаленого сайту.
- Батьківський контроль: Абоненти можуть застосовувати фільтри батьківського контролю до різних пристроїв вдома. Батьківський контроль реалізується шляхом перенаправлення запитів DNS з конкретного пристрою (збігається за MAC-адресою) на локальний dnsmasq, який пересилає їх до

зовнішньої служби батьківського контролю, наприклад FamilyShield OpenDNS або Akamai's AnswerX.

- Вимірювання пропускної здатності: Оператори можуть встановлювати пропускну здатність передньої та нижньої течії, доступну для абонентів.
- Діагностика доступу: Оператори можуть запускати прості інструменти діагностики на підключенні абонента. Діагностика реалізується за допомогою запуску вибраного інструмента (наприклад, ping, traceroute, tcpdump) всередині vSG та повернення виведеного тексту.
- Брандмауер: Оператори та абоненти можуть встановлювати правила брандмауера для всього трафіку в будинок абонента.

Віртуальний маршрутизатор (vRouter)

У застарілих середовищах пристрої широкосмугового мережевого шлюзу (BNG) пристрій агрегує абонентські з'єднання та маршрутизує трафік до основної мережі та з неї. У розділеній архітектурі CORD у нас більше немає пристрою BNG як шлюзу до системи CO, а натомість ми реалізуємо необхідну функціональність маршрутизації як послугу віртуального маршрутизатора (vRouter). Сервіс vRouter не реалізує всю функціональність, яка існує в традиційних BNG-пристроях. Однак він реалізує функціонал, необхідний для забезпечення доступу до Інтернету до CORD у розрізненому вигляді.

Метою послуги vRouter є створення шлюзу між інфраструктурою CORD та висхідною мережею та надання доступу до Інтернету абонентам та послугам в CORD. Логічно, це остаточне обслуговування в ланцюзі, яке трафік користувача проходить перед виходом із системи CORD. Фізично це інтерфейс між CORD та вищою мережею постачальника. Послуга vRouter надає Інтернет-послугу іншим службам в рамках CO. vRouter реалізується як програма управління мережею, що працює на ONOS.

Конструкція послуги vRouter розділена на дві основні частини, які відносно незалежні одна від одної - площину управління та площину даних.

vRouter має деякі пристрої плоскості даних, якими він керує, і з зовнішньої точки зору ці пристрої виглядають так, ніби вони є єдиним маршрутизатором, підтримуючи набір протоколів маршрутизації.

Основна функціональність vRouter стосується протоколів маршрутизації розмов із зовнішніми маршрутизаторами. Щоб уникнути необхідності впровадження протоколів маршрутизації в додатку ONOS, ми вирішили використовувати існуючий стек маршрутизації з відкритим кодом під назвою Quagga. Quagga підтримує широкий спектр протоколів маршрутизації, що дозволяє vRouter підтримувати ці протоколи без необхідності їх повторної імплементації. На даний момент vRouter розглядає середовища з відносно невеликою кількістю маршрутів, тому не виникає проблем щодо продуктивності, пов'язаних із масштабами Інтернету.

Quagga буде налаштований для зв'язку з маршрутизаторами вище за течією - у польовій пробній справі це буде використовувати OSPF та iBGP.

Quagga визначає інтерфейс під назвою FIB Push Interface (FPI), який дозволяє йому пересувати маршрути до зовнішньої сутності. Цей інтерфейс ми використовуємо для зв'язку маршрутів від Quagga до ONOS. Додаток ONOS vRouter виконує функції диспетчера планів переадресації (FPM) і може приймати та декодувати маршрути від Quagga. Потім програма vRouter зможе використовувати ці маршрути для відповідного запрограмування площини даних.

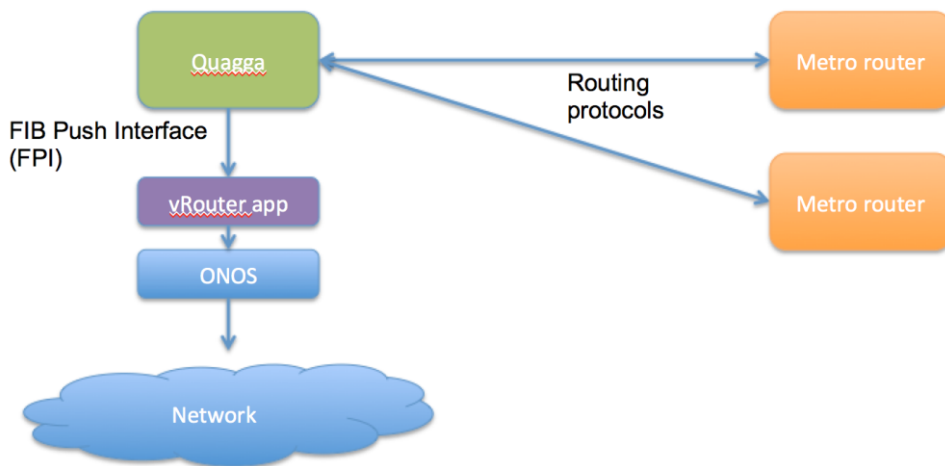


Рисунок 4.6: Архітектура площини управління vRouter

Конструкція площини управління аналогічна підходу, який використовується в застосуванні ONOS SDN-IP. Основна відмінність тут полягає в тому, що нам потрібно підтримувати більше, ніж просто пирінг BGP, тому що ми також повинні підтримувати IGP. SDN-IP використовує iBGP-з'єднання між Quagga та ONOS, тоді як з vRouter ми використовуємо інтерфейс FPM.

Контроль трафіку: для того, щоб Quagga мав можливість спілкуватися з маршрутизаторами вище, маршрутний трафік повинен протікати між сервером Quagga та зовнішнім маршрутизатором. Перш ніж обмінятися будь-якими маршрутами, першим завданням програми vRouter є запрограмувати площину даних, щоб цей трафік мав можливість протікати. Сервер Quagga підключений до порту на панелі даних vRouter, а вхідні / вихідні пакети маршрутизації спрямовуються на / з цього порту. Це обходить звичайну функцію маршрутизації vRouter, оскільки стосується трафіку управління площиною, який призначений для самого маршрутизатора.

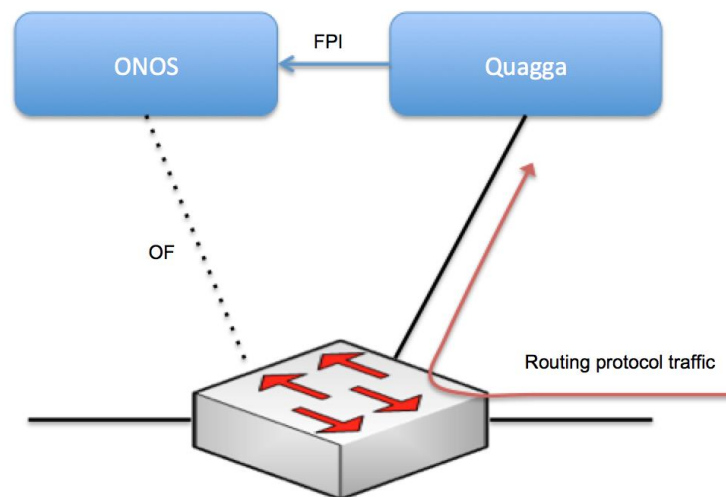


Рисунок 4.7: Керуючий трафік маршрутизації обробляється шляхом перенаправлення на екземпляр Quagga, підключений на площині даних

M-CORD

M-CORD - це базове рішення з відкритим кодом для операторів, що розгортають мобільні бездротові мережі 5G. Це хмарне рішення, побудоване на SDN, NFV та хмарних технологіях. Вона включає в себе як віртуалізацію функцій RAN, так і віртуалізований мобільний ядро (vEPC) для забезпечення можливості мобільних додатків та інноваційних сервісів за допомогою архітектури мікропослуг.

Побудована на інфраструктурній платформі CORD, яка приносить економіку центрів обробки даних та гнучкість у хмарі для операторських мереж, M-CORD перетворює мобільну мережу шляхом дезагрегації та віртуалізації функцій стільникової мережі, а також конкретних послуг для оператора. Це дає змогу створювати служби, що застосовуються до конкретних випадків, які можна динамічно масштабувати. M-CORD закладає основу для мереж і сервісів 5G через підтримку дезагрегованого і віртуалізованого розвинутого ядра пакетів, розрізнення від кінця до кінця від RAN до EPC, мобільних обчислень на крайовому рівні та програмованої мережі радіодоступу.

Stratum

Stratum - це операційна система комутаторів з відкритим кодом для програмно визначених мереж. Це створення відкритого, мінімально готового до виробництва розподілу для білих коробних комутаторів. Stratum відкриває набір інтерфейсів SDN нового покоління, що дозволяє взаємозамінність пристроїв переадресації та програмованість поведінки переадресації[13].

Stratum дозволяє уникнути блокування постачальників сучасних площин передачі даних (тобто власні інтерфейси кремнію та закриті програмні

API) та дозволяє легко інтегрувати пристрої в операторські мережі. Він пропонує повне рішення для перемикання білого поля для реалізації обіцянки SDN, визначеного програмним забезпеченням.

Проект Stratum розширює сферу використання SDN, включаючи повний інтерфейс управління життєвим циклом, конфігурацію та операційні інтерфейси.

ODTN

Проект "Відкрита та роз'єднана транспортна мережа" (ODTN) - це ініціатива, керована оператором для створення взаємозв'язків центру обробки даних, використовуючи розділене оптичне обладнання, відкриті та загальні стандарти та програмне забезпечення з відкритим кодом. ODTN має на меті стимулювати інновації та стати оптичною мережею за вибором, розбираючи компоненти мережі та надаючи відкрите програмне забезпечення для управління складанням компонентів, що постачаються кількома постачальниками.

ODTN дасть можливість оптимізованої екологічної системи «периферійних пристроїв», яка дозволяє поєднувати кілька компонентів та вбудовувати їх у комплексні рішення. Постачальники можуть зосередитись на створенні конкретного компонента (наприклад, транспондера), не будуючи повного рішення, що призводить до прискорених інновацій та зниження витрат. Оператори матимуть свободу вибору компонентів найкращого класу та уникати блокування постачальників, тим самим отримуючи гнучкість у міру зростання потреб їх мережі.[14]

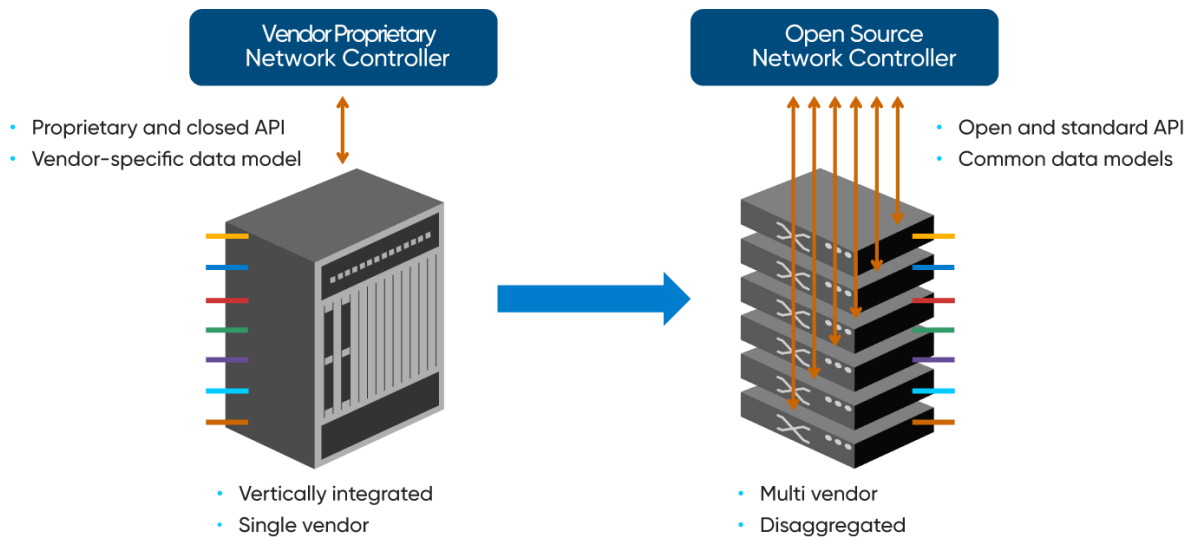


Рис. 4.8 ODTN Архітектура

Технічна складність, обумовлена аналоговим характером оптимічного зв'язку на великі відстані, зумовила необхідність сучасних вертикально інтегрованих рішень на ринку. Підхід ODTN покликаний змінити це, припускаючи, що кожне оптичне посилення використовуватиме збірну пару транспондерів від одного постачальника. Однак, на відміну від рішень одного постачальника, мережа може використовувати іншу марку транспондерів для кожної ланки, і ці транспондери можуть працювати над відкритою лінійною системою від іншого постачальника.

Використовуючи контролер ONOS SDN, ODTN автоматично та прозоро відкриє розчленовані компоненти та контролюватиме всю транспортну мережу як єдине ціле, тим самим даючи можливість вибору багатьох постачальників. ODTN покладається на відкриті галузеві стандарти, такі як TAPI (Transport API) і OpenConfig, щоб досягти справді нейтрального для постачальників рішення. Проект ODTN розпочнеться з відносно простих систем відкритої лінії "точка-точка" до складніших мережевих сценаріїв, і закінчиться мережевим мережа, що складається з розчленованого оптичного обладнання.

MININET

Mininet забезпечує віртуальну пробну версію та середовище розробки для програмно визначених мереж (SDN). Mininet дозволяє розробляти SDN на будь-якому ноутбукі чи ПК, а проекти SDN можуть безперешкодно переміщуватися між Mininet (що дозволяє недорогий і спрощений розвиток) та реальним обладнанням, що працює за лінійною швидкістю в реальному розгортанні. Mininet дозволяє

- Швидке прототипування програмно визначених мереж
- Комплексне тестування на топологію без необхідності підключення фізичної мережі
- Кілька одночасних розробників працювати самостійно над тією ж топологією

Mininet пропонує розширюваний API Python для створення мережі та експериментів. Він випускається під дозвільною ліцензією BSD Open Source та активно розробляється та підтримується спільнотою ентузіастів мереж та SDN.

Мережа Mininet складається з:

- ІЗОЛІТОВАНІ ДОМИ . Група процесів на рівні користувача перемістилася в мережевий простір імен, який забезпечує виключне володіння інтерфейсами, портами та таблицями маршрутизації.
- ЕМУЛОВАНІ ПОСИЛАННЯ. Linux Traffic Control (tc) примушує швидкість передачі даних кожного посилання для формування трафіку до налаштованої швидкості. Кожен емульований хост має власний віртуальний інтерфейс Ethernet.

- ЕМУЛЮВАНІ ВИМКНЕННЯ. За замовчуванням Linux Bridge або Open vSwitch, що працюють у режимі ядра, використовується для перемикання пакетів між інтерфейсами. Перемикачі та маршрутизатори можуть працювати в ядрі або в просторі користувача.

4.2 SDN та мережеві функції віртуалізації

Крім терміна SDN, я скажу що є термін NFV – Network Function Virtualization, тобто тут основною метою цього терміна полягає в тому що ми, по суті йдемо від дорогих апаратних мережних спеціалізованих пристроїв і ці функції ми перекладаємо на обчислювальну інфраструктуру по віртуалізації на сервера.

Віртуалізація мережевих функцій (NFV) - це концепція мережевої архітектури, яка використовує технології віртуалізації ІТ для віртуалізації функцій мережного вузла в будівельні блоки, які можуть з'єднуватися для створення служб зв'язку. Наприклад, NFV може ставитися до переміщення служб, таких як балансування навантаження і міжмережевий екран, із вибраного обладнання в віртуалізоване середовище. Рішення NFV в основному були розгорнуті в центрах обробки даних для хмарних платформ, що використовуються як на підприємствах, так і у постачальників послуг.



Рис.4.9 Архітектура SDN та NFV.

NFV і SDN - дві тісно пов'язані технології, які в деяких випадках доповнюють один одного. Більшість сучасних платформ NFV містять контролери SDN. З одного боку, хоча цілі NFV можуть бути досягнуті з використанням механізмів без SDN, при використанні принципів SDN це може підвищити продуктивність, спростити сумісність з існуючими розгорнутими і спростити процедури експлуатації і технічного обслуговування. З іншого боку, NFV може підтримувати SDN, надаючи інфраструктуру, на якій можна запускати програмне забезпечення SDN. Крім того, NFV тісно узгоджується з цілями SDN по використанню звичайних серверів і комутаторів. Очікується, що ці рішення можуть в кінцевому підсумку об'єднатися з системами оркестрації, такими як хмарні платформи управління або платформи оркестрації мережеслужб.

NFV — це частина змін, які відбуваються в роботі і взаємодії мережевого програмного і апаратного забезпечення. У поєднанні з SDN технологія NFV створює середовище з різноманітними можливостями автоматизації та програмування. Крім того, NFV дозволяє операторам і постачальникам послуг розгорнути більш клієнтоорієнтовані мережеві

інфраструктури, динамічно адаптуються до потреб та вимог клієнтів.

Оператори великих мереж звертаються до технології NFV, так як їх приваблює її програмовані відкриті стандарти. Крім того, вона позбавляє їх від обмежень, що накладаються власницькими мережевими платформами.

Всі ці функції можуть бути реалізовані за допомогою SDN. NFV можна вважати одним з основних випадків використання SDN у системах постачальників послуг. Також очевидно, що багато випадків використання SDN можуть включати поняття, ініційовані NFV. Наприклад, централізований контролер управляє розподіленою функцією пересилання пакетів, яка фактично може бути також віртуалізованою на чинному пристрої обробки або маршрутизації.

Не можна сказати, що перехід на технології SDN і NFV в мережах операторів буде гладким і безболісним. Тут також можна провести аналогію з переходом на цифрові станції від аналогових. Цей процес зайняв роки і навіть десятиліття. Багато хто пам'ятає повільне зростання «відсотка цифровізації» для операторських мереж. Точно також і зростання "NFV-зації" операторських мереж також буде поступовим. У міру вироблення терміну служби або морального старіння апаратних платформ IMS/EPC, мереж агрегації і доступу, буде відбуватися процес поступового переходу на «хмарні» платформи SDN/NFV, які будуть реалізовувати віртуалізацію vIMS/vEPC. Основна проблема, яка тут буде виникати – наявність власних дата-центрів у операторів, або доступність дата-центрів у провайдерів хмарних для аутсорсингу ресурсів стандартних серверів і систем зберігання даних.

Висновки до 4-го розділу:

1. Таким чином, можна зробити висновок, платформа SEBA надасть можливість удосконалити мережі доступу по всьому світу для обіцянки

відкритих, керованих програмним забезпеченням систем у нових ключових областях, таких як багатогігабітні мережі з волоконним доступом. А також R-CORD також включає в себе віртуальний шлюз абонентів (vSG) і використовує основний мережевий сервіс - віртуальний маршрутизатор (vRouter); перший реалізується контейнером, який прив'язаний до кожного абонента.

2. В даний час багато постачальників мереж інвестують в технології NFV і SDN. Драйвери і переваги технології NFV і SDN схожі, що не дивно, оскільки для досягнення своїх результатів NFV часто залежить від використання контролера SDN. Це може спростити управління мережею, прискорити надання нових послуг і потенційно скоротити витрати.

Висновок

В дипломній роботі було розглянуто концепцію побудови програмно-визначеної мережі, її основні елементи. Було досліджено архітектуру мережі SDN та її основні рівні. Більш детально було розглянуто елементи, які входять в дану телекомунікаційну мережу та які задачі вони виконують, також розглянули процес обслуговування пакетів SDN. Вивчили протоколи мережі SDN та відмінність між ними. Розглянули нові технології даної мережі на яких платформах вона реалізує своє майбутнє.

Чітко можна сказати, що програмно-конфігуруюча мережа використовує режим роботи, який іноді називають адаптивним або динамічним. Технологія SDN дозволяє значно підвищити ефективність телекомунікаційних мереж, пропускну здатність та якість обслуговування.

Список використаної літератури

1. Застосування SDN - рішень для оптимізації транспортних мереж мобільних операторів [Електронний ресурс] // Алексей Шалагинов. – – Режим доступу до ресурсу: <https://scinse.donntu.edu.ua/tks/volynskiy/diss/indexu.htm>
2. NGN [Електронний ресурс] // - Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/NGN>
3. ОСОБЛИВОСТІ ВЗАЄМОДІЇ КОНТРОЛЕРА І МЕРЕЖЕВИХ ПРИСТРОЇВ В МЕРЕЖАХ SDN [Електронний ресурс] // Навчальний посібник- Режим доступу до ресурсу: <http://conferenc.its.kpi.ua/2017/paper/download/6320/1650>
4. Особливості архітектури NGN [Електронний ресурс] // - Режим доступу до ресурсу: <http://www.znanius.com/3577.html>
5. Архітектура мереж зв'язку наступного покоління [Електронний ресурс] // - Режим доступу до ресурсу: <http://www.myshared.ru/slide/1412462/>
6. ВИКОРИСТАННЯ ПРОГРАМНО-КОНФІГУРОВАНИХ МЕРЕЖ [Електронний ресурс] // - Режим доступу до ресурсу: <http://elar.khnu.km.ua/jspui/bitstream/123456789/8722/1/27.pdf>
7. [Електронний ресурс] // - Режим доступу до ресурсу: http://tk-its.kpi.ua/sites/default/files/2020-02/%D0%94%D0%B8%D0%BF%D0%BB%D0%BE%D0%BC_%D0%9A%D0%BB%D0%B5%D1%86%D1%8C.pdf
8. Побудова SDN мереж [Електронний ресурс] // Навчальний посібник- Режим доступу до ресурсу: http://www.dut.edu.ua/uploads/1_1710_34882811.pdf
9. ONF Reference Design – SEBA [Electronic resource] // – Mode of access: <https://www.opennetworking.org/wp-content/uploads/2019/04/ONF-Reference-Design-SEBA-032919.pdf>
10. VOLTHA [Electronic resource] // – Mode of access: <https://www.opennetworking.org/voltha/>

11. SEBA [Electronic resource] // – Mode of access:
<https://www.opennetworking.org/seba/>
12. Trellis [Electronic resource] // – Mode of access:
<https://www.opennetworking.org/trellis/>
13. Stratum [Electronic resource] // – Mode of access:
<https://www.opennetworking.org/stratum/>
14. ODTN [Electronic resource] // – Mode of access:
<https://www.opennetworking.org/odtn/>
15. ONOS [Electronic resource] // – Mode of access:
<https://www.opennetworking.org/onos/>
16. Mininet [Electronic resource] // – Mode of access:
<https://www.opennetworking.org/mininet/>
17. What are SDN Controllers [Electronic resource] // – Mode of access:
<https://www.sdxcentral.com/networking/sdn/definitions/sdn-controllers/>
18. Residential CORD [Electronic resource] // – Mode of access:
<https://wiki.opencord.org/display/CORD/Residential+CORD>
19. Virtual Router [Electronic resource] // – Mode of access:
<https://wiki.opencord.org/pages/viewpage.action?pageId=1278093>
20. Virtual Subscriber Gateway [Electronic resource] // – Mode of access:
<https://wiki.opencord.org/pages/viewpage.action?pageId=1278090>
21. OpenFlow Switch Specification version 1.3.0 [Электронный ресурс] // Open Network Foundation - URL: <https://www.opennetworking.org/>
22. Mininet [Электронный ресурс]. - URL: <http://mininet.github.com/>
23. SDN [Электронный ресурс] // - Режим доступа до ресурсу: <https://sibsubis.ru/upload/c5c/%D0%91%D0%BE%D1%80%D0%BE%D0%B2%D0%B8%D0%BA%D0%BE%D0%B2%20%20%D0%90.%D0%90.%20%D0%98%D0%A3-323%20full.pdf>
24. Software-Defined Networking: The New Norm for Networks [Electronic resource] // Open Networking Foundation. – [2012]. – Mode of access: <https://>

www.opennet working. org/images/stories/downloads/sdn-resources/whitepapers/wp-sdn-newnorm.pdf.

25. Руководство по SDN и NFV [Электронный ресурс] // Алексей Шалагинов.

– – Режим доступа до ресурсу:

<https://shalaginov.com/2018/01/16/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE-%D0%BF%D0%BE-sdn-%D0%B8-nfv-1/>