НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут телекомунікаційних систем

Кафедра телекомунікацій

«На правах рукопису»

УДК 621.391

«До захисту допущено»
Завідувач кафедри
_____ Сергій КРАВЧУК
«___»_____2021 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою «Інженерія та програмування інфокомунікацій»
зі спеціальності 172 «Телекомунікації та радіотехніка»

на тему: «Широкосмуговий доступ з підтримкою SDN в базових телекомунікаційних мережах»

Виконав:
студентка ІІ курсу, групи ТЗ-01мп
Кочура М.Ю. _____

Керівник: директор НН ІТС,
д.т.н., професор кафедри ТК, Ільченко М. Ю. _____

Консультант з розділу: професор кафедри ТК
д.т.н, професор Романов О.І. _____

Рецензент:
доцент кафедри ІКТС НН ІТС
к.т.н., доцент Созонник Г.Д. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студентка _____

Київ – 2021 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій КРАВЧУК
«___»_____2021 р.

ЗАВДАННЯ
на магістерську дисертацію студентці
Кочурі Марині Юріївні

1. Тема дисертації «Широкосмуговий доступ з підтримкою SDN в базових телекомунікаційних мережах», науковий керівник дисертації директор ІТС, академік НАНУ, доктор технічних наук, професор, Ільченко Михайло Юхимович затверджені наказом по університету від «04» листопада 2021 р. № 3673-с.

2. Термін подання студентом дисертації 10.12.2021 р.

3. Об’єкт дослідження: мережа SDN

4. Предмет дослідження: Управління мережами SDN

5. Перелік завдань, які потрібно розробити :

a. Аналіз мереж попередніх поколінь
b. Аналіз архітектури мережі SDN
c. Аналіз принципу роботи протоколу OpenFlow
d. Аналіз принципу роботи протоколу OF-Config

e. Аналіз взаємодії контроллера і мережевих пристроїв в мережах SDN.
f. Внесення пропозицый щодо покращення роботи мережі SDN

6. Орієнтовний перелік ілюстративного матеріалу

7. Орієнтовний перелік публікацій: Міжнародно науково-технічна конференція «Перспективи телекомунікацій 2021»

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Аналіз напрямківірозвитку телекомунікаційної мережі	Романов О.І., д.т.н., проф. ІТС НТУУ "КПІ" каф. ТК	05.10.2020	25.10.2020
Структура ііосновні елементи мереж SDN	Романов О.І., д.т.н., проф. ІТС НТУУ "КПІ" каф. ТК	05.10.2020	28.01.2021
Сучасні платформиі побудови мереж SDN	Романов О.І., д.т.н., проф. ІТС НТУУ "КПІ" каф. ТК	05.10.2020	22.04.2021
Основні функціональні елементи	Романов О.І., д.т.н., проф. ІТС НТУУ "КПІ" каф. ТК	05.10.2020	09.08.2021
Розробленняістартап проєкту	Романов О.І., д.т.н., проф. ІТС НТУУ "КПІ" каф. ТК	05.10.2020	15.10.2021

9. Дата видачі завдання “ 15 ” вересня 2019 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Формулювання теми, мети, об’єкту та предмету дослідження у магістерській дисертації	03.09.2020-15.09.2021	виконано
2.	Формулювання проблематики та визначення актуальності магістерської роботи.	15.09.2020-25.09.2021	виконано
3.	Пошук інформації про протоколи OpenFlow, OF-Config	25.09.2020-01.10.2021	виконано
4.	Пошук інформації про рівень мережевих сервісів	02.10.2020-27.10.2021	виконано
5.	Аналіз процесу обслуговування пакетів у мережі SDN на основі комутаторів OpenFlow	28.10.2020-20.11.2021	виконано
6.	Висновки	21.11.2021-01.11.2021	виконано
7.	Оформлення готової роботи	02.12.2021-10.12.2021	виконано
8.	Надання МД з презентацією до комісії з захисту МД	10.12.2021-22.12.2021	

Студентка	Марина КОЧУРА
Науковий керівник дисертації	Михайло ІЛЬЧЕНКО

РЕФЕРАТ

Магістерська дисертація містить: 86 сторінок, 34 рисунків, 16 таблиць, 20 посилань.

Актуальність: На даний момент існує багато варіантів реалізації концепції SDN для різних сценаріїв використання, тому перед споживачами даного продукту постає складне питання вибору необхідної архітектури SDN, яка задовольнятиме вимоги до конкретної мережі. Для того, щоб обрати підходящу архітектуру необхідно розуміти принципи роботи, побудови та відмінності у роботі різних реалізацій SDN. Ключовим моментом роботи мережі SDN є метод , згідно з яким контролер управляє елементами мережі , адже від нього залежить швидкість реакції елементів на команди контролера, ефективність використання наявних ресурсів мережі, складність інтеграції архітектури SDN з існуючою інфраструктурою і навіть різноманітність доступного функціоналу.

Мета . Дослідити принцип роботи системи підвищення ефективності архітектури розробки рішень для операторського широкосмугового доступу.

Об'єкт дослідження: мережа SDN

Предмет дослідження: досліджений приницип роботи повинен дозволяти легко адаптувати нові технології, новий фундамент, що підтримує технології, і нові пристрої, які включають ці технології та фундамент у елементи , що розгортаються.

Ключові слова: SDN, OpenFlow ,маршрутизатор, SEBA

ABSTRACT

The work contains: 86 pages, 34 pictures, 20 links , 16 tables

Topicality. There are many options for implementing the SDN concept for different usage scenarios, so the consumers of this product face the difficult choose of the SDN architecture that will meet the requirements of a specified network. In order to choose the right architecture, it is necessary to understand the working principles, construction and differences in the operation of different SDN implementations. The key to the operation of the SDN network is the method by which the controller manages the network elements, as it depends on the response time of the elements to the controller commands, the efficiency of using available network resources, the complexity of integrating the SDN architecture to the existing infrastructure and even the variety of available functionality.

Goal. Investigate the principle of operation of the system to increase the efficiency of architecture development and solutions for operator broadband access.

Object of research: the researched principle of work should allow to adapt easily new technologies, the new base supporting technologies, and the new devices which include these technologies and the base in developing elements.

Key words: SDN, OpenFlow , router, network, SEBA

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

API	інтерфейс прикладного програмування
SDN	Software Defined Networks/Networking
vRouter	Virtual Router
NGN	Next Generation Network
ASG	Агрегація та сервісний шлюз
BBF	Broadband Forum
BMC	Контролер управління радою
BNG	Широкосмуговий мережевий шлюз
CAP	Carrier Automation Platform
CI/CD	Безперервна інтеграція/Безперервна доставка
CORD	Центральний офіс Перезареєстрований як центр обробки даних
CUPS	управління та поділу площини користувача
DOCSIS	Data Over Cable Service Interface Специфікація
DPU	Distribution Point Unit
EPON	Пасивна оптична мережа Ethernet
FCAPS	Управління несправностями, конфігурація, облік, продуктивність та безпека
FWA	Фіксований бездротовий доступ
NBI	Інтерфейс NorthBound
NEM	Посередник мережевих країв
NFV	Віртуалізація мережевої функції
NG-PON2	Пасивна оптична мережа наступного покоління 2
OLT	Термінал оптичної лінії OLT
ONAP	Відкрита платформа автоматизації мережі
ONOS	Відкрита мережева операційна система
ONT	Оптичний мережевий термінал
ONU	Оптичний мережевий блок

OSS	Підсистема операцій
PNF	Фізична функція мереж
POD	Точка доставки
PON	Пасивна оптична мережа
RD	Еталонний дизайн
SDN	Програмно-визначені мережі
SEBA	Широкосмуговий доступ з підтримкою SDN
VOLTHA	Віртуальна оптична лінія терміналу обладнання абстракції
xDSL	x Цифрова абонентська лінія

ВСТУП

Кожного року в технічному суспільстві все більше зростають вимоги до комп'ютерних мереж . Це пов'язано із стрімким розвитком технологій. В наслідок цього ми можемо спостерігати що кожного року вимог стає все більше, а мережі стають все складнішими. Мережі повинні бути швидшими та більш досконалими в моніторингу. Такий стан речей наштовхує людство на створення нових рішень , нових пропозицій щодо реалізації компютерних мереж.

Тому протягом останніх декількох років зрозстає популярність програмно -конфігуруючих мереж із застосуванням протоколу OpenFlow.

Головним завданням програмно-конфігуруючої мережі є розділення рівня додатків від рівня управління, та розділення рівня управління від рівня передачі данних. Завдяки цьому виникає можливість спрощення складності фізичних пристроїв. Виконування всіх логічних функцій переноситься на вищий рівень . Такий підхід дозволяє здешевити фізичні пристрої, покращити надійність системи та спростити керування мережею. В програмно-конфігуруючих мережах відпадає потреба у використонні маршрутизаторів, адже цю роль бере на себе мозок мережі-контроллер. [1]

РОЗДІЛ 1
АНАЛІЗ НАПРЯМКІВІРОЗВИТКУ 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].

З розвитком телекомунікаційних послуг стали все більш популярними обговорення різноманітних варіантів архітектури мережі NGN, які поєднуюють в єдину структуру мережі мобільного звязку, ресурси мережі інтернет тощо. Особливою рисою моделі, запропонованою сектором стандартизації електрозв'язку ITU-T, є функціональний розподіл на два рівні : послуг і транспортний (Рис 1.1). Рівень послуг виконує прикладні функції, наприклад організацію передачі мови, відеозображення або їх комбінацію. Транспортній рівень виконує функції доставки інформації різних типів між абонентами мережі

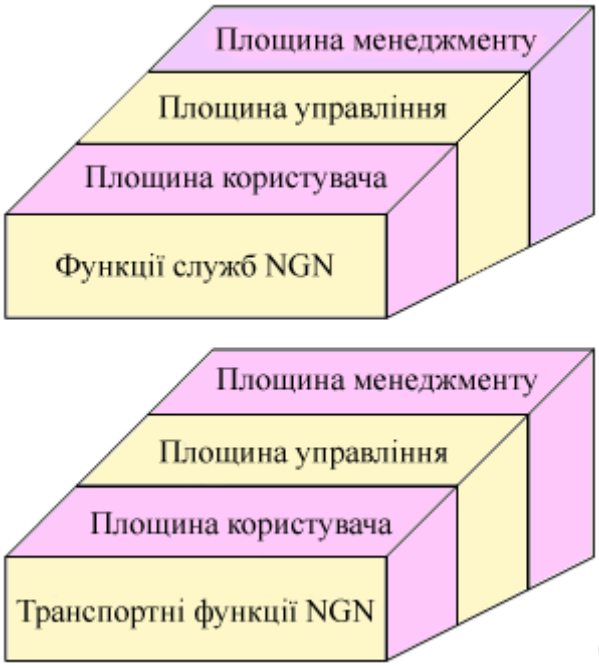

Рис 1.1 Розподіл функцій транспорту та служб в архітектурі NGN

На даний час великого поширення набула чотирирівнева архітектура

NGN, у якій рівень послуг та транспортний рівень замінили такі рівні як:

- Рівень управління послугами (четвертий рівень)
- Рівень мережного контролю й управління (третій рівень)
- Транспортний рівень (другий рівень)
- Рівень доступу (перший рівень)
- Термінальне обладнання (нульовий рівень)

Традиційна модель (Рис. 1.2) передбачає , що платформи для надання послуг підключатимуться до мережі за допомогою стандартних інтерфейсів ЗКС № 7 , SIP, Parlay, H.323

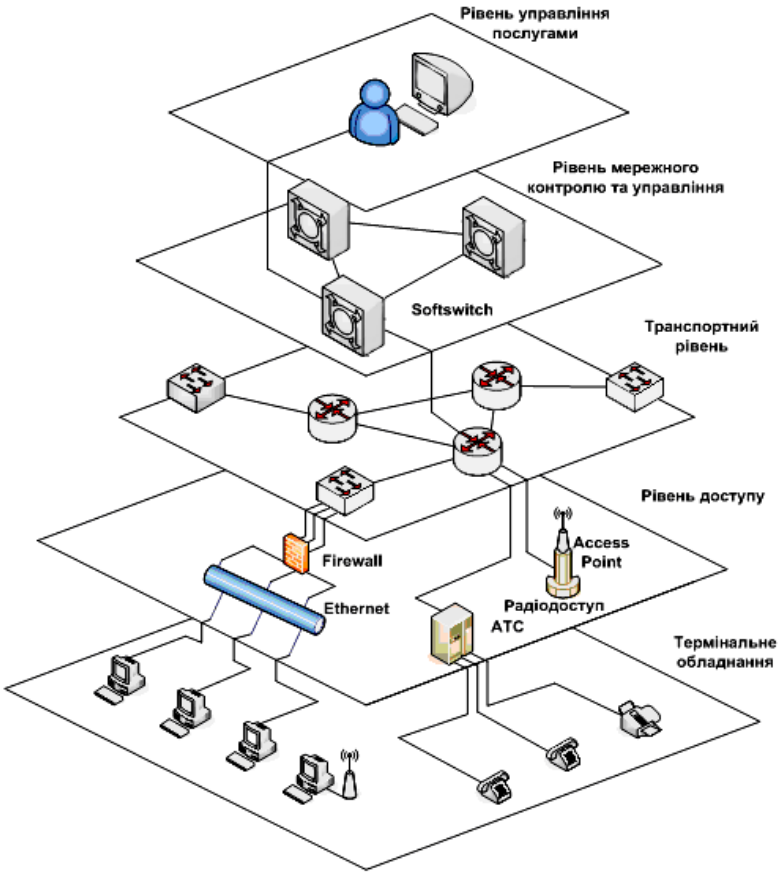

Рис. 1.2 Функціональна модель мережі NGN, адаптована для надання послуг NGS

Однак при впровадженні мережі NGN в експлуатацію виникє ряд недоліків таких як :

- Відсутність чіткої нормативної бази

- Проблема взаємодії обладнання різних виробників
- Недостатня надійність обладнання
- Недостатня зрілість послуг для мереж нового покоління.

Саме ці , та інші фактори стали головною причиною пошуку іншого рішення реалізації мережі операторів. 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.

Оператор повинен думати про керовану 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 неповинні 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.

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 маршрутизировать 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 канальної 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)

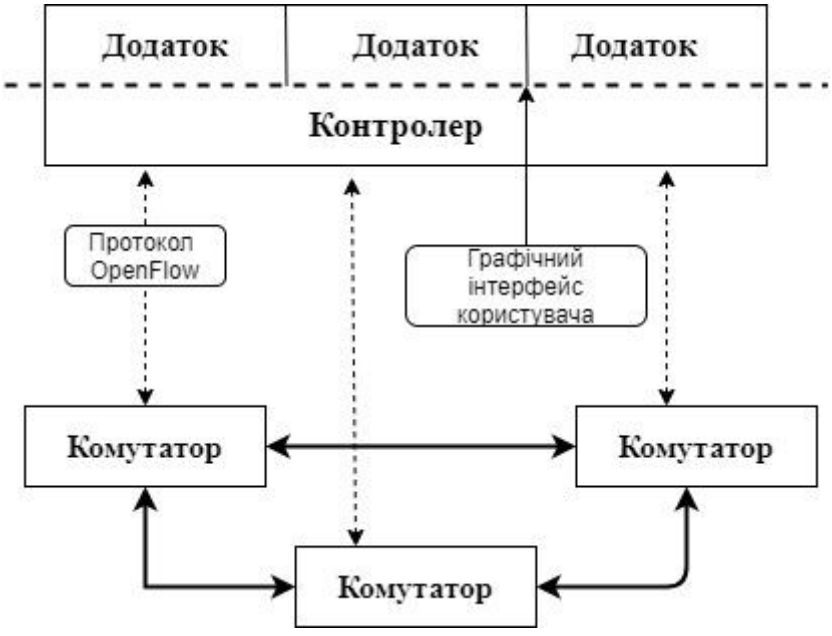

Рис. 1 .3 Принцип реалізації концепції SDN

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, безпеку 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 :

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

Також концепція програмно-визначених мереж (SDN) грунтовно 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.

РОЗДІЛ 2
СТРУКТУРА І ОСНОВНІ ЕЛЕМЕНТИ МЕРЕЖ SDN

2.1 Структура мережі SDN.

Програмно-конфігуруючі мережі на данний час представляють нову архітектуру побудови мережі, яка розділяє в собі управління та передачу трафіку . Організаіція «Open Networking Foundation (ONF)» займає лідируючу позицію у стандартизації SDN.

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

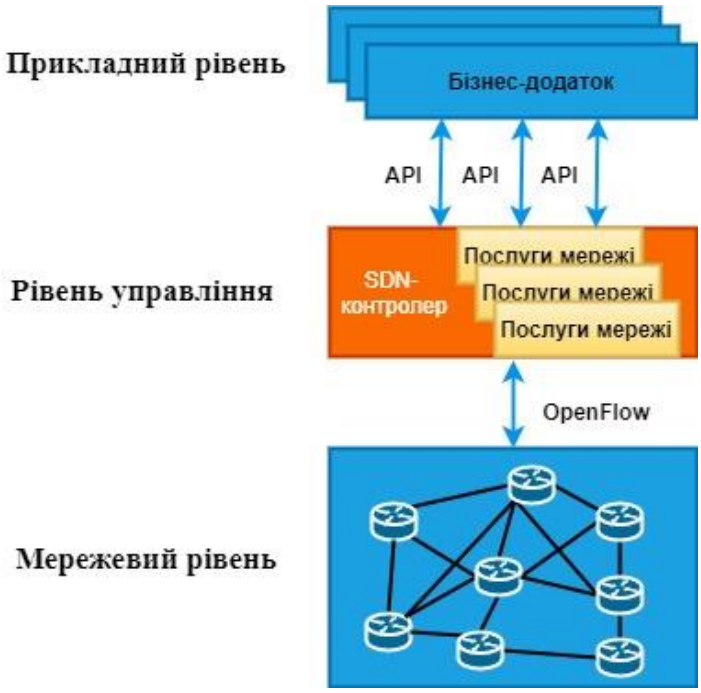

Рис. 2.1 Архітектура програмно-конфігуруючих мереж

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 аналіз 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.

показані на рис.1.2 та рис.1.3 :

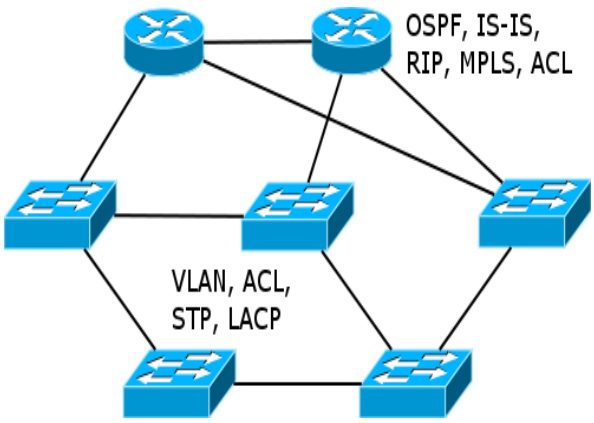

Рис.2.2 Традиційна мережа

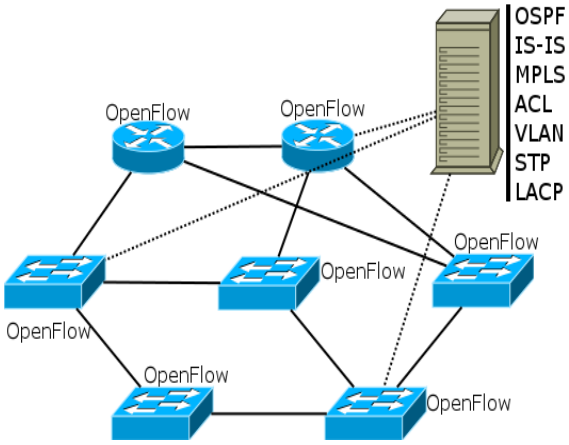

Рис.2 .3 Мережа по використанню SDN

На рис 2.2 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

Контролери в програмній мережі є головним «мозком» мережі. Ці додатки, які виступають в якості стратегічного контрольного пункту в мережі SDN, керує потоками для комутаторів, маршрутизаторів додатків та бізнес-логіки для розгортання інтелектуальних мереж . В останній час контролери

отримали здатність з’єднувати між доменами SDN Controller, використовуючи інтерфейси додатків , такі як відкрита база даних комутаторів та OpenFlow.

Платформа SDN Controller зазвичай має набір підключаючих модулів, які можуть використовувати мережеві задачі різних типів. Також додаються розширення, які покращують функціональність та підтримують деякі функції, такі як виконання алгоритмів для аналітики та організації нових правил а мережі.

Одним з найбільш відомим протоколом, який використовується в мережі SDN, є протокол OpenFlow.

SDN контролери мають унікальну здатність підтримувати глобальну видимість мережі. При керуванні від SDN - контролерів, функції передачі пакетів в мережевих елементах не займаються маршрутизацією пакетів. Цю задачу виконує SDN контролер. Саме тому функції передачі мають назву «білі ящики». Але це не виключає їх взаємодії з «маршрутизуючими ящиками» тобто з традиційними маршрутизаторами. Така архітектура має низку особливостей:

- Логічне централізоване керування
- Абстрагування рівнів мережі та програмування топології мережі.
- Підтримка забезпечення від різних постачальників за допомогою існуючих та нових протоколів.
- Можливість розгортання та активації послуг програмним шляхом.

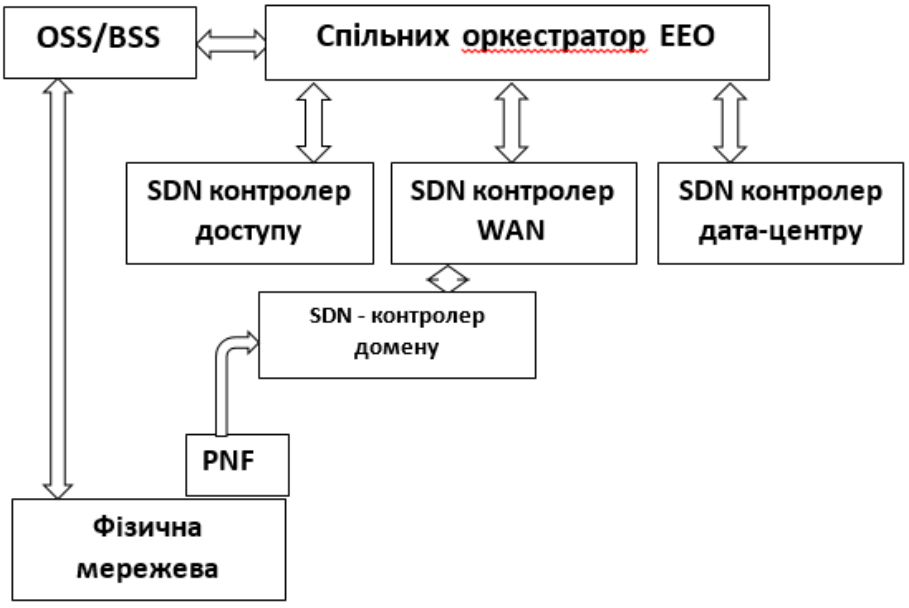

Рис. 2.4 Архітектура SDN-контролерів.

На рисунку 2.4 зображено місце різних контролерів програмн-конфікуруючої мережі в її архітектурі .

Основною особливістю цієї архітектури є те, що вона дозволяє існування декількох різних доменів мережі , кожен із яких має можливість керуватись своїм контролером. Міждомене керування виконується в двох форматах :

- Через спільний оркестратор EEO, у випадку, якщо домени відносно незалежні. Наприклад, для послуги віртуального пристрою користувача vCPE (virtual Customer Premise Equipment) може бути потрібною послуга IP-VPN в WAN, яка підключається до цепочки послуг в дата-центрі, щоб скерувати трафік через різноманітні функції, так як міжмережевий екран та NAT. Послуги в SDN-WAN і в дата -центрі можуть бути незалежні друг від друга . Оркестратор мережевих ресурсів, який пов’язує ці послуги , повинен ідентифікувати граничну точку передачі послуги та встановлювати зв'язок між ідентифікаторами. Другий приклад – сценарій , коли домен SDN являється частиною послуги під гібридним керуванням.
- Через SDN - контролер: ця форма крос-доменного керування застосовується у випадку керування ресурсами всього з’єднання. Наприклад у випадку повного з’єднання через оптоволоконну мережу DWDM по мережі , побудованій на приладах різних вендорів. В межах існуючого потоку неможливо керувати ресурсами оптоволоконної мережі одного, чи іншого веднора від стороннього контролера . Для такого керування потрібно мати свої контролери для кожного домена мережі, побудованого на приладах кожного вендора. Тому керування такими повними з’єднаннями виконує контролер WAN SDN. Він отримує інформацію про стан мережі від компонентів мережі, та використовує їїї для розрахунку оптимального маршруту. Після того він послідовно інструктує контролери кожного домену для встановлення з’єднання через сегменти мережі в кожному домені.

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.

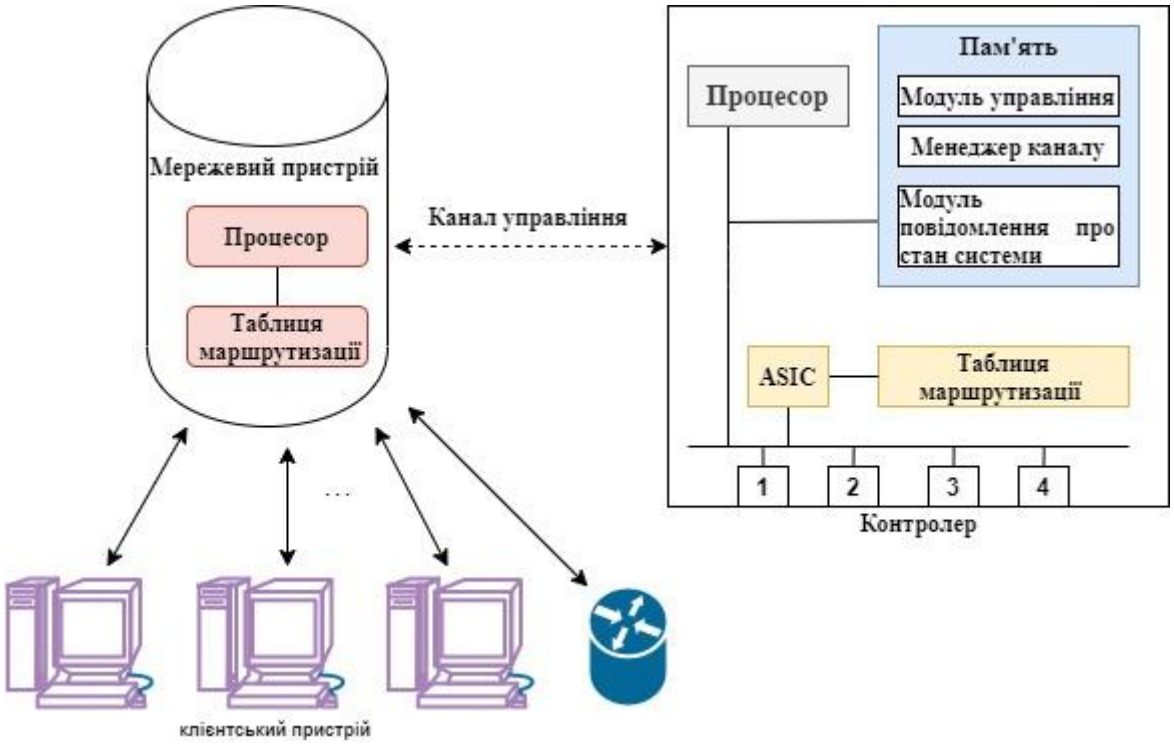

Рис. 2.5 Взаємодія мережевого пристрою SDN (МП-SDN) і контролера SDN (К-SDN).

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 необмежується тільки одним типом стану системи.

2.3 Структура комутатора SDN

Комутатор OpenFlow - це комутатор даних з підтримкою OpenFlow, який зв'язується по каналу OpenFlow з зовнішнім контролером. Він виконує пошук і пересилку пакетів відповідно до однієї або декількома таблицями потоків і таблицею груп. Комутатор OpenFlow зв'язується з контролером, і контролер управляє комутатором через протокол комутатора OpenFlow. Вони або засновані на протоколі OpenFlow, або сумісні з 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.

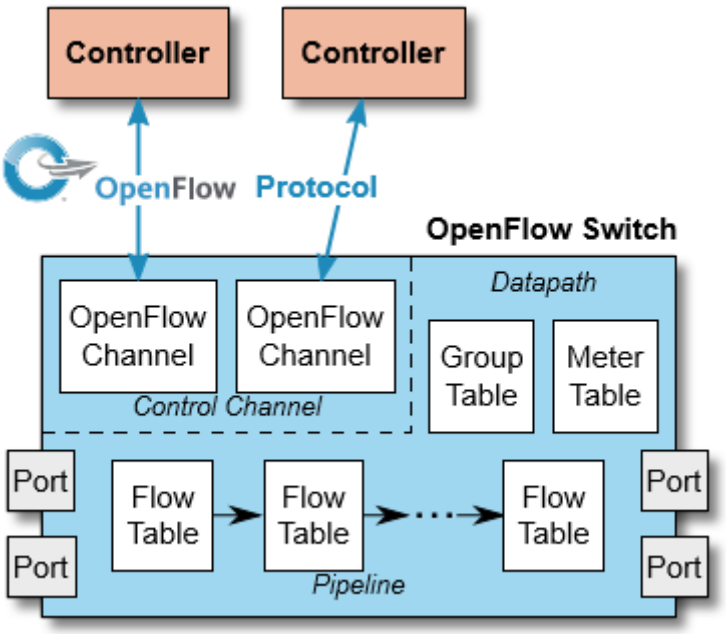

Рис. 2.6. Основні компоненти комутатора OpenFlow.

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 знайдена, 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, багатопроменеве 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.

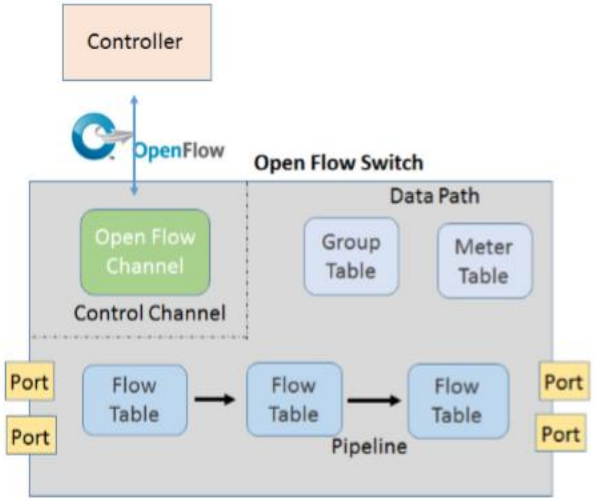

Рис. 2.7 Складові OpenFlow комутатора.

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 неасоціюються 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 відповідніть 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 неспрямовує 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.

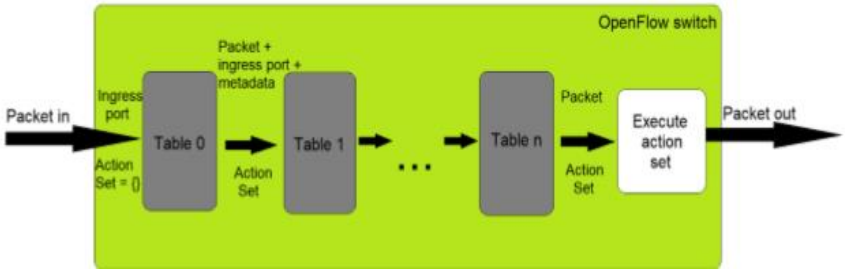

Рис. 2.8 Потік пакетів через конвеєрну обробку OpenFlow.

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 небуде 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.

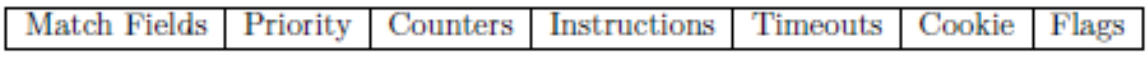

Рис. 2.9. Основні компоненти запису потоку в таблиці потоків.

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

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.

2.5 Протокол OpenFlow

Стандарт OpenFlow, створений в 2008 році, був визнаний першою архітектурою SDN, яка визначила, як елементи управління і площини даних будуть розділені і взаємодіяти один з одним за допомогою протоколу OpenFlow. Open Network Foundation (ONF) є органом , відповідальним за управління стандартами OpenFlow, які є відкритим вихідним кодом. Архітектура контролера OpenFlow складається з декількох рівнів, кожен з яких відповідає за ряд необхідних функціональних можливостей.

Основними рівнями OpenFlow контролера є рівень взаємодії з мережею ; рівень обробки OpenFlow повідомлень; рівень обробки подій ; рівень мережевих сервісів і внутрішніх додатків; інтерфейс для мережевих додатків контролера; рівень мережевих додатків . За своєю суттю контролер виступає в ролі елемента реагування який отримує повідомлення від комутаторів по каналах управління і виробляє відгуки, які змінюють вміст таблиць комутації. OpenFlow комутатор є сполучною ланкою між мережевою інфраструктурою та контролером . [3].

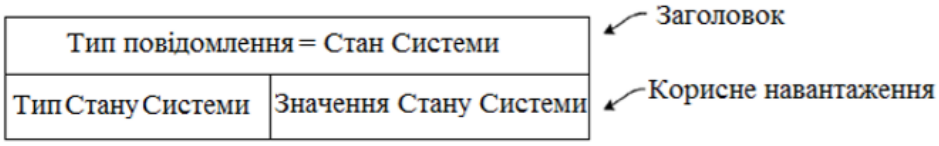

Рис. 2.10 Загальна структура повідомлення про стану системи (1)

На рис.2 .10, 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.

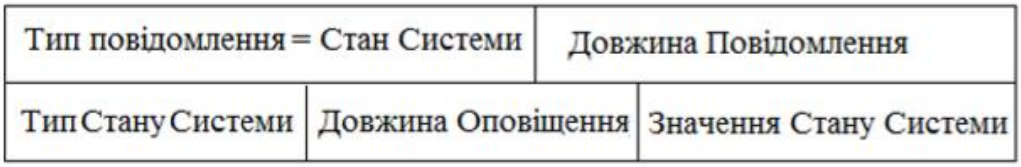

Рис.2.11 загальної структури для повідомлення про стану системи(2)

На рис. 2.11 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.

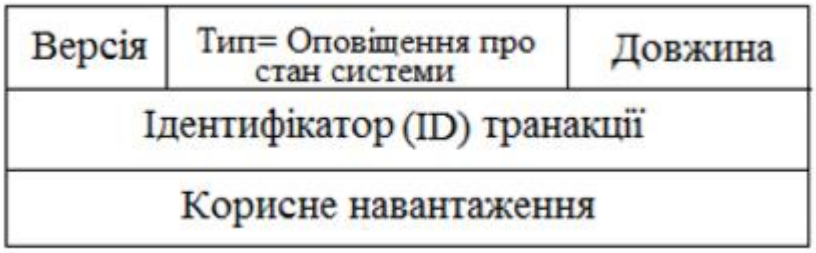

Рис. 2.12. Заголовок повідомлення про стан системи протоколу OpenFlow.

Рис.2.12 ілюструє конкретний приклад заголовку повідомлення про стан системи, використовуючи протокол OpenFlow більш докладно. Додатково до полів , вже описаним на рис.2.12 , заголовок на рис.2.13 включає в себе поле версії протоколу OpenFlow (OFP), яке вказує, що повідомлення відповідає OFP, а також може вказувати версію OFP, використовуваному, 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.2 .11, або на

рис. 2.12 .

Даний протокол розроблений для вирішення завдань більш високого, посравнению 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 .

2.6 Протокол OF-CONFIG

Протоколом OFCONFIG (OpenFlow Management and Configuration Protocol) передбачені наступні абстракції (Рис 2.14):

- 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].

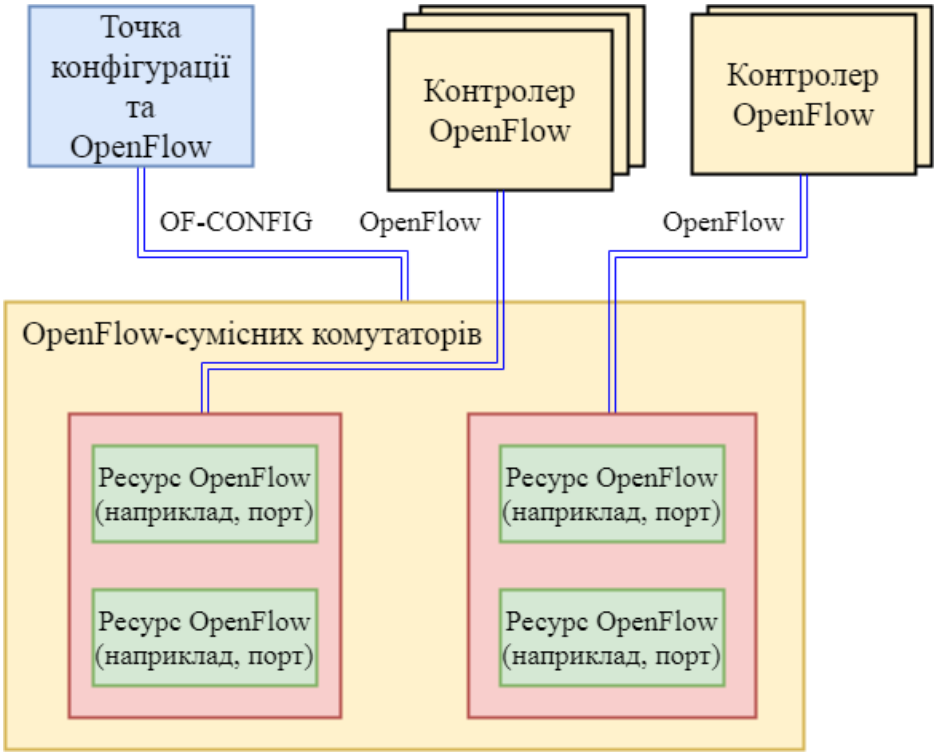

Рис. 2.13 - Основні абстракції протоколу OF-CONFIG

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.

2.7 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.2.14 представлена схема взаємодії різних частин системи конфігурації і управління перемикачем.

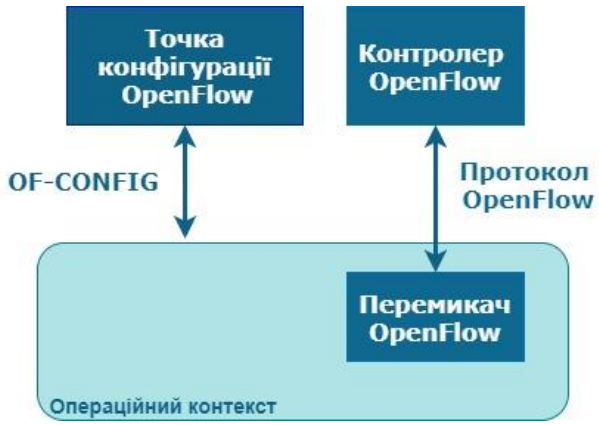

Рис .2.14 Точка конфігурації OpenFlow взаємодіє з операційним контекстом, який може підтримувати роботу перемикача OpenFlow, використовуючи конфігурацію і протокол управління (OF-CONFIG)

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 непотрібно вирішувати задачу в інтервалі часу проходження потоку і, отже , може виконуватися по протоколу - OF - CONFIG.

Висновки :

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 резевні порти .
6. 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.

РОЗДІЛ 3
СУЧАСНІ ПЛАТФОРМИ ПОБУДОВИ МЕРЕЖ SDN

3.1 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].

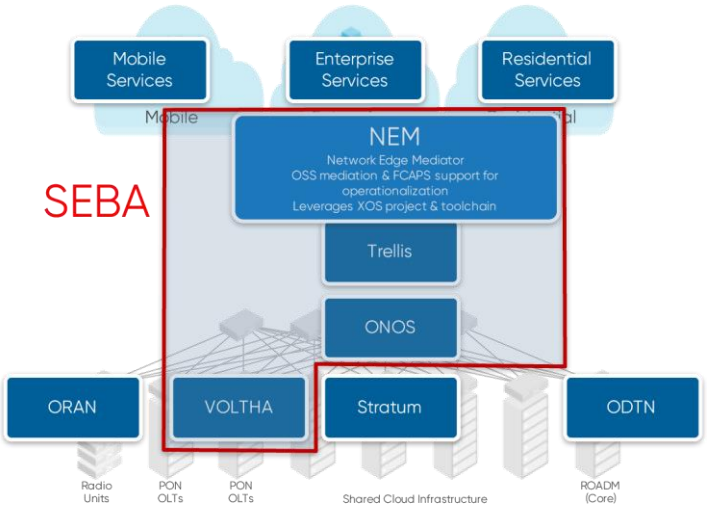

Рис. 3.1 програмне забезпечення SEBA

Як показано на рисунку 3.1, SEBA є об'єднанням проектів , таких як VOLTHA, ONOS і TRELLIS.

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 проєкта VOLTHA, ONOS і TRELLIS

ONOS

ONOS - контролер SDN, який може управляти VOLTHA 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.

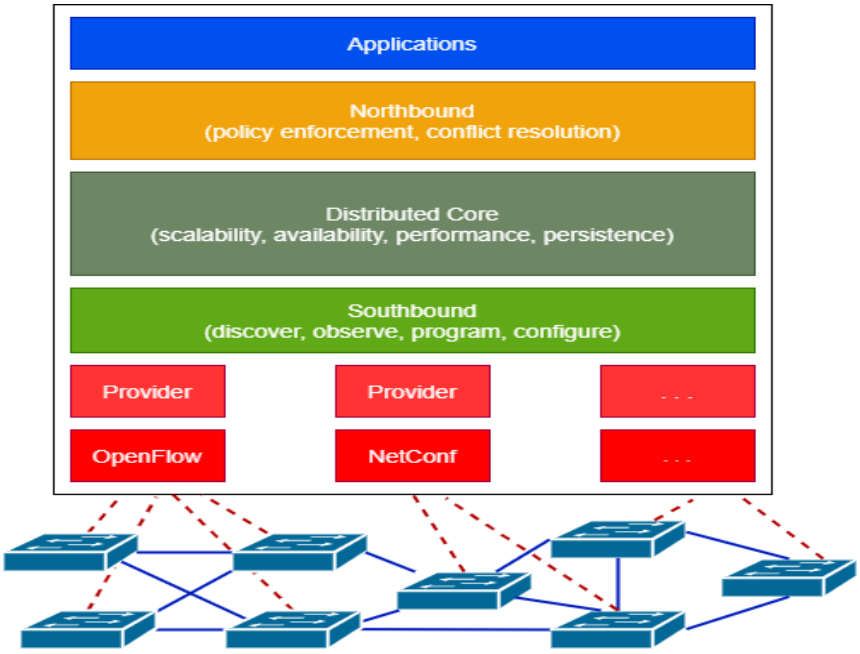

Рис. 3.2 Архітектура ONOS

1. Містить користувальницькі додатки, наприклад, реактивну переадресацію, ProxyARP, 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 кластеризацию. для підтримки HA і масштабованості

4. Надати абстрактний інтерфейс для управління мережевою інфраструктурою

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

VOLTHA

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

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

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

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

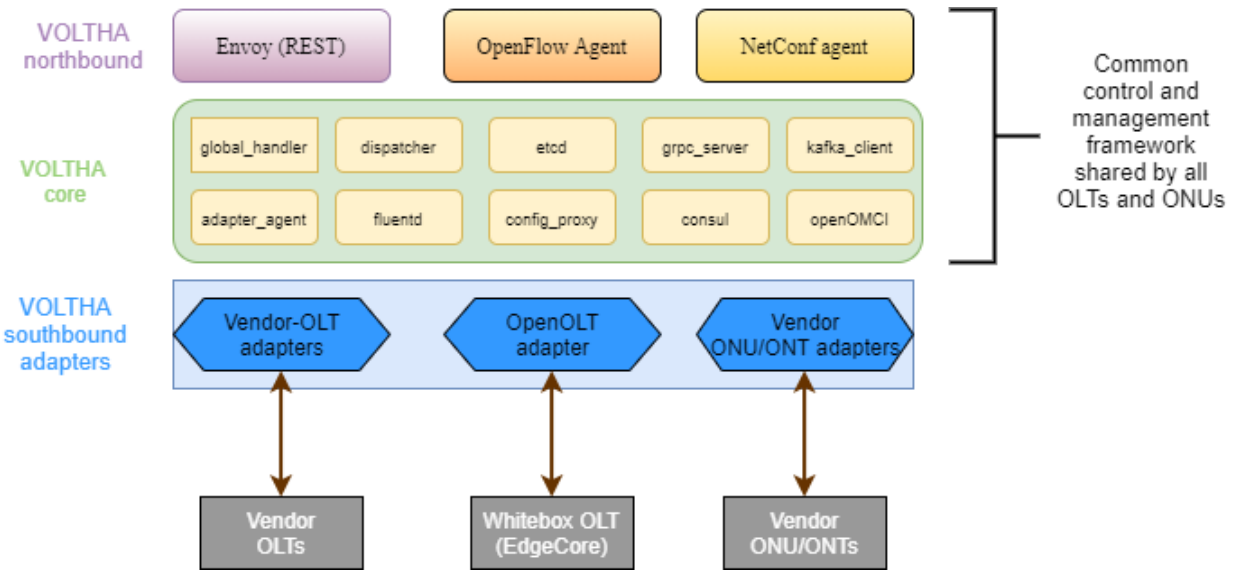

Рис 3.3 Архітектура VOLTHA

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.

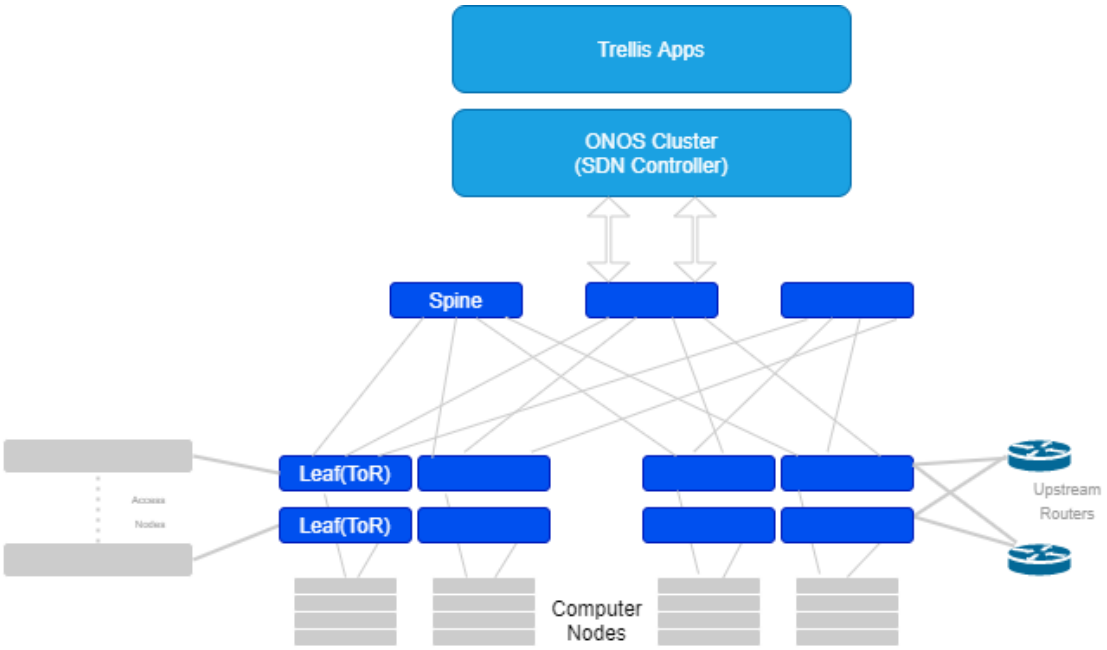

Рис 3.4 Архітектура Trellis

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 висувають 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 3.5 , 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)

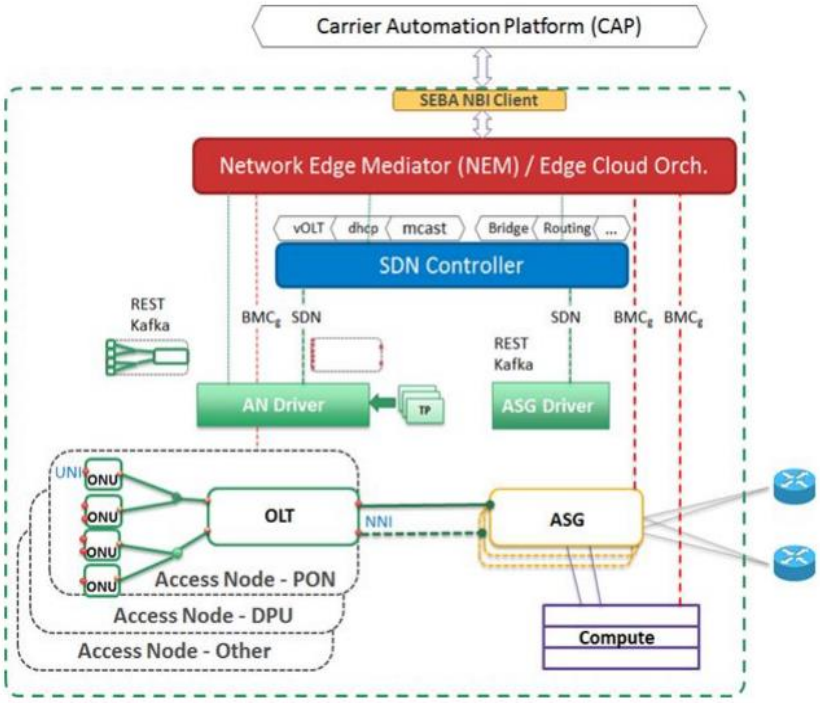

Рис .3.5 Діаграма цільової архітектури

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 api / 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)

Аналітика для CORD

Цілі цієї ініціативи:

(1) побудувати структуру моніторингу загального призначення, яка збирає, архівує та доставляє (за допомогою запитів чи сповіщень) зонди / показники ефективності фізичних пристроїв та програмних елементів у CORD;

(2) продемонструвати цю структуру в сервісі аналітичних програм, які динамічно адаптують CORD на основі навантаження та аномальних подій.

Основними особливостями A-CORD є:

- Забезпечує глибоку спостережливість як фізичних мережевих елементів, так і програмного елемента всередині CORD: пристрої з роз'єднаним доступом (GPON OLT), розблоковані основні пристрої (ROADM), комутатори білої панелі тканини, апаратне обчислення / зберігання, віртуальні комутатори програмного забезпечення (OVS), програмне забезпечення віртуалізовані послуги, що працюють на інфраструктурі
- Забезпечує не тільки моніторинг рівня інфраструктури, але й моніторинг рівня обслуговування, тим самим забезпечує можливість

спостережливості в кінці

- Програмні інтерфейси для контролю рівня спостережливості
- Розв'язує додатки Analytics від базових зондів , тим самим дозволяючи справжнє середовище для багатьох постачальників
- Дозволити багатьом програмам споживати однакові зібрані дані. тобто збирати один раз і користуватися багатьма
- Програми також можуть бути видавцями даних такими, що агреговані події можуть подаватися в систему, яка може споживатися іншими програмами

Інформацію про загальну архітектуру можна знайти в A-CORD_Architecture та в примітці проектування CORD - Служба моніторингу CORD

Служба моніторингу - компоненти програмного забезпечення

Служба моніторингу розроблена як багатокористувацька, масштабована послуга всередині CORD, використовуючи інструментарій побудови служб XOS, де кожна служба складається з "сервісного контролера" та набору "службових примірників".

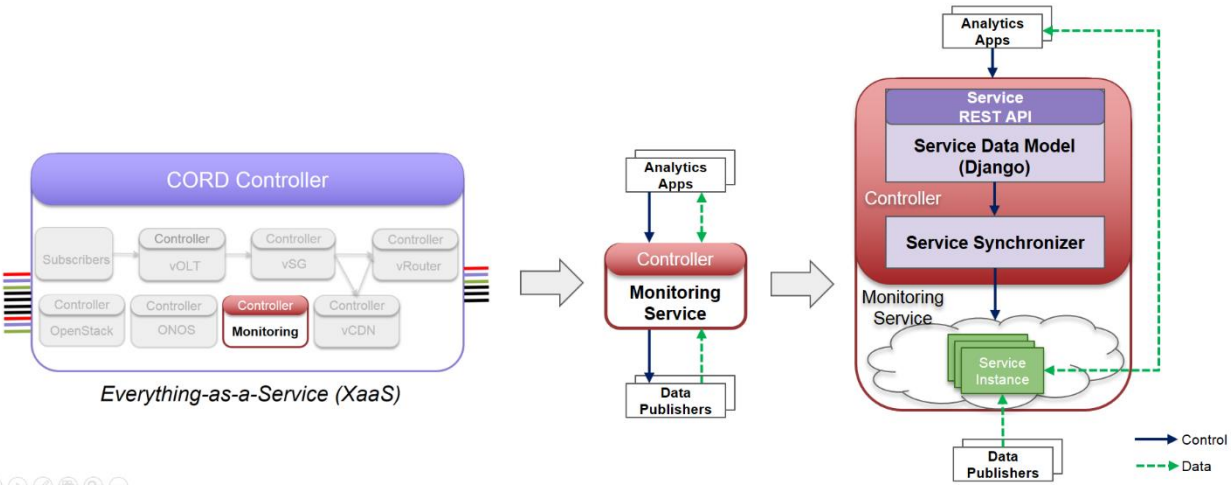

Рис. 3 .6: Служба моніторингу як послуга багатокористувача всередині CORD

Зазвичай компоненти "Service Controller" запускаються як частина платформи XOS на головному вузлі, а "Service Examences", які реалізують Службу, буде інстанціюватися на кластері CORD Compute.

"Контролер обслуговування" складається з:

- Модель Джанго, яка визначає авторитетний стан обслуговування
- Сервіс та конфігурація орендарів на основі TOSCA
- Синхронізатор, який підтримує бек-енд-системи , що реалізують послугу в синхронізації з авторитетним станом

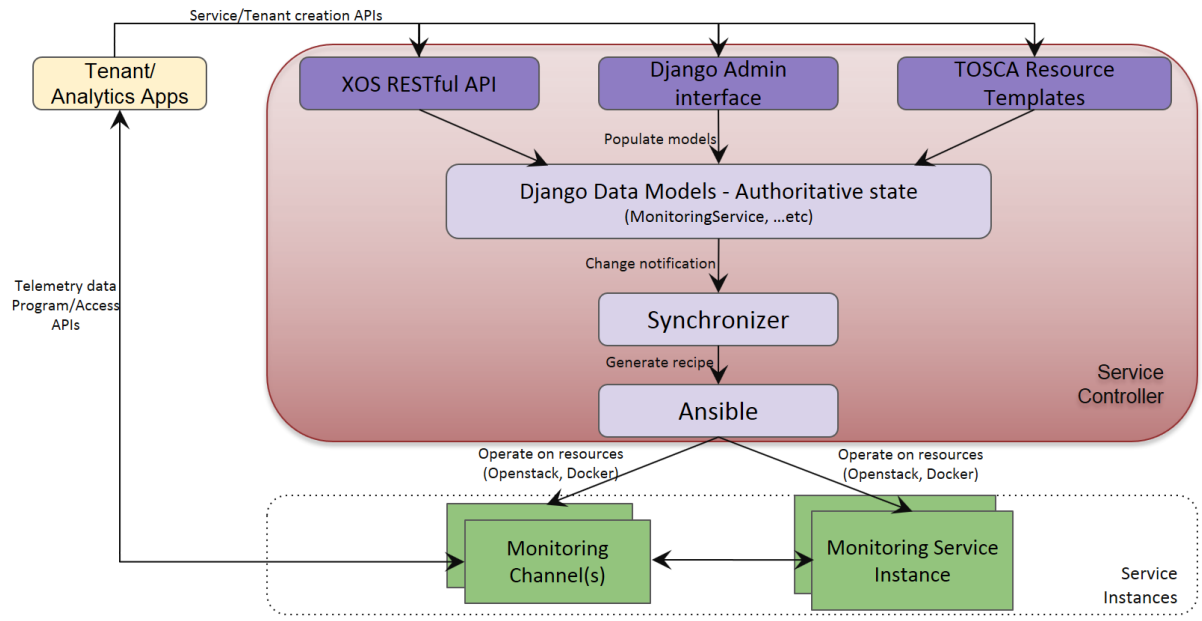

Рис. 3.7: Контроль сервісного контролера та екземплярів обслуговування

Коли служба моніторингу ініціюється через API XOS TOSCA або REST, "Контролер обслуговування" закручує екземпляри віртуальних машин служби моніторингу на вузлі обчислення.

Екземпляри віртуальної машини служби моніторингу запускаються з використанням попередньо вбудованого зображення , як визначено у "Власне зображення екземпляра служби моніторингу служби"

Перегляд на високому рівні різних програмних компонентів, що працюють у службі моніторингу екземпляра:

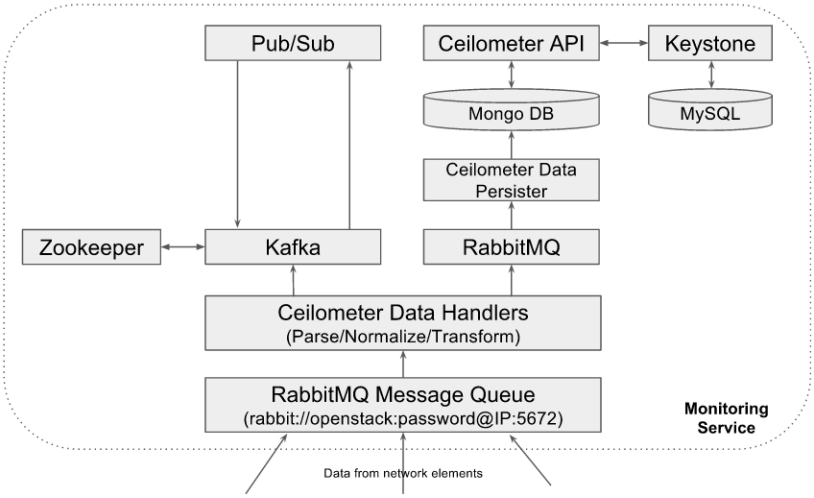

Рис. 3.8: Моніторинг компонентів службового примірника

- Ceilometer Data Handlers: Служба моніторингу використовує OpenStack Ceilometer Framework як основні компоненти збору та обробки даних. Він використовує демони сповіщення ceilometer-agent-сповіщення як оброблювачі даних, які отримують дані, опубліковані різними мережевими елементами, аналізують дані і нормалізують дані у загальних форматах вибірки целометрів. Якщо вони налаштовані, ці обробники даних перетворюють отримані дані метрик у різний формат метрик, таких як unit_conversion, rate_of_change, акумулятор ... тощо.
- RabbitMQ Message Queue: Служба моніторингу використовує черги повідомлень RabbitMQ для збору даних із елементів південної мережі, які потім будуть оброблятися келометровими обробниками даних. Черга повідомлень RabbitMQ також використовується для маршрутизації нормалізованих та перетворених зразків у бік даних про кілометр.
- Ceilometer Data Persister: Служба моніторингу використовує Openstack Дані цілометрів зберігають демон-целометр-збірник , який записує дані в налаштовану бази даних
- MongoDB: Бенкенд бази даних, який використовується в службі моніторингу, яка містить всі показники та дані про події
- Ceilometer API: інтерфейс API для доступу до даних із резервної бази даних

- Keystone: модуль OpenStack Keystone для аутентифікації доступу до API-метра келометра
- MySQL: Банк даних для OpenStack Keystone
- Черга повідомлень Kafka: служба моніторингу розмістить сервер обміну повідомленнями kafka для підтримки повідомлень про зібрані дані для зовнішніх програм. Копія всіх нормалізованих та перетворених зразків кілометрів буде опублікована в темі черги повідомлень kafka "xxxx", яка буде використовуватися модулем "Pub / Sub" служби моніторингу. Модуль "Pub / Sub" публікує дані для всіх підписаних програм, використовуючи ту саму чергу повідомлень kafka у визначених користувачем / додатком темах.
- Зоопарк: менеджер кластерів Kafka
- Модуль публікації / підписки: надає інтерфейс додаткам для підписки на підмножину телеметричних даних та отримання сповіщень, коли дані / подія доступні. Підписуючись, додатки можуть вказувати ціль, куди дані потрібно доставити . В даний час модуль "pub / sub" підтримує механізми доставки даних на основі kafka та UDP. Цей модуль використовує дані з теми "xxxx" черги повідомлень kafka та відкидає дані , якщо жодна програма не зацікавлена в цьому .

Архітектура є модульною таким чином, що компоненти в цьому стеку можна замінити деякими іншими високопродуктивними компонентами.

Деякі приклади, коли служба моніторингу інстанціюється за допомогою різних програмних даних:

Модель розгортання

Поточний реліз CORD підтримує розгортання всіх компонентів служби моніторингу в одній машині управління , як показано нижче:

Для підтримки виробничого середовища служба моніторингу може працювати в масштабній конфігурації (майбутня робота). Вид високого рівня моніторингу розгортання служби в масштабі:

- Перше і головне врахування масштабованого розгортання - це запуск бази даних на іншому хості, оскільки Ceilometer генерує багато ПИСЬМОСТІ , як описано в електронній таблиці Google
- Використовуйте шордінг і репліки-набори
- Розгорніть RabbitMQ в конфігурації кластера з декількома вузлами
- Розгортайте кратні агенти сповіщень на цілометрі з увімкненою функцією "workload_partitioning" і налаштовуйте "pipeline_processing_queues" зі значенням як мінімум рівним кількості агентів сповіщення

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.

Поточна мобільна інфраструктура:

- Побудований із закритими фірмовими вертикально інтегрованими системами
- Неефективне використання призводить до неоптимального використання радіоресурсів
- Неможливо підтримати налаштування для індивідуальних клієнтів
- Повільно підтримувати створення нових та інноваційних послуг
- Неможливо підтримати специфічні для галузі галузеві сценарії

Stratum

M-CORD: Включення мереж майбутнього

- Дезагрегований та віртуалізований RAN для високої гнучкості та масштабованості
- Розчленований ЕПК, який охоплює хмарну конструкцію масштабу для

незалежної масштабованості

Послуги Edge Multi-Access для індивідуальних служб та покращеного QoE

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]

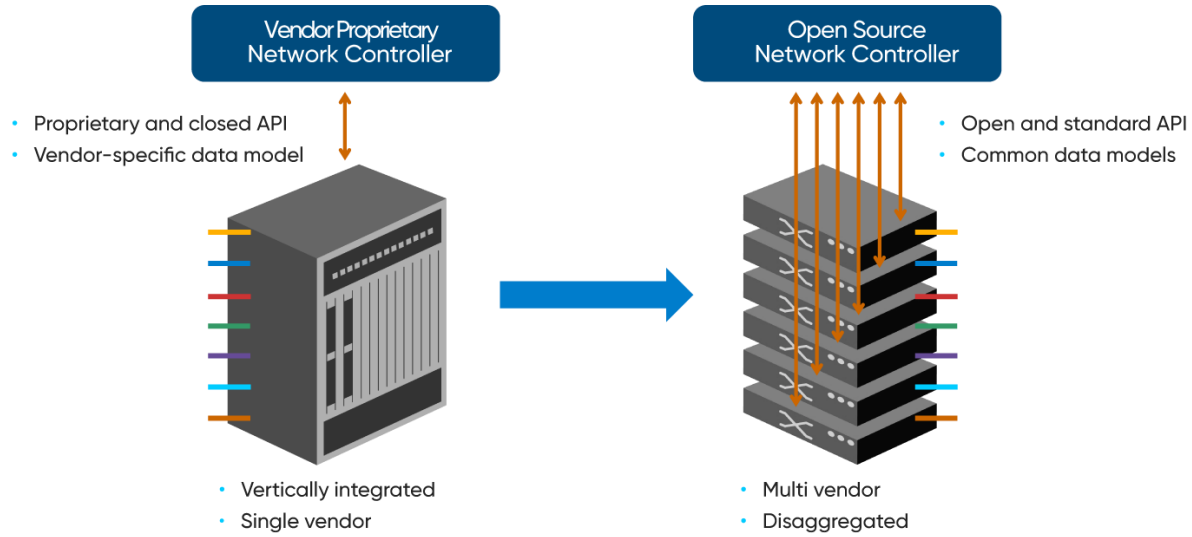

Рис. 3.9 ODTN Архітектура

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 посилання 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 ноутбуці 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 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.

Висновки :

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.

РОЗДІЛ 4
ОСНОВНІ ФУНКЦІОНАЛЬНІ ЕЛЕМЕНТИ

4.1 Призначення та сфера застосування

SEBA створена для надання шаблону архітектури для розробки рішень для широкосмугового доступу носіїв.

Спільність допомагає створити ефективність у розробці відкритих і білих коробок, а потім комерційних продуктів і підтримки для цих суб'єктів. Щоб рухатися до цієї мети, група операторів, що беруть участь у розробці цього РД, як очікується, буде використовувати отриманий продукт потоку реалізації в тій чи іншій формі або за межами просто лабораторних або польових випробувань.

Сфера застосування SEBA RD призначена для охоплення широкого набору бездротових і фіксованих технологій бездротового доступу і пов'язаних з ними можливостей Service Edge. До них відносяться, але не обмежуються: PON, XGS-PON, NG-PON2, EPON, майбутні технології PON, Gfast, Ethernet, фіксований бездротовий, DOCSIS і xDSL. Сфера застосування повинна дозволяти легко адаптувати нові технології, новий кремній, що підтримує ці технології та нові пристрої, які включають ці технології та кремній у елементи, що розгортаються . Це повинно бути можливо без переписування основних розділів підкомпонентів, які складають SEBA і не повинні вимагати нових фундаментальних взаємодій на північ від платформ автоматизації носіїв.

RD підтримує підхід POD, який означає, що всі необхідні компоненти для надання доступу до послуг покриті. На додаток до підтримки вузлів мережі простого доступу , RD також може включати інші елементи, такі як архітектура листового хребта. Перемикаюча тканина забезпечує функції агрегації та службового краю. RD підтримує вимоги операторів, які хочуть надавати широкосмугові послуги без можливостей Service Edge, а також операторів , що надають IP-послуги, включаючи можливості обслуговування в межах POD.

Пряма підтримка бездротового доступу до мобільності, як це визначено 3GPP, не в області цього РД, однак багато хто в групі операторів висловили бажання мати спільний інфраструктурний шар і загальні компоненти

підтримують обидва , тому пильне око віддається координації з майбутнім проектом (-ами) бездротового ONF.

4.2 Підходи до реалізації цілей

SEBA може бути реалізований різними способами, залежно від оператора та/або ситуації. Найбільш важливим в цьому плані є чітке відчуження і відокремлення «сервісу» від «інфраструктури ». Це також передбачає реалізацію SEBA на різних інфраструктурних платформах, включаючи, але не обмежуючись тими , які базуються на CORD. Модульність через хорошу функціональну декомпозицію повинна забезпечувати широке повторне використання в рішеннях ONF, що забезпечує високий обсяг , послідовні, SW збірки, можливі з екосистеми.

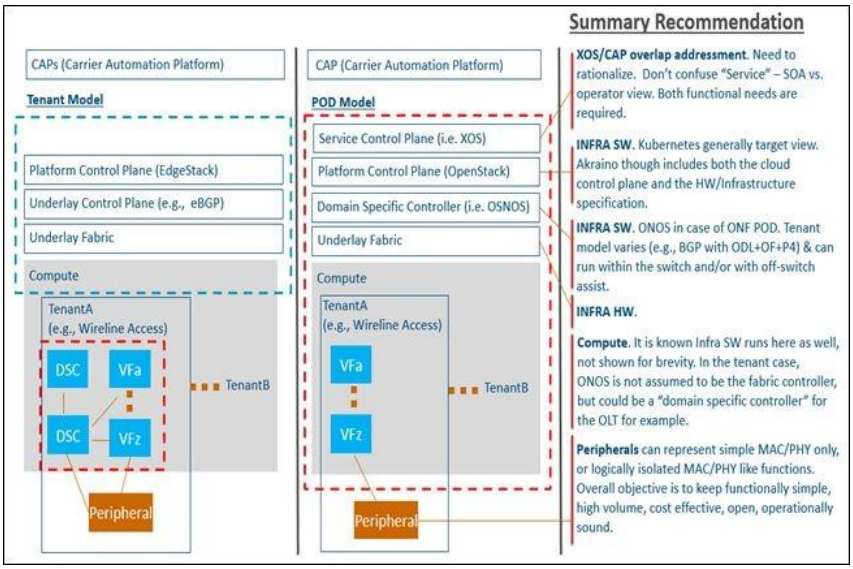

Рис. 4.1: Підходи до реалізації цілей

4 .2.1 Платформа автоматизації носіїв (CAP)

Платформа автоматизації перевізника, яка є зовнішньою і північною частиною SEBA, є надійною конструкційною рамкою, яка дозволяє специфікацію сервісу у всіх аспектах - моделювання ресурсів і відносин, які складають сервіс, вказуючи правила політики, які керують поведінкою сервісу, і програми, аналітика та події замкнутого циклу, необхідні для еластичного

управління сервісом. ONAP є прикладом CAP.

Система оркестрування та контролю (Service Orchestrator and Controllers) є рецептурою/політикою, спрямованою на забезпечення автоматизованої інсталяції сервісу при необхідності та управління вимогами до послуг еластичним чином.

Аналітичний фреймворк уважно стежить за поведінкою сервісу під час життєвого циклу сервісу на основі зазначеного дизайну, аналітики та політики, щоб уможливити відповідь відповідно до вимог системи управління, вирішувати ситуації , починаючи від тих, які вимагають зцілення, до тих, які вимагають масштабування ресурсів, щоб еластично пристосуватися до попиту варіації.

4.2.2 Рівень інфраструктури

Рівень інфраструктури включає обладнання в рішенні, включаючи пристрої, стелажі та полиці, обладнання та з'єднання для живлення, зовнішні волокна або електричні кабелі та інші пасивні пристрої.

Пристрої в POD включають пристрої ASG, обчислювальні сервери , OLT або інші пристрої, визначені технологією (наприклад , інші типи пристроїв для рішень Wireless або DOCSIS, які визначаються для SEBA).

Зовнішні волокна є частиною інфраструктури, оскільки вони підключаються до портів пристроїв і передають користувальницький трафік та /або керуючий трафік, а пристрої контролюють роботу сигналів і протоколів, що передаються оптоволоконним медіа .

Пасивні пристрої включають розгалужувачі PON, патч-панелі та інше обладнання, яке не відстежує сигнали , але виконує основні функції для підключення шляхів , комбінування сигналів, розділення сигналів або фільтрації сигналів .

Комбінація компонентів на рівні інфраструктури складає фізичний інвентар рішення, який постачальники планують доставку, постачальники рішень для виконання об'єднують у ланцюжок поставок , а монтажники розміщують і перевіряють.Комбінація компонентів в Інфраструктурному шарі

складається з фізичної інвентаризації рішення, яке постачальники планують для доставки, постачальників рішень виконання агрегувати в ланцюжку поставок, і що інсталятори розміщують і перевіряють. Рівень прикладних програм — це найближчий до користувача рівень взаємозв’язку відкритих систем (OSI). Цей рівень встановлює зв'язок між додатками та користувачем. Прикладом компонента SEBA на рівні прикладних програм є клієнт SEBA NBI.

4.2.3 Безпека

Як кероване віртуалізоване середовище, SEBA має максимально використовувати вимоги безпеки, архітектуру безпеки, найкращі методи безпеки та рішення безпеки з відкритим кодом від інших відкритих спільнот. Оператору потрібно буде адаптувати SEBA до своїх методів безпеки. SEBA може реалізувати деякі основні функції безпеки, які можуть допомогти реалізувати безпеку на периметрах, таких як API, ОС і Kubernetes.

Оператори повинні вказати реалізації безпеки для спільноти розробників SEBA, якщо ці реалізації приносять користь основним функціям безпеки, які краще підтримуються у відкритому коді SEBA. Наприклад , транзакції SEBA, які змінюють будь-які конфігурації або виконують будь-які дії оператора з будь-якого оператора або автоматизованого інтерфейсу (CLI, Northbound API, дія циклу керування), можуть використовувати ідентифікатор транзакції, який проходить через обробку SEBA дій конфігурації, щоб виконувати ефективні журнал аудиту безпеки та інше.

4.2.4 Надійність і стійкість

Надійність - це визначення ймовірності системи або компонента до функціонувати в заявлених умовах протягом певного періоду часу. Надійність також визначається в термінах доступності системи або компонента в термінах числа з «9», таких як «п'ять 9», що вказує на доступність 99,999 %.

Аналіз надійності рекомендується для компонентів POD, і весь POD передбачити доступність POD як системи .

Resiliency - це можливість відновлення сервера, мережевого компонента або POD від збою (наприклад, відключення живлення або збій обладнання) і

швидко резюме операцій .

Резервування або кластеризація використовується, щоб допомогти поліпшити як надійність POD, і стійкість операцій в POD.

4.2.5 Продуктивність системи

Вимірювання продуктивності системи у внутрішніх функціях обробки POD є важливими для вимірювання та розуміння щодо реакції системи на транзакції контролю та управління , а також для масштабованості системи, щоб забезпечити певну кількість операцій певного типу, поки система є працює під певним профілем навантаження (профіль різноманітних визначених операцій на визначених частотах протягом певного періоду часу).

Реалізація інструментів моніторингу для моніторингу продуктивності системи повинна бути розглянута та обрана, щоб мінімізувати їх власний вплив на продуктивність системи та ресурси.

4.2.6 Управління потенціалом

Управління потужністю забезпечує аналітику та звітність про ресурси, необхідні для рішення. Він отримує вимірювання потужності з елемента керування продуктивністю та з ресурсів, які визначають потужність елемента рішення.

Реалізація управління потужністю може відбуватися в платформі автоматизації оператора (CAP), яка отримує вимірювання продуктивності від POD(ів), і тому колекція Performance Management вимагає пересилання в CAP для цієї функції.

Провайдер може визначити, що функції керування потужністю реалізовані в локальному SEBA POD і надаються через інтерфейс локального оператора.

4 .2.7 Управління несправностями

Управління помилками застосовується на рівні SEBA POD. Платформа автоматизації оператора (CAP), яка підключається до SEBA POD через клієнт SEBA NBI, як правило, забезпечує дві функції на рівні підприємства для керування несправностями : (1) Показувати поточні активні тривоги та (2)

Показувати історію сигналів і подій.

Для цілей цього обговорення тривожним сигналом може бути також «постійний стан», який має стан встановленого/вибраного , але зіставлений на серйозність «не тривожний».

Бажано , щоб NEM забезпечував нормалізований колектор, реалізований у Kafka, для всіх несправностей для зовнішніх OSS/Оркестраторів, що рухаються на північ, щоб мати можливість отримувати потоки історії несправностей детермінованим способом . Як потокова платформа, Kafka надає можливості публікувати та підписуватися на потоки записів, зберігати потоки записів у надійний відмовостійкий спосіб та обробляти потоки записів у міру їх появи.

Хоча VOLTHA публікує помилки своєї шини Kafka, які NEM може надати своїм клієнтам, що рухаються на північ, інші несправності можуть бути отримані від інших API в SEBA POD, таких як Redfish для керування пристроями. NEM може розробити «трансформатор», якщо це необхідно, щоб експортувати дані з інших API в тему Kafka, і таким чином забезпечити всі помилки через нормалізований колектор у Kafka.

Північні OSS/Оркестратори можуть реалізувати та встановити агента як мікросервісу в клієнті SEBA NBI, який підписується на нормалізований колектор Kafka і перетворює помилки в потрібний формат (наприклад, VES для ONAP, IPFIX для іншого OSS тощо) для транспортування до Північного САП. Клієнт SEBA NBI для САР не є частиною SEBA, оскільки різні реалізації САР можуть використовувати різні протоколи керування.

Оскільки клієнт SEBA NBI може виявити або повторно підключитися до SEBA POD в будь-який час, він може залежати від кешованих сигналів і подій у SEBA POD і повторно синхронізуватися з цією додатковою історією, щоб правильно оновити його «Показати поточний «Активні сигнали», якщо і тільки тоді, коли клієнт SEBA NBI може повторно синхронізуватися з фактичної останньої тривоги або події з попереднього підключення до SEBA POD.

Реалізація шини Kafka в SEBA повинна дозволити клієнту SEBA NBI

визначити, чи може він синхронізуватися з останнім відомим станом керування несправністю SEBA POD. Якщо клієнт SEBA NBI не може синхронізуватися з останнім відомим станом , у нього будуть відсутні сигнали тривоги та події. У цьому випадку SEBA POD має забезпечувати функцію «Отримати всі активні сигнали з SEBA POD». SEBA надає абстрактні інтерфейси конфігурації для надання послуг на рівні абонента. SEBA можна використовувати в багатьох середовищах розгортання. Кожна реалізація повинна забезпечувати прості, прості у використанні API, що забезпечують мінімально необхідні параметри.

4 .2.8 Управління життєвим циклом програмного забезпечення

SEBA надає API, здатні керувати програмним забезпеченням для компонентів POD. Для типу PON AN це включає в себе фізичне обладнання резидента в POD, а також всі пристрої ONT, підключені до портів PON.

Компоненти POD управляються самостійно. Де новий послуги вимагають оновлення програмного забезпечення до різних компонентів POD тих служби не будуть налаштовуватися до тих пір, поки всі частини POD не будуть модернізовано.

Під час оновлення будуть надані API для відстеження прогресу. API повиннівміти визначати успіх діяльності з модернізації та визначати будь-які fallout діяльності , які є обов'язковими.

Повернення до попередніх компонентів програмного забезпечення повинно бути доступним, якщо критерії відмови зустрічаються.

Заходи, що впливають на обслуговування клієнтів, виконуються в технічному обслуговуванні періоди, зазвичай від 2 до 4 годин. Очікувана перерва в обслуговуванні абонентів менше 5 хвилин.

4 .2.9 Управління продуктивністю

VOLTHA публікує дані моніторингу продуктивності на своїй шині Kafka, яку NEM може перетворювати в масові колекції даних (наприклад, організовані за інтервалом збору).

NEM також може опитувати інші дані моніторингу продуктивності для інших функцій у SEBA POD, наприклад, використовуючи Redfish API для

моніторингу продуктивності пристрою. NEM може розробити «трансформатор» за потреби з іншого API на тему Kafka, і таким чином забезпечити весь моніторинг продуктивності через нормалізований колектор у Kafka.

Північні OSS/Оркестратори можуть реалізувати та встановити агент як мікросервіс у SEBA POD, який підписується на нормалізований колектор Kafka і перетворює колекції моніторингу продуктивності в потрібний формат (наприклад, VES для ONAP, IPFIX для іншого OSS тощо) для транспортування до північного OSS/ оркестратора.

SEBA NB API має надавати API для отримання всіх даних і показників PM для "поточних" інтервалів (наприклад, 15 хвилин і щодня).

Якщо можливо, щоб платформа автоматизації оператора (CAP) була несинхронізована з колекцією історичних інтервалів PM SEBA POD після виявлення або повторного підключення до SEBA POD, тоді потрібен механізм для повторної синхронізації CAP з шини Kafka підлягають визначенню та реалізації.

4.2 .10 Інвентаризація

Інвентаризація - це визначення компонентів і інтерфейсів системи. У а вид життєвого циклу , запланована інвентаризація включає в себе очікувані компоненти і інтерфейси, які повинні працювати в інфраструктурному шарі, щоб доставити POD функції. Фактичний або виявлений інвентар вимагає відкриття і перевірка на планову інвентаризацію для надання необхідних інфраструктура для сервісів і операцій POD.

4.2.11 Телеметрія, моніторинг, аналітика та функції політики

Телеметрія передбачає автоматичний запис і передачу даних з віддалені або недоступні джерела в систему управління для моніторингу та аналіз. Для цього рішення рекомендується направляти зібрану телеметрію дані до підсистеми управління моніторингом продуктивності для загальних обробка.

Зверніть увагу, що SEBA надає необов'язкову систему моніторингу та реєстрації. Дивіться інфраструктуру моніторингу та журналювання SEBA у документах дизайну SEBA.

Цей необов'язковий фреймворк включається через Helm заяви для моніторингу іфункції входу в стартап SEBA.

Зверніть увагу , що проект SEBA в даний час не включає або не планує включати Аналітичний рушій в рамках SEBA, такий як описано в запропонованій аналітиці для проекту CORD (A-CORD). При відсутності рушія Analytics з A-CORD проект, оператор може розробляти власні додатки Analytics в POD, який може взаємодіяти з SEBA Monitoring and Logging інфраструктура.

Зверніть увагу, що замість побудови додатків аналітики в POD оператор може будувати централізовані функції аналітики та політики у своїй північнонімецькій КЕП. CAP отримає несправності , телеметрію та моніторинг продуктивності від NEM адаптери (див. також розділи вище для управління несправностями та Моніторинг продуктивності), які підписуються на автобус Кафка і трансформатори в SEBA Monitoring & Logging Infrastructure, для того, щоб забезпечити централізовану обробку для управління несправностями, продуктивності Управління, телеметрія та будь-які функції аналітики та політики.

4 .2.12 Автоматизація та управління (включає в себе Зовнішні API)

У типовому сценарії розгортання буде до тисяч SEBA встановлено POD.

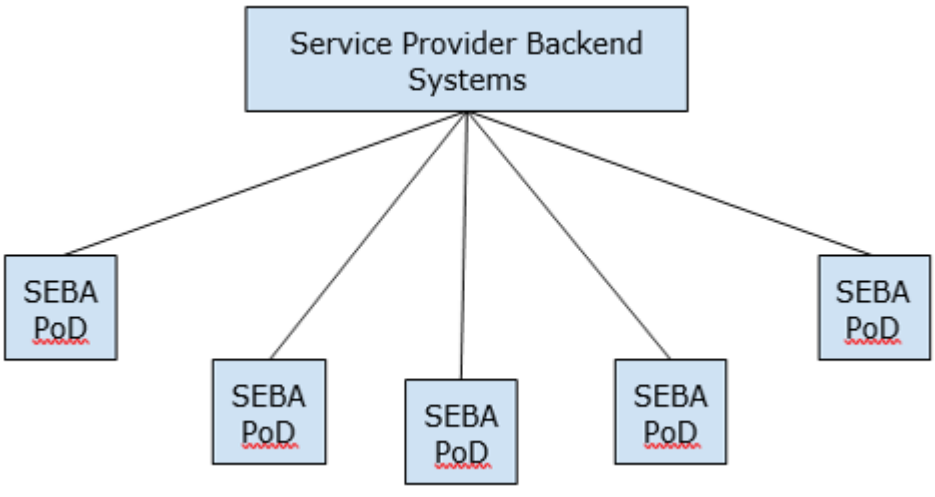

Рис.4.2 Серверні системи постачальника послуг для багатьох POD SEBA

SEBA має бути автономним і мати можливість працювати в будь-якому середовищі постачальника послуг. Щоб досягти цих цілей, нам потрібно мати зовнішню панель адаптації, яка розташована поверх SEBA Northbound API і може бути розміщена віддалено. Оператор може створити екземпляр клієнта SEBA NBI в SEBA POD.

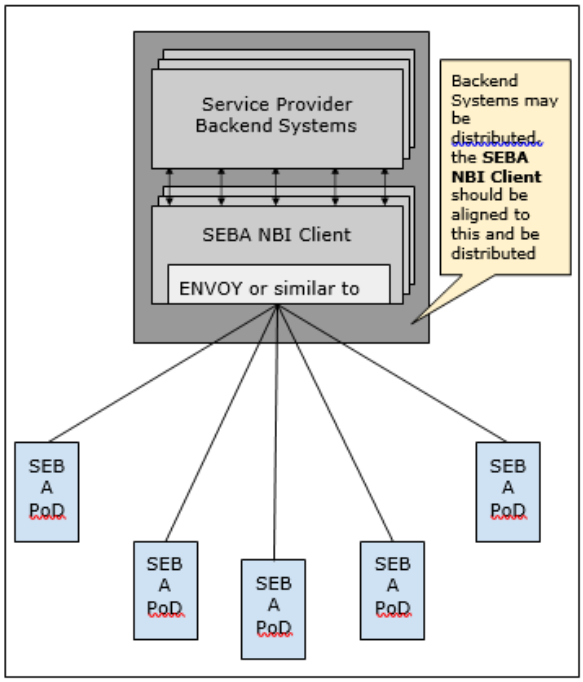

Рис. 4.3 Роль клієнта SEBA NBI

Клієнт SEBA NBI тісно пов’язаний із серверною частиною постачальника послуг і буде залежати від постачальника послуг.

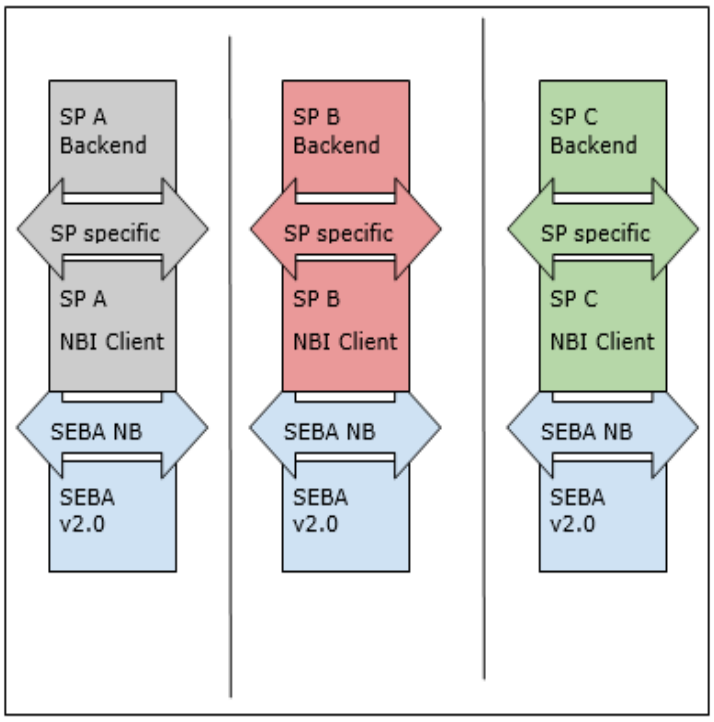

Рис.4.4: SP Backend, SEBA NBI Client (per SP) та SEBA NB API

SEBA повинен бути автономним і не впливати на будь -які зміни, які може виникати в середовищі розгортання.

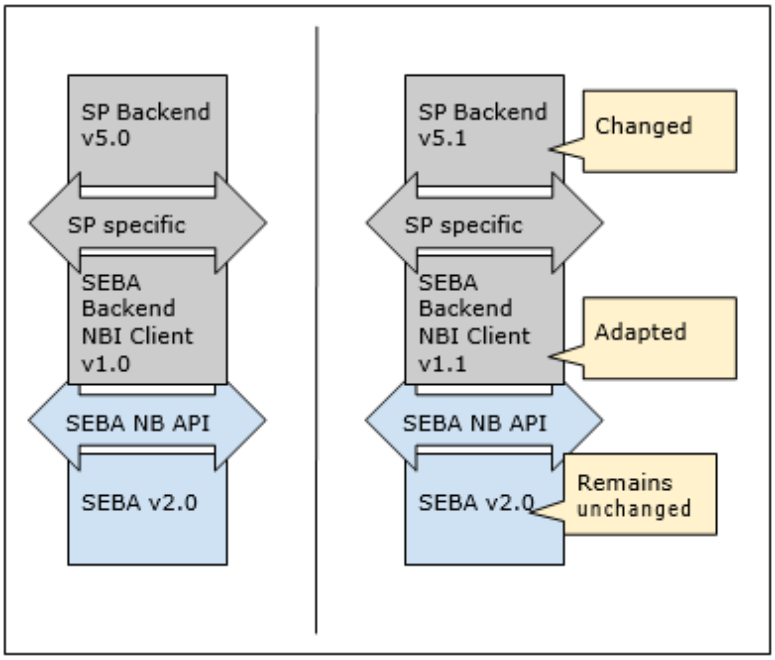

Рис.4.5: Приклад - Немає змін до SEBA NB API

Аналогічно, оновлення версії SEBA не впливають на внутрішні системи.

Як показано на малюнку нижче, версія SEBA NB API може змінитися. Релізи SEBA мають визначати версію NB API з документованим каталогом API та примітками до випуску.

API, каталог і версія SEBA NB повинні об'єднувати API всіх служб NBI. Наприклад, SEBA NBI має включати VOLTHA NB API, SADIS (Subscriber / Access Device Information Service) NB APIs та інші послуги NBI в SEBA.API API SEBA NB має залишатися зворотно сумісним. Виклик API може бути зворотно сумісним, лише додавши параметри або значення параметрів. Щоб запропонувати замінити API, має бути визначено новий API, який замінить поточний API. Лише після підтвердження відсутності користувачів API, які потрібно замінити, можна видалити колишній API. Примітки до випуску нової версії API повинні документувати всі покращення API, нові API, API, які пропонуються замінити, і видалені API.

Інтерфейси керування SEBA можуть скористатися стандартизованим моделюванням пристроїв і технологій доступу з використанням моделей YANG від організацій з розробки стандартів (SDO) – див. каталог github для SDO. У цьому каталозі github широкосмуговий форум (BBF)підтримує опубліковані моделі YANG для ITU-T PON (на BBF TR-385), FTTdp для G.fast і G.hn (TR-355) і Common YANG (BBF TR-383). При розробці SEBA NB API слід враховувати відповідність цим стандартизованим моделям YANG для інтеграції до Carrier Automation Platform (CAP).

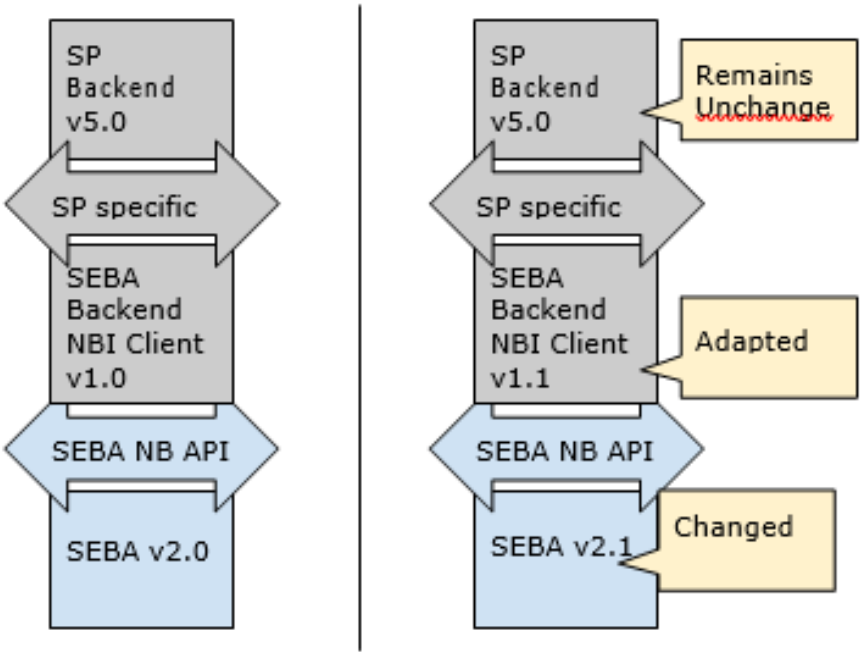

Рис 4.6: Приклад - Немає змін в SP Backend

Адаптер повинен викликати відповідні методи в SEBA NB API і реалізує набір API зворотного виклику для виклику SEBA. Віддалений виклик здійснюється через gRPC.

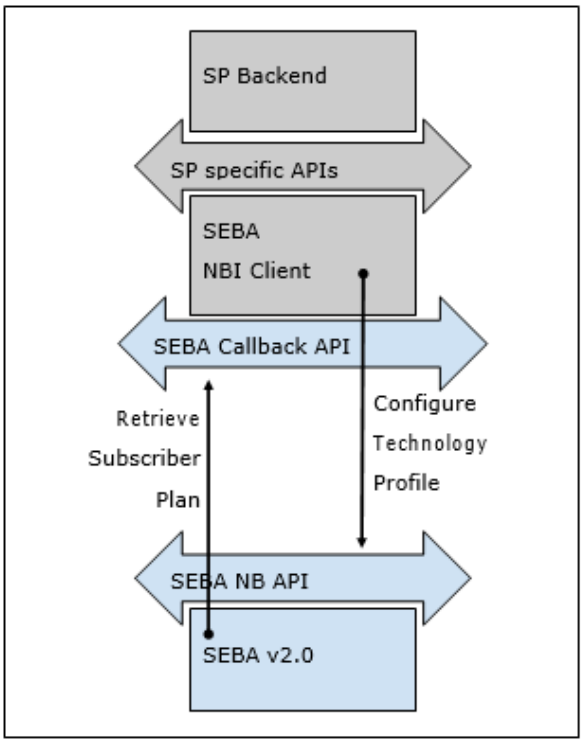

Рис 4.7: API зворотного виклику SEBA

Керування пристроями

Інтерфейс керування пристроями (DM) повинен підтримувати моделі даних та API, визначені спільнотою SEBA/VOLTHA. DM має включати необхідні механізми для виконання дій, перерахованих нижче .

- Управління запасами: має бути можливим автоматичне виявлення мережевих елементів, а також збір даних про конфігурацію та стан з них

- Моніторинг стану : DM повинен підтримувати моніторинг стану пристроїв, щоб завчасно виявляти та усувати проблеми (наприклад, температура пристрою, збій вентилятора)

- Управління журналами: має бути можливим керувати журналами та подіями, які походять із пристроїв, щоб допомогти налагодити системні проблеми .

- Обслуговування ПЗ пристрою: DM має надавати підтримку сценаріїв оновлення/пониження для всіх компонентів ПЗ , включаючи мікропрограму, ONIE, NOS тощо.

- Перезавантаження пристрою : DM має дозволити перезавантаження будь-якого пристрою для відновлення після помилок, які неможливо усунути іншими способами.

Висновки:

У даному розділі було розглянуто функціональні будівельні блоки , рівень управління, автоматизацію так управління, тобто SEBA повинна бути автономною і не впливати на будь-які зміни , які може виникати в середовищі розгортання. Оскільки драйвер AN і драйвер ASG забезпечують основні абстракції ресурсів і обробляють ключові операції, керування пристроями містить лише мінімальний набір (загальних) операцій , необхідних для керування пристроями SEBA, які ще не охоплені драйверами AN/ASG.

РОЗДІЛ 5
РОЗРОБЛЕННЯ СТАРТАП ПРОЄКТУ

SEBA призначена для підтримки мережевих та функціональних потреб декількох операторів із загальною архітектурою. SEBA RD забезпечує високий рівень шаблону або архітектури для підтримки широкосмугового доступу з мінімальним рецептом вибору технології.

5.1 Опис ідеї проєкту

Ідея проєкту полягає у визначенні спільного компонента інфраструктури, який буде вважатися недиференційованим як для операторів, так і для постачальників . Одночасність допомагає більш ефективно розробляти відкриті коди та білі скриньки, а потім комерційні продукти та підтримку цих організацій. Очікується, що для досягнення цієї мети група операторів, які беруть участь у розробці цього RD, використовуватиме отриманий в результаті робочий продукт потоку реалізації в тій чи іншій формі, крім лабораторних або польових випробувань.

Таблиця 5.1

Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система підвищення ефективності архітектури розробки рішень для операторського широкосмугового доступу.	1.Підвищення ефективності роботи платформ. 2.Зміни комерційних продуктів . 3.Визначення групи операторів, які будуть використовувати робочий продукт.	Оператори зможуть замінити неефективні або застарілі компоненти більш сучасними та легкими компонентами , щоб використовувати декілька потоків реалізації, які будуть відповідати вимогам SEBA.

Сфера застосування повинна дозволяти легко адаптувати нові технології, новий фундамент, що підтримує ці технології, і нові пристрої, які включають ці технології та фундамент у елементи , що розгортаються. Це повинно бути можливим без переписування основних розділів підкомпонентів, які складають SEBA, і не повинно вимагати нових фундаментальних взаємодій , пов'язаних з платформами автоматизації операторів. RD підтримує підхід POD, що означає, що покриваються всі необхідні компоненти для надання доступу до послуг .

5.2 Технологічний аудит ідеї проєкту

Технологічна здійсненність ідеї проєкту Таблиця 5.2

№ п/п	Ідея проєкту	Технології реалізації	Наявність технологій	Доступність технологій
1	Система розробки рішень для операторського широкосмугового доступу.	1. Робоче середовище Kubernetes	ONOS, VOLTHA, XOS і R-CORD.	Є доступними
2		2. Інструменти CI/CD для розробки, а також екземплярів розгортання.	наприклад , Jenkins і т.д.	Є доступними
Обрана технологія реалізації проєкту: Робоче середовище Kubernetes				

Для реалізації проєкту доступна SEBA, яка пов’язана з кількома зрілими проєктами в ONF, включаючи ONOS, VOLTHA, XOS і R-CORD. Оскільки ця робота взаємодіє з існуючими випущеними роботами в активних спільнотах ONF, цілком імовірно, що деякі процеси, визначені для нормальної нової роботи RD, можливо, потрібно буде відкоригувати, щоб забезпечити, щоб існуючі спільноти та їхня робота не були позбавлені прав.

5.3 Аналіз ринкових можливостей запуску стартап-проєкту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проєкту , та ринкових загроз, які можуть перешкодити реалізації проєкту, дає змогу спланувати напрями розвитку проєкту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проєктів-конкурентів.

При дослідженні ринкових можливостей, в першу чергу проведений аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку . Дані наведені у таблиці нижче.

Таблиця 5 .3

Попередня характеристика потенційного ринку стартап-проєкту

No п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж	100000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу	Немає
5	Специфічні вимоги для стандартизації, специфікації	Немає
6	Середня норма рентабельності в галузі, %	100%

Враховуючи сьогоднішню необхідність ринку рішень стосовно підвищення лояльності клієнта, за попереднім оцінюванням ринок є привабливим для входження.

Надалі визначають потенційні групи клієнтів, їх характеристики, та формують орієнтовний перелік вимог до товару для кожної групи (табл. 5.4).

Після визначення потенційних груп клієнтів проведений аналіз ринкового середовища : складені таблиці факторів, що сприяють ринковому

впровадженню проєкту, та факторів що йому перешкоджають.

Таблиця 5.4

Характеристика потенційних клієнтів стартап-проєкту

N_o п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Забезпечення зручного інструменту SEBA	Оператори	Оператори в свою чергу можуть визначати самостійно формат взаємодії та внести свої коригування.	Використання новітніх методів обробки даних для швидкості та правильності роботи системи.

Таблиця 5 .5

Фактори загроз

No п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Відсутність зацікавленості у продукті	Успіх системи залежить від підтримки малого бізнесу, адже з великою імовірністю всеукраїнські оператори зв'язку не будуть звертати увагу на нового гравця на ринку аналітики даних та рішення проблеми відтоку, а виберуть вже перевірені часом рішення.	Робота з малим бізнесом над підвищенням лояльності клієнта, що буде означати створення бази клієнтів, які будуть рекомендувати цей продукт.
2	Складність даних	Великою проблемою при роботи в великими масивами даних є їх складність, а саме: пропуски, надлишковість, різна структура даних. Це все призводить до того що ще до того як почати аналізувати та робити висновки, потрібно провести великий обсяг роботи для структуризації та отримання якогось поняття мають ці данні корисну інформацію для нас чи ні.	Використання нових методів математичного аналізу даних та комбінація різних алгоритмів для отримання найкращого результату.

Надалі проведений аналіз пропозиції: визначені загальні характеристики конкуренції на ринку (табл. 5.6)

Таблиця 5.6

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1 . Тип конкуренції: олігополія	На ринку представлені декілька компаній, що поставляють подібні послуги рішення проблеми відтоку	Підвищувати якість товару за рахунок використання передових технологій
2. Рівень конкурентної боротьби: національний /інтернаціональний	Першим етапом є боротьба за ринок України з подальшим виходом на ринки інших країн	Маркетингова компанія в першу чергу орієнтована на захоплення місцевого ринку
3. Галузева ознака : внутрішньогалузева	Економічна боротьба з конкурентами відбувається в одній галузі економіки, пропонуються аналогічні послуги, що мають архітектурні відмінності у функціонуванні	Слідкувати за новітніми технологіями, та розробками конкурентів

Таблиця 5 .7

Аналіз конкуренції в галузі за М . Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	ONF	Гнучкі ціни, Патент на продукт	Змінні витрати постачальників	Рівень чутливості до зміни цін	Ціна , лояльність споживачів
Висновки	Конкуренція не інтенсивна, кожен працює в окремому регіоні	Можливість входу в ринок висока. Потенційні конкуренти присутні	Постачальник може диктувати умови: ціни на послуги	Кожен з клієнтів потребує індивідуального підходу для вирішення його задач	Обмежень для роботи на ринку з боку товарів замінників на даний момент не існує

В результаті проведення аналізу таблиці 5 .7, можна зробити висновок, що можливість виходу на ринок з огляду на конкурентну ситуацію є високою. Для виходу на ринок товар в першу чергу повинен пропонувати унікальні характеристики , які відсутні у продуктах конкурентів.

Таблиця 5 .8

Обґрунтування факторів конкурентоспроможності

№ п/п	Фактори конкурентоспроможності	Обґрунтування
1	Динаміка галузі	Проблема відтоку користувачів наразі є дуже важливою, тому оператори зацікавлені у ній
2	Концепція товару і послуги	Система підвищення лояльності клієнтів дозволяє утримувати клієнтів впливаючи на них маркетинговими методами.
3	Після-продажне обслуговування	Підтримка щодо використання системи після її продажу

Таблиця 5.9

Порівняльний аналіз сильних та слабких сторін системи підвищення лояльності.

№ п/п	Фактори конкурентоспроможності	Бали 1 -20	Рейтинг товарів-конкурентів у порівнянні з власною системою						
			-3	-2	-1	0	+1	+2	+3
1	Динаміка галузі	20							✓
2	Концепція товару і послуги	20							✓
3	Після продажне обслуговування	20							✓

Таблиця 5.10

Альтернативи ринкового впровадження стартап-проєкту

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Оператори	Готові	Високий	Низька	Середня
2	Розробники систем	Готові	Дуже високий	Низька	Середня

Базові стратегії в обраних сегментах ринку представлені у таблиці 5.11

Таблиця 5.11

Визначення базової стратегії розвитку

№ п / п	Обрана альтернатива розвитку	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції	Базова стратегія розвитку
1	Динамічний розвиток з використанням маркетингу та встановлення бізнес-контактів	Підняття рейтингу компанії шляхом маркетингу , встановлення конкурентоспроможних цін	Незалежність від посередника, який утримує кошти за свої послуги	Стратегія лідерства по витратах

Продовження Таблиці 5.11

№ п/п	Обрана альтернатива розвитку	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції	Базова стратегія розвитку
2	Динамічний розвиток завдяки висвітленню унікальних характеристик наданих послуг	Унікальність послуг, для збільшення лояльності клієнта	Використання технології машинного навчання	Стратегія диференціації

В результаті аналізу обираємо стратегію диференціації. Наступним кроком є вибір стратегії конкурентної поведінки (таблиця 5 .12)

Таблиця 5.12

Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проєкт першопроходцем на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів	Чи буде компанія копіювати основні характеристики	Стратегія конкурентної поведінки
1	Ні	ТАК	ТАК	Стратегія «лідер»

Визначимо стратегію позиціонування (таблиця 5.13)

Таблиця 5.13

Визначення стратегії позиціонування

№ п /п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап -проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту
1	Висока доступність	Стратегія диференціації	Використання технології машинного навчання, що є новітнім підходом до рішення цієї задачі	Доступність , якість, швидкість

5.4 Розроблення маркетингової програми стартап-проєкту

Маркетингова програма - це намічений для планомірного здійснення, об'єднаний єдиною метою та залежний від певних строків комплекс завдань і адресних заходів соціального, економічного, науково-технічного, виробничого, організаційного характеру з визначенням ресурсів, що використовуються, а також джерел одержання цих ресурсів. Основну увагу слід приділяти вибору, значенню та формі інструментів маркетингу, їх об'єднанню в найбільш оптимальний з погляду визначеної мети комплекс, а також розподілу фінансових ресурсів у межах бюджетування маркетингу.

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару (таблиця 5.14).

Таблиця 5.14

Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентом
1	Можливість проведення аналізу даних.	Можливість, що забезпечується новітніми методами машинного навчання	Комплексний підхід до рішення задачі

Надалі розробляється трирівнева маркетингова модель товару: уточняється ідея продукту та послуги, його фізичні складові та особливості процесу (таблиця 5 .15).

Таблиця 5.15

Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
1. Товар за задумом	Товар забезпечує моніторинг відтоку операторів від застарілих платформ.
2. Товар у реальному виконанні	Властивості: доступність, цілісність, зручність , прозорість
	Товар представляє собою програмний комплекс виконаний за допомогою середовища Kubernetes та суміжних бібліотек .

Продовження Таблиці 5.15

2. Товар у реальному виконанні	Поставляється у вигляді застосунку для всіх популярних платформ.
3. Товар із підкріпленням	До продажу: відбувається інсталяція та конфігурування системи , проводяться презентація платформ для клієнта
	Після продажу : відбувається підтримка програмного забезпечення та його доопрацювання під потреби клієнта
Програмний комплекс, що забезпечує вирішення проблеми відтоку та підвищення лояльності клієнтів , розповсюджується на основі підписки, захисту підлягає програмний код та програмна реалізація.	

Аналіз системи збуту передбачає визначення ефективності кожного елемента цієї системи , оцінювання діяльності апарату працівників збуту. Аналіз витрат обігу передбачає зіставлення фактичних збутових витрат за кожним каналом збуту і видом витрат із запланованими показниками для того, щоб виявити необґрунтовані витрати, ліквідувати затрати, що виникають у процесі руху товарів і підвищити рентабельність наявної системи збуту.

Таблиця 5.16

Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення
1	Консервативна поведінка, але відкриті до нового	Соцмережі професійного спрямування, корпоративна пошта	Можливості отримання переліку параметрів та абонентів	Висвітлити унікальні характеристики продукту

У якості концепції маркетингових комунікацій були обрані інтегровані маркетингові комунікації, де компанія ретельно обмірковує і координує роботу своїх численних каналів комунікації, рекламу в засобах масової інформації, особистий продаж, стимулювання збуту, пропаганду, прямий маркетинг, упаковку товару.

Висновоки:

В даному розділі був проведений маркетинговий аналіз перспектив ефективного використання програмного середовища SEBA, та проведене оцінювання можливостей його ринкового впровадження.

В результаті дослідження визначено , що існує можливість ринкової комерціалізації проєкту в першу чергу завдяки використанню технологій машинного навчання , що дозволяє наділити продукт унікальними характеристиками, такими як швидкість, прозорість, якість отриманого рішення та великі можливості кастомізації проєкту.

Конкурентна ситуація надає перспективи впровадження продукту, так як продукція товарів -аналогів має лише частковий функціонал реалізованої

системи та володіє низкою критичних недоліків, через які рівень довіри до них залишається незадовільним. В результаті існуючі товари-аналоги не створюють прямої конкуренції на ринку України.

Проведений аналіз підтверджує , що подальша імплементація проєкту є доцільною.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

В магістерскій дисертації 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 ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Застосування SDN - рішень для оптимізації транспортних мереж мобільних операторів[Електронний ресурс] // Алексей Шалагинов. – – Режим доступу до ресурсу: https://scinse.donntu.edu.ua/tks/volynskyi/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/l_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[Електронний ресурс] // - Режим доступу до ресурсу:https://sibsutis.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 Net works [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 [Електронний ресурс] // Алексей Шалагинов.

– – Режим доступу 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/