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

Інститут телекомунікаційних систем

Кафедра Телекомунікацій

До захисту допущено:

Завідувач кафедри

_____ Сергій КРАВЧУК

«___»_____2021 р.

Дипломна робота

на здобуття ступеня <u>бакалавра</u>

<u>Спеціальності 172 «Телекомунікації та радіотехніка»</u>

на тему: «Побудова інформаційної моделі ресурсів мережі SDN з використанням інтерфейсу T-API»

Виконала: студентка <u>4</u> курсу, групи <u>ТЗ-71</u>

<u>Корнієнко Надія Ігорівна</u>
(прізвище, ім'я, по батькові) (підпис)

Керівник <u>д.т.н., професор Ільченко М.Ю.</u>
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант <u>1,2,3</u> <u>д.т.н., професор Романов О.І.</u>
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент <u>к.т.н., доцент Созонник Г.Д.</u>
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

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

Студентка _____
(підпис)

Київ – 2021 рік

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

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 172 «Телекомунікації та радіотехніка»

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

ЗАВДАННЯ
на дипломну роботу студенту

Корнієнко Надії Ігорівні
(прізвище, ім'я, по батькові)

1. Тема роботи: Побудова інформаційної моделі ресурсів мережі SDN з використанням інтерфейсу T-API.

керівник роботи Ільченко Михайло Юхимович, д.т.н., професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 14 квітня 2021 р. № 1007-с

2. Термін подання студентом роботи 7 червня 2021р.

3. Вихідні дані до роботи: SDN-мережа та інтерфейси; SDN-контролер з відкритою мережевою операційною системою ONOS; оптична транспортна мережа; модель та елементи транспортної мережі.

4.Зміст роботи:

1. SDN та її архітектура, стандартні протоколи та інтерфейси.
2. SDN-контролер на базі відкритої мережевої операційної системи ONOS.
3. Оптична транспортна мережа та принцип побудови моделі з використанням інтерфейсу T-API.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): презентація на тему: «Побудова інформаційної моделі ресурсів мережі SDN з використанням інтерфейсу T-API».

Слайд №1,2 – Тема, мета, актуальність роботи;

Слайд №3,4 – Загальні відомості про SDN;

Слайд №5 – T-API;

Слайд №6,7 – Топологічна та інформаційна моделі T-API;

Слайд №8 – Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Романов О.І. професор, д.т.н.	25.09.2020	16.11.2020
2	Романов О.І. професор, д.т.н.	16.11.2020	28.02.2021
3	Романов О.І. професор, д.т.н.	28.02.2021	01.05.2021

7. Дата видачі завдання 25 вересня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Пошук необхідної літератури, статей, специфікацій згідно теми дипломної роботи.	25.09.2020	
2.	Аналіз подальшого виконання завдання, підбір засобів для рішення поставленої задачі.	30.10.2020	
3.	Дослідження інформації щодо SDN-мереж, стандартних протоколів та інтерфейсів.	16.11.2021	
4.	Опрацювання відомостей стосовно SDN-контролера та відкритої мережевої операційної системи ONOS.	28.02.2021	
5.	Обґрунтування принципу побудови інформаційної моделі з використанням інтерфейсу T-API.	01.05.2021	
6.	Аналіз досліджених результатів.	10.05.2021	
7.	Оформлення пояснювальної записки.	17.05.2021	

Студентка Надія КОРНІЄНКО

Керівник Михайло ІЛЬЧЕНКО

РЕФЕРАТ

Робота виконана на 58 сторінках, містить 21 ілюстрацію та 1 таблицю. При написанні використовувалася література з 9 різних джерел.

Метою роботи є побудова інформаційної моделі ресурсів мережі SDN з використанням інтерфейсу T-API. Об’єктом дослідження є надання вірних рішень щодо принципу побудови моделі. Предметом дослідження є розробка та обґрунтування принципу побудови моделі.

У даній роботі розглянуто загальні відомості та архітектуру мережі SDN, північні та південні інтерфейси, SDN-контролер на базі відкритої операційної системи ONOS, проєкти ONF OTCC та ONF T-API, побудовано топологічну та інформаційну моделі. Також було проаналізовано переваги операційної системи рівня управління, програмно-конфігурованих та оптичних транспортних мереж. У результаті виконання роботи було проведено аналіз топологічної та інформаційної моделей T-API.

Ключові слова: SDN, інтерфейс, топологія, ONOS, T-API, YANG, модель, архітектура, контролер, операційна система, специфікація.

ABSTRACT

The work carried out on 58 pages, containing 21 illustrations, 1 table. Literature from 9 different sources was used in writing.

The main purpose of the work is to construction of the SDN Network Resource Information Model, using the T-API interface. The object of the study is to provide the right solutions on the principle of building a model. The subject of the study is the development and justification of the principle of model construction.

This paper considers the general information and architecture of the SDN network, northbound and southbound interfaces, SDN-controller based on the Open Networking Operating System, ONF OTCC and ONF T-API projects, construction the Topology and Information models. The advantages of the operating system management layer, software-defined and optical transport networks were also analyzed. As a result of the work, the analysis of Topology and Information models of T-API was performed.

Keywords: SDN, interface, topology, ONOS, T-API, YANG, model, architecture, controller, operating system, specification.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

API	Application Programming Interface, прикладний програмний інтерфейс
CIM	Common Information Model, загальна інформаційна модель
CORD	Central Office Rearchitected as a Datacenter, центральний офіс реогранізований як центр опрацювання даних
DWDM	Dense Wavelength Division Multiplexing, щільне мультиплексування з поділом по довжині хвилі
ESCON	Enterprise Systems Connection, підключення до корпоративних систем
FC	Fibre Channel, волоконний канал
FEC	Forward Error Correction, попередня корекція помилок
GbE	Gigabit Ethernet, гігабіт Ethernet
IoT	Internet of Things, Інтернет речей
IP	Internet Protocol, Інтернет протокол
ITU-T	International Telecommunication Union, Telecommunication sector, міжнародний союз електрозв'язку, сектор зв'язку
JVM	Java Virtual Machine, віртуальна машина Java
MEF	Metro Ethernet Forum, форум Metro Ethernet
MPLS	Multi Protocol Label Switching, багатопротокольна комутація за мітками
NBI	Northbound Interface, північний інтерфейс
NRM	Network Resource Modeling, моделювання мережевих ресурсів
NRP	Network Resource Provisioning, забезпечення мережевих ресурсів
OAM	Operations, administration and management, операції, адміністрування та обслуговування
OIF	Optical Internetworking Forum, форум оптичних Інтернет-мереж
ONF	Open Networking Foundation, фонд відкритих мереж
ONOS	Open Network Operating System, відкрита мережева операційна система
OSI	Open Systems Interconnection, модель взаємодії відкритих систем

	Open Transport Configuration & Control
OTCC	Optical Transport Networking, оптична транспортна мережа
OTN	Optical Tributary Signal
OTSi	Optical Tributary Signal Assembly
OTSiA ROADM	Reconfigurable optical add-drop multiplexer, реконфігурований оптичний мультиплексор з вводом-виводом
	Software Defined Networking, програмно-конфігурована мережа
SDN	Southbound Interface, південний інтерфейс
SBI	Synchronous Digital Hierarchy, синхронна цифрова ієрархія
SDH	Standards Development Organisation, організація з розробки стандартів
SDO SDK	Software Development Kit, комплект розробки програмного забезпечення
	Transport Application Programming Interface, транспортний
T-API	прикладний програмний інтерфейс
	Virtual Local Area Network, віртуальна локальна комп'ютерна мережа
VLAN	Wavelength Division Multiplexing, мультиплексування з поділом по
WDM	довжині хвилі

ВСТУП

Нині оператори орієнтуються на програмно-конфігуровані мережі SDN, щоб забезпечити програмованість своїх мереж - підвищення їх ефективності, швидкості розгортання і створення нових послуг, які приносять прибуток. Широке поширення парадигми програмованості залежить від наявності загальних або стандартизованих API, які дозволяють отримати доступ до специфічних для домену атрибутів та механізмів, не вимагаючи, щоб сам API був таким для постачальника або технології. Під час тестування було визначено, що T-API є критичним інтерфейсом для розгортання транспортних SDN у доменах з різним обладнанням та можливостями. Інфраструктура транспортної мережі дозволить повною мірою провести впровадження технології SDN вже найближчим часом.

Бізнес-моделі операторів зв'язку і їх мережеві архітектури залишалися відносно незмінними протягом багатьох років. Проте, поширення мобільних пристроїв і зростання необхідності послуг передачі даних, збільшення конкуренції з боку провайдерів, різке зростання вимог до пропускної здатності, а також до скорочення витрат і підвищення ефективності роботи є основними факторами, які змушують сервіс-провайдерів починати роботи по модернізації мереж і переглядати звичні підходи до ведення бізнесу. SDN відіграє у цій трансформації не останню роль.

Предметом дослідження роботи є мережева транспортна інфраструктура технології програмно-конфігурованих мереж SDN. Мета роботи полягає в побудові інформаційної моделі ресурсів мережі SDN з використанням інтерфейсу T-API. Проаналізувати можливості і перспективні рішення для побудови SDN мереж з використанням транспортного інтерфейсу.

Для здійснення мети необхідно виконати наступні задачі:

- Розглянути принципи побудови транспортних оптичних мереж.

– На основі принципів побудови запропонувати рішення у вигляді топологічної та інформаційної моделей.

РОЗДІЛ 1. SDN ТА ЇЇ АРХІТЕКТУРА, СТАНДАРТНІ ПРОТОКОЛИ ТА ІНТЕРФЕЙСИ

1.1 Загальні відомості про SDN

Програмно-конфігурована мережа (SDN) - це підхід до побудови мереж, який використовує програмні контролери або інтерфейси прикладного програмування (API) для зв'язку з базовою апаратною інфраструктурою і прямого трафіку в мережі.

Апаратне забезпечення панувало у світі мереж до появи програмно-конфігурованої мережі (SDN), категорії технологій, що відокремлюють площину управління мережею від площини переадресації, щоб забезпечити більш автоматизоване забезпечення та управління мережевими ресурсами на основі політик.

Витоки SDN можна прослідкувати у дослідницькій співпраці між Стенфордським університетом та Університетом Каліфорнії в Берклі, що в кінцевому підсумку дало протокол OpenFlow у 2008 році.

Мережі SDN відрізняються від традиційних мереж, в яких використовуються виділені апаратні пристрої (наприклад, маршрутизатори і комутатори) для управління мережевим трафіком. На основі SDN можна створити віртуальну мережу і керувати нею або керувати традиційним обладнанням через програмне забезпечення.

Майк Капуано, директор з маркетингу компанії Pluribus, відмітив, що в основі SDN лежить централізований або розподілений інтелектуальний об'єкт, який має повний огляд мережі та може приймати рішення про маршрутизацію і комутацію на основі цього огляду. Зазвичай маршрутизатори та комутатори знають лише про сусідні пристрої. Але завдяки правильно налаштованому середовищу, SDN контролер може контролювати все, починаючи від легко

змінюваних політик та спрощуючи конфігурацію та автоматизацію на всьому підприємстві.

За роки існування SDN перетворився на авторитетну мережеву технологію, пропоновану ключовими постачальниками, включаючи Cisco, VMware, Juniper, Pluribus та Big Switch. Open Networking Foundation також розробляє безліч технологій SDN з відкритим кодом.

Провайдери послуг зацікавлені в мережах SDN, адже вони сприяють:

- Зниженню капітальних та операційних витрат;
- Внесенню у свої мережі гнучкості та масштабованості хмарних технологій;
- Швидкому розгортанню сервісів;
- Зниженню складності експлуатації, підвищенню видимості;

Ідей, що лежать в основі розробки SDN, багато. Наприклад, значне спрощення автоматизації мережевих функції; спрощення представлення мережевих ресурсів і управління ними всюди, від центру опрацювання даних до кампусу або глобальної мережі.

Вектор розвитку програмно-конфігурованих мереж:

- Обслуговування десятків мільйонів фіксованих та сотні мільйонів бездротових кінцевих точок;
- Забезпечення високої продуктивності та низької затримки;
- Потрібна простота використання, створення і доставки послуг;
- Забезпечення безшовної міграції існуючого N/W (networking and the white box solutions), використовуючи white boxes;

1.2 Архітектура SDN

Програмно-конфігурована мережа складається з трьох рівнів: додатків, управління та інфраструктури, пов'язаних один з одним через північний і південний API-інтерфейси *(див. рис. 1.1).*

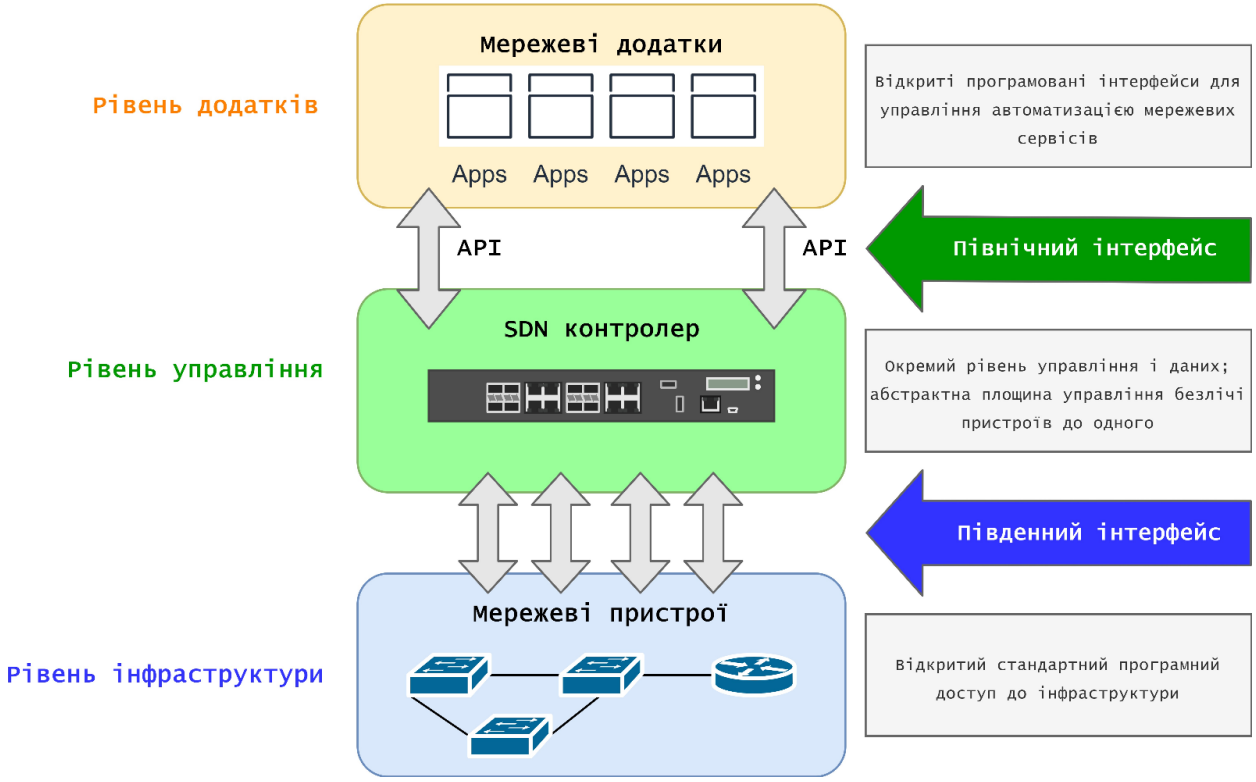

Рис. 1.1 Загальна архітектура SDN

Як видно з архітектури, крім класичного управління мережею командами системного адміністратора до контролера, SDN контролер підтримує запуск на собі додатків управління мережею. Кожен SDN додаток, по суті, являє собою інтерфейс оптимізації мережі під конкретний бізнес додаток і його основна роль - зміна мережі в реальному часі під поточні потреби програми, яка обслуговується *(див. рис. 1.2)*.

Якщо розглянути більш детально інформаційні потоки в архітектурі SDN, можна помітити два основних напрямки обміну інформацією: перший - між SDN додатками, другий для управління фізичними мережевими пристроями.

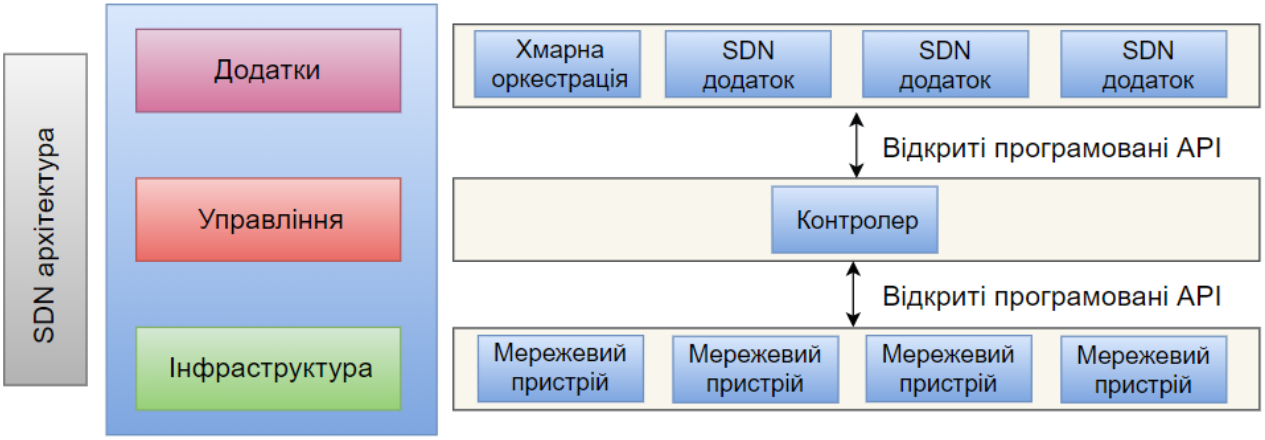

Рис. 1.2 Структура і компоненти SDN

Перший потік отримав назву «північний інтерфейс», а другий «південний інтерфейс». У якості «північного інтерфейсу» виступає протокол на основі REST API, а в якості «південного» зазвичай використовується протокол OpenFlow. *(див. рис. 1.3)*. [1]

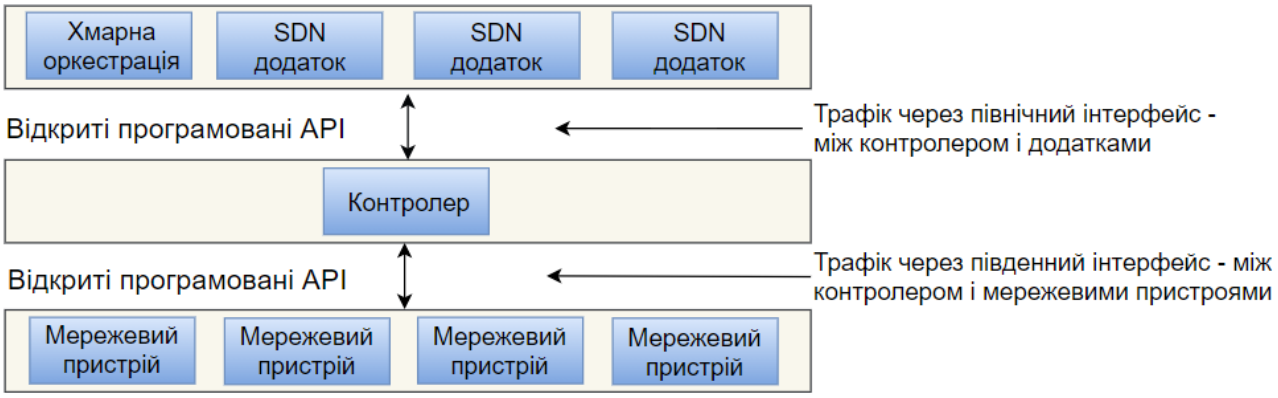

Рис. 1.3 Керуючі інформаційні потоки контролера SDN

1.3 Інтерфейси

1.3.1 Northbound Interface – Північний інтерфейс

Північні інтерфейси дозволяють взаємодіяти між компонентами більш високого рівня, на відміну від південного API. У той час як традиційні мережі використовують брандмауер або балансувальник навантаження для управління поведінкою площини даних, SDN встановлює додатки, які використовують контролер, і ці програми зв'язуються з контролером через його північний

інтерфейс. Це дозволяє мережевому адміністратору отримати доступ до SDN контролера, щоб налаштувати його або отримати з нього інформацію.

Крім того, північний API полегшує операторам мережі здійснювати нововведення або налаштовувати мережні елементи управління, і для обробки цього завдання не потрібна допомога фахівців, оскільки API може бути сконфігурований програмістом, який добре володіє такими мовами програмування, як Java, Python, або Ruby. API-інтерфейси Northbound використовуються в різних галузях, включаючи некомерційні сектори, освітні інститути, IT-компанії і багато інших *(див. рис. 1.4)*.

Північний інтерфейс використовується для доступу до самого контролера SDN. Це дозволяє адміністратору отримати доступ до SDN, щоб налаштувати його або отримати з нього інформацію. Це можна зробити за допомогою GUI, але він також пропонує API, який дозволяє іншим програмам отримувати доступ до контролера SDN. Це можна використовувати для написання сценаріїв і автоматизації мережевого адміністрування. Ось деякі приклади:

- Список інформації з усіх мережевих пристроїв у мережі.
- Відображення статусу всіх фізичних інтерфейсів в мережі.
- Додання нової VLAN на всі свої комутатори.
- Показ топології всієї мережі.
- Автоматичне налаштування IP-адреси, маршрутизації і списків доступу при створенні нової віртуальної машини.

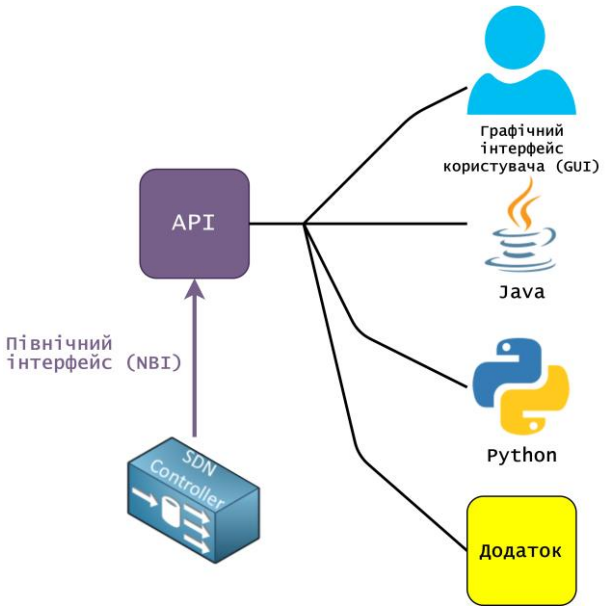

Рис. 1.4 Взаємодія додатків з SDN контролером

Деякі додатки через API можуть отримати доступ до контролера SDN:

- Користувач, який використовує графічний інтерфейс для отримання інформації про мережу від контролера SDN. Тобто, графічний інтерфейс підключений через API.
- Скрипти, написані на Java або Python, можуть використовувати API для отримання інформації з контролера SDN або налаштування мережі.
- Інші додатки можуть отримати доступ до контролера SDN. Наприклад, додаток, який автоматично налаштовує мережу після створення нової віртуальної машини на сервері VMware.

Багато завдань вирішується за допомогою наступних інтерфейсів та протоколів:

- FlowObjective – це протокол, який узагальнює правила потоку незалежним від конвеєра способом.
- FTP (File Transfer Protocol - Протокол передачі файлів) - це набір стандартних протоколів для передачі файлів по мережі. FTP передає паролі і вміст файлів у вигляді звичайного тексту. SSH FTP (SFTP) використовує протокол SSH для забезпечення безпечної передачі та обробки файлів. З SFTP паролі і дані шифруються під час передачі.

- gNMI (gRPC Network Management Interface - інтерфейс управління мережею gRPC): використовується для установки і отримання стану конфігурації. gNMI зазвичай поєднується з моделями OpenConfig YANG, які визначають структуру конфігурації і дерева станів.
- gNOI (gRPC Network Operations Interface - інтерфейси мережевих операцій gRPC): використовується для установки і отримання робочого стану, наприклад, для підтримки управління сертифікатами, тестування пристроїв, оновлення програмного забезпечення та усунення неполадок в мережі.
- GUI (Graphic User Interface - графічний користувацький інтерфейс) - це один із різновидів користувацьких інтерфейсів, елементи якого виконані у вигляді графічних зображень.
- REST API (Representational State Transfer - передача репрезентативного стану) - це інтерфейс програмування додатків (API або веб-API), який відповідає обмеженням архітектурного стилю REST і дозволяє взаємодіяти з веб-службами RESTful. REST - це архітектурний стиль для розподілених гіпертекстових систем.
- SNMP (Simple Network Management Protocol - простий протокол мережевого управління) зазвичай використовується в системах мережного управління для контролю підключених до мережі пристроїв на предмет умов, які вимагають уваги адміністратора.
- SYSLOG (System Log - системний журнал) - стандарт відправки і реєстрації повідомлень про події, що відбуваються в системі (тобто створення журналів про події), що використовується в комп'ютерних мережах і працює по протоколу IP.
- TACACS (Terminal Access Controller Access Control System) - це протокол автентифікації, загальний для мереж UNIX, який дозволяє серверу віддаленого доступу пересилати пароль входу користувача на сервер автентифікації, щоб визначити, чи можна дозволити доступ до певної системи.

1.3.2 Southbound Interface – Південний інтерфейс

Контролер SDN повинен взаємодіяти з мережевими пристроями, щоб програмувати площину даних. Це робиться через південний інтерфейс. Це не фізичний інтерфейс, а програмний інтерфейс.

У SDN південним інтерфейсом є специфікація протоколу OpenFlow (або альтернативного). Його основна функція - забезпечити зв'язок між контролером SDN і мережевими вузлами (як фізичними, так і віртуальними комутаторами, і маршрутизаторами), щоб маршрутизатор міг виявляти топологію мережі, визначати мережеві потоки і реалізовувати запити, передані йому через північний API.

Інтерфейси API Southbound дозволяють кінцевому користувачеві краще контролювати мережу і підвищують рівень ефективності контролера SDN, щоб розвиватися відповідно до вимог та потреб в реальному часі. Крім того, інтерфейс є галузевим стандартом, який виправдовує ідеальний підхід, при якому контролер SDN повинен взаємодіяти з площиною пересилання для зміни мереж, які дозволять йому поступово рухатися разом зі зростаючими потребами підприємства. Щоб скласти кращий мережевий рівень до вимог трафіку в реальному часі, адміністратори можуть додавати або видаляти записи у внутрішню таблицю потоків мережевих комутаторів і маршрутизаторів.

Опис найбільш використовуваних південних інтерфейсів та протоколів *(див. рис. 1.5):*

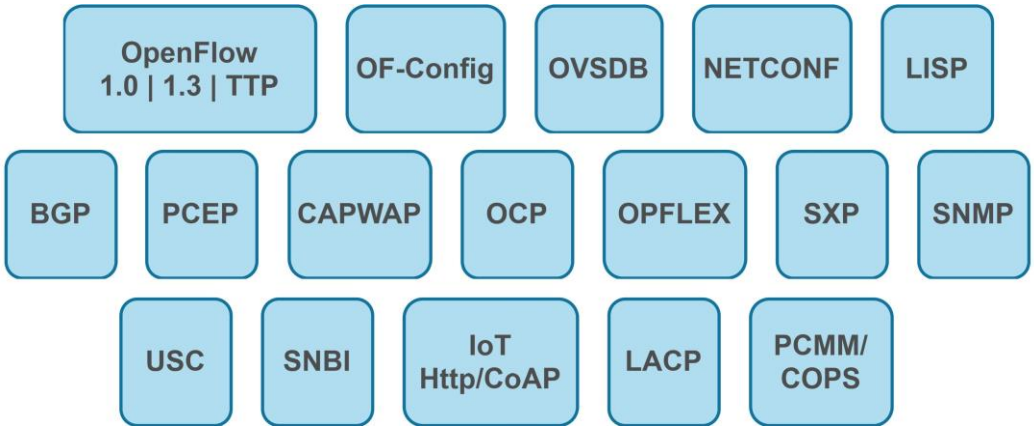

Рис. 1.5 Південні інтерфейси та протоколи

- BGP (Border Gateway Protocol) - це протокол маршрутизації за замовчуванням, який використовується маршрутизаторами для зв’язку з іншими маршрутизаторами щодо найкращого способу доступу до Інтернет-доменів.
- CAPWAP (The Control And Provisioning of Wireless Access Points - контроль і надання бездротових точок доступу) - це стандартний, сумісний мережевий протокол, який дозволяє центральному контролеру доступу до бездротової локальної мережі (AC) управляти набором точок завершення бездротової мережі (WTP), більш відомих як точки бездротового доступу. Специфікація протоколу описана в RFC 5415.
- CLI (Command Line Interface - інтерфейс командного рядка) він використовує протоколи, доступні на обладнанні поточного покоління, наприклад Telnet, SSH і SNMP.
- CoAP (Constrained Application Protocol - протокол обмежених додатків) - це спеціалізований протокол веб-передачі, який використовується з обмеженими вузлами та обмеженими мережами в Інтернеті речей.
- COPS (Common Open Policy Service - загальна служба відкритих політик) - протокол, який є частиною набору інтернет-протоколів, як визначено в RFC 2748 IETF. COPS визначає просту модель клієнт/сервер для підтримки управління політикою над протоколами сигналізації якості обслуговування (наприклад, RSVP). Політики зберігаються на серверах і використовуються точками прийняття рішень (PDP) і застосовуються на клієнтах, також відомих як точки застосування політик (PEP). Існує дві моделі COPS: модель аутсорсингу і модель забезпечення, що розглядається з точки зору клієнта або PEP.
- FTP (File Transfer Protocol - протокол передачі файлів) - це набір стандартних протоколів для передачі файлів по мережі. FTP передає паролі і вміст файлів у вигляді звичайного тексту. SSH FTP (SFTP) використовує протокол SSH для забезпечення безпечної передачі та обробки файлів. З SFTP паролі і дані шифруються під час передачі.

- gNMI (gRPC Network Management Interface - інтерфейс управління мережею gRPC): використовується для установки і отримання стану конфігурації. gNMI зазвичай поєднується з моделями OpenConfig YANG, які визначають структуру конфігурації і дерева станів. gRPC (gRPC Remote Procedure Calls - виклик віддалених процедур).
- gNOI (gRPC Network Operations Interface - інтерфейси мережевих операцій gRPC): використовується для установки і отримання робочого стану, наприклад, для підтримки управління сертифікатами, тестування пристроїв, оновлення програмного забезпечення та усунення неполадок в мережі.
- IOT HTTP/CoAP (англ. Hyper Text Transfer Protocol - протокол передачі гіпертексту) розроблений для задоволення потреб систем IoT на основі HTTP, CoAP покладається на протокол дейтаграм користувача (UDP) для встановлення безпечного зв'язку між кінцевими точками. Надаючи трансляцію і багатоадресну розсилку, UDP може передавати дані на кілька хостів, зберігаючи при цьому швидкість зв'язку та низьку пропускну здатність, що робить його гарним вибором для бездротових мереж, зазвичай використовуваних в середовищах M2M з обмеженими ресурсами. Ще одна річ, яку CoAP розділяє з HTTP, - це архітектура RESTful, яка підтримує модель взаємодії запит/відповідь між кінцевими точками програми. Більш того, CoAP використовує основні методи HTTP get, post, put і delete, завдяки яким можна уникнути неоднозначності під час взаємодії між клієнтами.
- LACP (Link Aggregation Control Protocol - протокол управління агрегацією каналів) - відкритий стандартний протокол агрегування каналів.
- LISP (Locator/Identifier Separation Protocol - протокол поділу локатора/ідентифікатора) являє собою протокол "зіставлення і інкапсуляції". Основна ідея поділу полягає в тому, що архітектура Інтернету об'єднує дві функції: локатори маршрутизації (коли клієнт

підключений до мережі) і ідентифікатори (хто є клієнтом) в одному просторі номерів: IP-адреса. LISP підтримує поділ адресного простору IPv4 і IPv6 відповідно до мережевої схеми відображення і інкапсуляції (RFC 1955). У LISP ідентифікатори і локатори можуть бути IP-адресами або довільними елементами, такими як набір координат GPS або MAC-адресу.

- NETCONF (Network Configuration - конфігурація мережі) є протоколом мережевого управління пристроями. NETCONF надає механізми установки, управління і видалення конфігурації мережевих пристроїв за допомогою віддаленого виклику процедур RPC. NETCONF використовує XML як засіб надання конфігурації і як засіб формування повідомлень протоколу, який реалізується поверх транспортного.
- OCP (Open Core Protocol - відкритий основний протокол) - протокол для обміну даними всередині підсистеми. Це відкрито-ліцензований протокол, орієнтований на ядро і визначає незалежний від шини налаштовуваний інтерфейс.
- OF-Config (OpenFlow Configuration and Management Protocol - протокол конфігурації і управління OpenFlow) - це спеціальний набір правил, який визначає механізм, для контролерів OpenFlow, для доступу і зміни даних конфігурації на комутаторі OpenFlow.
- OpenFlow - найпопулярніший інтерфейс на даний момент. Це протокол з відкритим вихідним кодом від Open Networking Foundation. Існує досить багато мережевих пристроїв і контролерів SDN, що підтримують OpenFlow. Він надає єдиний API, за допомогою якого адміністратори можуть програмувати роботу мережі, а також задавати правила маршрутизації пакетів, балансування навантаження і управління доступом. В цей API входять два основних компоненти: програмний інтерфейс для контролю пересилання пакетів через мережеві комутатори і набір глобальних інтерфейсів, на основі яких можна створювати високорозвинені інструменти управління.

- OpFLEX - цей протокол є новітнім доповненням ініціативи Cisco ACI, яка представляє альтернативний варіант SDN-платформи для організацій. Cisco Systems продовжує просувати свою, орієнтовану на додатки, інфраструктуру (Application Centric Infrastructure, ACI) в якості альтернативи з'являються на ринку платформ програмно-конфігурованих мереж (SDN).
- OVSDB (Open vSwitch Database - протокол управління базою даних) - це протокол конфігурації OpenFlow, призначений для управління реалізаціями Open vSwitch.
- PCEP (Path Computation Element Protocol - протокол зв'язку елемента обчислення шляху) - це спеціальний набір правил, який дозволяє клієнту обчислення шляху (PCC) запитувати обчислення шляху від елементів обчислення шляху (PCE). Протокол також дозволяє PCE повертати відповіді.
- PCMM - оптимізує квитирування IPC між лінійної картою кабельного інтерфейсу і процесором маршрутизації для маршрутизатора Cisco CMTS. Додаткові зміни інтерфейсу PCMM в порівнянні з PacketCable 1.x включають обробку інтерфейсу COPS і розподілених лінійних карт кабельного інтерфейсу.
- SNMP (Simple Network Management Protocol - простий протокол мережевого управління) використовується в системах мережного управління для контролю підключених до мережі пристроїв на предмет умов, які вимагають уваги адміністратора.
- SXP (SGT Exchange Protocol – протокол обміну SGT). Пристрої Cisco використовують протокол обміну SGT (SXP) Для поширення SGT на мережеві пристрої, які не мають апаратної підтримки Cisco TrustSec. SXP - це програмне рішення, що усуває необхідність в оновленні обладнання Cisco TrustSec на всіх комутаторах Cisco. Контролер підтримує SXP як частина архітектури Cisco TrustSec. SXP відправляє інформацію SGT на комутатори, щоб можна було активувати відповідні списки управління

доступом на основі ролей (списки RBAC). Це залежить від рольової інформації, представленої в SGT. Для реалізації SXP в мережі тільки вихідний комутатор розподілу повинен мати підтримку Cisco TrustSec. Всі інші комутатори можуть бути комутаторами, які не підтримують Cisco TrustSec.

- Тунельний протокол, який використовується для взаємодії контролера і точок радіодоступу. Заснований на LWAPP. CAPWAP передає трафік управління і даних:
  а) Трафік управління шифрується по протоколу DTLS (RFC 4347);
  б) Трафік даних опціонально шифрується по протоколу DTLS.

Отже, в обох інтерфейсів є свої ролі. Функції північного API в корпоративному центрі обробки даних полягають в розробці управлінських рішень для автоматизації, оркестрації і обміну дієвими даними між системами, в той час як південний - працює для доставки протоколів віртуалізації мережі, взаємодії з комутаційної фабрикою або інтеграції розподіленої обчислювальної мережі.

Висновок до розділу 1

У даному розділі було розглянуто основні положення та переваги програмно-конфігурованих мереж, проаналізовано їх архітектуру та детально описано протоколи та інтерфейси, які виступають у якості північних і південних у SDN мережах.

SDN дозволяє централізовано керувати поведінкою мережі за допомогою програмного забезпечення і відкритих API, розробляти програми з урахуванням пропускної здатності мережі, відстежувати її стан і автоматично підлаштовувати конфігурацію під конкретні умови.

Архітектура SDN складається з трьох рівнів, що відкриває ряд можливостей для інновацій в галузі управління процесом передачі даних і забезпечення інформаційної безпеки. Програмно-конфігуровані мережі

дозволяють взаємодіяти різним постачальникам і сприяють створенню комерційно нейтральної екосистеми, тим самим віщуючи нову епоху відкритості. Відкритість закладена в самій основі SDN. Замість інтерфейсів управління, прив'язаних до апаратного забезпечення, додатки взаємодіють з мережею через API.

РОЗДІЛ 2. SDN-КОНТРОЛЕР НА БАЗІ ВІДКРИТОЇ МЕРЕЖЕВОЇ ОПЕРАЦІЙНОЇ СИСТЕМИ ONOS

2.1 SDN-контролер

Рівень додатків містить програми та такі мережеві функції, як фаєрволи і балансування навантаження. Традиційні мережі використовують для цієї мети спеціальний пристрій, а програмно-конфігуровані - контролер для управління площиною передачі даних. Рівень управління керує політиками і трафіком в мережі. Рівень інфраструктури містить фізичні комутатори мережі.

Контролер SDN - це додаток в програмно-конфігурованій мережевій архітектурі, який управляє керуванням потоку для покращення управління мережею і продуктивністю додатків. Платформа контролера SDN зазвичай працює на сервері і використовує протоколи, щоб повідомити комутаторам, куди відправляти пакети. Контролери направляють трафік відповідно до політики переадресації, яка встановлена оператором мережі, тим самим зводячи до мінімуму ручне налаштування окремих мережевих пристроїв. Прибравши площину управління з мережевого обладнання та запустивши її замість програмного забезпечення, централізований контролер спрощує автоматичне керування мережею і спрощує інтеграцію, і адміністрування бізнес-додатків. По суті, контролер SDN служить операційною системою (ОС) для мережі. [2]

Отже, контролер - це ядро програмно-конфігурованої мережі. Він знаходиться між мережевими пристроями на одному кінці мережі і додатками на іншому кінці. Будь-який зв'язок між додатками і мережевими пристроями повинен здійснюватися через контролер.

Контролер взаємодіє :

- з додатками, такими як міжмережеві екрани або балансувальники навантаження, через північні інтерфейси;
- з окремими мережевими пристроями, використовуючи інтерфейс південного напрямку, зазвичай такий, як протокол OpenFlow. Ці південні протоколи дозволяють контролеру налаштовувати мережеві

пристрої і вибирати оптимальний мережевий шлях для трафіку додатків.

Основні завдання рівня управління покладаються на SDN контролер зі встановленою операційною системою ONOS.

2.2 Відкрита мережева операційна система ONOS

2.2.1 Історія створення

5 грудня 2014 року Open Networking Lab (ON.Lab) разом з іншими галузевими партнерами, включаючи AT & T і NTT Communications, випустила вихідний код ONOS, щоб створити спільноту відкритого вихідного коду. 14 жовтня 2015 року Linux Foundation оголосив, що ONOS приєднався до організації в рамках одного зі спільних проєктів.

Проєкт було розпочато приблизно в жовтні 2012 року під керівництвом Панкай Берде, архітектора ON.Lab. Назва ONOS була придумана Берде приблизно в кінці 2012 року. Ранній прототип був показаний в квітні 2013 року на Open Networking Summit (ONS), а шлях початкових ітерацій був представлений на ONS у 2014 році.

2.2.2 Загальні відомості

ONOS - це операційна система для управління SDN-мережами і їх компонентами, такими як комутатори та канали, виконуючи програми або модулі для надання послуг зв'язку кінцевим хостам і сусіднім мережам. Особливістю є те, що вона поєднує в собі функціонал SDN-контролера, мережевої і серверної ОС. За рахунок такої комбінації інструмент дозволяє стежити за всім, що відбувається в мережах, і спрощує міграцію від традиційної архітектури до SDN. Вразливим місцем платформи була безпека, але з 2015 року платформа отримала велику кількість оновлень, що підвищили безпеку середовища.

Рис. 2.1 Командний рядок ONOS

Багато компаній зробили свій внесок у проєкт ONOS. У ONOS є два рівня членства: партнери і співробітники.

Партнери: AT&T, China Unicom, Ciena, Cisco, Comcast, Deutsche Telekom, Ericsson, Fujitsu, Google, Huawei, Intel, NEC, Nokia, NTT Communications, Radisys, Samsung Electronics, Türk Telekom, Verizon та інші.

Співробітники: AARNET, ADARA Networks, Airhop Communications, Akamai, AmLight, BlackDuck, BTI Systems, Beijing University of Posts and Telecommunications, Cavium, ClearPath Networks, CNIT, CREATE-NET, Criterion Networks, CSIRO, ECI Telecom, ETRI, Consortium GARR, GEANT, Happiest Minds, Internet2, INSPIRE group, KAIST, KREONET, KISTI, NAIM Networks, NetCracker, OpenFlow Korea, Open Networking Foundation та інші.

ONOS був розроблений для задоволення потреб операторів, які бажають створювати рішення операторського класу, які використовують економіку напівпровідникового апаратного забезпечення для комерційних підприємств, пропонуючи гнучкість для створення і розгортання нових динамічних мережевих сервісів зі спрощеними програмними інтерфейсами. ONOS підтримує як налаштування, так і управління мережею в реальному часі,

усуваючи необхідність запуску протоколів управління маршрутизацією і комутацією всередині мережевої структури. Перенесення інтелекту в хмарний контролер ONOS дозволяє впроваджувати інновації: кінцеві користувачі можуть легко створювати нові мережеві додатки без необхідності змінювати системи передачі даних. [4]

ONOS включає та забезпечує:

- Платформу і набір додатків, які діють як розширюваний модульний розподілений контролер SDN.
- Горизонтально-масштабовану архітектуру для забезпечення відмовостійкості і масштабованості, необхідних для відповідності суворим умовам виробничого операторського середовища.
- Спрощене управління, налагодження та розгортання нового програмного забезпечення, обладнання та послуг.

ONOS надає деякі, аналогічні серверним операційнним системам, типи функцій, включаючи API-інтерфейси і абстракції (елементи), розподіл ресурсів і дозволу, а також програмне забезпечення, орієнтоване на користувача, таке як CLI, GUI і системні програми. ONOS керує всією мережею, а не одним пристроєм, що може значно спростити управління, налаштування і розгортання нового програмного забезпечення, обладнання та послуг *(див. рис. 2.2)*.

Найбільш важливою перевагою операційної системи є те, що вона забезпечує корисну і зручну платформу для програм, розроблених для конкретного додатку або варіанту використання. Додатки та сценарії використання ONOS часто складаються з налаштованих служб маршрутизації, управління або моніторингу для програмно-конфігурованих мереж. ONOS може працювати як розподілена система на декількох серверах, що дозволяє використовувати ресурси центрального процесора та пам'яті кількох серверів, забезпечуючи відмовостійкість у разі збою сервера і потенційно підтримуючи оновлення обладнання і програмного забезпечення в реальному часі або по черзі без переривання мережевого трафіку.

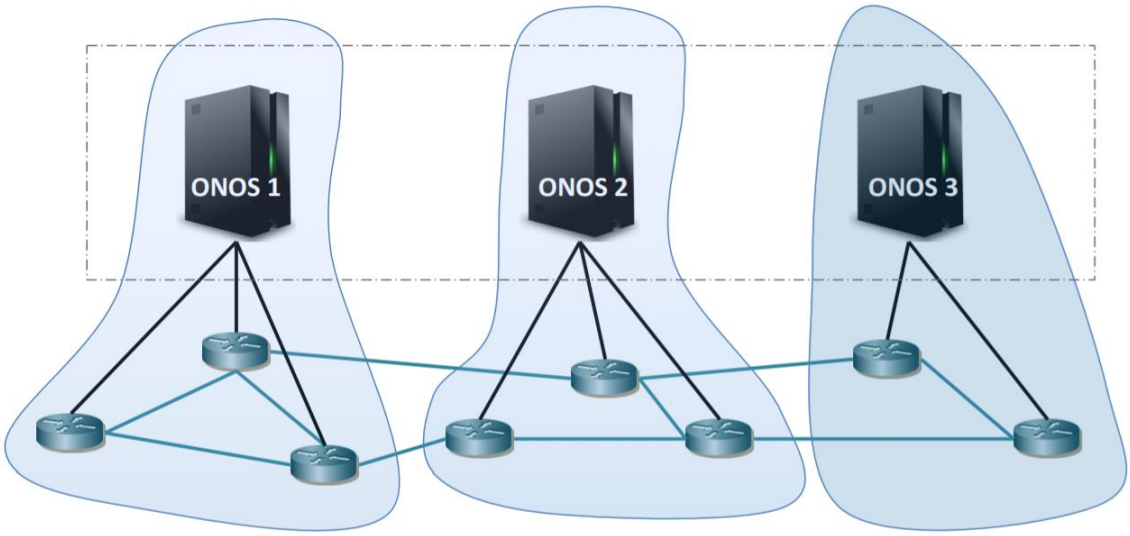

Рис. 2.2 Топологія SDN мережі з використанням контролерів ONOS

Рисунок 2.3 – це наочне зображення того випадку, коли один контролер через будь-яку причину не працює. Отже, це не є проблемою для мережі, так як інші контролери мають внутрішній протокол для комунікації, вони можуть виконувати функцію іншого.

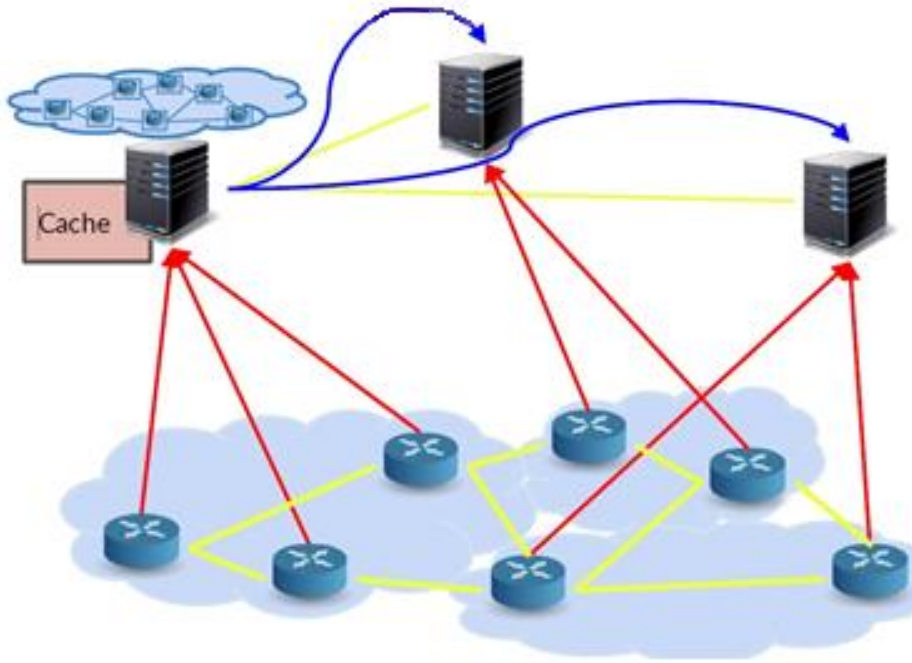

Рис. 2.3 Топологія SDN мережі у разі відмови одного з контролерів ONOS

2.2.3 Архітектура ONOS

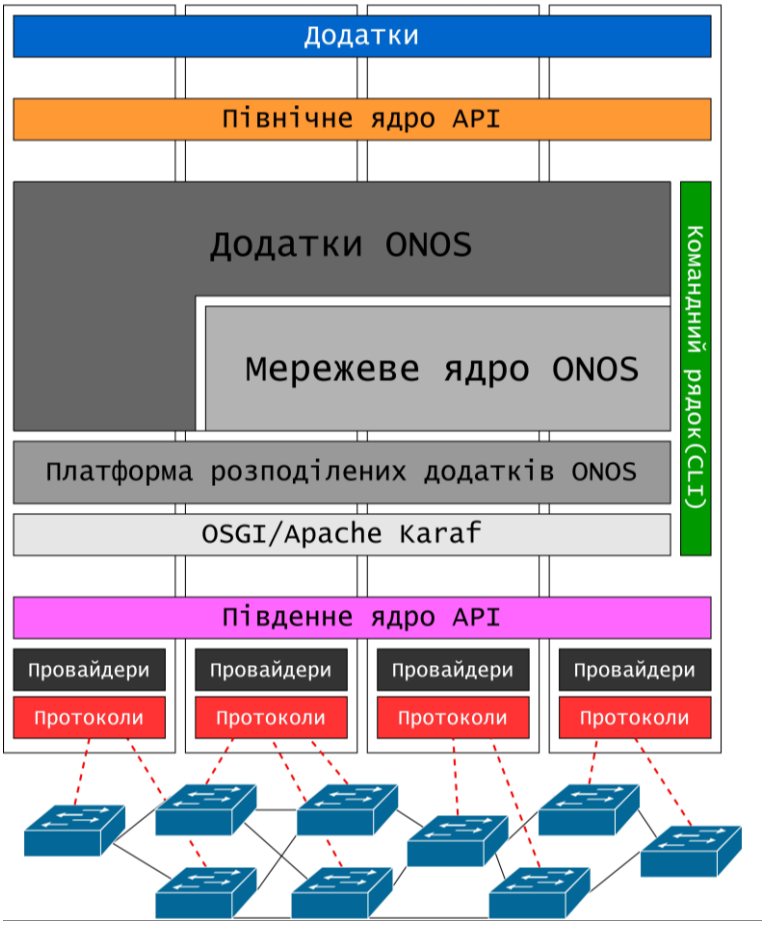

Рис. 2.4 Архітектура ONOS

Ядро ONOS і основні служби, а також додатки ONOS написані на Java у вигляді пакетів, які завантажуються в контейнер Karaf OSGi. Apache Karaf - це сучасний поліморфний контейнер, може використовуватися як автономний контейнер, підтримуючи широкий спектр додатків та технологій. Apache Karaf можна масштабувати від дуже легкого контейнера до повнофункціональної корпоративної служби: це дуже гнучкий та розширюваний контейнер, що охоплює всі основні потреби. Він працює на основі OSGi. OSGi - це компонентна система для Java, яка дозволяє встановлювати і запускати модулі динамічно в одній JVM. Оскільки ONOS працює в JVM, він може працювати на декількох базових платформах ОС. Система призначена для роботи у вигляді кластера вузлів, які ідентичні з точки зору їх програмного стека і можуть витримувати відмову окремих вузлів, не викликаючи порушень в її здатності

контролювати роботу мережі. Ядро ONOS відповідає за відстеження інформації про мережеве середовище і поширення його застосування або синхронно через запит, або асинхронно через зворотні виклики. Ядро також відповідає за збереження стану вибору і синхронізацію стану між вузлами кластера; забезпечує контроль станів, сповіщень, високу доступність та масштабованість. Тут відбувається основний логічний алгоритм.

Хоча ONOS в значній мірі спирається на стандартні протоколи і моделі, наприклад OpenFlow, NETCONF, OpenConfig, його системна архітектура безпосередньо до них не прив'язана. Замість цього ONOS надає свій власний набір елементів і моделей високого рівня, які він надає програмістам додатків. Ці моделі можуть бути розширені додатками під час виконання. Щоб система не була прив'язана до певної конфігурації або протоколу керування, будь-яке програмне забезпечення, яке перебуває в прямому контакті з бібліотеками конкретних протоколів і бере участь в прямій взаємодії з мережевим середовищем, навмисно ізольованим на своєму власному рівні, званому постачальником або драйвером. Точно так будь-яке програмне забезпечення, безпосередньо контактує з протоколами внутрішньо-кластерного зв'язку, навмисно ізольоване на своєму власному рівні, званому сховищем.

Платформа надає додаткам ряд високорівневих елементів, за допомогою яких вони можуть дізнаватися про стан мережі і можуть контролювати потік трафіку через мережу. Елементи мережевого графа надають інформацію про структуру і топології мережі. Flow Objective - це абстракція, орієнтована на пристрої, яка дозволяє програмам направляти потік трафіку через конкретний пристрій без необхідності знати конвеєр таблиці пристроїв. Intent - це мережевоцентрична абстракція, яка дає прикладним програмістам можливість управляти мережею, вказуючи, чого вони хочуть досягти, замість того, щоб вказувати, як вони хочуть це зробити. Це спрощує розробку додатків і в той же час надає платформі додаткові ступені свободи для вирішення того, що зазвичай вважається конфліктуючими запитами. Додатки (розширення ядра) можна завантажувати і вивантажувати динамічно, через REST API або

графічний інтерфейс, і без необхідності перезапускати кластер або його окремі вузли.

Підсистема управління додатками ONOS бере на себе відповідальність за поширення артефактів додатків по кластеру, щоб гарантувати, що всі вузли працюють з одним і тим же програмним забезпеченням. Базовий дистрибутив ONOS містить більше 175 додатків, які потрапляють в численні категорії, такі як програми для управління трафіком, драйвери пристроїв, утиліти, програми для моніторингу, готові моделі YANG. YANG - це мова моделювання даних для визначення даних, що передаються через протоколи мережевого управління, такі як NETCONF та RESTCONF. Мова моделювання даних може бути використана для моделювання як даних конфігурації, так і даних стану елементів мережі. Крім того, YANG можна використовувати для визначення формату повідомлень про події, що видаються елементами мережі.

Для того щоб взаємодіяти із зовнішнім світом, в ONOS є графічний інтерфейс та низка зовнішніх адаптерів, таких як REST API, CLI і розширюваний динамічний веб-інтерфейс. Такі інтерфейси, як gRPC для ONOS знаходяться в стадії активної розробки. gRPC - це система з відкритим вихідним кодом. Використовує HTTP/2 для транспорту, буфери протоколу в якості мови опису інтерфейсу і надає такі функції, як аутентифікація, двонаправлена потокова передача і управління потоком, а також скасування і тайм-аути. Він генерує кросплатформені (ті, що працюють більш ніж на одній програмній або апаратній платформі) прив'язки клієнта і сервера для багатьох мов.

Додатки ONOS виступають як розширення ядра, вони можуть бути бібліотеками протоколів, драйверами або попередньо скомпільованими моделями.

Програми та додатки, які будуть впроваджені в ONOS у майбутньому:

- Підсистема інструментів і динамічної конфігурації YANG;
- Графічні подання топології кількох регіонів;
- Дизагрегація по кодової базі;

- Інтеграція та підтримка P4 & gNMI;
- Оновлення програмного забезпечення в процесі експлуатації;
- gRPC API;
- Розгортання;
- Віртуалізація;
- Покращення процесу складання і випуску.

2.2.4 Підсистеми ONOS

Існують різні способи побудови розподілених систем. ONOS побудований, як симетричний, тобто всі вузли програмно однакові. Наприклад, один вузол може виконувати функцію та роботу іншого вузла. Так як ONOS модульна операційна система, вона складається із окремих підсистем *(див.рис. 2.5)*.

Рис. 2.5 Підсистеми ONOS

Ядро ONOS складається із двох частин: мережевої та немережевої. Мережева частина підтримує виконання таких функцій: збір статистики, аналіз топології, конфігурація пристрою, створення віртуальних мереж, веб-груп, загалом підтримка служб для роботи в Інтернеті. Немережева частина виконує функції, необхідні для керування фізичною частиною SDN контролера, а саме обмін повідомленнями між контролерами, керування пам'яттю пристрою, налаштування операційної системи локально, підтримка графічного інтерфейсу для спрощеного керування системою.

DHCP - це стандартний протокол прикладного рівня, який дозволяє комп'ютерам автоматично отримувати IP-адресу та інші параметри, необхідні для роботи в мережі. Для цього комп'ютер звертається відповідно - до DHCP-сервера.

Проксі ARP - це техніка, за допомогою якої проксі-сервер у даній мережі відповідає на запити протоколу роздільної адреси щодо IP-адреси, якої немає в цій мережі.

L2 переадресація - це протокол тунелювання, розроблений для встановлення віртуальних приватних мережевих підключень через Інтернет.

SDN-IP/BGP - це додаток ONOS, який дозволяє програмно-конфігурованій мережі підключатися до зовнішніх мереж в Інтернеті з використанням стандартного протоколу граничного шлюзу (BGP). Зовні, з точки зору BGP, мережа SDN виглядає як окрема автономна система (AS), яка поводиться як будь-яка традиційна AS. У AS додаток SDN-IP забезпечує механізм інтеграції між BGP і ONOS. На рівні протоколу SDN-IP поводиться як звичайний динамічний BGP. З точки зору ONOS, це просто додаток, який використовує свої служби для встановлення та оновлення відповідного стану пересилання в площині даних SDN.

Пакет/оптика - багатошарова мережа SDN, яка об'єднує пакетні і оптичні домени.

Northbound API та Southbound API в ONOS налаштовані так, щоб додатки мали доступ до будь-якої інформації із будь-якої точки кластера. NB API є дуже важливим фактором, адже тут знаходяться фактичні мережеві служби, які працюють в додатках. SB API забезпечує зв'язок між контролером SDN і мережевими вузлами (див. рис. 2.6). [3]

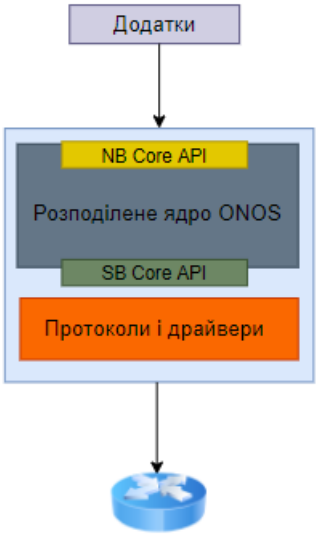

Рис. 2.6 Структура SDN із Northbound та Southbound API

Рисунок 2.7 показує взаємодію контролера ONOS із користувачем за допомогою таких протоколів та технологій. Нижня частина відображає інтерфейси/протоколи, через які контролер може керувати найнижчим рівнем архітектури, тобто рівнем інфраструктури.

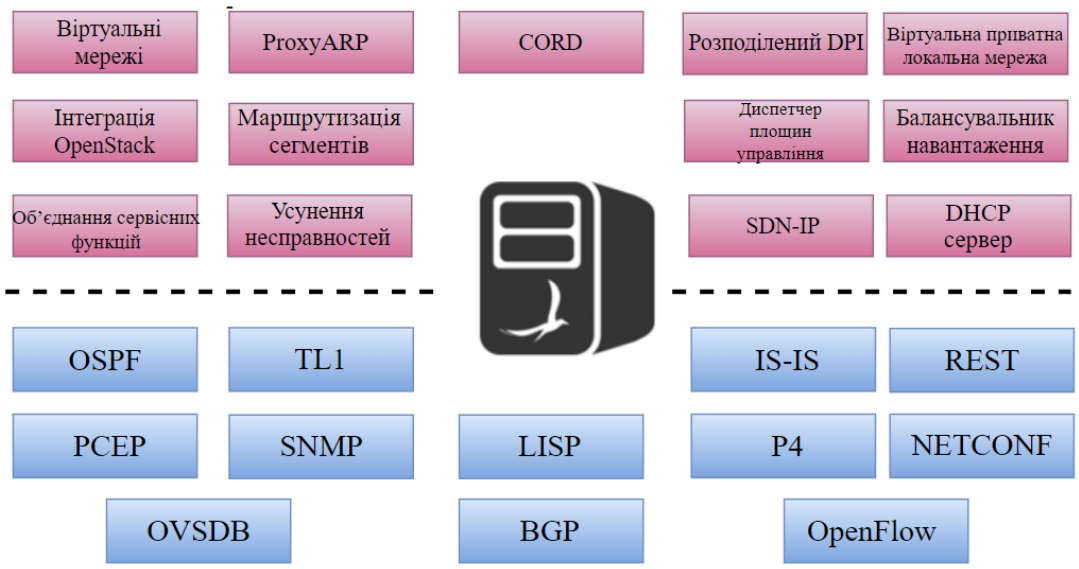

Рис. 2.7 Взаємодія контролера ONOS за допомогою протоколів

Висновок до розділу 2

У ході виконання даного розділу було розглянуто рівень управління та операційну систему ONOS. Рівень управління включає операційну систему, яка забезпечує додаткам сервіси та програмний інтерфейс для управління пристроями та мережею.

Контролер – це основний мережевий пристрій, який за допомогою встановленої операційної системи управляє пристроями рівня інфраструктури. Зв'язок між додатками і мережевими пристроями повинен здійснюватися через контролер.

Стандартні служби ONOS дають можливість управління, моніторингу та маршрутизації програмно-конфігурованих мереж. ONOS працює як розподілена система, що забезпечує відмовостійкість у разі збою. Відкрита мережева операційна система є зручною платформою із відкритим вихідним кодом, що полегшує розробникам програмного забезпечення та обладнання у створенні технологій, які можна буде інтегрувати в SDN.

РОЗДІЛ 3: ОПТИЧНА ТРАНСПОРТНА МЕРЕЖА ТА ПРИНЦИП ПОБУДОВИ МОДЕЛІ З ВИКОРИСТАННЯМ ІНТЕРФЕЙСУ T-API

3.1 Оптична транспортна мережа

Оптична транспортна мережа (OTN), також відома як технологія цифрової упаковки, - це протокол промислового стандарту нового покоління, який використовується як ефективний і загальноприйнятий спосіб мультиплексування різних послуг на оптичних світлових шляхах.

OTN - це технологія, аналогічна SONET/SDH, але розроблена з урахуванням поточних і майбутніх вимог до протоколу і смуги пропускання - стандартизована ITU в ряді специфікацій, в першу чергу, в стандарті G.709. Ці стандарти визначають інкапсуляцію корисного навантаження клієнта, службові дані OAM, упереджене виправлення помилок (FEC) та ієрархію мультиплексування. Разом ці функції забезпечують такий же надійний і керований транспорт, як SONET/SDH, але мають ряд поліпшень. Архітектура OTN зображена на рисунку 3.1, де показано, як клієнтський трафік різних протоколів інкапсулюється в цифрову оболонку, потім транспортується і управляється через інфраструктуру оптичної мережі. [5]

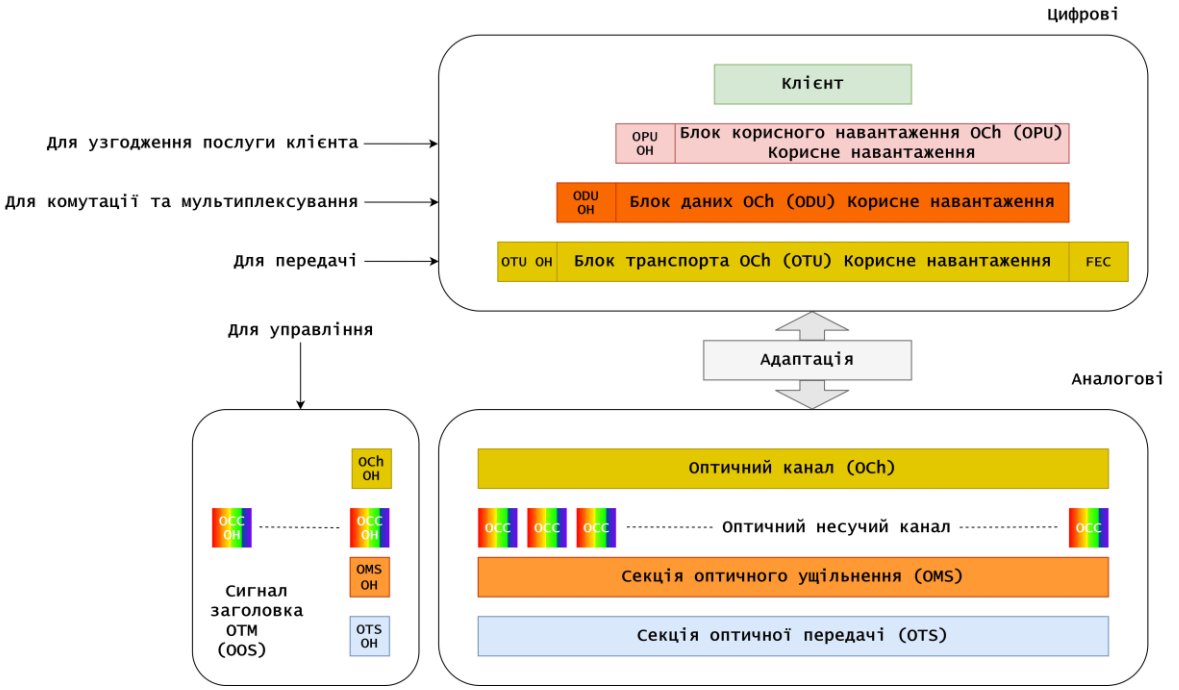

Рис. 3.1 Архітектура OTN

Кадр OTN зазвичай представляється у вигляді матриці, що складається з 4080 стовпців-байтів і 4-х рядків. Він складається з поля користувальницьких даних (Payload) і службових полів блоків OPU, ODU і OTU. Формат кадру залежить від рівня швидкості OTN. Поле для користувача даних розташовується з 17-го по 3824-й стовпець і займає всі 4 рядки кадру, а заголовок блоку OPU займає стовпці 15 і 16 також в 4-х рядках. При необхідності заголовок OPU OH може займати кілька кадрів підряд (наприклад, такий варіант зустрічається в тому випадку, коли потрібно описати структуру поля призначених для користувача даних, мультиплексуючу кілька блоків OPU нижчого рівня).

Подібно SONET/SDH, OTN визначає робочий канал, який називається загальним каналом зв'язку (GCC) та переноситься в службових байтах. Це використовується для функцій OAM, таких як моніторинг продуктивності, виявлення несправностей, а також команди сигналізації й обслуговування, для підтримки захисного перемикання, усунення несправностей та моніторингу на рівні обслуговування.

ITU визначив різні швидкості, як зазначено в таблиці 1. Ці швидкості рівні або трохи перевищують загальні швидкості передачі даних SONET/SDH, що дозволяє прозоро передавати ці сигнали.

Таблиця 1

Стандартні формати оптичних блоків даних OTN

Контейнер	**Бітова швидкість (приблизно)**	**Оптимізовано для**
ODU0	1.25 Гб/с	1GE
ODU1	2.7 Гб/с	2.5G (OC-48/STM-16)
ODU2	10 Гб/с	10G, 10GE WAN PHY
ODU2e	10.3 Гб/с	10GE LAN PHY
ODU3	40.1 Гб/с	40G
ODU4	104 Гб/с	100G
ODUFlex	Ціле число трибутарних слотів TS OPU_k (OPU2, OPU3, OPU4)	Будь-яка бітова швидкість

Мережа OTN складається з декількох мережевих рівнів у чітко визначеній ієрархії, як показано на рисунку 3.1. Рівень послуг надає сервіси кінцевого користувача, такі як GbE, SONET, SDH, FC або будь-який інший протокол. Для асинхронних сервісів, таких як ESCON, GbE або FC, сервіс передається через перетворювач GFP.

Блок корисного навантаження оптичного каналу (OPU) містить всі часові інтервали в кадрі OTN. Блок даних оптичного каналу (ODU) забезпечує транспортні функції OPU на рівні тракту. Він представлений тільки заголовком ODU OH. Блок OTU складається з заголовка OTU OH і кінцевика OTU FEC, що містить код корекції помилок FEC. Починається кадр з невеликого поля вирівнювання кадру, необхідного для розпізнавання початку кадру. OTU забезпечує службову інформацію на рівні розділу для ODU і надає байти GCC0.

Фізичний рівень відображає OTU в систему мультиплексування по довжинах хвиль або WDM. OTN має таку ж ієрархію, як SONET/SDH. Оптичний канал (OCh) проходить між усім, що перетворює послугу в сигнал

OTU-1/OTU-2, як показано на рисунку 3.2. Секція оптичного мультиплексування (OMS) знаходиться між двома пристроями, які можуть мультиплексувати довжини хвиль у волокні. Секція оптичної передачі (OTS) - це оптоволокно між усім, що виконує оптичну функцію сигналу. Волоконні підсилювачі, леговані ербієм (EDFA), вважаються обладнанням для лінійного посилення.

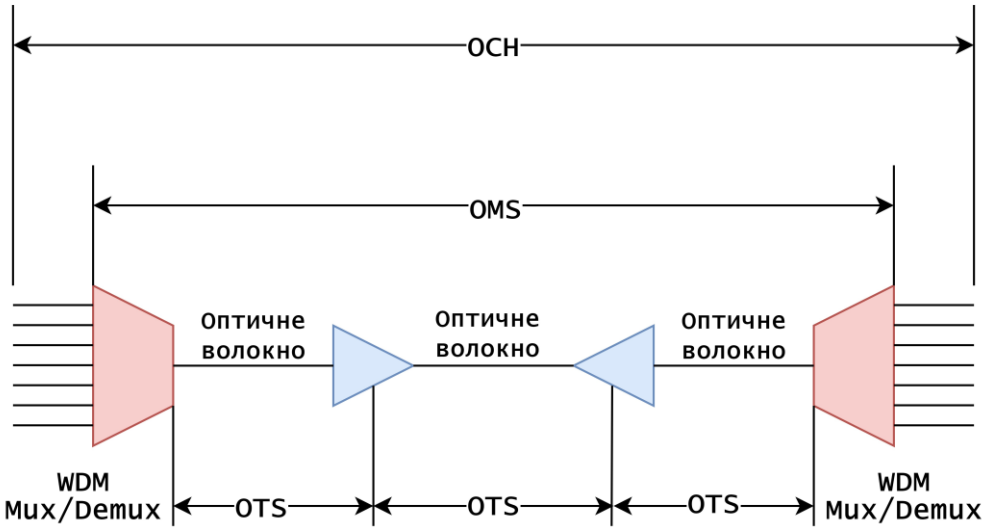

Рис. 3.2 Мережеві рівні OTN

OTN-магістралі і базові міські мережі забезпечують значні переваги в порівнянні з традиційними мережами на основі WDM, в тому числі підвищення ефективності та надійності і можливість надання приватних послуг на базі спектральних каналів. WDM використовується для передачі SDH/Ethernet/IP між регіонами або містами, або для агрегування трафіку там, де потрібна велика пропускна здатність. Інфраструктура IP на базі OTN також підвищує ефективність управління і моніторингу, скорочує число переходів, збільшує безпеку послуг і знижує витрати на обладнання. Крім нарощування швидкості до 100G і вище, OTN грає ключову роль в перетворенні мережі у відкриту і програмовану платформу, завдяки чому транспорт може стати такою ж важливою частиною інтелектуальної мережі ЦОД, як системи обчислення і зберігання.

OTN пропонує цілий ряд переваг, включаючи наступні:

- скорочення витрат на транспорт: завдяки можливості передачі декількох клієнтів на одній довжині хвилі OTN надає економічний механізм для заповнення спектральних каналів оптичних мереж;
- ефективне використання оптичного спектру: OTN забезпечує ефективне використання потужності каналів DWDM, підтримуючи узгоджену швидкість заповнення в масштабі мережі з використанням комутаторів OTN в оптичних переходах;
- детермінізм: OTN виділяє певну налаштовувану швидкість передачі для кожної послуги, групи послуг або сегмента мережі, гарантуючи для кожного клієнта певну пропускну здатність і продуктивність без конфліктів між одночасно працюючими послугами або користувачами;
- віртуалізовані мережеві операції: нові технології віртуалізації, такі як оптичні віртуальні приватні мережі, дозволяють надати клієнту виділений набір мережевих ресурсів, що не залежить від решти мережі;
- гнучкість: мережі OTN надають операторам можливість використовувати необхідні технології, дозволяючи адаптувати їх відповідно до вимог бізнесу;
- безпечна модель: мережі OTN забезпечують високий рівень конфіденційності і безпеки шляхом жорсткої розбивки трафіку по виділеним ланцюгам;
- надійність і простота операцій: адміністративні дані мережі OTN передаються по окремому каналу, який повністю ізольований від даних користувача додатків. Це означає, що через інтерфейсний порт клієнта набагато складніше отримати доступ до налаштувань мережі OTN і змінити їх.

3.2. OTCC

Проєкт OTCC спрямований на просування загальних інтерфейсів конфігурації і управління для транспортних мереж в SDN, визначаючи ці

інтерфейси за допомогою програмного забезпечення з відкритим вихідним кодом і програмно-визначених стандартів.

В рамках цього проєкту розробляється програмне забезпечення з відкритим вихідним кодом і супутні програмні стандарти для управління транспортними технологіями рівнів L0-L2+, включаючи оптичні та мікрохвильові. Рівень L0 - це рівень фотоніки, а точніше, технологія DWDM вважається рівнем L0 моделі OSI. [7]

Також планується розробка програмного забезпечення з відкритим вихідним кодом і супутніх програмних стандартів, що спрощують управління багатодоменними багатотехнологічними транспортними мережами постачальника послуг.

Проєкт OTCC координує свої дії з проєктами ONOS і ONF Packet-Optical для розробки і публікації стандартних інтерфейсів. Ця координація буде здійснюватися в двох напрямках:

- Сприяння впровадженню OTCC в ONOS;
- Використання методів та інструментів OTCC для моделювання інтерфейсів ONOS для створення моделей і API, які будуть опубліковані як програмно-визначені стандарти.

Використання цієї роботи постачальниками транспортного обладнання та партнерськими SDO (такими як OIF, MEF) в їх інформаційній моделі, розробці API та зусиллях по реалізації PoC/Interop сприятиме конвергенції галузі й уникненню непотрібної фрагментації в просторі транспортних API.

3.3 Транспортний API (T-API).

3.3.1 Загальні відомості про T-API

Проєкт T-API, розроблений згідно з ONF OTCC, був ініційований в 2014 році після успішної демонстрації прототипу архітектури транспортної SDN з декількома постачальниками, проведеної спільно з OIF. У 2019 році з’явився

детальний огляд на інтерфейс T-API. Цей проєкт не закінчений, над ним продовжують працювати.

T-API - це стандартний API, що дозволяє клієнту T-API (наприклад, платформі організації оператора або програмі замовника) отримувати інформацію з контрольованого домену транспортного мережного обладнання сервером T-API (наприклад, транспортний контролер SDN).

T-API розроблений як інтерфейс між контролерами на різних рівнях ієрархії SDN, пропонуючи контроль над мережевими ресурсами на різних рівнях абстракції. Типовим розгортанням T-API є інтерфейс між набором мережевих контролерів домену та оркестратором верхнього рівня. T-API розроблений, щоб дозволити мережевим операторам розгортати SDN на багатошаровій, багатодоменній та мультипостачальній транспортній інфраструктурі, розширюючи можливість програмування у своїх мережах наскрізно *(див.рис. 3.3)*.

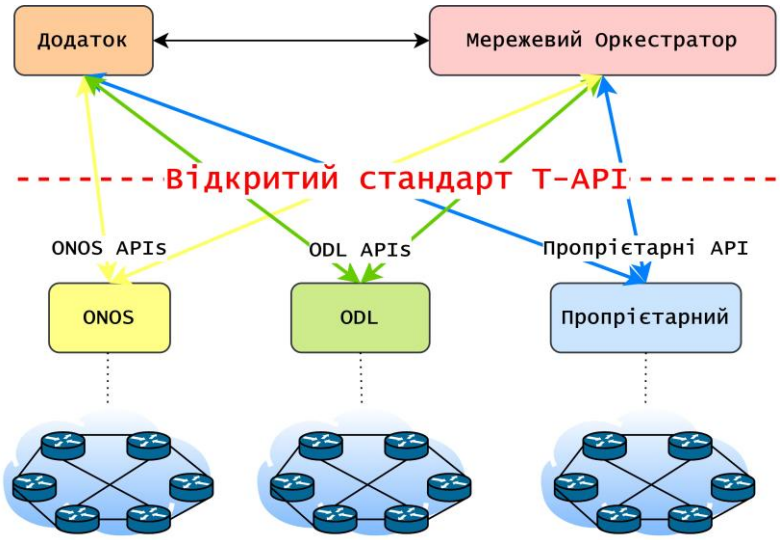

Рис. 3.3 Взаємодія оркестратора та контролерів за допомогою T-API

Існує набір сервісів, необхідних для управління транспортною мережею. Транспортна мережа має аналогічні служби підсистемі ONOS. [6]

- Топологія або отримання топології мережі.
- Наявність ресурсів.
- Інформація про стан.

- Послуга, що дозволяє клієнту створювати зв’язок між кінцевими точками служби.
- Обчислення шляху запитів для майбутніх служб зв’язку.
- Повідомлення, що надає інформацію про мережеві події та зміни.
- Віртуалізація або поділ мережі на окремі віртуальні розділи для конкретних клієнтів або додатків.

Цей набір реалізовано в ONF T-API такими способами, що дозволяють отримувати різний ступінь інформації та контролю на основі політики оператора. Даний діапазон дозволяє підтримувати різні ділові відносини через API, включаючи клієнтів, яким буде доступно обмежений перегляд мережі типу «чорний ящик» для внутрішніх додатків оператора *(див. рис. 3.4)*.

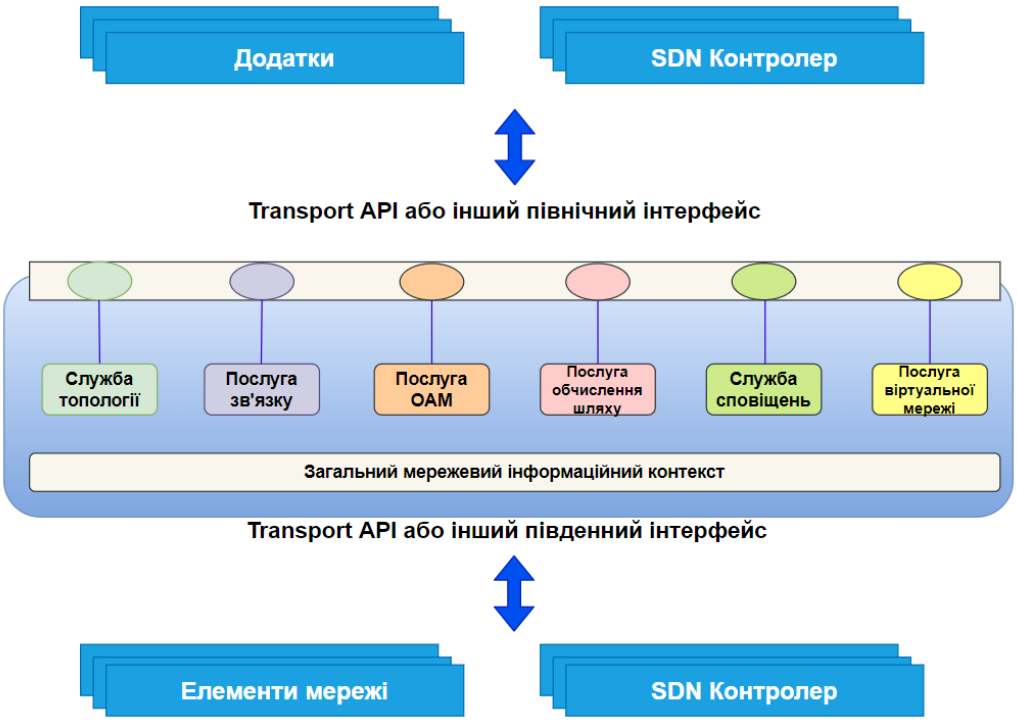

Рис. 3.4 Функціональна архітектура транспортного API

Робота T-API ONF базується на ONF CIM. ONF-CIM забезпечує представлення ресурсів площини даних з метою управління та контролю оператором. ONF T-API отримує свою інформаційну модель шляхом поділу і

рефакторингу ONF-CIM як цільової реалізації для управління транспортною мережею.

Потенційні програми T-API мають багато можливостей поєднувати управління та моніторинг оптичної транспортної мережі з додатками вищого рівня, такими як:

- Наскрізні послуги динамічної пропускної здатності через багатодоменну мережу несучих з підтримкою стійкості та повторної оптимізації;
- Взаємозв'язок декількох сайтів CORD через багатодоменну мережу;
- Підтримка послуг віртуальної транспортної мережі, що пропонують динамічно керовані та відстежувані віртуальні ресурси, які з'єднують віддалені сайти великих клієнтів. Підтримка поділу мережі, що забезпечує можливість підключення до 5G-сервісів з великою пропускною здатністю або наднизькою затримкою за допомогою ізольованих та захищених віртуальних підмножин мережі.

На рисунку 3.5 показано строго ієрархічну структуру мережі SDN, на верхньому рівні (додатків) знаходиться мережевий оркестратор, який здійснює керування нижчими рівнями ієрархії за допомогою додатків. У свою чергу мультидоменний контролер має зв'язок з іншими контролерами та керує групою доменних контролерів, які йому підпорядковуються. Основним інтерфейсом для зв’язку між усіма рівнями є транспортний програмний інтерфейс.

Також зображено операторів та вендорів, які зробили внесок у проєкт, а саме надання послуг зв’язку для тестування мережі та мережевого обладнання рівня інфраструктури.

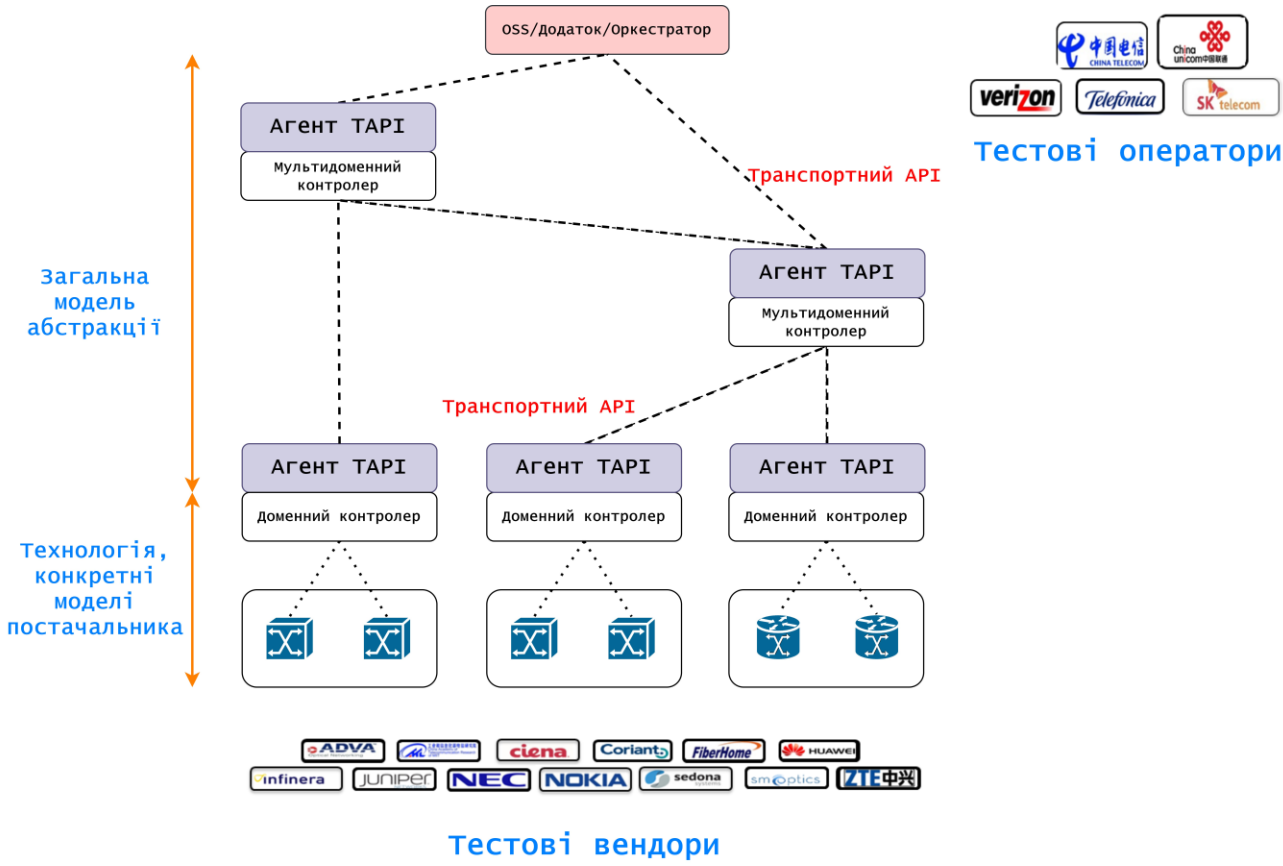

Рис. 3.5 Мультидоменна архітектура

3.3.2 Приклади використання

Прийняття T-API такими галузевими органами як:

– **MEF**

Модель T-API та SDK використовуються у проєктах моделювання мережевих ресурсів (NRM) та забезпечення мережевих ресурсів (NRP), де вони вдосконалюють T-API з розширеннями, специфічними для MEF. Потім MEF планує протестувати, продемонструвати та сертифікувати ці API як частину довідкових проєктів реалізації MEF OpenCS (наприклад, Optical Transport та OpenCS Packet WAN).

– **OIF**

OIF прийняв специфікації T-API як основу для тестування на сумісність транспортних рішень SDN у 2016 році та продовжував подальше тестування взаємодії у 2018 році.

- **ITU-T**

ITU-T узгодив свою нейтральну інформаційну модель протоколу з метою подання площини управління (Рекомендація ITU-T G.7718) з роботою CIM в ONF як основу для моделювання транспортних мереж для цілей управління.

3.3.3 Особливості T-API 1.0, T-API 2.0

Особливості T-API 1.0

Технічні характеристики T-API 1.0 та SDK мають такі функції:

- ***Служба топології***

Служба топології підтримує отримання інформації про топологію від контролера у вигляді деталей Node, Link & Edge-Point. Ця інформація може бути використана для обчислення шляхів, планування, аналізу та підтримує віртуалізацію мережевих ресурсів для певних клієнтських програм.

- ***Служба зв'язку***

Послуга підключення дозволяє клієнту отримувати інформацію та вимагати нову послугу підключення "точка-точка", "точка-багатоточка" та "багатоточка-багатоточка" через транспортну мережу. Включається підтримка як одношарових, так і багатошарових служб підключення. Зв'язок - це передбачений потенціал для переадресації (схема/пакет) між двома або більше крайніми точками вузла.

- ***Сповіщення***

Служба сповіщень дозволяє клієнтові підписатись і відфільтрувати автономні сповіщення від сервера про такі події, як зміна стану ресурсу чи

служби, збій або погіршення. Автономні повідомлення передаються за допомогою транспортного механізму, такого як веб-сокети.

– ***Обчислення шляху***

Служба обчислення шляхів дозволяє клієнту зробити запит на обчислення та оптимізацію шляхів.

– ***Служба віртуальної мережі***

Служба віртуальної мережі дозволяє клієнту створювати, оновлювати, видаляти топології віртуальної мережі. Клієнт вимагає підтримки матриці трафіку, що описує вимоги до трафіку між точками інтерфейсу клієнтського сервісу, а контролер реагує шляхом розподілу необхідних ресурсів у віртуальній мережі клієнта.

Особливості T-API 2.0

T-API 2.0 розширює T-API 1.0 за допомогою виправлень та вирівнювань на основі тестування взаємодії 2016 року, а також покращень моделі для підтримки більш складних обчислень шляху підключення, додаткових можливостей сповіщення, стійкості служби підключення та служб OAM.

– ***Виправлення та вирівнювання***

Протягом 2016 року тестування взаємодії в спільній демонстрації OIF/ONF T-API було зафіксовано кілька виправлень та вирівнювань до поточного використання YANG. Таких як формат списків атрибутів, використання регістру LISP , підтримка розширюваного переліку та інших. На основі виправлень модель T-API 2.0 YANG була протестована та перевірена кількома компіляторами YANG на правильність.

Крім того, модель Spec була вдосконалена як за допомогою спрощень, так і розширень, а також було внесено деякі оновлення імен та рефакторингу для покращення загальної інформаційної моделі T-API.

Отже, на основі спільних обговорень із MEF було зроблено ряд змін і розширень моделі, для уникнення створення проблем з термінологією та послугами. Одним з основних прикладів є те, що T-API 1.0 ServiceEndPoint (SEP) було перейменовано в ServiceInterfacePoint (SIP) у T-API 2.0.

– ***Обмеження та показники підключення вузлів***

Одним висновком тестування 2016 року було те, що початкова модель не надавала достатньо детальної інформації для обчислення шляху служби зв’язку між вузлами, особливо будь-яких показників зв’язку від порту до порту, пов’язаних з вузлом.

Завжди існував варіант виставити детальну суб-топологію всередині вузла, використовуючи ті самі вузли та конструкції, однак це в деяких випадках може додати небажаних витрат і складності, а в інших випадках може бути недостатньо для вираження певних типів обмежень.

Натомість T-API 2.0 додає можливість визначати правила, що стосуються портів або NodeEdgePoints у вузлі, як це виражено в наборі NodeRuleGroups. Ці NRG можуть ідентифікувати правила, що впливають на переадресацію, потужність, вартість, час чи ризик, і виражати обмеження між використанням NEP, пов'язаних з певною NRG. Крім того, можна висловити обмеження щодо переадресації між NRG, використовуючи InterRuleGroup.

Можна вказати NRG між кожною парою портів у вузлі, виражаючи просте правило переадресації МОЖЕ або НЕ МОЖЕ, та додатково обмежити на вартість або потужність між портами.

Використовуючи NodeRuleGroups для груп портів, можна висловити обмеження більш компактно або альтернативно.

– ***Автономне сповіщення про оновлення та телеметрію***

Для вказівки зміни стану T-API 1.0 забезпечував базову підтримку автономних повідомлень від контролера до клієнта. T-API 2.0 покращує ці можливості, визначаючи сигнали тривоги та попередження про перехід порогових значень.

Сигнали можна кваліфікувати за їх серйозністю, причиною та чи впливають вони на послуги, тоді як попередження про перетин порогових значень кваліфікують за параметром порогу та відповідним значенням.

- ***Стійкість та захищені зв'язки***

T-API 2.0 запроваджує підтримку стійкості з'єднання за допомогою захисту та відновлення. Модель стійкості використовує конструкцію Switch, яка представляє селектор переадресації та дозволяє змінювати її між альтернативними шляхами для досягнення стійкості. Також модель представляє елемент керування циклу, управління стійкістю, який контролює поведінку, оцінює її, ідентифікуючи необхідні зміни конфігурації, і застосовує їх для внесення необхідних коригувань до пересилання для досягнення запланованої поведінки відновлення.

- ***Послуги OAM***

T-API 2.0 представляє послуги OAM як особливість. Модель T-API 2.0 розширена, включаючи об'єкти, групи та кінцеві точки технічного обслуговування, проміжні точки обслуговування домену відповідно до стандартних визначень ITU-T та MEF, що дозволяють клієнту визначати, де можуть бути пункти моніторингу, а також починати, припиняти, вмикати та вимикати послуги вимірювання між визначеними точками з'єднання. OAM є дуже важливим компонентом надання послуги, яка відповідає угодам про рівень обслуговування, а також підтримує локалізацію та ізоляцію несправностей при їх виявленні. [8]

3.3.4 Майбутні релізи T-API

T-API 2.x:

- Оновлення фотонної моделі на основі відгуків ODTN;
- Узгодження ETH, L1 і OAM за рахунок взаємодії з проєктами MEF (NRM-OAM/SOAM, L1);
- Обладнання (фізичний інвентар).

T-API 3.0:

- Удосконалення пакетів топології/типів даних (ємність, вартість, затримка, параметри ризику);
- Покращення обмежень маршрутизації і стійкості;
- Удосконалення YANG і OpenAPI/RESTConf.

3.4. Топологічна модель T-API

Фізична модель мережі, представлена на рисунку 3.6, являє собою мережу з трьома мультиплексорами ROADM, оснащену матрицями OTN для виконання обробки клієнтських сигналів Ethernet 1G і 10G в інтерфейси 100G на стороні лінії, що передають кольорові оптичні хвилі OTSi.

OTN-комутація забезпечує ефективне використання смуги пропускання завдяки усуненню фрагментації і підтримує вищий рівень заповнення довжини хвилі при передачі трафіку. Використовуючи OTN-комутацію на основних вузлах, можна видалити з системи зустрічно-паралельні мультиплексори і скоротити число необхідних довжин хвиль. Впровадження OTN-комутації в точках установки оптичних мультиплексорів вводу-виводу (ROADM) забезпечує автоматичне угруповання сервісів і скорочення числа необхідних довжин хвиль завдяки використанню загальної ємності мережі.

OTN-комутатор дозволяє оператору управляти розподілом трафіку клієнтських портів по транспортним каналам. При цьому кожен клієнтський порт не пов'язаний жорстко з певним транспортним каналом: трафік можна перекидати на будь-який транспортний інтерфейс через систему управління, або взагалі налаштувати автоматичний перерозподіл трафіку в залежності від завантаження каналів.

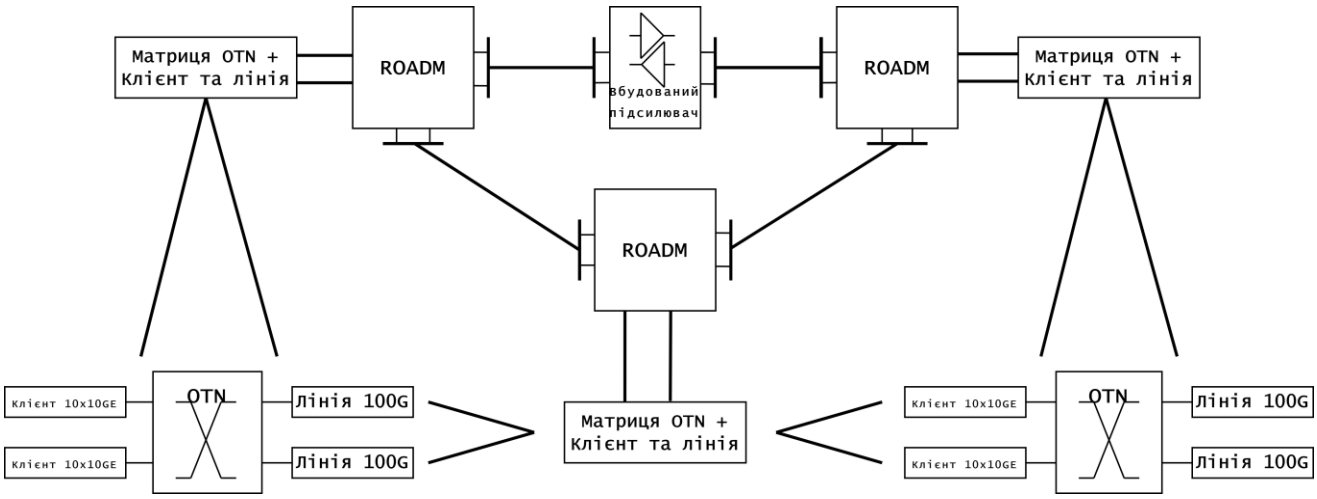

Рис. 3.6 Топологічна модель мережі

Багато компаній розгортають OTN для більш ефективного використання інвестицій в мережу шляхом збільшення пропускної здатності спектральних каналів. При цьому вони отримують додаткові переваги OTN, які дозволяють підвищити гнучкість мережі. Велика частка трафіку міської мережі залишається в зоні його виникнення, тому розміщення OTN-комутаторів ближче до границі мережі дозволяє уникнути передачі трафіку через ядро міської мережі і тим самим підвищує її загальну пропускну здатність. Додання технології OTN-комутації в існуючу транспортну мережу OTN є порівняно простим процесом, який забезпечує швидку окупність інвестицій. Після цього організації можуть перестати використовувати ручні оптоволоконні підключення для угруповання трафіку. Функція комутації також забезпечує більш ефективне й економічне управління каналами.

ROADM у волоконній оптиці є різновидом оптичного мультиплексора вводу-виводу, який дає можливість віддаленого перемикання трафіку з системи мультиплексування з поділом по довжині хвилі (WDM) на рівні довжини хвилі. Це досягається за рахунок використання модуля виборчого перемикання по довжині хвилі. Це дозволяє додавати і/або видаляти окремі або кілька довжин хвиль, що несуть канали даних, з транспортного волокна без необхідності перетворювати сигнали по всім каналам WDM в електронні сигнали і назад в оптичні сигнали.

Переваги ROADM:

- Планування всього призначення смуги пропускання не повинно виконуватися під час початкового розгортання системи. Конфігурація може бути виконана у міру необхідності, не зачіпаючи трафік,який вже проходить ROADM.
- ROADM забезпечує віддалене налаштування і зміну конфігурації.
- У ROADM, оскільки заздалегідь неясно, куди потенційно може бути направлений сигнал, існує необхідність в балансуванні потужності цих сигналів. ROADM забезпечує автоматичне балансування потужності.
- Функціональність ROADM спочатку з'явилася в обладнанні для щільного мультиплексування з поділом по довжині хвилі (DWDM). І до 2005 року вона почала з'являтися в оптичних системах великих міст (metro network) через необхідність побудови великих міських мереж для обробки трафіку, викликаного зростаючим попитом на пакетні послуги.
- Функції перемикання або реконфігурації ROADM можуть бути реалізовані з використанням різних технологій перемикання, включаючи мікроелектромеханічні системи (MEMS), рідкокристалічні, термооптичні і напрямні промені перемикачі в планарних хвилеводних схемах і технологію перебудованих оптичних фільтрів.

Вбудований підсилювач:

EDFA (Erbium Doped Fibre Amplifier) або підсилювач іншого типу, розміщується у лінії передачі для посилення ослабленого сигналу та для передачі на наступну, віддалену ділянку. Вбудований підсилювач - це повністю оптичний пристрій. Він встановлюється у волоконно-оптичному каналі кожні 80–100 км, як показано на рисунку 3.7, має помірний коефіцієнт підсилення та вихідну потужність, подібну потужності підсилювача.

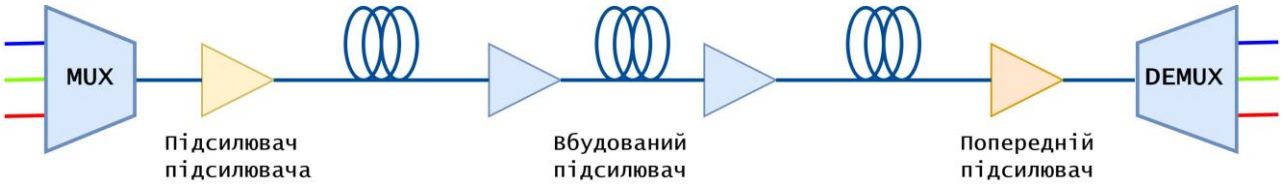

Рис. 3.7 Вбудований підсилювач

T-API надає інтерфейс для інтегрованого управління SDN на рівнях 0, 1 і 2. Новітня версія T-API включає в себе модель Photonic Media Model, що дозволяє отримати доступ до докладної інформації про мережеві фотонні можливості та конфігурацію, а також створити канали фотонних медіа по мережі. T-API 2.1 заснований на існуючій підтримці T-API для мережевих топологій, служб підключення і повідомлень, пов'язаних з продуктивністю мережі, подіями і сигналами тривоги. Завдяки новим фотонним можливостям, доданим до T-API 2.1, T-API тепер охоплює всі основні компоненти сучасних транспортних мереж, включаючи Ethernet, OTN і фотонну комутацію.

3.5 Інформаційна модель T-API

Інформаційна модель T-API заснована на єдиному поданні плоскої топології, яка об’єднує всі мережеві рівні в єдину топологію *(див. рис. 3.8.)*.

Вона включає:

- Домени переадресації ODU/DSR, представлені у вигляді багаторівневої і багатошвидкісної ***tapi-topology:node,*** що дозволяє представити внутрішнє відображення між DSR і ODU, і мультиплексування/демультиплексування на різних швидкостях ODU. Переходи ODU-OTSi представлені як транзитні ланки.
- Домени переадресації OTSi: представляють оптичну сторону оптичних терміналів (транспондери/мукспондери). Вони представлені однорівневою ***tapi-topology:node***, що дозволяє представити логічну агрегацію з'єднань OTSi в агрегацію OTSiA та відображати з'єднання OTSiA у з'єднання ODU. Зв'язок OTSI з вузлом Photonic-Media представлено як посилання OMS.

– Домени переадресації Photonic-Media: представляють мережу Photonic Media (Open Line System - OLS). Ці домени можуть бути співставленні з мережевими елементами OLP, ROADM/FOADM та ILA, зв'язок яких завжди представлено як посилання OMS або OTS. Ці домени переадресації надають можливість створення з'єднань Media Channel і служб зв'язку між його кінцевими точками.

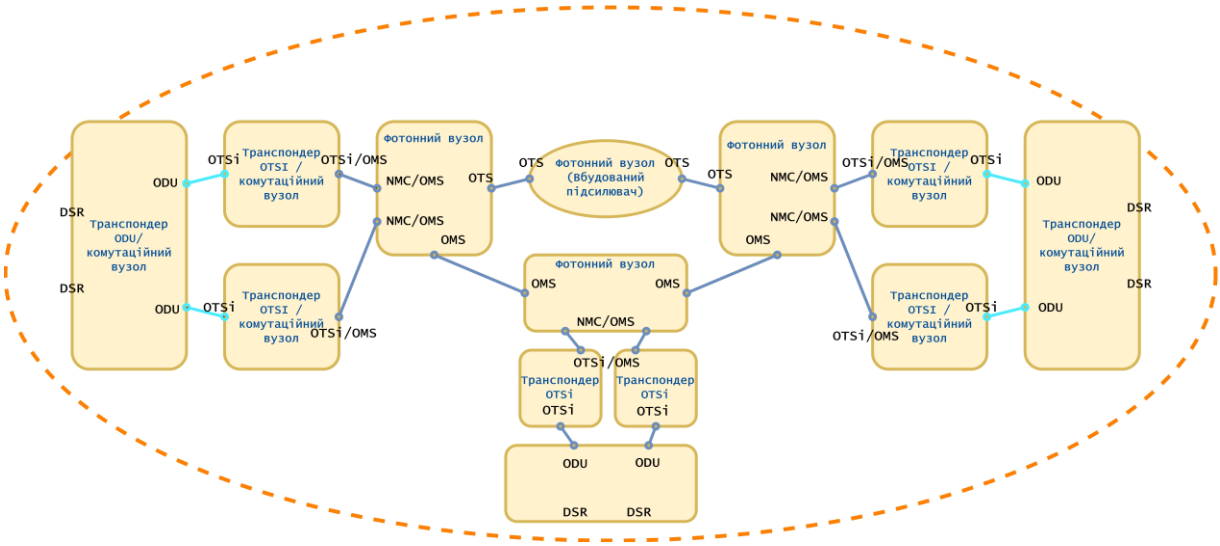

Рис. 3.8 Інформаційна модель T-API

Інформаційна модель T-API представляє собою чотирьохрівневу топологію, яка складається з таких рівнів *(див. рис. 3.9):*

– рівень клієнта DSR/ODU;
– рівень сервера DSR/ODU;
– рівень OTSi/OTSiA;
– рівень Photonic Media (медіаканали, OMS і OTS).

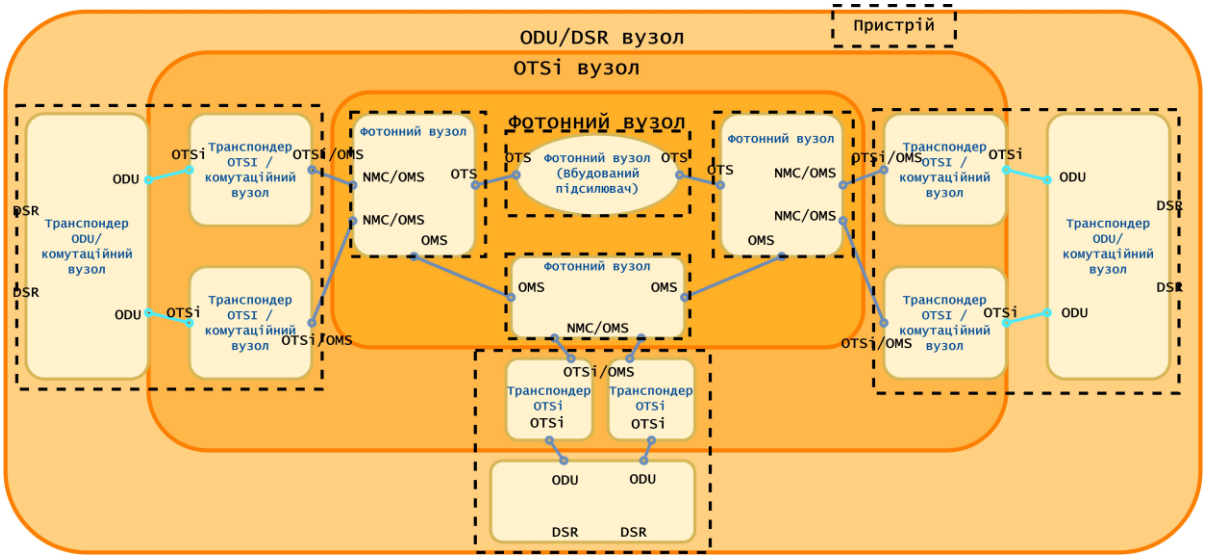

Рис. 3.9 Ієрархія інформаційної моделі T-API

Кожна топологія представлена як об'єкт ***tapi-topology: topology*** в контексті T-API.

Кожна топологія (за винятком абстрактної топології DSR/ODU, яка не має додаткового клієнтського рівня) представляє клієнтський рівень і можливість з'єднання з серверним рівнем через перехідні посилання. Топологія сервера представлена у вигляді абстракції з одним вузлом.

Принцип проєктування представляє кожен домен пересилання на кожному рівні як агрегований об'єкт ***tapi-topology: node***.

На кожному рівні включений агрегований вузол для подання потенційної пересилки між кінцевими точками клієнтської мережі на цьому рівні.

Кожен агрегований вузол включає посилання на топологію наступного рівня за допомогою атрибута ***tapi-topology:encap-topology***. [9]

Висновок до розділу 3

У ході виконання даного розділу було розглянуто концепцію OTN, на основі якої було створено проєкт ONF OTCC та ONF T-API, побудовано топологічну та інформаційну моделі транспортної мережі.

OTN інтегрує функції транспортування, мультиплексування, маршрутизації, управління, моніторингу та будує клієнтські з'єднання OTN

(наприклад, SONET/SDH, IP, ATM). У даний час OTN широко застосовується в метрополітені, регіональних і далекомагістральних DWDM-пакетно-оптичних транспортних мережах. Зазначимо, що відмінною характеристикою OTN мережі є її здатність транспортувати будь-який цифровий сигнал незалежно від специфіки клієнта. Це забезпечується тим, що границя OTN мережі розташовується між рівнем клієнта та оптичним каналом.

Проєкт ONF T-API, розроблений згідно з ONF OTCC, відповідальний за розробку SDK як проєкту з відкритим вихідним кодом. T-API є компонентом транспортної SDN, він дозволяє програмно керувати транспортною мережею для підтримки більш швидкого та гнучкого розподілу мережевих ресурсів для дотримання вимог додатків (наприклад, пропускна здатність або затримка).

Топологічна та інформаційна моделі пов’язані між собою, адже блоки інформаційної моделі чітко відповідають елементам топологічної. Топологічна модель має кільцеву топологію, що властиво для оптичних транспортних мереж, адже «кільце» має багато переваг.

Інформаційна модель побудована на основі мови моделювання YANG. Блоки, вузли, зв’язки програмно відображають дані про стан мережевих елементів.

ЗАГАЛЬНІ ВИСНОВКИ

У результаті виконання даної роботи було побудовано топологічну та інформаційну моделі ресурсів мережі SDN з використанням інтерфейсу T-API. Для цього було розглянуто основні положення та переваги програмно-конфігурованих мереж, проаналізовано їх архітектуру та детально описано протоколи та інтерфейси, які виступають у якості північних і південних. Досліджено рівень управління та операційну систему ONOS. Розглянуто концепцію OTN, на основі якої було створено проєкт ONF OTCC та ONF T-API.

SDN - це підхід, при якому для направлення трафіку в мережі і взаємодії з базовою апаратною інфраструктурою використовуються контролери на базі програмного забезпечення або API-інтерфейси. Така мережа відрізняється від традиційних, в яких для управління мережевим трафіком використовуються маршрутизатори і комутатори. SDN можна використовувати для створення та адміністрування віртуальної мережі або для управління звичайною апаратною мережею за допомогою програмного забезпечення. У той час як віртуалізація мережі забезпечує можливість поділу однієї фізичної мережі на різні віртуальні або об'єднувати пристрої з різних фізичних мереж в одну віртуальну.

Рівень управління включає операційну систему, яка забезпечує додаткам сервіси та програмний інтерфейс для управління пристроями та мережею. Особливість полягає у тому, що ONOS поєднує в собі функціонал SDN-контролера, мережевої і серверної ОС. За рахунок такої комбінації інструмент дозволяє стежити за всім, що відбувається в мережах, і спрощує міграцію від традиційної архітектури до SDN.

Проєкт ONF T-API, розроблений згідно з ONF OTCC, започаткований після успішної демонстрації прототипу багатопрофільних носіїв транспортної архітектури SDN спільно з OIF. Інтерфейс T-API використовується у транспортних оптичних мережах. Тобто, він є компонентом транспортного SDN, який дозволяє програмно керувати транспортною мережею для підтримки більш швидкого та гнучкого розподілу мережевих ресурсів для підтримки

вимог додатків (наприклад, пропускна здатність або затримка). До переваг відносять зменшення вартості, завдяки спрощенню операцій, та затримок з впровадженням нового обладнання та послуг, а також можливість розробляти та пропонувати нові послуги, що приносять прибуток, такі як поділ мережі та віртуалізація для додатків 5G та IoT.

Для технічної реалізації SDN на транспортному рівні розглядаються технічні рішення і можливості сучасних транспортних мереж на всіх рівнях: рівні оптичної кабельної структури, DWDM волоконно-оптичних систем передачі з мультиплексуванням по довжині хвилі, OTN оптичних транспортних мереж, IP / MPLS технології багатопротокольної комутації по мітках. Отже, для того, щоб стала можлива технічна реалізація технології SDN необхідно впровадити на кожному рівні нові технології.

На основі топологічної моделі мережі здійснено побудову інформаційної моделі ресурсів. Адже, кожен блок топологічної відповідає інформаційній. При побудові використано мову YANG, яка має великі переваги при моделюванні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. Компания "MUC". Сетевые технологии SDN – Software Defined Networking [Електронний ресурс] / Компания "MUC". – 2015. – Режим доступу до ресурсу: https://habr.com/ru/company/muk/blog/251959/.
2. SDN controller(software-defined networking controller) [Електронний ресурс] – Режим доступу до ресурсу: https://searchnetworking.techtarget.com/definition/SDN-controller-software-defined-networking-controller.
3. ПОБУДОВА SDN-КОНТРОЛЕРА НА БАЗІ ВІДКРИТОЇ МЕРЕЖЕВОЇ ОПЕРАЦІЙНОЇ СИСТЕМИ ONOS // XV Міжнародна науково-технічна конференція "Перспективи телекомунікацій" ПТ-2021: Збірник матеріалів конференції / Н. І. Корнієнко, А. О. Романов. – Київ: КПІ ім. Ігоря Сікорського, 2021. – С. 95–97.
4. Open Network Operating System (ONOS) [Електронний ресурс] / [T. Vachuska, A. Campanella, C. Cascone та ін.]. – 2021. – Режим доступу до ресурсу: https://opennetworking.org/onos/.
5. Оптическая транспортная сеть / П. Литтлвуд, Ф. Масуд, Э. Фоллис. – Ганновер: Ciena, 2014. – 40 с.
6. GitHub ONF T-API [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://github.com/OpenNetworkingFoundation/tapi.
7. OTCC [Електронний ресурс] / I. Busi, G. Cazzaniga – Режим доступу до ресурсу: https://opennetworking.org/open-transport/.
8. T-API Overview [Електронний ресурс] // Open Transport Configuration & Control. – 2019. – Режим доступу до ресурсу: https://wiki.opennetworking.org/display/OTCC/TAPI+Overview.
9. TAPI v2.1.3 Reference Implementation Agreement [Електронний ресурс] // Open Networking Foundation – Режим доступу до ресурсу: https://opennetworking.org/wp-content/uploads/2020/08/TR-547-TAPI-v2.1.3-Reference-Implementation-Agreement-1.pdf.