НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК _____

«ПОГОДЖЕНО»	**«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»**
Декан факультету інформаційних технологій	Завідувач кафедри комп'ютерних систем і мереж
Глазунова О.Г., д.п.н., професор	Лахно В.А., к.т.н. професор
_____	_____
___ _____ **2021 р.**	___ _____ **2021 р.**

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему «Дослідження засобів відтворення програмно-конфігурованої мережі»

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма «Інформаційні управляючі системи та технології»

Орієнтація освітньої програми освітньо-професійна

Керівник магістерської роботи

доцент, кандидат технічних наук _____ Сагун Андрій Вікторович
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав _____ Верба Ярослав Ігорович
(підпис) (ПІБ)

КИЇВ-2021

НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних систем та мереж
Лахно В.А.., д.т.н., доцент
_____
"___" _____ 20____ року

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Верба Ярослав Ігорович
(прізвище, ім'я, по батькові)

Спеціальність ____123 «Комп'ютерна інженерія»_____
Освітня програма _____ Інформаційні управляючі системи та технології _____
Орієнтація освітньої програми_____ освітньо-професійна _____
Тема магістерської кваліфікаційної роботи _____
__Дослідження засобів відтворення інфраструктури програмно-конфігурованої мережі ____ затверджена наказом ректора НУБіП України від "_23_" ___10____ 2020р. № 1578 «С»
Термін подання завершеної роботи на кафедру _____13.12.2021 _____
Вихідні дані до магістерської кваліфікаційної роботи _____
_____
_____
_____
_____

Перелік питань, що підлягають дослідженню:

№ з/п	Питання, що підлягає дослідженню	Строк виконання	Примітка
1.	Аналіз предметної області.	17.02.2021	
2.	Дослідження технології SDN	23.04.2021	
3.	Проектування топології мережі	15.07.2021	
4.	Проведення тестування топології	21.09.2021	
5.	Попередній захист	30.11.2021	
6.	Захист	15.12.2021	

Дата видачі завдання "__19__"_____жовтня____2020_ р.

Керівник магістерської кваліфікаційної роботи _____ Сагун А.В.
(підпис) (прізвище та ініціали)

Завдання прийняв до виконання _____ Верба Я.І.
(підпис) (прізвище та ініціали студента)

РЕФЕРАТ

Дипломний проект містить: 83 сторінок, 22 рисунки,2 додатока, 9 джерел

ПРОГРАМНО-КОНФІГУРОВАНА МЕРЕЖА,АРХІТЕКТУРА МЕРЕЖІ, ПЕРЕДАЧА ДАНИХ, УПРАВЛІННЯ ТРАФІКОМ

Метою проекту є проектування програмно-орієнтованої мережі для зниження капітальних та операційних витрат, сукупної вартості володіння мережею та збільшення шкидкодії передачі даних

У процесі виконання дипломного проекту були підняті навички проектування топології мережі та ознайомлення з новими протоколами.

В ході виконання дипломного проекту були спроектовані мережі дата-центрів, також було налагоджено SDN-контроллер.

Результатом роботи є розробка хмарного сховища данних мережі і побудована топологія мережі.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ONF - Open Networking Foundation

SDN - Software Defined Networking

EIGRP - Enhanced Interior Gateway Routing Protocol

VLAN - Virtual Local Area Network

VTP - VLAN Trunking Protocol

RSTP - Rapid spanning tree protocol

DNS - Domain Name System

HTTPS - HyperText Transfer Protocol Secure

ВСТУП

Безлік сучасних служб та додатків, особливо те, що пов'язані з хмарою, не працюють без SDN. SDN дозволяє легко переміщати дані між розподіленими середовищами, що дуже важливо для хмарних програм.

Крім того, SDN дозволяє швидко переміщати робочі навантаження через мережу. Наприклад, розбиття віртуальної мережі в розділи за допомогою віртуалізації мережевих функцій дозволяє операторам зв'язку переміщувати служби для замовників менш дорогі сервери або навіть на власні сервери замовників. Постачальники послуг можуть використовувати інфраструктуру віртуальної мережі для перенесення робочих навантажень із приватних хмарних інфраструктур у публічні при необхідності та для миттєвого надання нових служб замовникам. Крім того, SDN підвищує гнучкість і спрощує масштабування будь-якої мережі з додаванням або видаленням ВМ адміністраторами мережі, незалежно від того, чи є ці ВМ локальними або хмарними.

Нарешті, завдяки пропонованим рівням швидкості та гнучкості SDN може підтримувати нові тенденції та технології, наприклад обчислення на периметрі та Інтернет речей, яким потрібна швидка та зручна передача даних між віддаленими середовищами.

1.СИСТЕМА АРХІТЕКТУРИ ПРОГРАМНО КОНФІГУРОВАНИХ МЕРЕЖ

1.1Основні проблеми сучасної архітектури.

Програмно-конфігуровані мережі (Software Defined Networks, SDN) — одна з найновіших на сьогодні технологій реалізації мережі, при якому рівень управління мережею і рівень передачі даних поділяються за рахунок переносу функцій управління (виконуваних в традиційній мережі маршрутизаторами та комутаторами) на окремий центральний пристрій, що називається контролером , проте, незважаючи на те, що тема ще відносно нова, навколо неї вже сформувалося кілька полярних думок: від повного захоплення до скепсису.

Комп'ютерна мережа та основа інфраструктури — фактор розвитку сучасних мереж, архітектура мережі і принципи її побудови були закладені наприкінці 60-х років, тому зараз вона не здатна виконувати сучасні завдання і обробляти той обсяг інформації, який вимагають великі компанії, або доводиться будувати великі топології, що є ресурсозатратним. Збільшення кількості пристроїв з бездротовим підключенням та збільшення технологій їх підключення та керування, призвели до того, що на даний момент кількість таких користувачів перевищила кількість користувачів зі статичним підключенням. Але використання нових технологій мобільних терміналів веде за собою збільшення обчислювальної ємності додатків, що вимагає збільшення пропускної спроможності каналів зв'язку — обсяг трафіку зростає в геометричній прогресії, а вид трафіку стає все більш різноманітнішим. За даними виробників мережевого обладнання, кожні дев'ять місяців кількість трафіку збільшується у два рази, що призведе до збільшення навантаження у декілька разів через лічені роки.

Для вирішення питання навантаження бездротовими користувачами обладнання мережі повинно буду щільно розміщено, і якщо зробити соту невеликою, наблизивши клієнта до базової станції, то це збільшить пропускну спроможність соти та зменшить кількість користувачів у ній. Щоб реалізувати дану ідею, по оцінкам експертів, щільність покриття мережі треба збільшити в 20 разів, але сучасна архітектура погано пристосована для підтримки такого щільного покриття.

По-перше, неможливо постійно збільшувати щільність покриття — подібні станції треба буде розгортати всюди. По-друге, управління такою інфраструктурою має певні складності, окрім того вона зазнаватиме великих навантажень, взаємних впливів сот та інших факторів. По-третє, подібні топології мереж дуже дорогі в установці та підтримці.

Засоби побудови мереж сьогодні пропрієтарні, їхній основний функціонал реалізований апаратно та закритий для змін з боку власників мереж.

Зростання кількості та різнорідності контенту, розвиток сервісів та масштабів їхнього охоплення призвели до зміни парадигми організації обчислень — на місце клієнт-серверної архітектури прийшли хмари, а файлові системи та бази даних трансформувалися у мережі зберігання даних. Однак обсяг трафіку в Інтернеті за останні п'ять років зріс утричі, а пропускна спроможність сучасних каналів зв'язку при існуючих методах та засобах управління трафіком у мережах вже близька до вичерпання — нинішні темпи зростання пропускної спроможності мережі не в змозі задовольняти потреби користувачів. Починаючи з 2007 року щорічні темпи зростання пропускної спроможності мереж у всьому світі становили близько 60%, проте дослідження фахівців показують, що пропускну спроможність каналів зв'язку потрібно збільшувати вдвічі раз на два роки.

Одночасно зі зростанням кількісних показників навантаження на мережі ускладнилися завдання управління мережами — збільшився їхній перелік, значущість та критичність, причому на тлі підвищення вимог до безпеки та надійності. Мережі будуються на базі пристроїв, які постійно ускладнюються, оскільки змушені підтримувати все більше розподілених стандартних протоколів (сьогодні кількість протоколів, що активно використовуються, та їх версій перевищила 600), одночасно використовуючи закриті (пропрієтарні) інтерфейси. У таких умовах провайдери не можуть оперативно запроваджувати нові сервіси, а виробники мережного обладнання не можуть швидко модернізувати свої вироби для задоволення вимог замовників. Як наслідок, підтримка та управління складною мережевою інфраструктурою стали мистецтвом, а не інженерією, що частково підтверджується збільшенням кількості мережевих атак, вірусів та інших мережних загроз, що свідчать про те, що питання безпеки досі не мають надійних рішень.

У 70-80-ті роки в СРСР велися роботи над своїми стандартами та засобами побудови мереж. Найчастіше вони були несумісні зі стандартами, які ухвалювалися тим, що пізніше стало Інтернет Спільнотою ,зокрема, це призвело до того, що в країні не виникло свого виробництва мережного обладнання.

Отже, можна назвати такі проблеми сучасних комп'ютерних мереж.

науково-технічні – сьогодні неможливо контролювати та надійно передбачати поведінку таких складних об'єктів, як глобальні комп'ютерні мережі;

економічні — мережі дорогі, складні та вимагають для свого обслуговування висококваліфікованих спеціалістів;

проблеми розвитку - в архітектурі сучасних мереж є суттєві бар'єри для експериментування та створення нових сервісів. Відповіддю на кризу комп'ютерних мереж стала поява принципово нового підходу до їх побудови - програмно-конфігурованих мереж.

1.2Особливості програмно конфигурованої мережі

За рахунок такого поділу контроль стану мережі та управління мережею логічно централізовано на контролері. Крім того, такий підхід дозволяє рівню управління абстрагуватися від фізичної мережевої інфраструктури рівня передачі даних, використовуючи деяке логічне уявлення мережі. ідеї, які закладалися в ПКМ, полягають в наступному:

Поділ рівня передачі та рівня управління даними.

Єдиний,уніфікований,незалежний від постачальника інтерфейс між рівнем управління та рівнем передачі даних.

Логічно централізований рівень управління даними.

Віртуалізація ресурсів мережі.

Архітектура ПКС має три рівні (згідно рисунку 1)

Рівень інфраструктури мережі включає в себе набір мережевих пристроїв (комутаторів, маршрутизаторів) і каналів передачі даних.

Рівень управління, на якому відстежується та підтримується глобальне уявлення мережі (ГПС). Під глобальним уявленням мережі розуміється топологія мережі та стан мережних пристроїв. Рівень керування надає програмний інтерфейс (API) для мережних програм.

Рівень мережевих додатків, у яких реалізуються різні функції управління мережею: управління потоками даних у мережі, управління безпекою, моніторинг трафіку, управління якістю сервісу, управління політиками тощо.

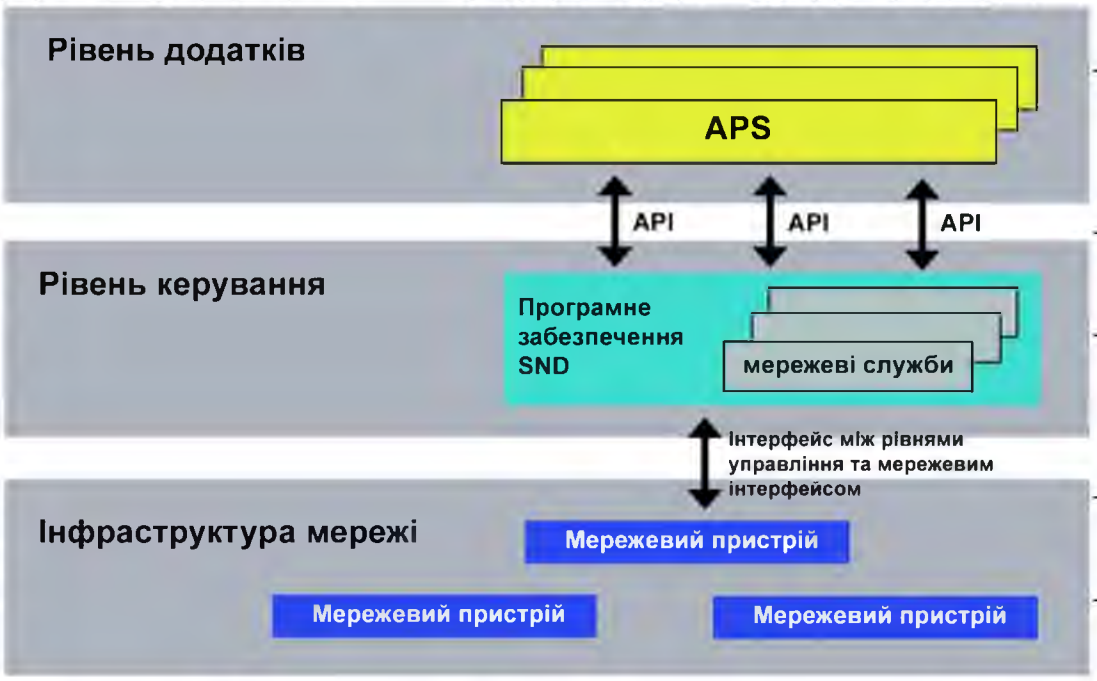

Рисунок 1.1 — Схема роботи

У SDN рівні керування мережею та передачі даних поділяються за рахунок перенесення функцій керування (маршрутизаторами, комутаторами тощо) у додатки, що працюють на окремому сервері (контролері). Така ідея була сформульована фахівцями університетів Стенфорда та Берклі у 2006 році. Це інноваційне дослідження було підримане не тільки в академічних колах, але и вплинули на провідних виробників мережвого обладнання. У 2011 році Google , Deautsche Telekom, Facebook,Mierosoft ,Verizon, Yahoo створили Open Networking Foundation (ONF). Після чого до ONF приєдналися ще велика кількість компаній. Першим – хто запровадив систему SDN була компания Nicira, яка після увійшла до складу VMware.

Така зацікавленість великих компаній пояснюється збільшенням продуктивності мереж в 20-30% при використанні СДН, окрім це економія на експлуатацію обладнання становить 30%, змінити управлення мережею з мистецтва на інженерію,

поліпшити захист інформації і дати користувачам можливість програмно створювати нові сервіси у мережеве обладнання.

Ключові частини у розробці та дослідженні SND пов'язані з програмною GENI (Global Environment for Network Innovations), яка активно розвивається в США. Близько 40 спеціалізованих університетів беруть участь у цій програмі та виконують дослідження та розробки у сфері Internet2.

Першу велику мережу SDN було реалізовано в 2012 р. компанією Google на базі комутаторів власної розробки. Таким чином їй вдалося зняти ті обмеження, які притаманні рішенням традиційних операторів. Трафік перенаправляється між ЦОДами так, як це зручно та вигідно в даний момент. Крім Google, технологію SDN використовують фірми NTT, Pertino, AT&T, Telecom Italia та інших компаній.

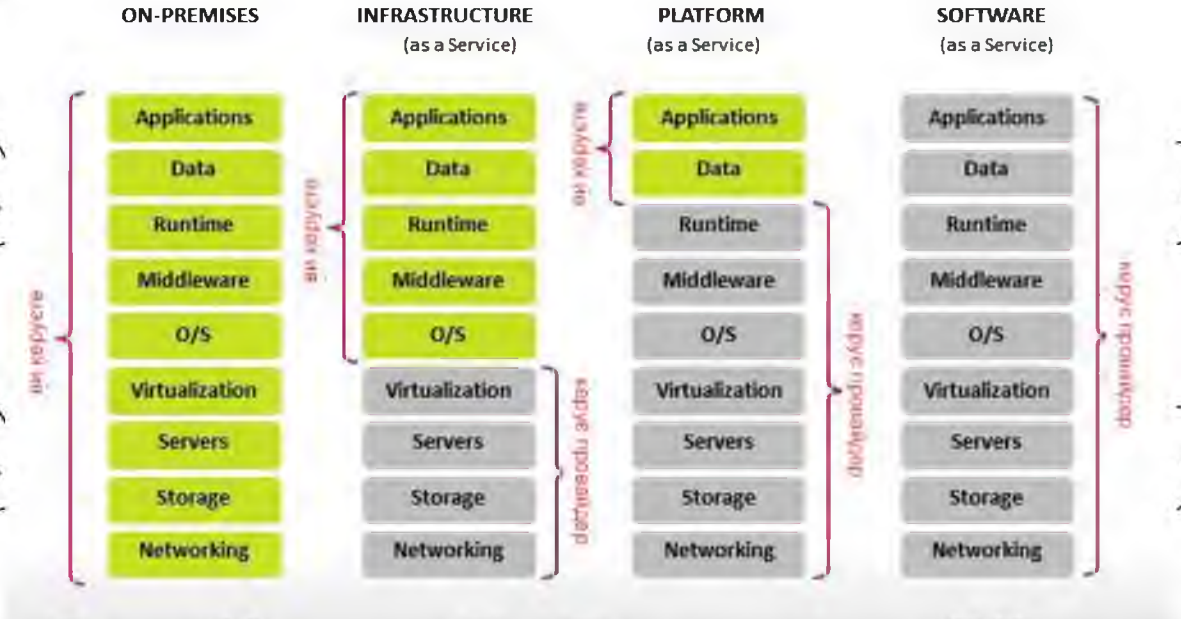

Рисунок 1.2 — Хмарне сховище за технологією SDN

Складається два стратегічні напрями впровадження SDN, NFV та хмар. Перше пов'язане з підвищенням ефективності мережі та гнучкості послуг. Головна мета - зниження вартості експлуатації мережі та скорочення часу виходу на ринок. Друге націлене отримання переваг від поєднання нових бізнес-можливостей. Мета у разі інша — формування нових диференційованих хмарних сервісів і динамічне, залежить від поточного профілю попиту їх надання. Першим шляхом йдуть такі компанії, як німецька Deutsche Telekom і іспанська Telefonica, другим — японська NTT і американська AT&T.

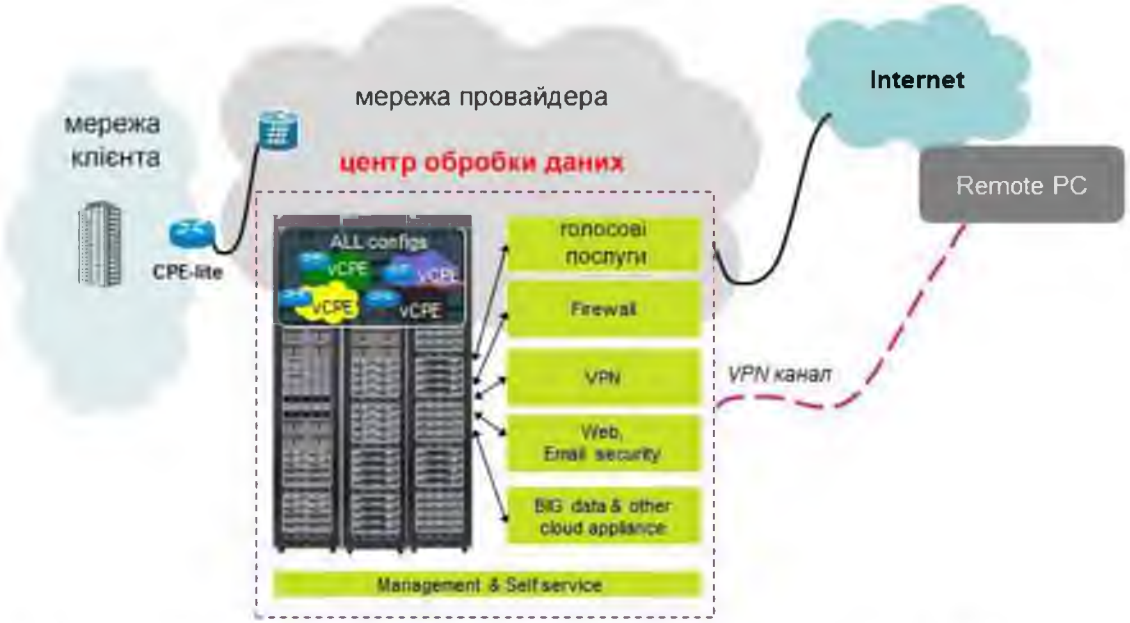

Рисунок 1.3 — Загальна схема роботи SDN

Практичний ефект від впровадження NFV/SDN для клієнта B2B

- Управління послугами з Особистого Кабінету
- Отримання мережевих функцій як послуг з чітким SLA.
- Зниження витрат на обслуговування власних мережевих функцій та IT систем за рахунок їхнього перенесення на «бік» оператора

- Отримання доступу до Послуг у режимі 24/7 навіть за зміни фізичного місця перебування офісу
- Отримання доступу до Послуг за мінімальний час при підключенні додаткового офісу
- Можливість тестової експлуатації послуги без необхідності її реалізації
- Практичний ефект від використання NFV/SDN для провайдера:
- Зниження вартості підключення – послуги віртуалізовані та не потребують виділеного обладнання
- Використання COTS обладнання (стандартне обладнання x86 архітектури)
- Скорочення або повне скасування виїздів до клієнта для підключення дод. послуг
- Доступ до Послуг у режимі 24/7 навіть при переїзді клієнта
- Зниження часу як на підключення нового клієнта, так і на додавання нових послуг
- Зниження вартості експлуатації
- Швидке та еластичне масштабування послуг залежно від потреб
- Уніфіковані включення на мережі – зниження кількості різнотипного кінцевого обладнання

Рисунок 1.4 — Обслуговування клієнтів за технологією SDN

Інтегратор як Сервіс Провайдер

- Відсутність витрат за виведення послуг ринку – платформа може бути розміщена у сторонньому Дата Центрі чи своїх серверах інтегратора розміщених у Оператора.
- Використання моделі Revenue Sharing: є клієнт – є прибуток, немає клієнта – немає витрат
- Накопичений досвід та експертиза дозволяють реалізовувати та виводити на ринок нові послуги за найкоротші терміни
- Гнучке використання як «вендорських» так і Open Source рішень для реалізації функцій мережі
- Підтримка та розвиток рішення знаходиться на боці інтегратора

Найбільш перспективним і активно розвивається стандартом для SDN є OpenFlow - відкритий стандарт, в якому описуються вимоги до комутатора, що підтримує протокол OpenFlow для віддаленого управління.

За допомогою сучасних маршрутизаторів зазвичай вирішуються дві основні завдання: передача даних (forwarding) - просування пакета від вхідного порту на певний вихідний порт і управління даними - обробка пакета та ухвалення рішення про те, куди його передавати далі, на основі поточного стану маршрутизатора. Це відповідає рівню передачі даних, на якому зібрані засоби передачі (лінії зв'язку, комутаційне обладнання, маршрутизатори, комутатори) та рівню управління станами засобів передачі даних (рис. 2). Розвиток маршрутизаторів досі йшло шляхом зближення цих рівнів, проте з ухилом на передачу (апаратне прискорення, вдосконалення ПЗ та впровадження нових функціональних можливостей для збільшення швидкості прийняття рішення щодо маршрутизації кожного пакета), тоді як рівень управління залишався досить примітивним і спирався на складні розподілені алгоритми маршрутизації та хитромудрі інструкції щодо конфігурування та налаштування мережі. Зрозуміло, ПЗ маршрутизаторів, що реалізує рівень управління, було пропрієтарним та закритим.

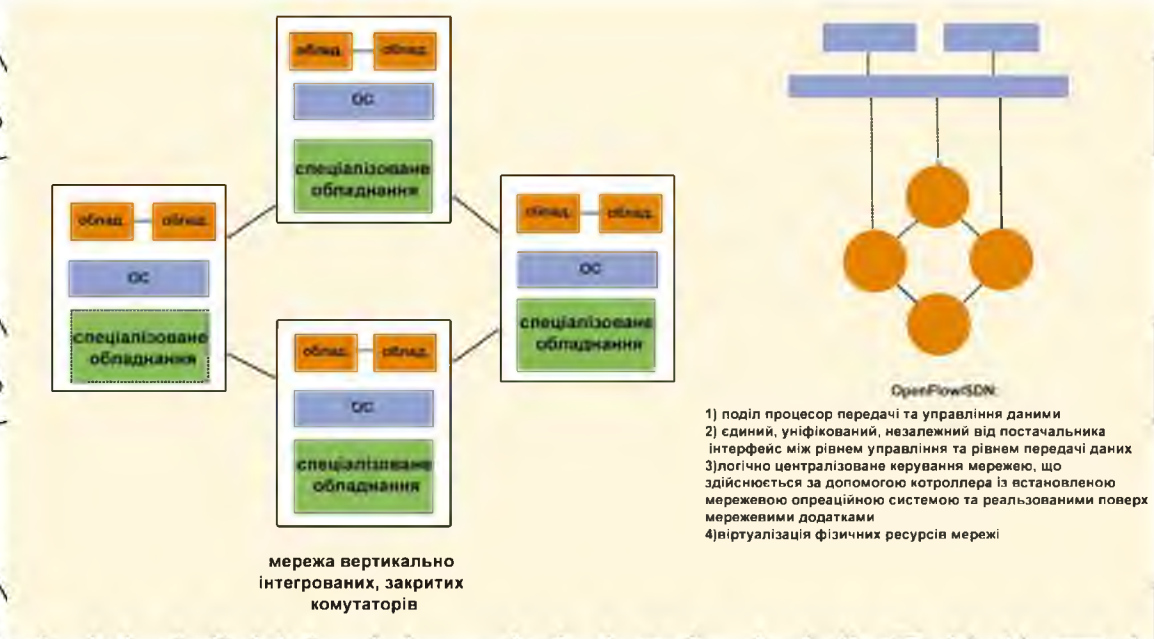

Рисунок 1.5 — Принципи роботи протокола OpenFlow

Відповідно до специфікації 1.3 стандарту OpenFlow, взаємодія контролера з комутатором здійснюється за допомогою протоколу OpenFlow – кожен комутатор повинен містити одну або більше таблиць потоків (flow tables), групову таблицю (group table) та підтримувати канал (OpenFlow channel) для зв'язку з віддаленим контролером - сервером. Специфікація не регламентує архітектуру контролера та API для його додатків. Кожна таблиця потоків у комутаторі містить набір записів (flow entries) про потоки чи правила. Кожен такий запис складається з полів-ознак (match fields), лічильників (coun-ters) та набору інструкцій (instructions).

Механізм роботи комутатора OpenFlow досить простий. У кожного пакету «вирізається» заголовок (бітовий рядок певної довжини). Для цього бітового рядка в таблицях потоків, починаючи з першого, шукається правило, у якого поле ознак найближче відповідає (збігається) заголовку пакета. За наявності збігу над пакетом і його заголовком виконуються перетворення, що визначаються набором інструкцій, зазначених у знайденому правилі. Інструкції, асоційовані з кожним записом таблиці,

описують дії, пов'язані з пересиланням пакета, модифікацією його заголовка, обробкою в таблиці груп, обробкою в конвеєрі та пересиланням пакета на певний порт комутатора. Інструкції конвеєра обробки дозволяють пересилати пакети в наступні таблиці для подальшої обробки та у вигляді метаданих передавати інформацію між таблицями. Інструкції також визначають правила модифікації лічильників, які можуть бути використані для збирання різноманітної статистики.

Якщо потрібного правила в першій таблиці не виявлено, то пакет інкапсулюється і відправляється контролеру, який формує відповідне правило для пакетів даного типу і встановлює його на комутаторі (або на наборі керованих ним комутаторів) або пакет може бути скинутий (залежно від конфігурації комутатора).

Запис про потік може наказувати переслати пакет у певний порт (звичайний фізичний або віртуальний порт, призначений комутатором, або зарезервований віртуальний порт, встановлений специфікацією протоколу). Зарезервовані віртуальні порти можуть визначати загальні дії пересилання: відправка контролеру, широкомовне (лавинне) розсилання, пересилання без OpenFlow. Віртуальні порти, визначені комутатором, можуть точно визначити групи агрегування каналів, тунелі або інтерфейси зі зворотним зв'язком.

Записи про потоки можуть також зазначати групи, у яких визначається додаткова обробка. Групи являють собою набори дій для широкомовної розсилки, а також набори дій пересилання з більш складною семантикою, наприклад, швидка зміна маршруту або агрегування каналів. Механізм груп дозволяє ефективно змінювати загальні вихідні дії потоків. Таблиця груп містить записи про групи, що містять список контейнерів дій зі спеціальною семантикою, яка залежить від типу групи. Дії в одному або декількох контейнерах дій застосовуються до пакетів, що надсилаються до групи.

Розробники комутаторів можуть бути вільними у реалізації їх внутрішньої начинки, проте процедура перегляду пакетів та семантика інструкцій повинні бути

для всіх однакові. Наприклад, в той час як потік може використовувати всі групи для пересилання в кілька портів, розробник комутатора може вибрати для реалізації цього єдину бітову маску всередині апаратної таблиці маршрутизацій. Інший приклад - це процедура перегляду таблиць: конвеєр фізично може бути реалізований за допомогою різної кількості апаратних таблиць. Встановлення, оновлення та видалення правил у таблицях потоків комутатора здійснюються контролером. Правила можуть встановлюватися реактивно (у відповідь на пакети, що прийшли) або проактивно (заздалегідь, до приходу пакетів).

Управління даними в OpenFlow здійснюється не лише на рівні окремих пакетів, але в рівні їх потоків. Правило в комутаторі OpenFlow встановлюється з участю контролера лише першого пакета, та був інші пакети потоку його використовують.

Наявні на сьогоднішній день фізичні комутатори SDN відповідають поки що специфікації OpenFlow 1.0 і містять лише одну таблицю потоків.

Протокол OpenFlow

Ідея SDN про створення уніфікованого, незалежного від виробника мережного обладнання, програмно-керованого інтерфейсу між контролером та транспортним середовищем мережі знайшла відображення у протоколі OpenFlow, що дозволяє користувачам самим визначати та контролювати, хто з ким, за яких умов та з якістю може взаємодіяти в мережі. Протокол підтримує три типи повідомлень: контролер-комутатор, асинхронні та симетричні.

Повідомлення типу контролер-комутатор ініціюються контролером і використовуються для безпосереднього керування та стеження за станом комутатора. Повідомлення цього типу можуть використовуватися контролером для встановлення параметрів конфігурації комутатора, для збору статистики, додавання, видалення та модифікації записів у таблицях потоків.

Асинхронні повідомлення ініціюються комутатором для оповіщення контролера про мережні події (прибуття пакетів або видалення запису з таблиці за тайм-аутом) та зміни стану комутатора або помилки.

Симетричні повідомлення можуть ініціюватися комутатором або контролером без запиту і використовуються при встановленні з'єднання, а також при вимірюванні затримок, пропускної здатності з'єднання контролер-комутатор або для перевірки живучості з'єднання.

Мережева ОС

Логічно-централізоване управління даними у мережі передбачає винесення всіх функцій управління мережею окремий фізичний сервер, званий контролером, який перебуває у віданні адміністратора мережі. Контролер може керувати як одним, так і декількома OpenFlow-комутаторами і містить мережну операційну систему, що надає мережеві сервіси з низькорівневого управління мережею, сегментами мережі та станом мережевих елементів, а також програми, що здійснюють високорівневе управління мережею та потоками даних.

Мережева ОС (СОС) забезпечує додаткам доступу до управління мережею і постійно відстежує конфігурацію засобів мережі. На відміну від традиційного тлумачення терміна ОС, під СОС розуміється програмна система, що забезпечує моніторинг, доступ та управління ресурсами усієї мережі, а не її конкретного вузла.

Подібно до традиційної операційної системи, СОС забезпечує програмний інтерфейс для додатків управління мережею та реалізує механізми управління таблицями комутаторів: додавання, видалення, модифікацію правил і збір різноманітної статистики. Таким чином, фактично вирішення завдань управління мережею виконується за допомогою додатків, реалізованих на основі API мережної операційної системи, що дозволяють створювати додатки в термінах високорівневих абстракцій

(наприклад, ім'я користувача та ім'я хоста), а не низькорівневих параметрів конфігурації (наприклад, IP- та MAC -Адрес). Це дозволяє виконувати керуючі команди незалежно від базової топології мережі, проте вимагає, щоб СОС підтримувала відображення між абстракціями високого рівня і низькорівневими конфігураціями.

У кожному контролері є хоча б одна програма, яка управляє комутаторами, з'єднаними з цим контролером, і формує уявлення про топологію фізичної мережі, що знаходиться під управлінням контролера, тим самим централізуючи управління. Подання топології мережі включає топологію комутаторів, розташування користувачів і хостів та інших елементів і сервісів мережі. Уявлення також включає прив'язку між іменами і адресами, тому однією з найважливіших завдань, розв'язуваних СОС, є постійний моніторинг мережі. Таким чином, СОС дозволяє створювати додатки у вигляді централізованих програм, що використовують високорівневі імена, на основі таких алгоритмів, як, наприклад, алгоритм Дейкстри пошуку найкоротшого шляху у графі, замість складних розподілених алгоритмів на кшталт алгоритму Беллмана – Форда, у термінах низькорівневих адрес, які використовуються у сучасних маршрутизаторах.

На даний момент є 28 реалізацій мережевих ОС для програмно-визначуваних мереж: NOX, POX, Beacon, Maestro, Trema, BigSwitch, FloodLight та ін.

Для контролерів у SDN дуже важливою є вимога того, що всі додатки одного контролера в кожний момент часу повинні мати однакове уявлення про топологію мережі. Однак перехід від розподіленого управління мережею до централізованого має й низку недоліків. Наприклад, зниження надійності, стійкості до відмов, масштабованості.

Сьогодні отримали розвиток кілька підходів до побудови розподіленого масштабованого контролера: HyperFlow , Onix та Kandoo . Однак, згідно з результатами досліджень ЦПД КС, найбільш перспективним є альтернативний підхід (рис. 3).

Оскільки кожен контролер може бути з'єднаний з декількома комутаторами, а кожен комутатор - з декількома контролерами, то контролери, що керують одним і тим самим комутатором, можна об'єднати в груповий контролер (ГК). Усі контролери однієї й тієї ж ДК повинні мати узгоджене уявлення про топологію тієї частини мережі, до якої вони забезпечують доступ. Як видно із рис. 3, C1 – C3 – контролери, S1 – S4 – комутатори, а V1 – V3 – фрагменти мережі, до яких забезпечує доступ комутатор S1, S2, S3 відповідно. Тоді ДК1 утворюють контролери C1 і C2, ГК2 - C2 і C3, а всі додатки в ГК1 повинні мати узгоджене уявлення про топологію V1 і V2, всі додатки в ГК2 - про топологію V2 і V3. У разі виходу з ладу, наприклад, контролера C1 його може замінити C2, взявши він управління V1. Уявлення простан відповідної частини мережі контролери можуть узгоджувати через комутатор S4, або через S1, S2 і S3.

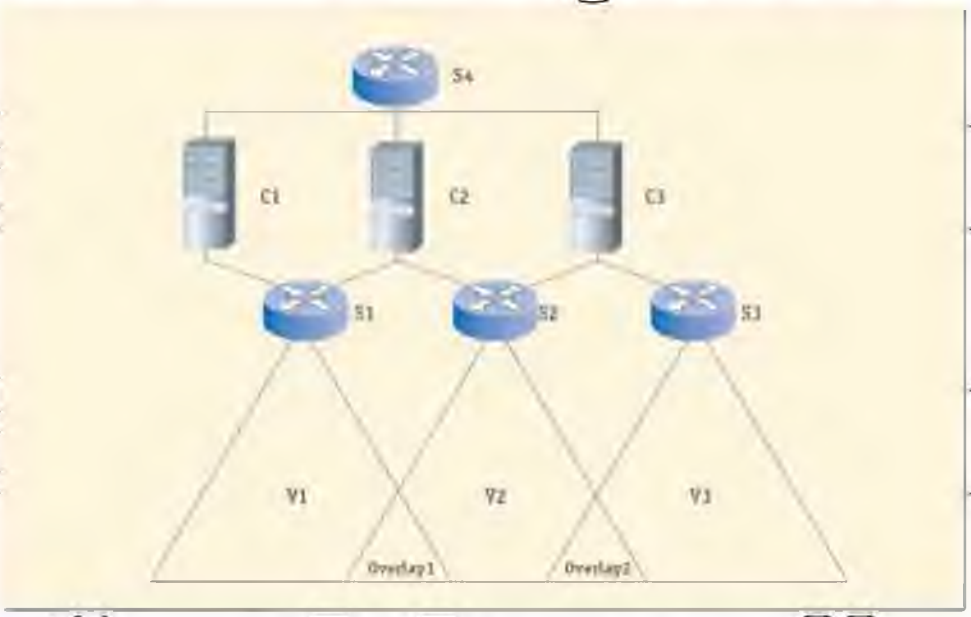

Рисунок 1.6 — Схема мережі з застосуванням OpenFlow

Альтернативный подход к построению распределенного масштабируемого контроллера

Такий підхід до побудови розподіленого контролера вирішує проблему масштабованості та підвищує відмовостійкість SDN.

Віртуалізація в SDN

Одна з ідей, що активно розвивається в рамках SDN, - це віртуалізація мереж з метою ефективнішого використання мережевих ресурсів (рис. 4). Під віртуалізацією мережі розуміється ізоляція мережного трафіку — групування (мультиплексування) кількох потоків даних із різними характеристиками у межах однієї логічної мережі, яка може розділяти єдину фізичну мережу коїться з іншими логічними мережами чи мережевими зрізами (network slices). Кожен такий зріз може використовувати свою адресацію, алгоритми маршрутизації, управління якістю сервісів тощо.

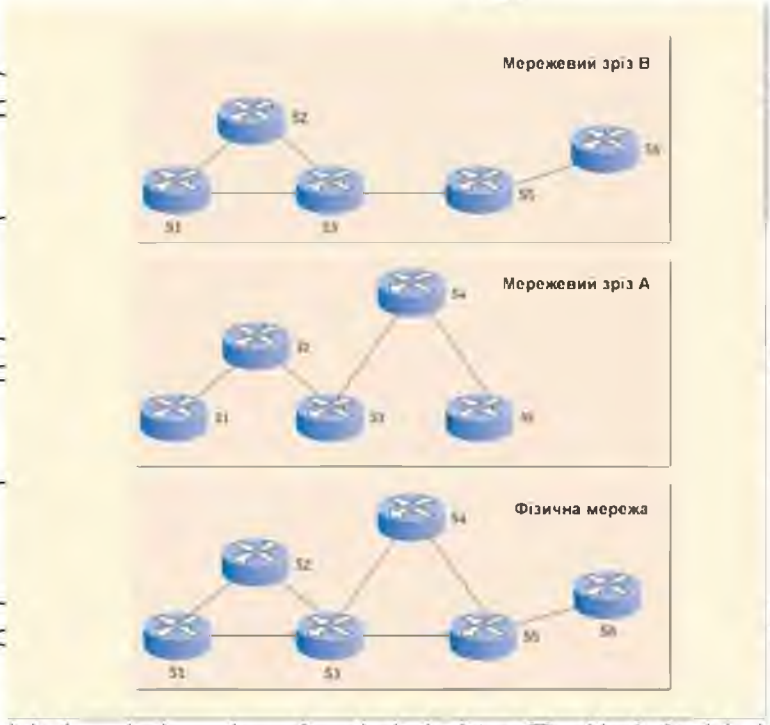

Рисунок 1.7 — Віртуалізація SDN

Віртуалізація мережі дозволяє: підвищити ефективність розподілу мережевих ресурсів та збалансувати навантаження на них; ізолювати потоки різних користувачів та додатків у рамках однієї фізичної мережі; адміністраторам різних зрізів використовувати свої політики маршрутизації та правила управління потоками даних; проводити експерименти у мережі, використовуючи реальну фізичну мережеву інфраструктуру; використовувати у кожному зрізі лише сервіси, які необхідні конкретним додаткам.

Одним із прикладів віртуалізації ресурсів SDN, поділу мережі на зрізи та управління ними є FlowVisor [6] — програма-посередник (proxy), що діє на рівні між OpenFlow-коммутаторами та різними контролерами SDN. За допомогою FlowVisor можна створювати логічні сегменти мережі, що використовують різні алгоритми керування потоками даних, забезпечуючи ізоляцію даних мереж один від одного. Це означає, що кожен контролер керує лише своєю логічною мережею і не може впливати на функціонування інших. Для контролера, що взаємодіє з обладнанням OpenFlow через FlowVisor, весь обмін повідомленнями виглядає так само, якби контролер взаємодіяв зі звичайною мережею SDN. Усю необхідну модифікацію повідомлень, потрібну для підтримки різних ізольованих сегментів мережі, виконує FlowVisor. Тобто для контролера логічної мережі не потрібно модифікації - це може бути будь-який контролер SDN, наприклад мережна операційна система NOX з довільним набором програм.

Переваги SDN

Завдяки зняттю з комутаторів навантаження з обробки тракту управління, SDN дозволяє цим пристроям направити всі свої ресурси на прискорення переміщення трафіку, що значно підвищує продуктивність. При цьому за рахунок віртуалізації управління мережею знижуються витрати на їх побудову та супровід. За результатами тестів, проведених на базі найбільших провайдерів США, використання SDN дозволяє на 20–30% збільшити утилізацію ресурсів ЦОД та у кілька разів знизити експлуатаційні витрати.

Програмні засоби SDN дозволяють адміністраторам додавати нову функціональність до вже наявної мережевої архітектури. При цьому нові функції працюватимуть на багатьох платформах - їх не доведеться реалізовувати наново у вбудованому програмному забезпеченні комутаторів кожного постачальника.

На централізованому контролері SDN системний адміністратор може спостерігати всю мережу в єдиному уявленні, за рахунок чого підвищуються зручність управління, безпека та спрощується виконання інших завдань. Дійсно, оскільки адміністратор бачить усі потоки трафіку, то йому легше помічати вторгнення, призначати пріоритети різним типам трафіку та розробляти правила реагування мережі при заторах та проблемах з обладнанням.

Теоретично необмежені можливості мереж SDN до розширення дозволяють будувати реальні хмари, що масштабуються в залежності від завдань, що вирішуються. При цьому мережа має необхідну «інтелектуальність», необхідну, зокрема, для оркестрування роботи великих груп комутаторів.

Перспективы SDN

Сьогодні на ринку рішень SDN помітні дві тенденції: спостерігається активна поява перспективних стартапів та відбувається орієнтація лідерів ринку ІКТ на SDN, що виражається у відкритті власних науково-дослідних підрозділів, що працюють на цю тематику, та окремих лінійок продуктів, заснованих на новому підході.

Перший комерційний проект у галузі SDN виконала у 2007 році компанія Nicira, заснована Ніком Маккеоном, Мартіном Касадою та Скоттом Шенкером. У Nicira розробили власну платформу віртуалізації мереж (Network Virtualization Platform, NVP), якою дуже швидко зацікавилися клієнти AT&T та NTT, а потім і такі компанії, як eBay, DreamHost, Fidelity Investments та Rackspace. В результаті кількох раундів інвестицій, у липні 2012 року компанія була куплена VMware, що започаткувало формування ринку рішень SDN.

Інша компанія - BigSwitch - була заснована професором Стенфордського університету Гуїдо Аппенцеллером і колишнім співробітником Cisco Кайлом Форстером, причому в рамках першого раунду інвестицій в компанію вклав свої кошти венчурний фонд Khosla Ventures, утворений Вінодом Кошла, співзасновником Sun

Microsystems. Аналітики вважають, що компанія BigSwitch, як і Nicira, незабаром отримає пропозицію про покупку від лідерів IT-ринку. Крім того, наприкінці липня 2012 року було оголошено про те, що Oracle досягла угоди про покупку компанії Xsigo Systems, яка займається розробкою програмного забезпечення для SDN.

Говорячи про другу тенденцію, слід зазначити, що ряд виробників вже мають готові до продажу власні рішення в галузі SDN. Наприклад, Cisco Systems, крім запуску лінійки комутаторів Nexus і Catalyst 35XX, здатних працювати в традиційних мережах та SDN, анонсувала платформу Open Network Environment (ONE), спеціально призначену для підтримки рішень SDN. Крім цього, компанія оголосила про розробку пілотної версії програмного забезпечення для контролерів, а також пілотну версію агента OpenFlow для збору відомостей про роботу мережевих інфраструктур SDN. Компанія Juniper Networks додала опцію OpenFlow в операційну систему JunOS SDK, а в червні оголосила про реалізацію цієї технології в лінійці комутаторів серій EX і MX. Компанії NEC, Pronto та Marvell пропонують комутатори, що реалізують лише протокол OpenFlow, а IBM випустила контролер IBM System Networking Programmable Network Controller як програмний додаток на Linux-платформі на основі OpenFlow. У HP реалізується стратегія HP Virtual Application Networks, що передбачає випуск контролера, програми, а також послуг та рішення на основі SDN, а Bro-cade представила перші продукти з підтримкою SDN, зокрема комутатор Brocade VDX 8770. До гонки приєдналася і компанія Intel, яка продемонструвала на IDF своє рішення для комутатора SDN та ПЗ з підтримкою протоколу OpenFlow на базі Linux.

У квітні 2012 Урс Хольце, старший віце-президент з технічної інфраструктури Google, заявив, що компанія перевела всю внутрішню мережу G-Scale для обміну трафіком між ЦОД Google по всьому світу на SDN, самостійно виготовивши комутатори OpenFlow, оскільки існуючі аналоги на ринку були на той час для компанії недоступні. Комутатори Google OpenFlow здатні масштабуватися до сотень портів 10-

Gigabit Ethernet, що не блокуються. Для Google використання SDN дозволило вибирати обладнання, яке суворо відповідає необхідному ПЗ, здійснювати централізоване управління мережею та потоками даних, оптимізувати процеси тестування та моніторингу.

Водночас говорити про формування повноцінного ринку рішень SDN поки що передчасно, проте, за оцінками аналітиків, до 2017 року він сформується і його обсяг може зрости до 2,1 млрд дол. проти 198 млн дол. у 2012 році. Основними рушійними силами цього ринку названо такі фактори, як зростаюча потреба в мобільності, потреба в новій мережній архітектурі при переході на хмарні послуги та використання різного виду трафіку. Головними локомотивами ринку SDN будуть поки що телекомунікаційні компанії, яким ця технологія дає гнучкість у наданні нових послуг та досягненні необхідної продуктивності.

1.3 Базові принципи побудови архітектури мережі

Для найшвидшого та найефективнішого розгортання SDN необхідно насамперед визначити проблеми, з якими фахівці можуть зіткнутися при впровадженні нових технологій.

По-перше, необхідно визначити цілі, які переслідує використання SDN. Керівники проектів повинні чітко розуміти, для чого це необхідно та забезпечувати координацію спільної роботи підрозділів, які у минулому, можливо, ніколи не взаємодіяли між собою.

По-друге, забезпечити новий підхід до моніторингу мережі. Оскільки мережа є єдиним, логічним цілим, особливі вимоги пред'являються синхронізації кінцевих пристроїв і контролера. Завдання моніторингу каналів зв'язку між компонентами SDN повинні бути вирішені за допомогою самого рішення SDN або сторонніми системами. До того ж моніторинг необхідний для пошуку місця появи тієї чи іншої проблеми, яка може виникнути на стику різних технологій.

Таким чином, перед початком розгортання SDN IT-фахівці повинні зрозуміти, яка інфраструктура вже є і що необхідно зробити, щоб системи, що лежать в її основі, могли швидко реагувати на зміни, пов'язані з впровадженням SDN-технологій.

Основні драйвери та стримуючі фактори впровадження

Вибір на користь SDN пояснюється багатьма причинами. По-перше, класичні підходи до вирішення проблем мереж на основі їхньої віртуалізації відстають від рівня розвитку віртуалізації серверів та СГД. Через війну мережі виявляються статичними і відповідають швидкої динаміці розвитку IT-бізнесу.

По-друге, при масштабуванні мереж з'являється велика кількість розподілених мережних пристроїв. У умовах, що змінилися, засоби традиційного управління стають великоваговими і неефективними.

По-третє, традиційна прив'язка до того чи іншого мережевого вендора, який заздалегідь опрацьовує необхідні заходи у разі тих чи інших трансформацій мережі, також виявляється неспроможною. Головна проблема – бізнесу не гарантується підтримка для майбутніх додатків та сервісів, що позбавляє його гнучкості при виборі майбутнього шляху розвитку.

- Щомісячний світовий IP-трафік у 2016 р. становитиме 110 ЕБ, CAGR = 32%, за оцінками аналітиків.
- На відео прийде 55% трафіку, на web – 23%, на обмін файлами – 21%, на голос – лише 1%.
- ARPU неухильно знижується.
- При цьому ростуть CAPEX та OPEX!

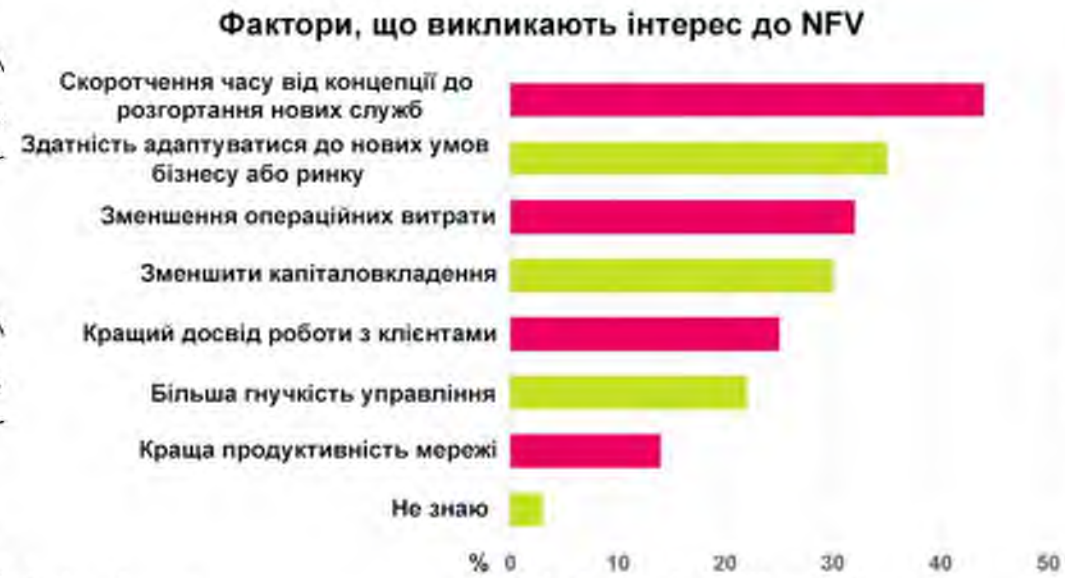

Рисунок 1.8 — Фактори,які викликают інтерес

До основних драйверів розвитку ринку SDN/NFV відносяться:

1. Зниження капітальних та операційних витрат, сукупної вартості володіння мережею

У разі мобільного зв'язку розмір економії на CAPEX особливо суттєвий у тому випадку, якщо оператор має досить розвинену оптоволоконну мережу. І тут компанія може скоротити капітальні витрати під час використання C-RAN до 60%. Інакше економія на CAPEX становитиме близько 30%. Скорочення CAPEX, зокрема, відбувається рахунок зменшення базових блоків (BBUs). З тієї ж причини відбувається зниження OPEX – за рахунок нижчого енергоспоживання[1] та зменшення витрат на обслуговування. За оцінкою China Mobile, "зелена" альтернатива у вигляді хмарних мереж радіодоступу зменшує рахунки на електроенергію на 71% порівняно з традиційними мережами.

За даними NEC, впровадження віртуалізованого пакетного ядра (vEPC) у рамках концепції NFV дозволить оператору мобільного зв'язку суттєво знизити сукупну вартість володіння (TCO).

Рисунок 1.9 — Економія при використанні SDN

Швидкість впровадження та адаптації послуг

Разом з тим, одним з основних факторів, що стримують для розвитку SDN – це відсутність єдиного стандарту і прагнення низки вендорів нав'язати ринку «своє» рішення, хоча такий підхід абсолютно суперечить основним принципам SDN. В результаті в SDN ще багато невизначеності, і потенційні споживачі програмно-конфігурованих рішень зайняли вичікувальну позицію, стежачи за появою успішних великих проектів у цій галузі.

Загальні цілі, що стимулюють розвиток напряму SDN, включають:

- гнучкість при створенні VPN, розподілі смуги пропускання та виділенні сегментів мережі;
- інтерфейси, що дозволяють користувачам вибирати стандартні мережеві шаблони;
- створення інтерфейсів систем;
- швидке виявлення та заміна з'єднань, що відмовили;

- потужний брандмауер між користувачами та зовнішнім світом;
- значне скорочення витрат людино-годин на управління мережею;
- автоматичне масштабування відповідно до розв'язуваних завдань та обсягу трафіку.

Обґрунтування економічного ефекту SDN все ще є видом мистецтва. Кількісний вимір нематеріальних речей, таких як вищий рівень безпеки або швидша реакція на вимогу внесення змін, — це складна справа. В результаті увагу волюють концентрувати на речах відчутних, таких як зниження витрат на обслуговування мережі, скорочення витрат на придбання продуктів і т. д.

Рисунок 1.10 — ЦОД SDN

Гетерогенність - реальність хмар, нею треба вміти керувати, тому наступний після хмар прорив в індустрії у бік підвищення ефективності буде неможливим без стандартизації. Для інтеграції та масштабування хмар не вистачає стандартів, єдиної, відкритої для всіх вендорів платформи, загального стеку технологій. Звідси випливає складність і висока вартість гібридних конфігурацій, зібраних з безлічі хмар. У разі програмного підходу можна створити відкриту платформу, побудовану за модульним принципом. Питання вже не в неможливості зробити це, а у потрібній кваліфікації

проектувальників ЦОД. Майбутнє — за програмно-визначуваними ЦОДами, які замінять портали Service Desk, автоматизувавши процеси керування. Шматкове управління мережею, пам'яттю, безпекою, обчисленнями та додатками буде замінено інтегрованим управлінням процесами доставки додатків на запит без втручання IT-служби.

Головна відмінність — традиційний ЦОД є набором апаратних пристроїв (сервери, мережні пристрої, системи зберігання даних, обчислювальні та програмні керуючі ресурси), тоді як SDDC вибудовується як надбудова над існуючою апаратною інфраструктурою, де всі підсистеми ЦОДу віртуалізовані та зібрані в захищену програмну систему. Налаштування, керування та обслуговування віртуальних компонентів ЦОД здійснюється програмним шляхом, потім необхідні команди автоматично переносяться на апаратні ресурси [8].

Завданням технології програмно-конфігурованих центрів обробки даних (SDDC), що стала популярною останнім часом, є покращення продуктивності центру за рахунок оптимізації на рівні додатків та гіпервізора. Проте аналітики Forrester вважають (осінь 2013 року), що при управлінні центрами слід прагнути до оптимізації на рівні конкретних бізнес-процесів — обробки фінансових даних, вирішення завдань постачання і так далі — а не на рівні окремих додатків, будь то ERP-системи, CRM , HCM та інші.

Навіщо переходити із традиційного ЦОДу на SDDC?

Головним системоутворюючим елементом у традиційному ЦОД часто називають комутатор. Цей пристрій відповідає за виконання трьох основних функцій: управління підключеними пристроями, управління трафіком і фізична передача даних.

При переході на програмно-визначувану модель ЦОД функції управління пристроями і трафіком централізуються і переводяться в програмну форму. Їхні команди

забезпечують злагоджену роботу всієї інфраструктури SDDC. Перед комутатора залишається лише функція передачі.

В результаті змін комутатор стає більш простим. Натомість ЦОД отримує додаткові можливості: спрощуються завдання масштабування інфраструктури, функції налаштування та управління стають гнучкішими, з'являються додаткові ресурси для роботи з прикладним навантаженням, оптимізації, виправлення помилок.

Перехід на SDDC дозволяє також отримати більш високу обчислювальну потужність, нарощувати ресурси зберігання даних та мережевої комутації, причому це досягається без виділення додаткової території під ЦОД або встановлення нових стійок.

Без збору актуальної інформації про роботу ЦОД немає руху вперед

Реальність та конкуренція на ринку ведуть до того, що власникам ЦОДів доводиться постійно оновлювати власний парк обладнання, нарощувати обчислювальні потужності та домагатися підвищення ефективності його управління. Для цього їм необхідно мати повну інформацію щодо поточної інфраструктури. Якщо набір зібраних даних виявляється неповним, приймати обґрунтовані рішення вкрай важко. Без ясного розуміння, що відбувається у ЦОДі, обладнання періодично потрапляє у стан простою.

З розвитком віртуалізації у ЦОДах вимоги до його обладнання також зростають. Для забезпечення доступності інфраструктури та контролю витрати обчислювальних ресурсів не обмежуються лише збиранням інформації. Потрібно отримувати її у повному обсязі та бути впевненим у актуальності зібраних даних. Якщо цих умов дотримано, можна проводити оптимізацію фізичної інфраструктури та переходити на програмне управління ЦОДом.

Зібрану інформацію про обладнання в традиційних ЦОД часто зберігають у вигляді електронних таблиць. Коли настає час для інвентаризації та модернізації, саме

з них починається пошук вільного місця. За цими «документами» звіряються резерви електричних потужностей, перевіряється достатність ресурсів для охолодження, наявність вільних портів для підключення.

Однак тепер такі способи збору даних стають все менш неефективними, особливо в умовах зростання популярності хмар та віртуалізації. При оновленні ЦОД з його оптимізацією необхідно використовувати спеціалізовані інструменти для збору та аналізу даних.

Програмно-визначені ЦОД (SDDC) з NFV пристроями

Рішення полягає в заміні апаратних компонентів інфраструктури ЦОД сукупністю x86 серверів, пов'язаних єдиним комунікаційним полем та утворюючих обчислювальне середовище, та набором Virtual appliances, що реалізують функціональність цих апаратних компонентів

На відміну від звичних всім ЦОДів, основою яких є «залізо» – сервери, системи зберігання даних, мережеві пристрої та ін. – SDDC є надбудовою над існуючою інфраструктурою, керування якою здійснюється програмним шляхом. А значить і захищати такі центри обробки даних треба інакше – традиційні рішення у сфері ІБ надто ресурсомісткі та гальмують роботу бізнес-додатків. Особливо це помітно в моменти сканування та оновлення антивірусних баз, що запускаються віртуальною машиною. У той же час, відключення віртуальної машини на тривалий період спричиняє появу слабкого місця в системі безпеки, оскільки встановлені на ній ІБ-компоненти перестають працювати.

Orange Business Services та AT&T розроблять стандарти SDN

Orange Business Services та AT&T підписали влітку 2016 року угоду про співпрацю в галузі розробки ініціатив з використання відкритого коду та стандартизації, які прискорять прийняття стандартів для технологій програмно-визначуваних

мереж (software-defined networking – SDN) та віртуалізації мережевих функцій (network function NFV). Компанії поділяють стратегічне бачення, згідно з яким не тільки обладнання, а й мережі мають ставати більш інтелектуальними, завдяки чому знижуватимуться витрати та складність експлуатації. Спільні зусилля партнерів наблизять появу більш маневрених, гнучких та оперативно реагують на потреби користувачів майбутніх мереж для індустрії та бізнес-замовників.

Розгортання нових віртуальних мережевих сервісів та функцій сьогодні надмірно ускладнене. Постачальникам мережевих послуг та іншим мережним компаніям доводиться мати справу з приватними стандартами, закритими архітектурами та обладнанням від багатьох різних постачальників, що орієнтуються на різні платформи та специфікації. AT&T та Orange організують обговорення проблем галузевої стандартизації, щоб разом рухатися до їх вирішення. Прийняття загальних стандартів та інтерфейсів допоможе індустрії спростити технологічну інтеграцію, підвищити операційну ефективність та знизити витрати, що прискорить процеси інновацій та розробки.

Коли технології SDN та NFV будуть засновані на загальних, відкритих та функціонально-сумісних технологічних стандартах, це допоможе подолати труднощі надання мережевих послуг з високим ступенем безпеки та власним інтелектом, що враховують особливості додатків, що використовуються. Поява екосистеми функціонально-сумісних сервісів та постачальників обладнання позитивно позначиться як на технологіях програмно-визначуваних мереж, так і на бізнес-замовниках, які зможуть швидше та простіше розгортати сервіси, налаштовувати їхню інфраструктуру в реальному масштабі часу та створювати інновації.

Взявши за основу сетецентричний підхід, AT&T та Orange мають намір зробити переваги свого бачення технологій SDN та NFV доступнішими як для бізнес-замовників, так і для індустрії. Компанії зосередяться на наступних завданнях:

Досягти того, щоб як телекомунікаційне обладнання, розташоване на території замовників, так і мережеві сервіси стали дійсно універсальними завдяки створенню загальних специфікацій мережевої інфраструктури і могли працювати в будь-яких середовищах програмно-визначених мереж з різним програмним забезпеченням.

Спростити та зробити більш ефективним процес впровадження технології NFV завдяки загальним рекомендаціям та шаблонам, які зроблять екосистему постачальників цієї технології зрілішою, а саму технологію – простішою у використанні.

Розробити стандартизовані інтерфейси прикладного програмування, які дозволять архітектурам програмно-визначуваних мереж різних постачальників взаємодіяти один з одним, роблячи розгортання віртуалізованих мережевих функцій та сервісів більш швидким та легким.

12 березня 2015 року стало відомо про плани компанії Huawei спільно з Open Network Operating System (ONOS) та фондом Open Networking Foundation (ONF) створення відкритої інноваційної галузевої екосистеми SDN, яка допомагає операторам підвищити прибутковість від мереж SDN.

Надшвидкісні з'єднання та курс на зниження витрат призвели до того, що програмно-визначені мережі (SDN) відіграли ключову роль у трансформації архітектури мереж операторів.

Про співпрацю компанії оголосили на прес-конференції, в рамках якої керівники обмінялися ідеями щодо розвитку відкритого, інноваційного середовища SDN та погодили процес створення платформи SDN з відкритим вихідним кодом.

На брифінгу компанія Huawei оголосила, що її рішення SDN повністю підтримують платформу ONOS. Huawei має намір підтримувати тісну взаємодію з ONOS,

ONF та Open Platform for NFV (відкрита платформа для NFV), спрямоване на створення єдиної, відкритої та програмованої мережевої архітектури SDN.

TechRepublic опитав навесні 2012 року 111 респондентів про нові технології, які вони збираються запровадити протягом наступних 12 місяців. Як видно на графіку нижче, SDN не є їх першочерговим вибором, але це не стало сюрпризом через те, що половина опитаних респондентів були незнайомі з SDN. Натомість серед тих, хто, як мінімум, щось чув про SDN, 64%, можливо, планують запровадити їх наступного року.

Багато організацій повідомили, що планують застосовувати SDN, але мало хто дійсно це зробив — лише 5% повідомили про впровадження, як видно з першого графіка. Цікаво, що 56% респондентів зацікавлені в SDN наступного року, значить малоймовірно, що це потрапить до їхніх бюджетів 2014р.

VMware придбала в 2012 році низку технологій та компаній з метою побудови стеку рішень для ЦОД майбутнього. Це, перш за все, куплений за досить велику суму в 1,26 млрд дол. забезпечення системи віртуалізації мережного обладнання Ще одне придбання - компанія DynamicOPs, що займається розробкою засобів автоматизації для хмар. Її продукти дозволяють контролювати процеси надання ресурсів та управління сервісами у гетерогенних середовищах: приватних та публічних хмарах, фізичних інфраструктурах, структурах на базі різних гіпервізорів та веб-сервісів. Все це в комплексі уможливлює наскрізне управління віртуальними, фізичними та мультихмарними середовищами при забезпеченні інтеграції з уже існуючими процесами та системами.

Nicira восени 2012 року готова бета-версія рішення для корпоративних ЦОД на базі віртуальних мереж (Virtual Extensible Local Area Network, VXLAN), що дозволяє масштабувати сегменти локальних мереж поверх мереж з хмар.

Обсяг світового ринку програмно-визначених інфраструктур (software-defined infrastructure, SDI) у 2020 році досяг $12,17 млрд, збільшившись на 5% порівняно з 2019-м. Такі дані 9 червня 2021 року оприлюднили в аналітичній компанії IDC.

У дослідженні зазначається, що динаміка виявилася нижчою порівняно з попередніми роками, але вищою щодо витрат на інші «ключові технології на тлі важкого пандемійного року». Основними сегментами ринку SDI аналітики вважають:

- програмно-визначувані обчислювальні рішення (software-defined compute /SDC/, їх частка у 2020 році - 53%);
- програмне зберігання даних (software-defined storage /SDS/, 36%);
- програмно-визначені мережі (software-defined networking /SDN/, 11%).

За словами директора з досліджень напряму Software-Defined Compute в компанії IDC Гері Чена (Gary Chen), обчислювальні технології, що програмно-визначаються, стали стандартом у дата-центрах завдяки віртуалізації серверів. Однак ринок продовжує розвиватися, і нещодавні проекти модернізації змістили зростання ринку у бік хмарних систем і контейнерів, зокрема, додав він.

Технології SDC забезпечують віртуалізацію груп фізичних обчислювальних модулів. Таке програмне забезпечення найчастіше продається з іншими інфраструктурними рішеннями, прикладними платформами та керуючим софтом.

Експерти зазначають, що підвищувати ефективність ЦОДу за рахунок підвищення ефективності його окремих компонентів стає все складніше через досягнення практичної стелі останніх. Боротьба тут іде за частки відсотка, і вести цю боротьбу стає економічно невигідно, якщо зважити на необхідні інвестиції. Натомість залишається потенціал підвищення ефективності об'єкта загалом за рахунок підвищення цифрової зв'язаності систем та вузлів, застосування спеціального програмного забезпечення, що дозволяє диригувати компонентами ЦОДу як злагодженим оркестром.

Наводиться приклад із системою охолодження в дата-центрі, можна отримати дуже відчутну економію, якщо програмно-апаратний «мозок» ЦОД управляє режимами роботи системи генерації холоду та електричної системи в залежності від діючого навантаження та ще прогнозує її зміни.

Обсяг ринку програмно-визначуваних мереж та дата-центрів - $51,7 млрд

У 2019 році обсяг світового ринку програмно-визначуваних мереж та дата-центрів (технології SDN, SD-WAN та SDDC) досяг $51,7 млрд. Про це свідчать дані аналітичної компанії MarketsandMarkets.

Експерти не уточнили динаміку щодо 2018 року, але кажуть, що ринок зростає і залишиться таким. Очікується, що продажі витрат на програмно-визначувані рішення у глобальному масштабі збільшуватимуться на 25,5% щорічно, а до 2024 року вони досягнуть $160,8 млрд.

За словами аналітиків, підйому ринку сприяє кілька факторів, серед яких збільшення попиту на віртуалізацію та хмарні обчислення в дата-центрах. Завдяки їм компанії отримують єдине керування всіма компонентами ЦОДів, такими як мережа, сервер, сховище, безпека та інші ресурси. Власники великомасштабних IT-інфраструктур, такі як постачальники хмарних послуг, оператори зв'язку та корпорації, все частіше використовують програмно-визначувані технології, зазначено в дослідженні.

Найсильніше піднімається попит на SD-WAN, і це багато в чому завдяки потребам бізнесу, що зростають, у простих засобах управління мережевим трафіком. SD-WAN допомагають роз'єднувати площини даних та керування ними, а також забезпечують централізоване керування при адмініструванні мережі.

Крім того, розвитку сегменту SD-WAN допомагають проекти цифрової трансформації, збільшення трафіку в мережах і технології, що розвиваються, на зразок 5G, інтернету речей і M2M-комунікацій.

Найшвидше витрати на програмно-визначені дата-центри та мережі зростають у телекомунікаційних операторів та хмарних провайдерів.

Експерти IDC підтверджують зростаючий попит на програмно-визначені технології. Наприклад, продаж рішень SDS (software-defined storage) збільшуватиметься на 13,5% щорічно і перевищить $16 млрд до 2021 року.

Аналітики пов'язують зростання сегмента з тенденцією переходу, що прискорюється, від традиційних IT-інфраструктур, які переважно використовують класичну архітектуру побудови масивів зберігання з двома контролерами, до хмарних середовищ на основі стандартного обладнання. Основними драйверами сегменту SDS у світі виступають три напрями: об'єктні та файлові сховища даних, а також гіперконвергентна інфраструктура (hyperconvergence infrastructure, HCI), зазначають у IDC.

Подальший розвиток світового ринку програмно-визначуваних технологій, як очікується, стимулюватиметься завдяки дедалі глибшому проникненню хмарних технологій у всі сфери, пов'язані з обробкою та зберіганням інформації. Ще одним важливим фактором, який вплине на розвиток сегмента у найближчому майбутньому, аналітики називають впровадження мереж мобільного зв'язку п'ятого покоління.

За прогнозами дослідників, мережі 5G призведуть до різкого зростання обсягів даних, їх обробка та зберігання вимагатиме нових підходів до автоматизації IT-процесів, які у свою чергу, найпростіше забезпечити за допомогою програмно визначених рішень. Що стосується розподілу по регіонах, то за прогнозами, більшу частину глобального ринку подібних систем займатиме Північна Америка (в основному США), за якою йдуть EMEA, Азіатсько-тихоокеанський регіон і американські країни.

У MarketsandMarkets зазначають, що США стали лідером за рахунок великомасштабних цифрових перетворень у компаніях різних розмірів. Темпи впровадження технологій на американському ринку настільки високі, що це допомагає організаціям

віртуалізувати свою IT-інфраструктуру та спрощує управління мережею, звідси і сплеск попиту на програмно-визначені технології.

Проектування топології мережі для побудови архітектури

Технологія SDN заснована на розділенні рівень керування з рівня даних (пристрої пересилання) для полегшення масштабованості мережі та управлінські операції. Однак процес виявлення топології мережі покладається на співпрацю між рівнями SDN-Controller і пристроїв пересилання, і відповідальність лежить насамперед на рівні контролю. Виявлення топології рівня керування.Завдання в основному складається з трьох основних операцій: виявлення SDN-Switch, виявлення посилання (тобто зв'язки між SDN-коммутаторами) та виявлення хосту .Оскільки нам потрібен OpenFlow, щоб діяти як конвеєр даних між керуючим рівнем і рівень даних, завантаження процесу виявлення топології повністю лежить на рівні контролера. Тому протокол OpenState був запропонований як модифікація OpenFlow намагається розділити навантаження процесу виявлення топології між SDN-контролером і SDN-перемикачі. В результаті можна безпосередньо програмувати перемикачі OpenFlow SDN, що дозволяє їм реалізовувати правила переспрямування, не покладаючись виключно на пульт дистанційного керування. Як майбутній покращений випуск OpenFlow, OpenState ще не розгорнуто.

Архітектура OpenFlow поєднує два шари за допомогою використання таблиць потоку в шар пристроїв пересилання. Крім того, кожен запис таблиці потоків містить три функції (правило, action, and statistics) і кожна таблиця потоків містить поля дій, які пов'язані з кожен вхід потоку. Дані та команди цих таблиць потоку передаються між двома шарами за допомогою каналу управління.

Відповідно до версії 1.5.1, кожен комутатор SDN повинен складатися з набору таблиць, а саме, таблиці потоків, таблиці відповідності та відсутньої таблиці, а також каналу керування для моніторингу зміни потоку в різних SDN-перемикачі через SDN-

контролер. Тому в SDN-OpenFlow процеси маршрутизації базуються на стандартних таблицях потоків, а не на адресах (тобто IP або MAC), як у традиційних мережах. Таблиця потоків використовується разом з логічною структурою даних, де пакети обробляються на основі списку пріоритетних записів у цих таблицях потоків. У кожному потоці можуть зберігатися до 15 полів у OpenFlow версії 1.10 , 5 з яких є обов'язковими, а решта необов'язковими.

Найпоширенішими полями є відповідність, дія, пріоритет, тайм-аут і лічильник. Більше того, SDN-контролер закріплює ці записи потоку в таблицю потоків двома підходами: реактивним і проактивним, а SDN-контролер визначає, який із них базується на настанні деяких подій. Коли починається мережева активність, SDN-контролер у реактивному підході не ініціалізує таблицю потоків жодними правилами. SDN-Controller буде вставляти правила в таблицю потоків щоразу, коли дані надходять на комутатори під час роботи мережі.

Для проактивного підходу, SDN-Controller закріпить записи потоку в таблицю потоків заздалегідь, коли мережа буде запущена. Вибір правил має важливе значення для оптимізації продуктивності мережі, особливо у великомасштабних мережах. Коли пакети досягають комутатора під час роботи мережі вхідний потік пакетів відповідає записам потоку в таблиці потоків. Якщо збіг не знайдено, SDN-коммутатор зателефонує до SDN-контролера, щоб запросити записи, щоб дозволити пакету досягти місця призначення. Це включає часті з'єднання між SDN-контролером і SDN-коммутатором, а також затримки перед тим, як пакет може бути переданий на наступний стрибок. Проактивна техніка була впроваджена, щоб зменшити кількість часу, необхідного для спілкування SDN-коммутаторів і SDN-контролерів.

Крім того, кожен SDN-коммутатор OpenFlow налаштовується з IP-адресою та номером порту TCP SDN-контролера. Таким чином, щоб приєднатися до мережі, OpenFlow SDNSwitch створює сеанс TCP за допомогою тристороннього рукостискання (SYN, SYN/ACK, ACK), щоб ініціювати зв'язок з SDN-контролером. Далі SDN-

Controller надсилає повідомлення OFPT_Features_Request до SDN-Switch із запитом його поточної конфігурації як адреси керування доступом до медіа (MAC-адреси) та мережевих інтерфейсів. Потім SDNSwitch відповість повідомленням OFPT_Features_Reply, яке містить запитану інформацію. SDN-Controller зберігає та використовує таку інформацію для майбутніх завдань керування мережею, включаючи повторну обробку виявлення топології. На малюнку 3 представлено процес встановлення виявлення SDN-Switch в мережах SDN.

Після процесу встановлення підключення OpenFlow SDN-Switch таблиці потоків будуть містити інформацію заголовка SDN-Switch та дії (відповідальні за надання команд). Як показано на малюнку 2, коли пакети передаються з площини даних на комутатор SDNSwitch, кожен вхідний пакет буде перевірятися з таблицями потоків, коли відповідний заголовок пакета зустрічається з одним у таблицях потоків конвеєра, правила, пов'язані з записом потоку буде запущено. Для кожного успішного збігу між записами вхідного пакета та таблиці потоків поле лічильника буде збільшено. Коли потік пакетів досягає таблиці вхідного потоку (вхід) і виконується збіг із записами потоку, якщо збігу не відбувається, він переходить до наступної таблиці потоку за допомогою інструкції GoTo-Table, а потім знову виконує збіг із записами потоку. Цей процес продовжується плавно до тих пір, поки не будуть закінчені всі таблиці потоків, і, отже, пакет буде розглядатися як miss_flow, якщо він не отримає відповідності в одній з таблиць потоків. Відповідно до інструкцій у записі miss_flow, він або скидає пакет, або повторно надсилає його до іншої таблиці потоків. Потоки пакетів через OpenFlow SDN-Switch в обох напрямках (вхідному та вихідному). У випадку, коли мережевий трафік є величезним і складним, багато невідомих потокових пакетів буде надходити до вузла пересилання, таким чином, він буде виробляти велику кількість пакетних повідомлень. З іншого боку, надсилання запиту потоку (повідомлення про введення пакетів) до SDNController для кожного невідомого пакета призведе до замишання SDN-Controller, оскільки SDNController повинен обчислити

правила пересилання для кожного нового пакета, а потім встановити його до таблиць потоків у всі вузли пересилання даних (SDN-перемикачі). Такий великий обсяг трафіку та накладні обчислювальні витрати призведуть до накладних витрат SDN-Controller і збільшать час, необхідний для розміщення правил потоку, впливаючи на ефективність і масштабованість мережі. Більше того, стверджується, що дата-центр з 4К сервером може обробляти до 200 000 потоків в секунду. Інше дослідження показало, що середня ширина потоку становила приблизно 20 пакетів на потік із затримкою між потоками менше 30 мілісекунд. Ці витрати дуже високі, але доступна пам'ять для зберігання записів надсилання обмежена. Однак записи для типових таблиць потоку OpenFlow SDN-Switch зберігалися в потрійній адресованій пам'яті вмісту (TCAM), типі високошвидкісної пам'яті, яка дозволяє шукати безперервний запис потоку в тактовому циклі. Хоча пошук TCAM швидкий, його ємність обмежена кількома тисячами записів. З іншого боку, збільшення розміру TCAM викликає додаткові проблеми, такі як вартість, і вимагатиме високого споживання енергії. Тому дослідники спробували оптимізувати графік потоків, щоб скористатися ним у повній мірі.

У процесі спілкування та обміну пакетами (вхідними та вихідними) у протоколі OpenFlow, а також коли OpenFlow SDN-Switch отримує пакет, який не відповідає жодному правилу потоку в його таблиці потоків. У цьому випадку одна з можливостей полягає в тому, що новий хост підключається до мережі. Новий хост починає надсилати пакети на SDN-Switch, потім OpenFlow SDN-Switch інкапсулює цей пакет з повідомленням Packet_In і надішле його до SDN-контролера. SDN-контролер, у свою чергу, використовуватиме це повідомлення Packet-In, щоб виявити хости в мережі, потім SDN-контролер витягне місцезнаходження хоста (тобто, до якого порту SDN-Switch він підключений), IP-адресу хоста, і його MAC-адресу з повідомлень Packet_In.

Метою процесу виявлення зв'язку є виявлення існуючих зв'язків між підключеними SDN-перемикачами OpenFlow, а також ефективне виявлення змін у топології

мережі. Додавання та видалення є одними з найпоширеніших прикладів зміни топології мережі. Видалення посилання відбувається, коли існуюче посилання видаляється (фізично) або доступ не вдається через інші причини. Видалення посилання також відбувається, коли наявний комутатор SDN видалено або не вдалося отримати доступ з різних інших причин. Більше того, процеси додавання посилань будуть схожі на процеси видалення. У всіх цих процесах зміна зв'язку безпосередньо пов'язана з контролером SDN.

У мережах SDN немає стандартного протоколу для виявлення зв'язків між SDN-коммутаторами, і більшість існуючих SDN-контролерів використовують протокол виявлення рівня каналів (LLDP), такий як OpenDaylight, Floodlight, POX, Beacon, Cisco Open SDN-Controller і Open Network Operating System (ONOS), щоб знайти ці посилання. LLDP вважається протоколом рівня 2 (канал даних), який використовується мережевими пристроями для їх ідентифікації, потужності та сусідніх пристроїв у локальній мережі (LAN) на основі IEEE 802. Крім того, кожне повідомлення про виявлення LLDP вбудовано в рівень -2 кадр, який є типом Ethernet і блоку даних, який називається (LLDPUD). Дані, отримані LLDP, поміщаються в базу даних управлінської інформації комутаторів SDN, які потім можна запитувати під час сканування вузлів мережі для отримання топології мережі за допомогою протоколу керування мережею. На пізнішому етапі NOX SDN-Controller реалізував процес розробки LLDP, щоб покращити виявлення зв'язку між SDN-перемикачами та створив першу версію протоколу OFPD. OFDP не залежить від централізованого керування, такого як LLDP (тобто комутатори автономно надсилають та отримують рекламні оголошення LLDP). Це протокол виявлення запиту-відповіді, який може надіслати повідомлення про вхід пакетів до контролера SDN, щоб отримати зібрану інформацію про виявлення. Тим не менш, OFDP використовує формат пакетів LLDP з невеликими модифікаціями і працює дещо іншим чином, ніж протокол LLDP для сумісності з архітектурою SDN, де логіка управління є центральною для SDNController. Таким чином, комутатори

SDN-OFDP не ініціюють рекламу LLDP, але SDN-контролер має повний контроль над процесом виявлення каналу. У таблиці 1 представлені основні відмінності між протоколами LLDP та OFDP.

Контролер SDN ініціює процес виявлення в OFDP, надсилаючи повідомлення про виявлення LLDP, інкапсульоване в повідомленні Packet Out, до пересилачів (батьківських комутаторів SDN), які безпосередньо пов'язані з OpenFlow за допомогою багатоадресної адреси. Коли пристрій пересилання отримує повідомлення-повідомлення, він заповнює всі свої порти повідомленням про виявлення LLDP, і єдиний комутатор SDN, який підтримує OpenFlow, оновлює свою таблицю OFDP. Щоб проілюструвати, як OFDP використовує рекламні повідомлення LLDP, LLDP розглядає кадр рівня 2, який складається із заголовка та корисного навантаження, як показано на малюнку 5. У частині заголовка кадру поле Ethertype встановлено на 0x88cc, а цільовий MAC-поле адреси встановлюється на багатоадресну адресу, як ми обговорювали в Таблиці 1. Поле Ethertype використовується перемикачами OpenFlow SDN, щоб відрізняти кадри LLDP від інших. Частина корисного навантаження називається LLDPDU, яка затінена сірим кольором. Корисне навантаження складається з ряду полів зі структурою Type-Length-Value (TLV) і закінчується полем «End of LLDPDU TLV». Деякі поля TLV в LLDPDU є обов'язковими, а інші – необов'язковими. Обов'язкові поля містять інформацію, яку SDN-коммутатор хоче повідомити своєму сусідові, а саме: Chassis-ID (який є унікальним ідентифікатором комутатора), Port ID (який є його вихідним портом) і час життя.

Тому, щоб проілюструвати функцію OFDP, ми пояснюємо взаємодію між секціями ODFP на основі один одного, як показано на малюнку 6. SDN-контролер надсилає кожен визначений період (тобто 10 с) пакет LLDP, інкапсульований із виходом пакету, повідомлення до кожного активного порту кожного SDN-коммутатора. Як показано на малюнку 6, комутатор SDN (s1) має три активних порти, що означає три пакети LLDP, і кожен з цих пакетів має TLV-ID порту та відповідно налаштований

ідентифікатор шасі. У протоколі OpenFlow, якщо SDN-контролер хоче надіслати пакет до OpenFlow SDN-Switch, він інкапсулює його у повідомлення PacketOut. Структура повідомлення Packet_Out містить поле, яке називається полем інструкції. Це поле відповідає за рішення, що SDN-коммутатор повинен робити в цьому пакеті. Таким чином, пакет LLDP отримується з усіх портів SDN-Switch, крім порту, підключеного до SDN-Controller. Цей порт зв'язується з критролером SDN через повідомлення OpenFlow Packet-In. Більше того, він використовується для збору та агрегації всієї інформації про порти в кожному SDN-коммутаторі в єдине повідомлення «Packet-In» та надсилання його до SDNController відповідно до проактивного правила, встановленого в таблицях потоків усіх комутаторів SDN. У сценарії, наведеному на малюнку 6, пакет LLDP на порту-ID 1 надсилається від SDN-Switch (s1) і отримується SDN-Switch (s2) також через порт 1.

Крім того, повідомлення Packet_Out має поле, яке називається інструкцією, яке налаштовано для пересилання інкапсульованого LLDP з відповідного порту на комутаторі SDN. Далі, комутатор SDN (s1), у свою чергу, отримуватиме повідомлення Packet-Out, декомпілює пакет LLDP із повідомлення Packet-Out і пересилає лише пакет LLDP, який вийшов із кожного відповідного порту (на основі MAC-адреса порту в LLDPDU). Коли SDN-Switch (s2) отримує пакет LLDP, він аналізує пакет LLDP, записує його ідентифікатор шасі SDNSwitch і додає Port-ID (тобто порт, через який комутатор SDN отримав пакет LLDP). Потім SDN-Switch (s2) інкапсулює пакет LLDP у повідомлення Packet_In і надішле його до SDN-контролера. SDN-контролер, у свою чергу, аналізує повідомлення Packet_In і виявить нові посилання, представлені відображенням між s1 і s2. Однак цей же метод повторюється для виявлення решти посилань у мережі.

Кожен логічний SDN-перемикач OpenFlow підключений до OpenFlow SDN-контролера через канал OpenFlow. SDN-Controller встановлює та підтримує SDNSwitch за допомогою цього посилання, збирає події від SDN-Switch та передає

дані від SDN-Switch до SDN-Controller. Канал управління SDN-Switch може обробляти один канал OpenFlow з одним контролером SDN або кілька каналів OpenFlow з кількома контролерами SDN, які спільно використовують управління SDN-Switch. Крім того, комунікаційне з'єднання між потоком даних і каналом OpenFlow керується незалежно, але воно повинно відповідати правилу протоколу OpenFlow SDN-Switching. Крім того, канал OpenFlow зазвичай працює через TCP і захищений за допомогою TLS.

Крім того, якщо комутатор SDN підключається до групи SDN-контролерів, оновлення статусу SDN-контролерів має бути надіслано на комутатор SDN лише з одним SDNController та іншими контролерами SDN в режимі очікування, якщо перший контролер зупиняється. . Крім того, комутатор SDN повинен забезпечувати покращену індикацію статусу контролера для всіх контролерів SDN, коли канал OpenFlow повторно підключається. Більше того, якщо SDNSwitch втрачає з’єднання з усіма контролерами SDN з різних причин, включаючи тайм-аути запиту echo, час очікування сеансу TLS або інші роз’єднання, він повинен переключитися в «Fail-Safe Mode» або «Fail-Standalone Mode», залежно від Конструкція та конфігурація SDN-перемикача. Єдина відмінність у поведінці перемикача SDN у безпечному режимі полягає в тому, що пакети та повідомлення, адресовані контролерам SDN, відкидаються. У Fail-Safe режимі записи потоку повинні закінчуватися відповідно до їх тайм-аутів, тоді як у “Failure Standalone” SDNSwitch використовує зарезервований порт OFPP_NORMAL для обробки всіх пакетів. Іншими словами, SDN-коммутатор діє як старий Ethernet-коммутатор або маршрутизатор SDN. Крім того, SDN-Switch може використовувати таблиці потоків будь-яким способом, коли він перебуває в «Fail-Standalone», і може видаляти, додавати або редагувати будь-який запис потоку. До речі, тільки гібридні комутатори SDN зазвичай мають режим Fail-Standalone.

Крім того, згідно з повільним каналом управління, значно зменшує пропускну здатність рівня даних і час відгуку, а також загрожує доступності мережі. Тому було

запропоновано кілька останніх робіт, які допомагають підтримувати доступність OpenFlow.

Однією з найважливіших послуг SDN-Controller є надання оновленої та всеосяжної топології мережі під його контролем. Усі мережеві програми повністю залежать від топології мережі, отриманої від SDN-контролера. Таким чином, будь-які проблеми з роботою цієї служби в SDN-контролері негативно вплинуть на продуктивність всього SDN. Ці проблеми з продуктивністю більш помітні у випадку динамічних і великих мереж. Що стосується служби виявлення топології, було проведено кілька експериментів на різних SDN-контролерах із запущеним лише одним блоком виявлення. Результати показали, що коли кількість SDN-перемикачів (тобто розмір мережі) досягає певної межі, відбувається значне збільшення використання ЦП SDN-контролера та значне зниження продуктивності мережі. Автори емпірично оцінили продуктивність OpenDaylight і ONOS SDN-Controllers з точки зору оновлення процесу виявлення топології. Автори використовували час виявлення топології та пропускну здатність як показники продуктивності, і їхні результати показали, що ONOS працював краще з точки зору пропускної здатності мережі у разі зміни топології, тоді як OpenDaylight перевершував ONOS за часом виявлення топології. Тому в цьому підрозділі ми проаналізуємо, як розмір мережі впливає на продуктивність SDN-контролера під час процесу виявлення топології мережі.

Тим не менш, немає багато досліджень, які б аналізували продуктивність OFDP в мережах SDN, і більшість дослідників в аналізі продуктивності OpenFlow зосередилися на типах SDN-контролерів. Тому аналіз продуктивності ODFP був би доречним, щоб відкрити це питання для дослідників. для пошуку в цих доменах. У цій роботі ми обговоримо проблеми продуктивності виявлення топології в протоколі OpenFlow та її вплив на контролер SDN, де OFDP буде відігравати головну роль.

Автори досліджували продуктивність OFDP для транспортних мереж (тобто мереж постачальників магістральних послуг) і виявили, що коли вимоги до рангу оператора задовольняються, транспортні мережі повинні відновлюватися після збою каналу протягом максимум 50 мс. Таким чином, щоб досягти часу відновлення 50 мс для носіїв на основі OpenFlow, служба виявлення топології на контролері SDN повинна працювати приблизно кожні 10 мс. Таким чином, SDNController повинен щосекунди перевіряти сотні повідомлень LLDP Packet_Out на активний порт SDN-Switch, щоб виявити один напрямок на посилання. Більше того, воно також повинно отримувати та обробляти двісті повідомлень LLDP Packet_In щосекунди для кожного каналу та наскрізного тунелю. Крім того, транспортні мережі мають сотні ланок і тисячі тунелів. Це означає, що SDN-Controller повинен обробляти мільйони повідомлень в секунду лише для контролю стану мережі. Це, безсумнівно, накладає велике навантаження на контролер SDN, а також великі витрати на мережу керування, особливо для внутріцньосмугових каналів управління. Крім того, ще один тип мережевого середовища, де OFDP показує проблеми з продуктивністю, — це хмарні центри обробки даних із багатьма орендарями. У таких середовищах топологія мережі є динамічною, оскільки орендар може будувати та змінювати свою мережу; Вони можуть у будь-який час додавати та видаляти SDN-перемикачі або посилання. Це означає, що топологія може динамічно і безперервно змінюватися. Отже, SDN-контролер повинен бути достатньо ефективним, щоб підтримувати оновлену топологію мережі. Однак для мереж OpenFlow, де OFDP є протоколом виявлення зв’язку, SDN-Controller виявляє топологію лише через періодичні, постійні та відносно довгі інтервали (контролер Floodlight, наприклад, кожні 15 с. Таким чином, SDN-Controller реалізує лише нову топологію зміни в кожному раунді виявлення, і якщо виникне помилка, виправлення помилок має чекати наступного раунду виявлення топології, який є занадто довгим. В результаті мережеві програми рівня програми, такі як маршрутизація, використовуватимуть неправильну конфігурацію мережі принаймні до наступний раунд топології.

Як ми обговорювали раніше, три основні об’єкти беруть участь один з одним у протоколі OFPD для виконання роботи: контролер SDN, комутатори SDN та канали керування між контролером SDN та комутаторами SDN. Контролер SDN буде надсилати, отримувати та обробляти повідомлення, пов’язані з процесом виявлення. Аналогічно, комутатори SDN будуть отримувати, надсилати, обробляти повідомлення, і всі ці повідомлення використовуватимуть канал керування як аргументи. Зрештою, ці процеси є накладними витратами для всіх учасників. Накладні витрати, пов’язані з OFDP, складаються з підключення OFDP до роз’єму (каналу керування) з одного боку та накладних витрат на обробку контролера SDN та комутаторів SDN з іншого боку. До цього моменту не існує жодних показників продуктивності, які були б прийняті будь-якою зі стандартних організацій для вимірювання продуктивності служби виявлення топології в SDN, але є деякі дослідження, які пропонують і використовують деякі загальновизнані показники продуктивності. Тому ми обговоримо використовувані показники продуктивності у цих дослідженнях та запропонуйте деякі інші щодо їх важливості, а саме:

- Середнє використання ЦП SDN-контролера.Автори використовують цю метрику для вимірювання того, наскільки OFDP використовує SDN-контролер для отримання топології. SDN-Controller використовує свій ЦП для створення, надсилання повідомлень LLDP Packet_Out та обробки повідомлень LLDP Packet_In. Середня завантаженість ЦП збільшується, коли збільшується кількість пакетів, надісланих і отриманих контролером SDN.
- Акумулятивне використання ЦП SDN-коммутаторів. Автори використовували цю метрику для вимірювання того, наскільки OFDP використовує ЦП для SDN-коммутаторів. SDN-Switch є важливою частиною виявлення топології. SDN-Switch отримує повідомлення LLDP Packet_Out від SDN-Controller і надсилає їх на свої активні порти. В результаті кількість пакетів, надісланих або отриманих SDN-коммутаторами, також збільшить коефіцієнт використання ЦП.

- Пропускна здатність, споживана OFDP. Як описано в методології OFDP, існує два типи з'єднань: між самими комутаторами SDN і між SDN-перемикачами та SDN-контролером. Ця метрика визначається розміром обмінюваного пакета OFDP для підтримки топології. Автори використовували цю метрику для оцінки OFDP. Таким чином, ця пропускна здатність може бути особливо важливою для вимірювання продуктивності великих мереж і внутрішньосмугових каналів управління.
- Час навчання. Деякі дослідники використовували цю метрику для оцінки OFDP та ефективності виявлення топології. Час навчання – це час, який контролер SDN повинен дізнатися про зміни топології. Процес виявлення буде повторюватися з кожним інтервалом виявлення. Інтервал відкриття – це інтервал часу між двома раундами виявлення. Проблема полягає в тому, що коли відбувається зміна топології, SDN-контролер буде чекати наступного раунду виявлення, щоб дізнатися про нові зміни топології. Це означає, що час навчання принаймні дорівнює інтервалу відкриття.

Останні дослідження продуктивності виявлення топології SDN

У літературі дослідження, які обговорюють питання продуктивності в OFDP або OpenFlow, вважаються досить мізерними, а більшість досліджень, що стосуються підтримки безпеки OpenFlow. Тому я зосереджуся на OpenFlow, і зокрема на протоколі OFDP, а також на факторах, які безпосередньо впливають на його продуктивність. Докладно наведу підсумки цього дослідження:

У цьому підрозділі ми обговоримо запропоновані рішення, пов'язані з виявленням каналів OFPD. Ці пропозиції можна розділити на дві категорії залежно від їх процедурного випуску (подієвий чи періодичний).

Періодичні. Цей домен називається періодичним, тому що процес відкриття виконується періодично для кожного періоду. Автори запропонували новий підхід під

назвою OpenFlow Discovery Protocol версії 2 (OFDPv2) для підвищення продуктивності OFDP. OFDPv2 зменшує кількість повідомлень LLDP Packet_Out лише до одного повідомлення LLDP Packet_Out на SDN-коммутатор замість активного порту. Вони запропонували дві версії OFDPv2. OFDPv2-A для SDNController і OFDPv2-B для SDN-Switch. У OFDPv2-A контролер SDN встановить додатковий набір правил потоку в кожен комутатор SDN за допомогою повідомлення OFPT_FLOW_MOD для пересилання пакетів LLDP з кожного порту комутатора SDN. Повідомлення OFPT_FLOW_MOD використовується контролером SDN для обробки таблиць потоків SDNSwitches OpenFlow. Він може додавати, оновлювати або видаляти записи потоку з таблиць потоків SDN-перемикачів OpenFlow. Однак додані правила споживають багато тернариої адресної пам'яті (TCAM), яка вже є дефіцитним ресурсом у SDN-коммутаторах. У OFDPv2 B вони не встановлювали правила потоку на комутатори SDN, але надсилали список дій (тобто набір інструкцій для пересилання пакету LLDP для кожного порту) з повідомленням LLDP Packet_Out. Однак це змушує OFDPv2-B витримувати велику пропускну здатність, особливо для внутрішньосмугових каналів керування.

S1 отримає повідомлення Packet_Out і витягне LLDP пакет з нього. Потім він пересилає пакет LLDP до всіх активних портів SDN-коммутатора і замінює вихідну MAC-адресу пакету LLDP на MAC-адресу вихідного порту. S2 отримає пакет LLDP і інкапсулює пакет LLDP з повідомленням Packet_In і надішле його до SDN-контролера. SDN-контролер розбере вхідний пакет Packet_In і дізнається про нове посилання. Нарешті, результати показують, що OFDPv2 використовує на 63–80% менше повідомлень LLDP Packet_Out, ніж той самий процес у стандартному OFDP. Крім того, при вимірюванні накладних витрат ЦП між OFDP і OFDPv2 результати показують, що OFDPv2 зменшив використання ЦП контролера SDN до 45% порівняно зі стандартним OFDP. Крім того, в Україні автори представили легкий, ефективний та

безпечний підхід для виявлення зв'язків між SDN-коммутаторами в SDN, який називається Secure and Lightweight Link Discovery Protocol (SLDP). Загалом, у пропозиції використовується новий формат пакету для виявлення посилань, використовуючи мінімальні можливості фрейму та видаляючи непотрібні функції зі стандартного кадру LLDP. Більше того, для кожної ітерації процесу виявлення зв'язку, SDN-контролер генерує пакет SLDP і надсилає його з випадковим джерелом MAC-адреси на комутатори SDN. Потім SDN-контролер встановлює запис потоку в кожну таблицю потоків комутаторів SDN, щоб згенерувати пакет із цим випадковим джерелом і погодитися з цими значеннями, коли повідомлення повертається до контролера SDN. Коротше кажучи, SDN-контролер спочатку надсилає пакет SLDP на кожен порт SDN-Switch в мережі, а наступні ітерації виявлення отримають лише порти, які підходять для пакетів SLDP. Як наслідок, тільки легітимні пакети SLDP будуть надсилатися до SDN-контролера для побудови топології, і це зменшить кількість пакетів, що використовуються в процесі виявлення, і запобіжить некваліфікованим портам отримувати пакети SLDP.

Для оцінки метрик автори використали емулятор Mininet і порівняли його з OFDP в різних мережевих топологіях з різною кількістю SDN-коммутаторів, хостів і посилань. Крім того, вони використовували кількість пакетів, надісланих контролером SDN, центральним процесором SDN-контролера та часом перевірки. За всіма цими показниками SLDP перевершує стандартний OFDP. Крім того, автори запропонували інший підхід під назвою Efficient and Secure Link discovery Scheme (ESLD), також обмежуючи передачу пакетів LLDP до портів SDNSwitch, підключених до комутаторів SDN.

Основна ідея ESLD полягає в тому, щоб класифікувати порти SDN-Switch на два класи: «коммутатор» або «хост». Порти «коммутатор» - це порти, підключені до SDN-перемикачів, і порти «хост», підключені до користувачів. Крім того, ESLD використовує деякі добре відомі повідомлення OpenFlow, такі як повідомлення «Feature-Replay», «State-Reply» та «Port-Status», щоб позначити ці порти в обох типах («Host»

або «Switch»). Тому ефективність ESLD безпосередньо залежить від кількості портів SDN-Switch в SDN. Для оцінки автори використовували масштаб хоста в різних сценаріях з різними топологіями і порівнювали ESLD з фактичними OFDP і OFDPv2. Крім того, ряд пакетів LLDP, які обробляються контролером SDN, використання ЦП для SDN-контролера та комутатори SDN, були використані як показники продуктивності. Рисунок 9. Методологія ESLD. Основна ідея ESLD полягає в тому, щоб класифікувати порти SDN-Switch на два класи: «Комутатор» або «Хост». Порти «коммутатор» - це порти, підключені до SDN-перемикачів, і порти «хост», підключені до користувачів. Крім того, ESLD використовує деякі добре відомі повідомлення OpenFlow, такі як повідомлення «Feature-Replay», «State-Reply» та «Port-Status», щоб позначити ці порти в обох типах («Host» або «Switch»). Тому ефективність ESLD безпосередньо залежить від кількості портів SDN-Switch в SDN. Для оцінки автори використовували масштаб хоста в різних сценаріях з різними топологіями і порівнювали ESLD з фактичним OFDP.

Основна ідея ESLD полягає в тому, щоб класифікувати порти SDN-Switch на два класи: «коммутатор» або «хост». Порти «коммутатор» - це порти, підключені до SDN-перемикачів, і порти «хост», підключені до користувачів. Крім того, ESLD використовує деякі добре відомі повідомлення OpenFlow, такі як повідомлення «Feature-Replay», «State-Reply» та «Port-Status», щоб позначити ці порти в обох типах («Host» або «Switch»). Тому ефективність ESLD безпосередньо залежить від кількості портів SDN-Switch в SDN. Для оцінки автори використовували масштаб хоста в різних сценаріях з різними топологіями і порівнювали ESLD з фактичними OFDP і OFDPv2. Крім того, ряд пакетів LLDP, які обробляються контролером SDN, використання ЦП для SDN-контролера та комутатори SDN, були використані як показники продуктивності. Рисунок 9. Методологія ESLD. Основна ідея ESLD полягає в тому, щоб класифікувати порти SDN-Switch на два класи: «Комутатор» або «Хост». Порти «коммута-

тор» - це порти, підключені до SDN-перемикачів, і порти «хост», підключені до користувачів. Крім того, ESLD використовує деякі добре відомі повідомлення OpenFlow, такі як повідомлення «Feature-Replay», «State-Reply» та «Port-Status», щоб позначити ці порти в обох типах («Host» або «Switch»). Тому ефективність ESLD безпосередньо залежить від кількості портів SDN-Switch в SDN. Для оцінки автори використовували масштаб хоста в різних сценаріях з різними топологіями і порівнювали ESLD з фактичними OFDP і OFDPv2. Крім того, ряд пакетів LLDP, які обробляються контролером SDN, використання ЦП для SDN-контролера та комутатори SDN, були використані як показники продуктивності. В іншому дослідженні автори запропонували два нових застосування OFDP для покращення процесу виявлення каналів: розширену службу виявлення топології (ETDPSDN) і ETDP-гібрид [45]. У ETDP-SDN служба Discovery централізована всередині SDN-Controller, і SDN-Controller ідентифікує кожну ітерацію кореневого SDNSwitch, а потім надсилає пакет виявлення ETDP на кореневий SDN-коммутатор. SDN-Switch, у свою чергу, передає цей пакет на всі свої порти, за винятком порту, де отримано ETDP.

Далі SDN-контролер встановлює правило потоку, щоб змусити SDN-Switch відправити Packet_In до SDN-контролера, а інше правило потоку заповнює кадр ETDP. Далі, після того як SDN-контролер отримає повідомлення Packet_In, він надішле додатковий FLOW_Mod, щоб видалити правило flood, яке було раніше встановлено для запобігання циклам. Однак продуктивність ODFP покращується шляхом відправки одного пакета ETDP, інкапсульованого з повідомленням Packet_Out, до кореня SDN-Switch з SDN-Controller. Це зменшить кількість повідомлень Packet_Out на частоту виявлення лише до одного пакета. ETDP також забезпечує шлях мінімальної затримки між будь-якими двома SDN-комутаторами. У другій запропонованій програмі (ETDP-Hybrid) служба виявлення використовується спільно між SDN-контролером і SDN-комутаторами. Перемикачі SDN у цій програмі не залежать від SDN-контролера

для запуску процесу виявлення, він ініціює процес виявлення та встановлює запис потоку в свою таблицю потоків, щоб надіслати інформацію про топологію до SDN-контролера. Під час обговорення результатів автори оцінили їх застосування за допомогою Mininet в різних топологіях і різному часі експерименту, щоб обчислити середнє значення та стандартне відхилення. Повідомлення Packet_out count, Packet_In та Flow_Mod count були використані як показники продуктивності. Для ETDP-SDN результати показали, що лише одне повідомлення Packet_out на SDN-коммутатор. Очікувалося, що кількість повідомлень flow_Mod буде вдвічі більшою за кількість повідомлень flow_mod у стандартному OFDP, але це не так, тому що механізм блокування (щоб припинити заливання кадрів і запобігти циклам) недостатньо швидкий, щоб зупинити дратівливі пакети ETDP перед встановленням блоку. правила потоку. У ETDP-Hybrid не було повідомлень Packet_Out і Flow_Mod, надісланих контролером SDN.

Аналогічно, автори представили протокол виявлення ресурсів SDN (SDN-RDP) як рішення для спільного керування станом мережі між кількома контролерами SDN. Кожен контролер SDN виявляє частину топології мережі для підтримки управління розподіленим вузлом і підвищення точності протоколу. Представлений підхід є асинхронним, не вимагає повної інформації про мережу та не потребує глобального кроку запуску. За результатами моделювання запропонований спосіб ефективно зменшує перевантаження SDN-контролера.

Інші методи виявлення топології використовували централізовані методи, такі як обчислювальний елемент шляху (PCE). Автори запропонували централізований алгоритм виявлення топології під назвою Generalized TOPolgy (GTOP) для PCE. GTOP працює, дозволяючи PCE механічно створювати топологію мережі без використання глобального протоколу маршрутизації, такого як протокол Open Short Path (OSPF). GTOP використовує позасмуговий канал керування для проактивного збору витрат на тополомію з комутаторів SDN. Крім того, він використовує той самий канал керування для інтерактивного оновлення топології. Однак, що стосується системи тестування,

загальний час для запропонованого протоколу становив 10 мс для оновлення змін топології.

Недоліком цієї роботи було використання позасмугового каналу керування, який може бути неможливим для розгортання у великомасштабних державах. Виявляючи топологію SDN-Optical Network (SONT), автори прийняли послідовність перевірки сигналів для виявлення посилань по одному. Ця техніка вписується в категорію попереднього обслуговування випадків виявлення конвергенції рівня 1 і забезпечує правильне відображення зв'язків у SDN-Controller, незважаючи на обмеження масштабованості та ефективності часу, особливо у великих мережах. Тому ті самі автори використовували паралельний режим, а не послідовний, щоб подолати свої обмеження.

2. ПРОЕКТУВАННЯ ТОПОЛОГІЇ МЕРЕЖІ ДЛЯ ПОБУДОВИ АРХІТЕКТУРИ

2.1 Побудування схеми топлогії

Щоб розпочати роботу з вітропарним кабелем (UTP), необхідно правильно підібрати відповідний інструмент та матеріал. Сама робота займає зовсім мало часу – менше хвилини. З необхідного обладнання та матеріалу: обтискний пристрій, кручена пара та два роз'єми RJ-45.

Виходячи з того, що протокол 10/100baseT передбачає використання тільки 2-х пар, то перевагу потрібно віддати 2-х або 4-х парному кабелю. Дві не використані пари, при цьому складають резерв.

Кабелі відрізняються кількістю та якістю провідників. Так в одному випадку він може складатися з монолітного дроту завтовшки 0,5 - 0,65 мм, а в другому - з декількох тонких, завтовшки 0,2 мм. Перевагу варто віддати першому варіанту, так як другий тип кабелю, в основному, застосовують для комутаційних шнурів. Пов'язано це з тим, що багатодротяні конструкції мають набагато нижчі електричні показники.

Підібрати роз'єм потрібно виходячи з кабелю, що використовується. Провідник з монолітного дроту та багатодротяна конструкція мають різну конструкцію врізного контакту. Тут не можна припускатися помилок, оскільки вони незмінно призведуть до погіршення контакту.

Наступним моментом, на який потрібно звернути увагу - це роз'єми, що використовуються. Вони мають бути відповідної категорії: третьої чи п'ятої. Відмінність у цих роз'ємах носить естетичний вигляд, але з технічне призначення. Перед заведенням дротів усередину роз'єму в п'ятій категорії передбачався пластиковий вкладиш, який

служив для того, щоб створити мінімальну довжину розплетення витої пари. Таким чином, покращувалася електрична характеристика середовища.

Зараз виробники випускають ці роз'єми з однаковою конструкцією. Для проводків призначені жолоби у корпусі роз'єму. У зв'язку з цим монтажнику потрібно більше часу на роботу, але сам процес складності представляти не повинен.

Наступний момент - це обтискний інструмент, що використовується. Ціна його знаходиться у проміжку від $5 до $50. Відмінність за ціною впливає на його довговічність та зручність у роботі, у всьому іншому він абсолютно однаковий. Найзастосовнішим вважається Hanlong (HT-210). Він має всі необхідні функції, включаючи ножі для обрізки кабелю та зняття ізоляції.

Для вирішення цього завдання існують різні способи. Відмінність полягає у довговічності, надійності та, звичайно ж, вартості. Який спосіб все-таки вибрати?

Це питання ми стосувалися в четвертому розділі, де використовувався традиційний підхід побудови кабельних мереж. Цей метод спричинив додаткові фінансові вкладення, а як і потрібно було задати додаткові умови. Цей процес проілюстрований на малюнку нижче.

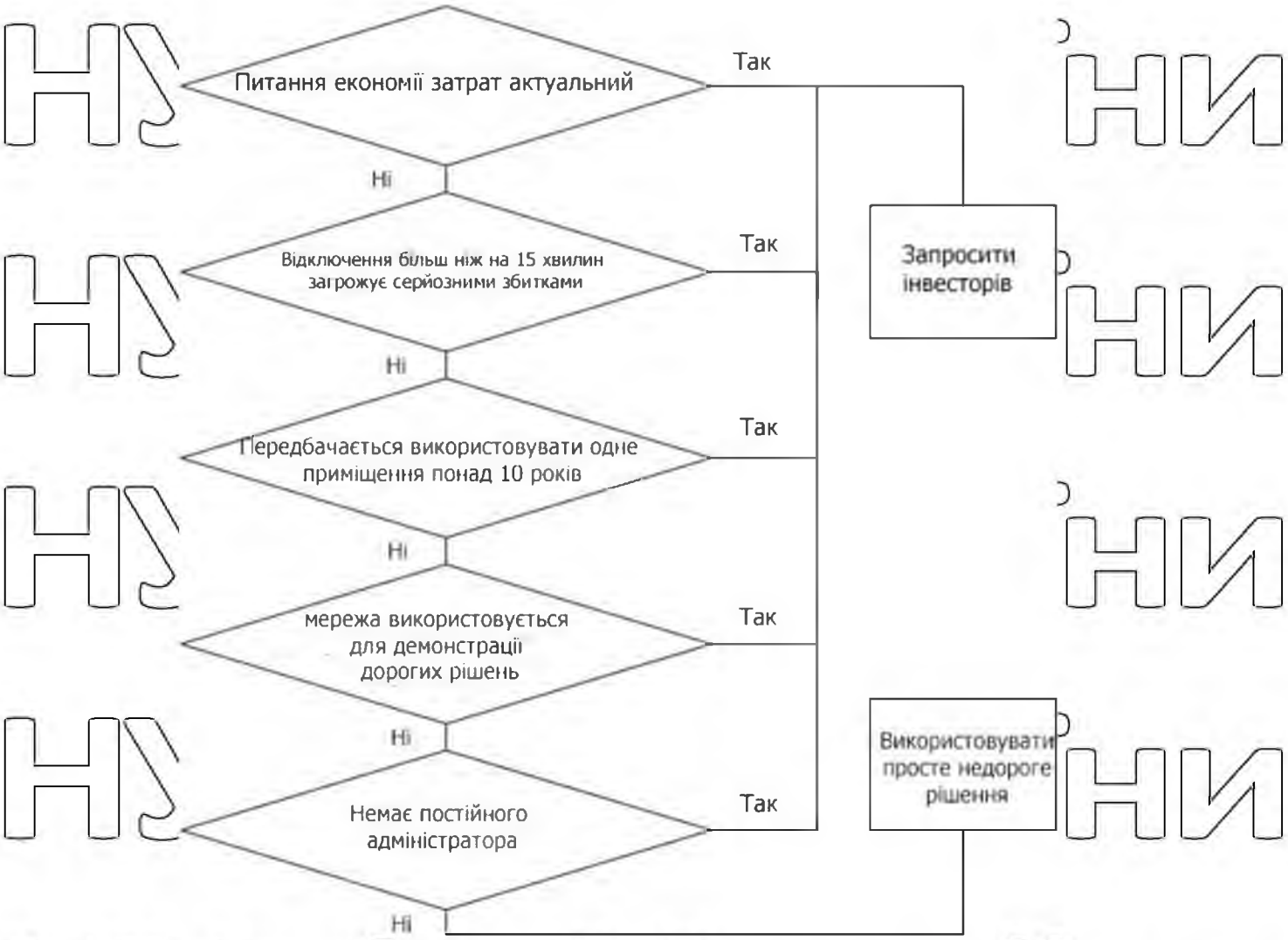

Рисунок 2.1 — Блок-схема вибора рішення відтворення мережі

Побудова мережі даним способом неможливо без підключення системних інтеграторів. Але, оскільки наша мета створити недорогу мережу самотужки - даний варіант не з кращих. Системні інтегратори, без сумніву, швидко побудують таку мережу, але це коштуватиме, аж ніяк, не дешево. Якщо їхні послуги значно перевищують запланований бюджет, то ці завдання краще вирішити іншими способами.

Коли мережа вибудовується із залишків коаксіального кабелю та списаних мережевих адаптерів. Такий підхід може виправдати себе, хіба що у шкільних кабінетах інформатики.

То який, все-таки, спосіб створення мережі віддати перевагу? Відразу слід відмовитися від використання коаксіального кабелю та накладати обмеження на трафік у 10 мегабіт. Не варто також використовувати некеровані комутатори, краще зупинити свій вибір на хабах, тим більше, їх вартість буде однаковою. Забудьте про телекомунікаційні шафи та кроси, за цих умов їх використання не доцільно. Рекомендовано використовувати настінні короби та розетки.

Почати слід з визначення місця прокладання силових кабелів - це, перш за все, питання безпеки вас самих. Зазвичай, при будівництві мережі такі кабелі вже прокладені в стінах, а якщо силове проведення роблять паралельно з комунікаціями ЛОМ, то все стає значно простіше. Якщо випадок типовий - розведення вже зроблено, то з усією уважністю поставтеся до цього пункту. Будівельники не завжди дотримуються вимог ГОСТів.

Існують спеціальні датчики електромагнітного поля, що визначають приховану проводку. Вони коштують не дорого, легкі у використанні, результатам можна довіряти. Обов'язково придбайте такий прилад, але якщо у вас його все ж таки не виявилося, скористайтеся стандартними ознаками наявності електричної проводки. По-перше - силові кабелі зазвичай знаходяться на відстані 10-15 см. нижче стелі, по-друге - це наявність розеток, коробок, вимикачів.

Крім того, що пошкодження силової проводки є небезпечним для здоров'я, є ще одна небезпека. Ця небезпека полягає в наступному, справа в тому, що струм високої напруги, що проходить по силовому кабелю, розташованому в безпосередній близькості до крученої пари, даватиме наведення і спричинить перебої у зв'язку.

У зарубіжних національних стандартах із цього приводу є жорстка інструкція – рознесення кабельних систем на 60 см. один від одного. Якщо всередині офісу проводка має потужність менше 2 кіловат, то вона вважається прийнятною, яка не несе

шкоди цілісності потоку даних. У такому випадку, мережі можна розміщувати поруч і монтуватися в тому самому коробі.

Навряд чи при побудові подібних систем, ви обійдетесь без свердління отворів у стінах та перегородках. Так само, не поспішайте одразу розпаковувати всю фурнітуру та активну апаратуру, зробити це можна лише після завершення "брудної" частини роботи. Тут вас може підстерігати ще одна неприємність – намічене на плані місце для отвору, може не вийти. Пов'язано це може бути, наприклад, із непрохідністю стіни саме в цьому місці. Тоді отвір доведеться переробляти, що помітно зіпсує зовнішній вигляд комунікацій.

Підбирати обладнання потрібно з поставлених завдань. Наприклад, побутовий дриль використовується для виготовлення отворів у тонких цегляних або дерев'яних стінах. А от якщо стоїть завдання подолати перешкоду завтовшки від 15 сантиметрів до 1 метра, то зробити це без перфоратора важко. Зазвичай, це устаткування беруть у найм, т.к. його вартість для покупки є досить високою.

Не завжди найкращим рішенням для прокладання мережевого кабелю є отвори в сінах. Іноді краще обійти капітальну стіну у дверях. У більшості випадків вартість зекономленого таким чином кабелю не окупає вартості професійного будівельного обладнання для свердління стін.

Тепер черга за створенням трас, якими буде прокладено кабель. У нами прикладі, коли мережа не велика і не дорога, встановлюються короби. Для потреб такої мережі немає сенсу використати інші методи. Скажімо відразу, дешевий варіант, не виключає проведення кабелю під підлогою та підвісною стелею. Такий тип прокладки дозволяє зробити дроти не видимими, що значно покращує естетичне сприйняття мережі. При цьому цей спосіб простий та економічний. Оскільки в такій мережі кабелів буде мало, немає сенсу спеціально готувати для них трасу. Той самий підхід використовується і в питанні кріплення.

Короби. Коробів існує безліч, вони різняться зовні і за ціною, але спосіб їх установки однаковий. На початку, до стіни прикручується основа короба, потім туди укладається кабель і все це закривається кришкою, яка носить декоративний характер. Дорогіший варіант має розетку, яка може бути частиною цього короба або кріпитися додатково. Цей варіант надійніший.

Кріплення для короба вибирається залежно від типу стіни (від саморізів до двостороннього скотчу). Виконувати горизонтальні прогони потрібно на певній висоті від підлоги, зазвичай це 60-80 см. Ще на стадії закупівлі матеріалів необхідно придбати конструкційно-декоративні елементи, які дозволять правильно з'єднувати прогони. Зазвичай, щоб не помилитись при їх придбанні, користуються порадами продавця-консультанта.

Сам механізм прокладання кабелю не складає труднощів і не вимагає особливої кваліфікації. Єдине, на що потрібно звернути особливу увагу – це вигини з малим радіусом та пошкодження зовнішньої оболонки. Короби бувають тонкі та товсті. В обох випадках після укладання та закріплення (якщо це передбачено конструкцією короба) кабелю закривають спеціальною декоративною кришкою.

Якщо кабель укладається по стелі, під підвісною стелею, слід пам'ятати, що стандартами забороняється укладати кабель прямо на його каркас. Пов'язано це з тим, що якщо порушити цей стандарт підвищується ризик пошкодження комунікацій, створюється додаткове навантаження тощо.

Правила пропонують нам кріпити кабель тільки до стін та спеціальних струн. Насправді ж, ці правила рідко дотримуються, т.к. 2-3 кручені пари не коштують таких трудовитрат. У результаті, в більшості невеликих офісів кабелю лежать за стелею, не створюючи ні перешкод, ні навантаження.

Також популярні варіанти прокладки між гіпсокартонними перегородками, під підлогою і т.п. Від чого хочеться застерегти, так це від прокладки кабелю плінтусом.

В офісі, з його частими збираннями, такий варіант вкрай не бажаний, але не забороняється.

Спеціальні роз'єми в розетках призначені для підключення до них кінці кабелю. Самі розетки бувають двох видів: що встановлюються на стіну та встановлюються в коробо. Модельний ряд настінних розеток представлений телефонними розетками та розетками з різним видом кріплень - "під гвинт" та врізними контактами через ізоляцію.

Топологія мережі складається з чотирьох кімнат:

- кімната для технічних працівників
- кімната для налаштування SDN
- кімната для керування
- серверна

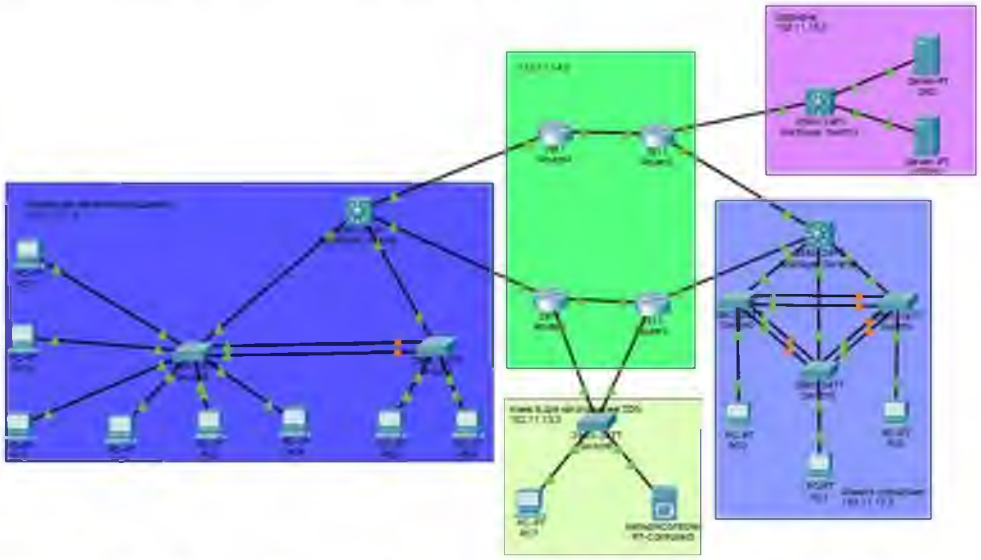

Рисунок 2.2 — Загальна схема мержі

Кімната для технічних робітників призначена для штатних співробітникі компанії, які виконують поставлені завдання. Кімната має 8 персональних комп'ютерів, 2 комутатора другого рівня та один комутатор третього рівня.

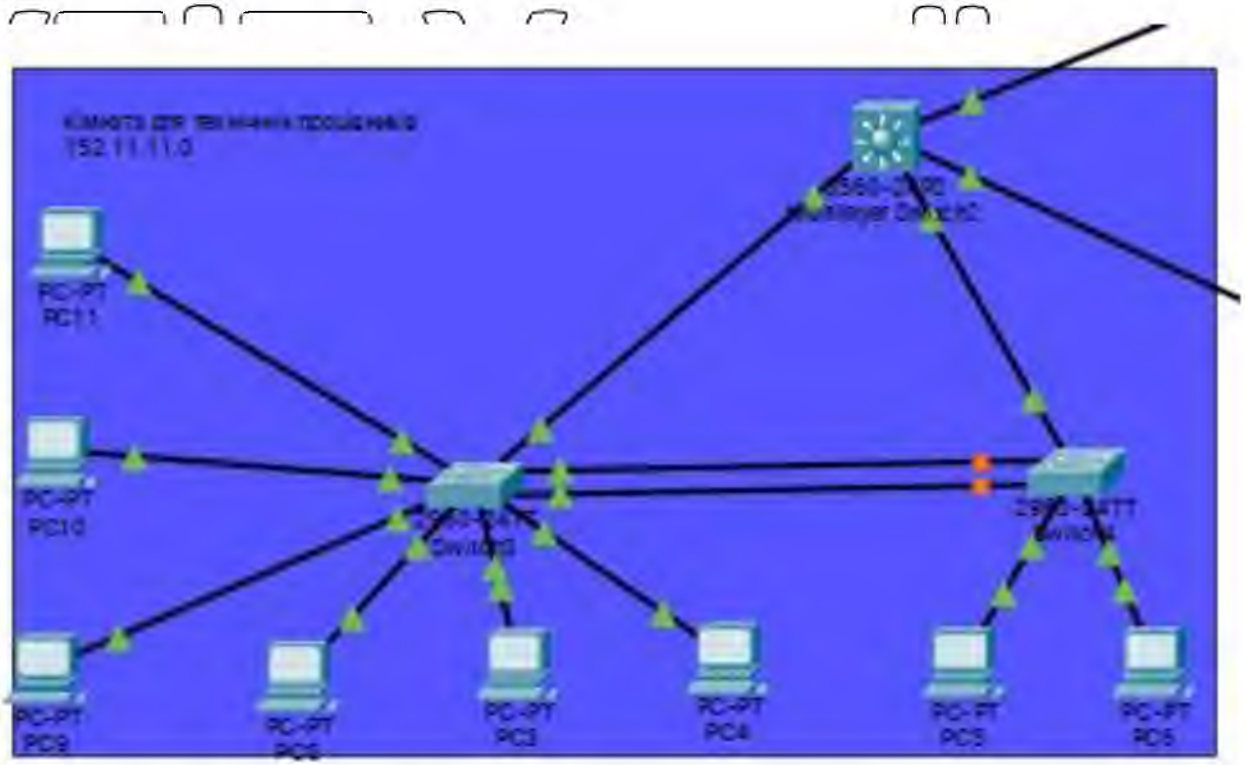

Рисунок 2.3 — Кімната технічних працівників

У цій кімнаті було налагоджено ряд протоколів, які дозволяють мережі правильно упарвувати трафіком.

VLAN Trunking Protocol (VTP) – пропрієтарний протокол компанії Cisco Systems, призначений для створення, видалення та перейменування VLAN на мережевих пристроях. Передавати інформацію про те, який порт знаходиться в якому VLAN, він не може.

VLAN (Virtual Local Area Network, віртуальна локальна мережа) - це функція в роутерах та комутаторах, що дозволяє на одному фізичному мережному інтерфейсі

(Ethernet, Wi-Fi інтерфейсі) створити кілька віртуальних локальних мереж. VLAN використовують для створення логічної топології мережі, яка не залежить від фізичної топології.

RSTP (Rapid STP, англ. Rapid spanning tree protocol) - версія протоколу STP з прискореною реконфігурацією дерева, що використовується для виключення петель у з'єднаннях комутаторів Ethernet з дублюючими лініями.

Агрегування каналів - технологія, яка дозволяє об'єднати кілька фізичних каналів в один логічний. Таке об'єднання дозволяє збільшувати пропускну здатність та надійність каналу. Агрегування каналів може бути налаштовано між двома комутаторами, комутатором та маршрутизатором, між комутатором та хостом.

Наступна кімната призначена для керівництва компанії, вони мають доступ до всіх частин мережі і можуть вносити там зміни.

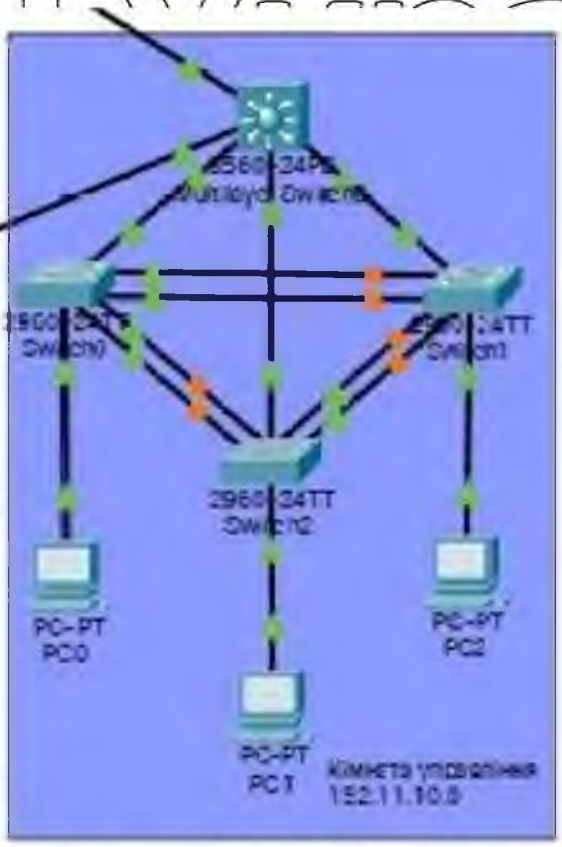

Рисунок 2.4 — Кімната управління

У цій кімнаті так встановлені вищезазначені протоколи, але особливість цієї під мережі полягає в більш надійній передачі даного, строгому настроюванні аркушів доступу в мережу та оптимальному розподілу трафіку.

ACL (Access Control List) - це набір текстових виразів, які щось дозволяють або щось забороняють. Зазвичай ACL дозволяє або забороняє IP-пакети, але також він може заглядати всередину IP-пакета, переглядати тип пакета, TCP і UDP порти. Також ACL існує для різних мережевих протоколів (IP, IPX, AppleTalk тощо). В основному застосування списків доступу розглядають з точки зору пакетної фільтрації, тобто пакетна фільтрація необхідна в тих ситуаціях, коли у вас стоїть обладнання на межі Інтернет та вашої приватної мережі і потрібно відфільтрувати непотрібний трафік.

Наступна частина мережі являє собою зв'язок маршрутизатор, який є скетелом мережі. Усі маршрутизатори об'єднані протоколом EIGRP, який дозволяє відкрити зони видимості решти під мереж інших маршрутизаторів, природно поираючи на правила аркушів доступу.

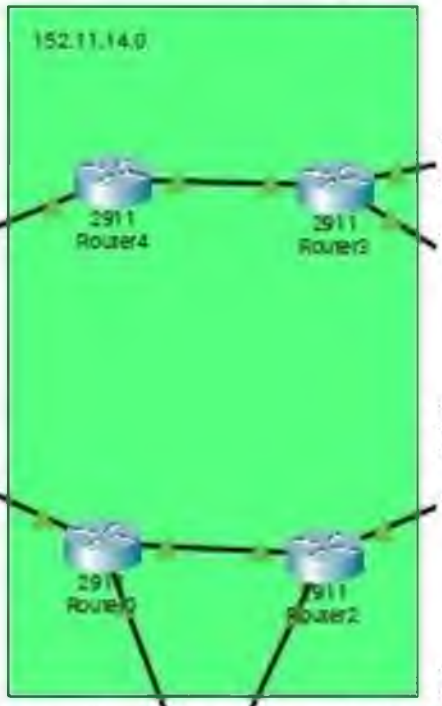

Рисунок 2.5 — Кімната маршрутизаторів

EIGRP – удосконалений дистанційно-векторний протокол динамічної маршрутизації, розроблений компанією Cisco.

Основні характеристики EIGRP:

- Швидка збіжність (порівняно з іншими дистанційно-векторними протоколами)
- Підтримка VLSM
- Часткові оновлення
- Підтримка різних протоколів мережного рівня (IP, IPX, AppleTalk)
- Одинакові налаштування протоколу при використанні різних протоколів канального рівня (наприклад, у OSPF налаштування відрізняються для Ethernet та Frame Relay)
- Складна метрика
- Використання multicast (224.0.0.10) та unicast адрес, замість широкомовного розсилання

Наступна серверна кімната. У ній знаходиться 2 сервери: HTTPS, DNS.

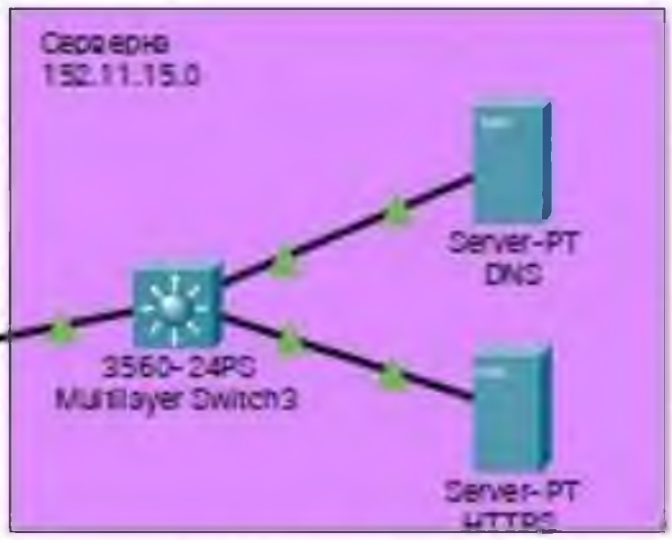

Рисунок 2.6 — Серверна кімната

HTTPS — розширення протоколу HTTP для підтримки шифрування для підвищення безпеки. Дані у протоколі HTTPS передаються поверх криптографічних протоколів TLS або застарілого у 2015 році SSL. На відміну від HTTP із TCP-портом 80, для HTTPS за замовчуванням використовується TCP-порт 443.

DNS-сервер — це сервер DNS (Domain Name System), який відповідає за зіставлення імен доменів Інтернету з IP-адресами комп'ютерів, на яких ці домени фізично знаходяться. DNS-сервери дозволяють користувачам набирати в браузері звичайні адреси сайтів та позбавляють необхідності запам'ятовувати IP-адреси.

Ця серверна призначена для того, щоб робітники змогли розміщувати сайти на внутрішніх серверах компанії та переконатися в їх готовності та працездатності.

Остання частина мережі призначена для налаштування контролера за допомогою стаціонарного комп'ютера.

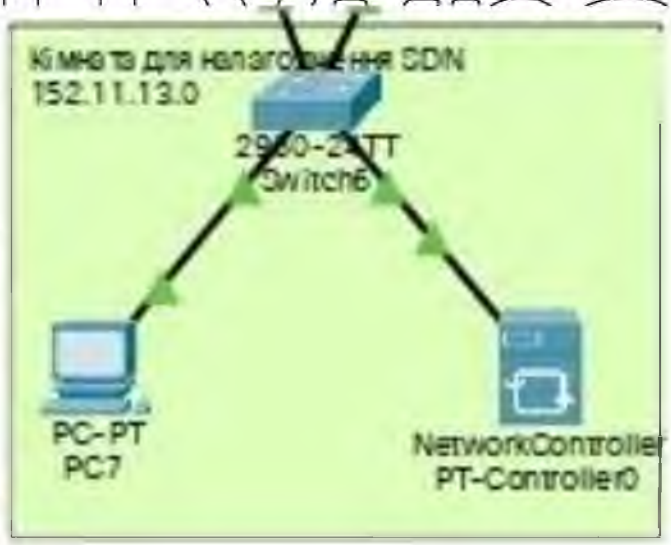

Рисунок 2.7 — Кімната з SDN-контроллером

2.2 Налаштування SDN контролера

Налаштування SDN контролера відбувається за допомогою підключення до хмарного сховища, ввівши в комп'ютері IP-адресу контролера можна зайти в конфігурації SDN контролера.

Циско дає можливість користуватися своїми сховищами. Спочатку потрібно зареєструватися і тоді з'являться всі можливості налаштування мережі.

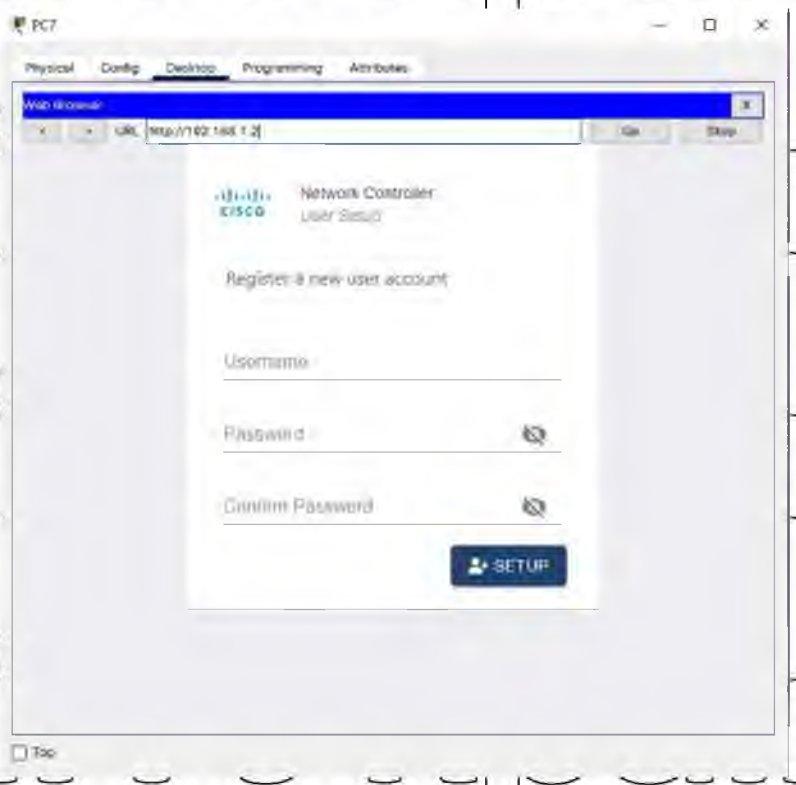

Рисунок 2.8 — Етап авторизації\реєстрації

Після авторизації нас зустрічає панель нашого хмарного сховища. Ми можемо спостерігати такі показники:

- Відсоток хостів , які можна отримати через ping
- Відсоток мережевих пристроїв, які перебувають у керованому стані
- Qos — Quality of Service (якості обслуговування)

- Host
- Кількість пристроїв мережі

Рисунок 2.9 — Інтерфейс налаштування контроллера

Далі треба зайти у вкладки Discovery та Credentials та налаштувати обліковий запис нашого хмарного сховища. За допомогою цього облікового запису наш SDNконтролер сам знаходить IP-адреси, які бачить у мережі. Він проводить пошук пристроїв, так сам перевіряє їх роботу та швидкість дії.

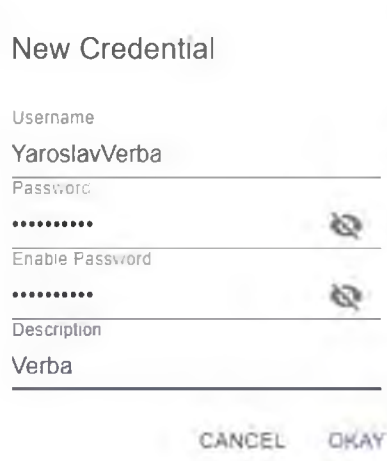

Рисунок 2.10 — Налаштування облікого запису сховища

Після певного часу наш SDN контролер знайде всі під'єднані аксесуари до нього. Після цього проведе тестування та визначити працездатність мережі

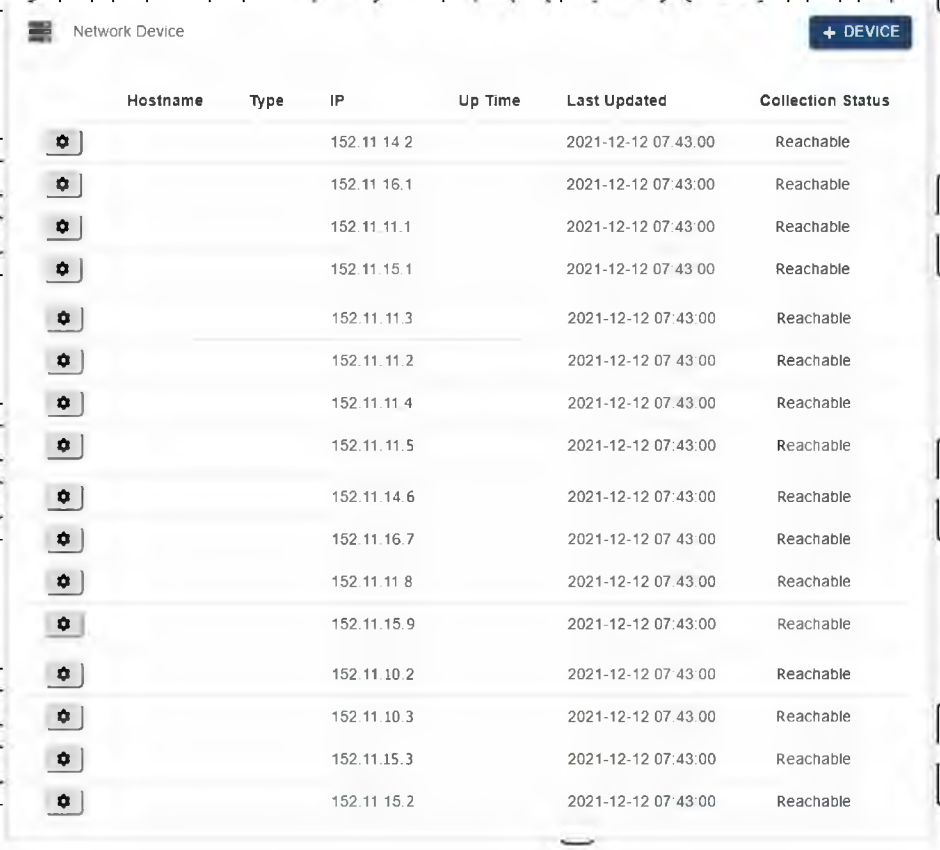

Рисунок 2.11 — Список усіх знайдених девайсів

На условній панелі ми побачимо всі показники, які були порожні після заходу в обліковий запис.

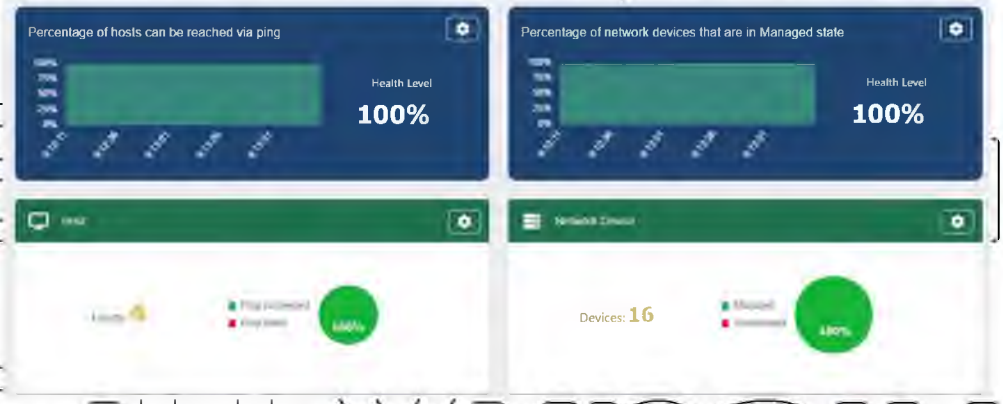

Рисунок 2.12 — Заповнена панель с мережевими пристроями

3. Тестування налагодженого хмарного сховища

3.1 П’ять етапів тестування SDN мережі

У першому тесті мультирівневий контролер SDN запитує у доменних контролерів топологію з використанням протоколу REST. Рішення відображає повну мультидоменну топологію з:

- оновленою інформацією про топологію мережі, отриману від доменних контролерів.
- інтерфейс користувача, що представляє мультивендорну і мультидоменну топологію мережі.
- таблицею існуючих сервісів.
- логічною топологією сервісів, що відображається поверх фізичної топології мережі.

У другому тесті мультирівневий контролер SDN обчислює мультидоменний маршрут мережевими доменами з використанням вбудованого механізму обчислення колії. Цей процес описаний у наступних кроках:

- користувач запитує надання сервісу на оптичній мережі через інтерфейс.
- мультирівневий контролер SDN звертається за API до доменних контролерів, запитуючи маршрути між оптичними портами.
- використовує отриману інформацію як вхідні дані для блоку обчислення шляхи (PCE) та розраховує можливість створення наскрізного маршруту.
- прокладає наскрізний сервіс.
- відображає створений сервіс.

У третьому тесті доменний контролер забезпечує відновлення трафіку при виникнення аварії всередині одного з доменів. Цей процес описаний у наступних кроках:

- користувач запитує надання сервісу за маршрутом відновлення на оптичній мережі через інтерфейс.
- мультирівневий контролер SDN звертається за API до доменних контролерів запитуючи маршрути між оптичними портами.
- використовує отриману інформацію як вхідні дані для блоку обчислення шляхи (PCE) та розраховує можливість створення наскрізного маршруту.
- прокладає наскрізний сервіс.
- відображає створений сервіс, як у другому тесті.
- після виникнення розриву мережі в графічному інтерфейсі відображається помилка.
- доменний контролер продає відновлення сервісу.

У четвертому тесті Мультирівневий SDN контролер виконує оптимізацію сервісу після спрацювання відновлення у третьому кейсі. Цей процес описаний у наступних кроках:

- користувач запитує оптимізацію сервісу оптичної мережі через користувальницький інтерфейс.
- мультирівневий контролер SDN звертається за API до доменних контролері запитуючи маршрути між оптичними портами.
- використовує отриману інформацію як вхідні дані для блоку обчислення шляхи (PCE) і розраховує можливість створення оптимізованого наскрізного маршруту.
- прокладає оптимізований наскрізний сервіс.
- відображає створений сервіс.

У п'ятому та останньому тестовому випадку мультирівневий SDN-контролер від забезпечує відновлення сервісу під час обриву мережі між доменами. Цей процес описаний у наступних кроках:

- користувач запитує надання сервісу заздалегідь розрахованим маршрутом відновлення на оптичній мережі через інтерфейс.
- мультирівневий контролер SDN звертається за API до доменних контролерів, запитуючи маршрути між оптичними портами.
- використовує отриману інформацію як вхідні дані для блоку обчислення шляхи (PCE) та розраховує можливість створення наскрізного маршруту.
- прокладає наскрізний сервіс.
- відображає створений сервіс.
- після виникнення розриву на мережі між доменами, у графічному інтерфейс відображається помилка.
- багаторівневий контролер SDN відновлює сервіс.
- відображає відновлений сервіс.

3.2 Висновки з проведенних тестувань SDN мережі

Тестування ілюструє ефективне надання мультидоменних сервісів з єдиного графічного інтерфейсу за участю кількох вендорів, а також швидке відновлення мережевих сервісів в одному або кількох доменах. Це призведе до значного прискорення усунення несправностей, надання та відновлення сервісів порівняно з тими самими діями через традиційні системи управління мережею.

Завдяки цим компонентам користувачі мережі можуть самостійно вибирати нові транспортні послуги на вимогу, які надаватимуться автоматично та оптимізуватися динамічно відповідно до вимог до пропускної спроможності та продуктивності мережі. Інтегрована мультивендорна мережа із централізованим наданням сервісів

призведе до гнучкості обслуговування та дозволить операторам створювати інноваційні програми для конкретних замовників, щоб прискорити мережеві інновації та надання нових послуг для своїх кінцевих користувачів.

Також може бути досягнуто значної економії. Ієрархічне управління мультивендорні мережі можуть зменшити операційні витрати, оскільки це дозволяє значно швидше розгортати сервіси та скорочує час їх відновлення. Терміни замовлення сервісу будуть зменшені з поточних кількох тижнів або місяців до кількох годин, також спроститься процес налаштування. Відновлення Сервіс після аварії на мережі може знизитися в середньому до декількох хвилин. Також можуть бути знижені капітальні витрати за рахунок того, що різні відділи всередині організації сервіс-провайдера можуть спільно використовувати мережу, роблячи її по-справжньому багатоцільовою.

Використовуючи мультивендорний SDN контролер, оператори зможуть уникнути залежності від виробника обладнання за допомогою підтримки кількох доменних контролерів T-SDN, які працюють із інфраструктурою різних постачальників.

Інноваційна демонстрація сприяла кращому розумінню комерційних можливостей мережевого керування з підтримкою SDN, яке допоможе спростити. Використання SDN механізмів в оптичних мережах.

ВИСНОВКИ

Ідея програмованої мережі спочатку почала матеріалізуватися в концепції «активної мережі», в якій було закладено бачення SDN, але ця концепція не набула широкого поширення. Потім проводилися спроби розділити площини управління та передачі з суто практичних міркувань, для кращої маршрутизації трафіку.

Нарешті робота над OpenFlow та мережевими операційними системами досягла оптимального балансу між візіонерством та прагматизмом. Такий баланс між широким і ясним баченням і прагматичною стратегією широкого застосування отримав визнання, коли SDN стала застосовуватися з метою віртуалізації мережі.

SDN продовжує розвиватися і практичне застосування цієї технології поширюється. Хоча дуже часто SDN проголошується як панацея від усіх проблем на мережі, необхідно пам'ятати, що SDN – не більше ніж інструмент для полегшення вирішення проблем мережевого управління. SDN дає можливість розробляти нові програми та вирішення «довгограючих» проблем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies [Книжка]
2. Software Defined Networking (SDN): Anatomy of OpenFlow Volume I [Книжка]
3. SDN: Software Defined Networks [Книжка]
4. Software Networks: Virtualization, SDN, 5G and Security [Книжка]
5. Software Defined Networks [Книжка]
6. Network Function Virtualization [Книжка]
7. MPLS in the SDN Era: Interoperable Scenarios to Make Networks Scale to New Services 1st Edition [Книжка]
8. Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud 1st Edition [Книжка]
9. Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture [Книжка]

Додаток А

Додаток Б

НУБіП України

НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України
НУБіП України