

MEANS OF TELECOMMUNICATIONS ЗАСОБИ ТЕЛЕКОМУНІКАЦІЙ

УДК 621.396.967.2

DOI:10.30837/rt.2022.3.210.16

*О.І. РОМАНОВ, д-р техн. наук, І.В. СВІД, канд. техн. наук,
Н.І. КОРНІЄНКО, А.О. РОМАНОВ*

УПРАВЛІННЯ ОПТИЧНОЮ МЕРЕЖЕЮ КОНТРОЛЕРОМ SDN НА БАЗІ ONOS

Постановка проблеми

Одним з найбільш перспективних напрямків розвитку сучасних телекомунікацій є технологія SDN. Питання побудови мереж на базі цієї технології знаходяться в центрі уваги представників науково-дослідних організацій, університетів і операторів мобільного зв'язку.

Архітектура програмно-конфігурованої мережі складається з трьох рівнів: додатків, управління та інфраструктури, пов'язаних один з одним через відкриті API-інтерфейси (рис. 1).

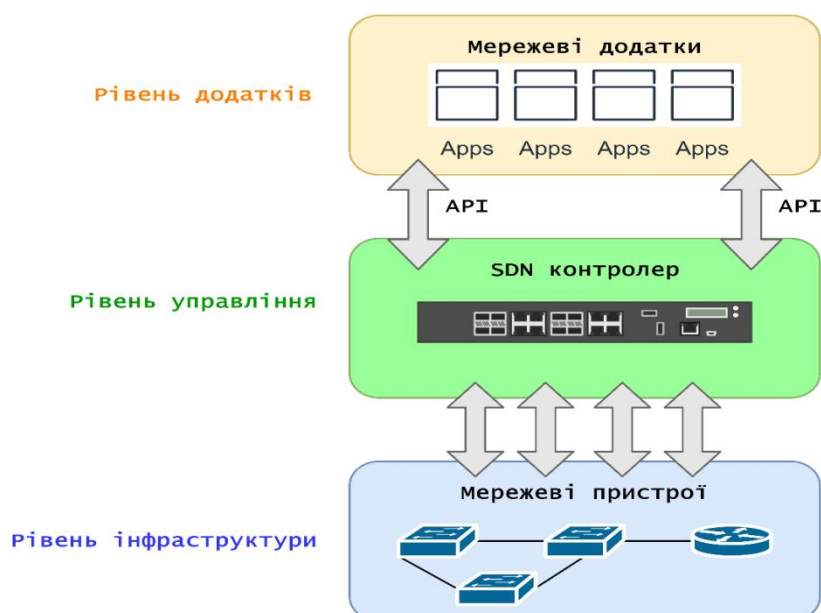


Рис. 1. Загальна архітектура SDN

Рівень додатків містить функціональні блоки з набором програмного забезпечення, яке може вирішувати окремі завдання управління, надавати сучасні сервіси користувачам, обробляти статистичні дані про стан мережевих елементів, забезпечити автоматизовану зміну протоколів при налаштуванні мережі, віртуалізувати мережеві функції, балансувати навантаження та інше.

Рівень управління побудований на базі контролера, який є централізованим органом управління мережею. На цьому рівні забезпечується управління політиками і трафіком в мережі. Рівень інфраструктури містить як фізичні, так і віртуальні елементи мережі, у яких є тільки виконавчі функції. Усі задачі управління вирішує контролер, який доводить ці рішення до рівня інфраструктури.

Аналіз останніх досліджень і публікацій

Сьогодні ведуться дослідження і розробляються пропозиції щодо практичної реалізації цієї технології в рамках ряду проєктів. Ці проєкти охоплюють створення інфраструктури широкосмугового доступу наступного покоління, побудову сервісної платформи (services

platform) надання послуг, платформи для побудови SD-RAN і Core RAN мереж операторів мобільного зв'язку 5G/6G. Найбільший внесок в розвиток даного напрямку вносять представники консорціуму Open Networking Foundation (ONF). ONF стоїть на чолі всіх проєктів з побудови SDN і займається прискоренням процесів їх практичної реалізації [1].

Опубліковано ряд документів, що описують принципи побудови і функціонування мереж SDN. У роботах [2 – 6] розглянуто загальні вимоги, системні підходи і узагальнено архітектуру мереж SDN. У роботах [2, 7, 8, 11] розглянуто завдання і особливості протоколів, які використовуються при вирішенні різних завдань в мережах SDN. У роботах [9, 10, 12] описано особливості побудови елементів мережі SDN та порядок їх взаємодії в процесі обслуговування потоків інформації. У роботах [13 – 17] розглянуто принципи побудови оптичної транспортної мережі та надано рекомендації щодо забезпечення безпеки їх функціонування. Найбільш повно і систематизовано матеріал викладено в роботах [18 – 21]. У статтях [22 – 26] досліджуються різні аспекти обслуговування інформаційних потоків і акцентується увага на необхідності виконання вимог щодо забезпечення безпеки мереж.

Постановка завдання

Найбільш складні завдання в мережах SDN стоять перед рівнем управління. Вирішення цих завдань покладається на контролер, основним елементом якого є мережева операційна система. Розглянемо особливості побудови відкритої мережевої операційної системи (ONOS), функціональний склад її елементів, протоколи та інтерфейси, що дозволяють представити мережу SDN у вигляді моделі.

Виклад основного матеріалу

Функціональна структура контролера SDN. Контролер забезпечує обслуговування трафіку відповідно до політики, яка встановлена оператором мережі. Прибравши площину управління з мережевого обладнання, контролер реалізує централізовану систему керування, спрощує автоматичне керування мережею і забезпечує інтеграцію і адміністрування бізнес-додатків [3, 27 – 29].

Архітектура SDN (рис. 2) не визначає внутрішній дизайн або реалізацію контролера SDN. Це може бути:

- один монолітний процес;
- конфедерація ідентичних процесів, організованих для розподілу навантаження або захисту один одного від збоїв;
- набір окремих функціональних компонентів, які працюють за певним набором правил;
- контролер SDN може використовувати зовнішні сервіси для виконання деяких функцій, наприклад для розрахунку шляху.

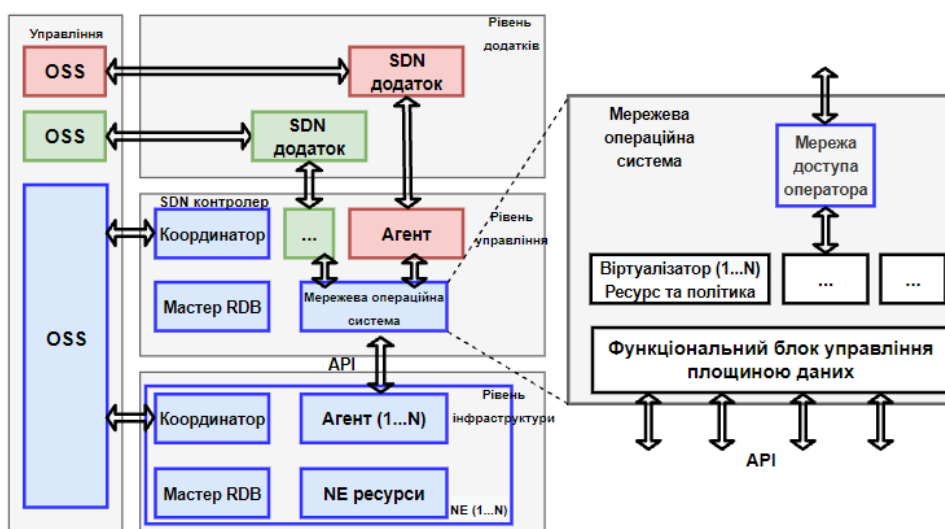


Рис. 2. Функціональні ресурси контролера

Допускається будь-яка комбінація цих альтернатив: контролер SDN розглядається як чорний ящик, який виконує певні функції. Компоненти контролера можуть розміщуватися на довільних обчислювальних платформах, включаючи обчислювальні ресурси в центрах обробки даних.

Так само як OSS управляє ресурсами і станами, контролер підпорядковується тією ж вимогою координації з будь-якими контролерами SDN, які залучені для спільного використання. Кілька компонентів менеджера або контролера можуть мати спільний доступ до мережевих ресурсів, але для виконання вимог SDN вони повинні бути:

- налаштовані на управління непересічними наборами ресурсів;
- синхронізовані один з одним, щоб ніколи не виникали суперечності або суперечливі команди.

Функціональний блок управління площиною даних ефективно володіє доступними йому підлеглими ресурсами і управляється відповідно до інструкцій OSS/координатора або віртуалізатора. Ресурси записуються у вигляді примірника інформаційної моделі. Доступ до моделі здійснюється через агента на підпорядкованому рівні. Оскільки область дій контролера SDN може охоплювати кілька мережевих елементів або навіть кілька віртуальних мереж, функціональний блок управління площиною даних повинен забезпечити скоординовану роботу. Ця функція зазвичай називається оркестрацією.

База даних ресурсів (RDB) моделює поточний екземпляр інформаційної моделі і необхідні можливості.

Координатор є функціональним компонентом, який діє від імені людини оператора. Він забезпечує надання інформації про політику обслуговування в мережі. Він передає керуючу інформацію для елементів мережі: моделі даних, площини управління і площині додатків. Тому функціональні блоки координатора розміщені повсюдно.

В архітектурі SDN віртуалізація – це виділення віртуальних абстрактних ресурсів для конкретних додатків. Контролер SDN пропонує сервіси додатків у вигляді примірника інформаційної моделі, яка абстрактно описує доступні ресурси і політику їх використання.

Віртуалізатор – це функціональний об'єкт, в якому зберігається примірник інформаційної моделі ресурсів і політики їх використання.

Віртуалізатор створюється OSS/координатором для кожного клієнтського додатку. OSS/координатор розподіляє ресурси і визначає політику, які повинен використовувати віртуалізатор в процесі надання послуг з додатками через інтерфейс API. Далі віртуалізатор створює агента, в який записується свій екземпляр інформаційної моделі доступних ресурсів і політики його використання для надання сервісу певної програми.

Будь-який протокол повинен закінчуватися функціональним об'єктом. Модель контролер-агент підходить для відносин між контрольованим і контролюючим об'єктом. Контрольований об'єкт призначається агентом, функціональним компонентом, який представляє ресурси і можливості клієнта в середовищі сервера. Агент в контролері SDN на рівні N представляє ресурси і дії, доступні з додатком на рівні N+1.

Архітектура відкритої мережевої операційної системи (ONOS). Розглянемо принципи побудови архітектури відкритої мережевої операційної системи ONOS та її особливості (рис. 3). Слід зазначити, що ONOS є основним функціональним блоком контролера SDN. Тому в науково-технічній літературі ONOS часто ототожнюють з контролером мережі SDN. Слід мати на увазі, що це не зовсім так. Контролер складається з ряду функціональних блоків, одним з яких є ONOS. Однак у зв'язку з тим, що ONOS відповідає за вирішення всіх інтелектуальних завдань управління оптичною мережею, в її архітектурі використовуються підходи, описи та рівні, які використовувалися як при описі архітектури SDN, так і при описі контролера цієї мережі.

Слід зазначити, що розробка ONOS є окремим проектом ONF з відкритим кодом, випущеним під ліцензією Apache 2.0 [30]. Крім того, опис системи ONOS є загальнодоступним [31].

ONF вважає, що платформа ONOS розроблена для задоволення потреб операторів у створенні рішень операторського рівня, які забезпечують гнучкість створення та розгортання нових динамічних мережевих служб із спрощеними API. ONOS має забезпечувати підтримку як конфігурації, так і керування мережею в реальному часі. Такий підхід позбавить від необхідності запускати протоколи керування маршрутизацією та комутацією в мережевій інфраструктурі оператора.

Одним із можливих способів подальшого підвищення ефективності управління є перенесення ONOS у хмару. Це дозволить кінцевим користувачам створювати нові мережеві програми без необхідності змінювати площину даних.

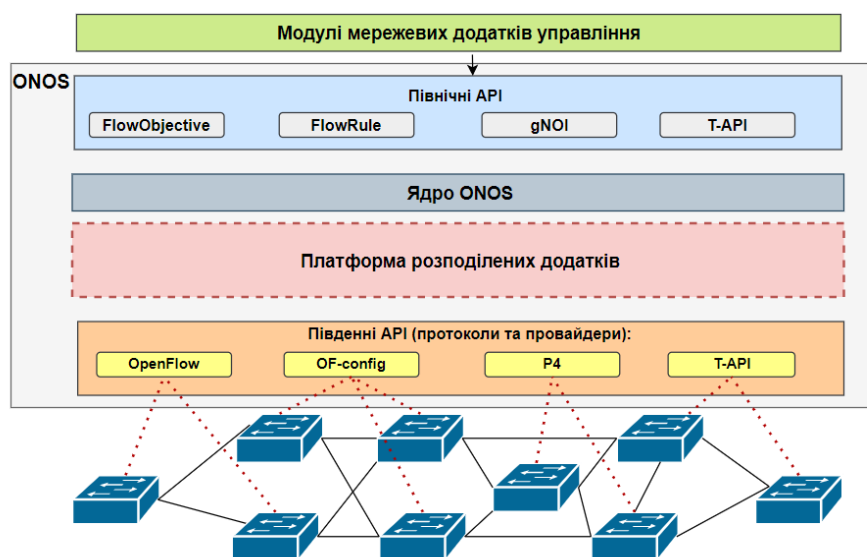


Рис. 3. Архітектура ONOS

Можливий варіант архітектури ONOS показаний на рис. 3. Видно, що архітектура ONOS є модульною за дизайном. Такий підхід дозволяє забезпечити оптимальне використання ресурсів, оскільки для кожного конкретного розгортання буде визначено оптимальну кількість необхідної підмножини модулів. Архітектура ONOS включає:

- ядро відкритої операційної системи, що відповідає за вирішення всіх інтелектуальних завдань управління мережею;
- платформу розподілених додатків, яка являє собою набір додатків для вирішення завдань управління оптичною мережею. Ця платформа побудована як розширювана модульна система. Крім того, кожен модульний функціональний блок вирішує певну задачу управління. Зі збільшенням переліку завдань управління, які необхідно вирішити, буде збільшуватися кількість модулів у платформі розподілених додатків;
- набір відкритих мережевих інтерфейсів. При цьому, як і в архітектурі SDN, пропонується використання двох типів інтерфейсів: відкритий південний інтерфейс (SBI), який є набором модулів для взаємодії з рівнем інфраструктури; відкритий північний інтерфейс (NBI), який являє собою набір модулів для взаємодії з функціональними блоками прикладного рівня.

Слід зазначити, що південний інтерфейс включає набір протоколів та інтерфейсів, таких як OpenFlow, OF-CONFIG, P4, T-API та інші. Завдання, які вирішують ці елементи системи, та їх важливість, визначаються умовами функціонування мережі. Наприклад, розглянемо роботу протоколів OF-CONFIG і OpenFlow.

OF-CONFIG (протокол керування та конфігурації OpenFlow) – це протокол для налаштування та керування робочим контекстом комутаторів OpenFlow.

Протокол OpenFlow – це протокол керування обробкою даних, що передаються через мережу передачі даних маршрутизаторами та комутаторами, який реалізує програмно-

конфігуровану мережеву технологію. Протокол OpenFlow передбачає, що комутатор OpenFlow (наприклад, комутатор Ethernet, який підтримує OpenFlow) налаштовано з різними артефактами, такими як IP-адреси контролерів OpenFlow та інших елементів мережі. Тобто комутатор попередньо налаштований.

Протокол OF-CONFIG призначений для віддаленого налаштування комутаторів OpenFlow. Він працює в повільнішому режимі порівняно з протоколом OpenFlow. Наприклад, протокол OF-CONFIG вирішує проблему створення матриць маршрутизації, яка згодом буде вирішуватися протоколом OpenFlow в реальному часі при обробці вхідних пакетів. Іншим прикладом протоколу OF-CONFIG може бути ввімкнення/вимкнення порту, що також не має відношення до обробки пакетів у реальному часі. OF-CONFIG представляє перемикач OpenFlow як абстракцію під назвою логічний перемикач OpenFlow (віртуальний перемикач). Протокол OF-CONFIG дозволяє конфігурувати параметри логічного комутатора OpenFlow так, щоб контролер OpenFlow міг спілкуватися та контролювати логічний комутатор OpenFlow через протокол OpenFlow. OF-CONFIG дозволяє динамічно пов'язувати ресурси з певними логічними перемикачами OpenFlow. Цей протокол може створити кілька віртуальних комутаторів з одного фізичного комутатора та призначити кожному певний ресурс. ONOS використовує ці протоколи для взаємодії з площиною даних, щоб реалізувати завдання керування мережею.

Останнім часом починають використовуватися P4, T-API. Вони все частіше використовуються для побудови мережевих моделей для вирішення задач керування. При цьому P4 демонструє більшу ефективність, ніж OpenFlow. Необхідно, щоб набір відкритих мережевих інтерфейсів і протоколів ONOS був універсальним для всіх сегментів і елементів мережі. Це дало б можливість будувати однорідні, масштабовані, універсальні системи з високим рівнем взаємодії всіх елементів.

Ця архітектура ONOS дозволить:

- спростити процес перевстановлення та оновлення програмного забезпечення на елементах мережі;
- знизити витрати на впровадження нових технологій та послуг всієї мережі оператора;
- поетапно збільшити кількість завдань управління, що вирішуються шляхом додавання нових функціональних блоків у складі платформи розподілених додатків;
- створити горизонтально-масштабовану систему, яка забезпечуватиме високий рівень відмовостійкості, що дуже важливо для забезпечення виконання заданих вимог до надійності функціонування централізованих систем управління.

Отже, можна сформулювати вимоги до архітектури ONOS:

По-перше, горизонтальна площина NBI повинна бути достатньо великою. Це пояснюється тим, що будь-який доступ до основного обладнання здійснюватиметься через ONOS. Таким чином, сукупність усіх північних API повинна бути достатньою для налаштування, підтримки та керування мережею.

Крім того, повинна бути багаторівнева система додатків і сервісів, що працюють на ONOS. Залежно від пріоритету завдання, яке вирішує додаток, і частоти його використання слід визначити рівень, що визначає ефективність завдання. Наприклад, додатки, які визначають головний функціональний блок керування у разі відмови елементів мережі, повинні мати пряме з'єднання з ONOS. А програми, які забезпечують встановлення нового програмного забезпечення, сертифікатів, параметрів конфігурації і в своїй роботі використовують протокол OF-CONFIG, можуть розташовуватися на рівні з високою затримкою.

Ще однією вимогою до ONOS є забезпечення одночасного двостороннього обміну інформацією через NBI та SBI. Як видно з рис. 3, програми використовують ONOS через NBI для керування мережею, а через SBI південні модулі передають інформацію про стан основної мережі до ядра ONOS.

Взаємодія між ядром ONOS і мережевими пристроями забезпечується набором протоколів і інтерфейсів, таких як OpenFlow, OF-CONFIG, P4, T-API, які забезпечують деталізацію

взаємодії з пристроями, тим самим ізолюючи ядро ONOS і додатки, що працюють на ньому, від деталей пристроїв мережі.

Ядро ONOS складається з ряду підсистем, кожна з яких відповідає за певний аспект функціонування мережі. Кожна підсистема підтримує власну абстракцію служби, яка відповідає за поширення параметрів стану мережі в кластері.

Сервіси ONOS побудовані за допомогою розподілених таблиць, які реалізовані за допомогою розподіленого сховища ключів/значень. ONOS використовує Atomix для зберігання. Це система на основі Java, яка включає: розподілену структуру даних; опис алгоритму прямого обміну повідомленнями між елементами; опис координації взаємодії, включаючи блокування небажаних функцій і обрання керівника; управління членством у групі.

Важливою особливістю Atomix є координація всіх екземплярів ONOS. Тут вирішуються дві задачі:

- кількість екземплярів ONOS, що працюють у будь-який момент часу, залежить від робочого навантаження та кількості реплікацій, необхідних для забезпечення доступності у разі збою. Примітив членства в групі Atomix використовується для визначення набору доступних екземплярів ONOS. Це дозволяє відстежувати справні та несправні екземпляри ONOS;

- основним завданням кожного екземпляра ONOS є моніторинг і підтримка підмножини фізичних комутаторів у мережі, де він обраний лідером. Визначення лідера здійснюється за допомогою функціонального блоку Atomix. Усі екземпляри ONOS можуть контролювати стан комутаторів. Однак управляти перемикачами може тільки лідер. Якщо примірник ONOS виходить з ладу, Atomix гарантує, що для перемикачів буде обрано нового лідера. Такий же підхід використовується при підключенні нового комутатора до мережі.

Сервіси та підсистеми ONOS. Хоча ONOS в значній мірі спирається на стандартні протоколи і моделі, наприклад OpenFlow, NETCONF, OpenConfig, його системна архітектура безпосередньо до них не прив'язана. ONOS модульна операційна система, вона складається із окремих підсистем (рис. 4).

Підсистема управління додатками ONOS бере на себе відповідальність за поширення артефактів додатків по кластеру. Вона забезпечує те, що всі вузли працюють з одним і тим же програмним забезпеченням. Базовий дистрибутив ONOS містить більше 175 додатків, які потрапляють в численні категорії, такі як програми для управління трафіком, драйвери пристроїв, утиліти, програми для моніторингу, готові моделі YANG.

Для того щоб взаємодіяти із зовнішнім світом, в ONOS є графічний інтерфейс та низка зовнішніх адаптерів, таких як REST API, CLI і розширюваний динамічний веб-інтерфейс. Інтерфейс gRPC з відкритим вихідним кодом використовує HTTP/2 для транспорту. Він надає такі функції, як аутентифікація, двонаправлена потокова передача, управління потоком, скасування і тайм-аути.

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

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

Найбільш типовими є служби: хост, пристрій, посилення, топологія, майстер, кластер, конфігурація мережі, конфігурація компонентів, пакет. Дані служби використовуються багатьма додатками, оскільки вони надають інформацію про мережеві пристрої і їх топології. Однак сервісів існує набагато більше. У тому числі ті, які дозволяють додаткам програмувати поведінку мережі з використанням різних конструкцій і різних рівнів абстракції. Такі сервіси як: маршрут, Mcast, група, лічильник, правило потоку, ціль потоку, намір. Вони забезпе-

чують підтримку з'єднання по відповідним шляхам. А потім безперервно відстежують мережу, змінюючи шляхи з плином часу з метою забезпечення відповідності цілям, запропонованим наміром, в умовах мінливої мережевої ситуації.

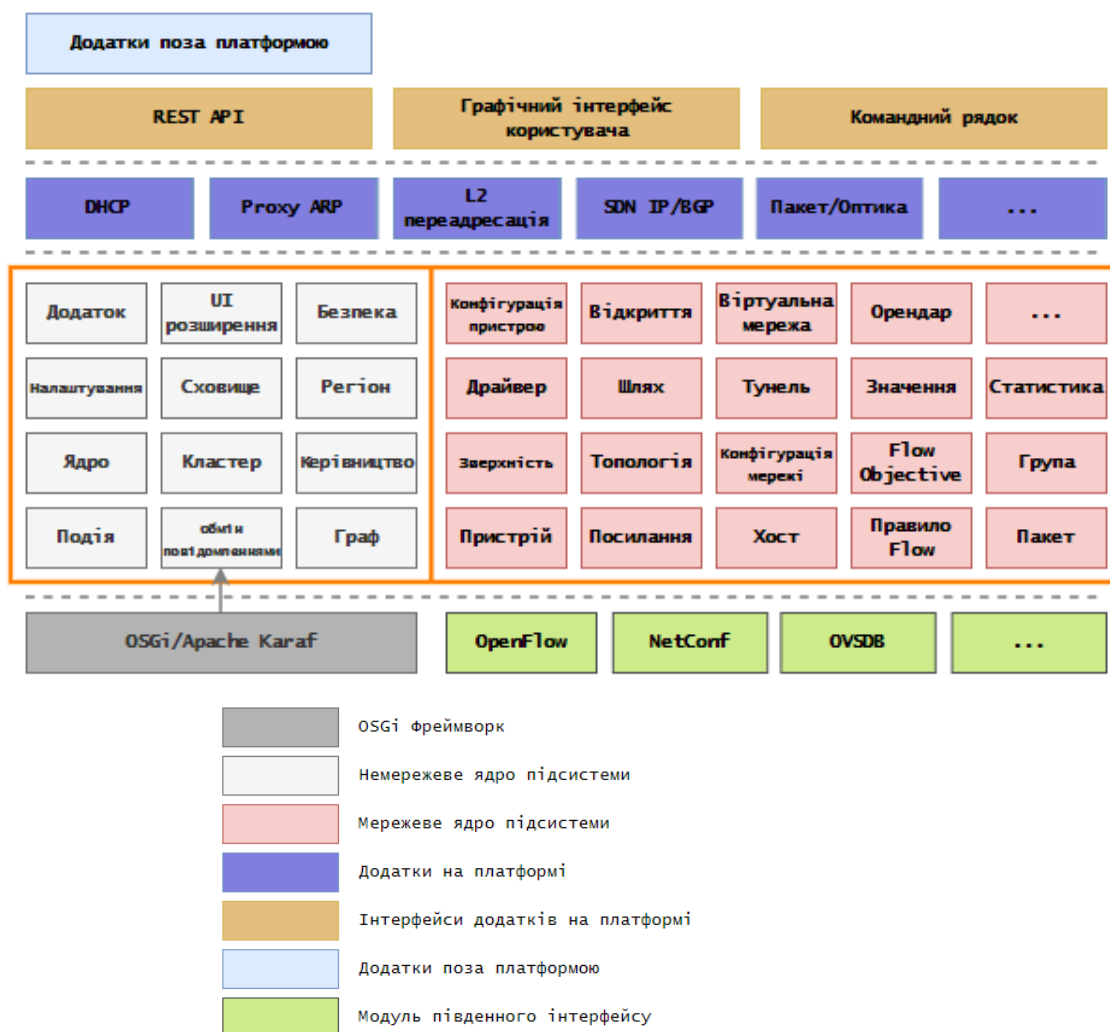


Рис. 4. Підсистеми ONOS

Кожна служба має власну розподілену базу даних і можливості обміну повідомленнями. Деякі програми можуть безкоштовно розширювати цей набір власними службами [32, 33].

Висновки

ONOS – це зручна платформа з відкритим вихідним кодом, яка дозволяє різним командам розробників спільно брати участь у проєктах з оновлення та вдосконалення системи управління. Використання ONOS дозволяє побудувати логічну централізовану площину управління в мережах SDN. Існуючий набір функціональних модулів, сервісів та інтерфейсів в ONOS дозволяє виконувати завдання управління оптичною мережею. Для подальшого розвитку ONOS необхідна розробка математичних моделей і методів оптимального вирішення задач керування в різних умовах експлуатації, які в майбутньому стануть програмними модулями прикладного рівня.

Список літератури:

1. Open Network Foundation. Accelerating the Adoption of SDN & NFV, 2021.
2. Open Network Operating System (ONOS) SDN Controller for SDN/NFV Solutions // Open Networking Foundation, 2021. [Online]. Available: <https://opennetworking.org/onos>.

3. What is SDN controller (software-defined networking controller)? // Definition from WhatIs.com, SearchNetworking, 2021. [Online]. Available: <https://searchnetworking.techtarget.com/definition/SDN-controller-software-defined-networking-controller>.
4. ONOS – Wikipedia, En.wikipedia.org, 2021. [Online]. Available: <https://en.wikipedia.org/wiki/ONOS>.
5. Apache License, Version 2.0 | Open Source Initiative, Opensource.org, 2021. [Online]. Available: <https://opensource.org/licenses/Apache-2.0>.
6. ONF TR-525 SDN Interoperability Event Technical Issues Report AppFest 2015.
7. K. Pentikousis. IETF RFC 7426. Request for Comments: 7426. ISSN: 2070-1721 EICT.
8. O. I. Romanov, M. V. Oryshuk and Y. S. Hordashnyk. Computing of influence of stimulated Raman scattering in DWDM telecommunication systems // 2016 International Conference Radio Electronics & Info Communications (UkrMiCo), 2016, pp. 1-4, doi: 10.1109/UkrMiCo.2016.7739622.
9. K. Pentikousis. ONOS. Security and Performance. Analysis // Report No. 1. September 19, 2017.
10. O. Romanov and V. Mankivskiy, Optimal Traffic Distribution Based on the Sectoral Model of Loading Network Elements // 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 2019, pp. 683-688, doi: 10.1109/PICST47496.2019.9061296.
11. O. Lemeshko and O. Yeremenko Linear optimization model of MPLS Traffic Engineering Fast ReRoute for link, node, and bandwidth protection // 2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2018, pp. 1009-1013, doi: 10.1109/TCSET.2018.8336365.
12. O. Romanov, M. Nesterenko, L. Veres, R. Kamarali and I. Saychenko. Methods for Calculating the Performance Indicators of IP Multimedia Subsystem (IMS) // Advances in Information and Communication Technology and Systems, pp. 229-256, 2020. doi: 10.1007/978-3-030-58359-0_13.
13. O. Lemeshko, J. Papan, O. Yeremenko, M. Yevdokymenko and P. Segec. Research and Development of Delay-Sensitive Routing Tensor Model // IoT Core Networks Sensors, vol. 21, no. 11, p. 3934, 2021. doi: 10.3390/s21113934.
14. C. C. O'Connor, T. Vachuska, and B. Davie. Software-Defined Networks // A Systems Approach, 2021, p. 152.
15. K. Phemius, M. Bouet and J. Leguay. DISCO: Distributed multi-domain SDN controllers // 2014 IEEE Network Operations and Management Symposium (NOMS), 2014, pp. 1-4, doi: 10.1109/NOMS.2014.6838330.
16. J. Lam, S. Lee, H. Lee and Y. Oktian. Securing SDN Southbound and Data Plane Communication with IBC // Mobile Information Systems, vol. 2016, pp. 1-12, 2016. doi: 10.1155/2016/1708970.
17. O. I. Romanov, D. M. Fediushyna and T. T. Dong. Model And Method Of Li-Fi Network Calculation With Multipath Light Signals // 2018 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), 2018, pp. 1-4, doi: 10.1109/UkrMiCo43733.2018.9047550.
18. I. Obod, I. Svyd, O. Maltsev, G. Zavolodko, D. Pavlova and G. Maistrenko. Fusion the Coordinate Data of Airborne Objects in the Networks of Surveillance Radar Observation Systems // Data-Centric Business and Applications, pp. 731-746, 2020. doi: 10.1007/978-3-030-43070-2_31.
19. O. Romanov, E. Siemens, M. Nesterenko and V. Mankivskiy. Mathematical description of control problems in SDN networks // International Conference on Applied Innovations in IT (ICAIIIT), pp. 33-39, 2021. Available: 10.25673/36582.
20. I. Svyd, I. Obod, O. Maltsev, T. Tkachova and G. Zavolodko. Optimal Request Signals Detection in Cooperative Surveillance Systems // 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), 2019, pp. 1-5, doi: 10.1109/UKRCON.2019.8879840.
21. I. Obod, I. Svyd, O. Maltsev, G. Maistrenko, O. Zubkov and G. Zavolodko. Bandwidth Assessment of Cooperative Surveillance Systems. // 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), 2019, pp. 1-6, doi: 10.1109/AIACT.2019.8847742.
22. D. Sanvito, D. Moro, M. Gulli, I. Filippini, A. Capone and A. Campanella. ONOS Intent Monitor and Reroute service: enabling plug&play routing logic // 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), 2018, pp. 272-276, doi: 10.1109/NETSOFT.2018.8460064.
23. D. Comer and A. Rastegarnia. Externalization of Packet Processing in Software Defined Networking // IEEE Networking Letters, vol. 1, no. 3, pp. 124-127, Sept. 2019, doi: 10.1109/LNET.2019.2918155.
24. O. Romanov, M. Nesterenko and V. Mankivskiy. The Method of Redistributing Traffic in Mobile Network // Data-Centric Business and Applications, pp. 159-182, 2021. doi: 10.1007/978-3-030-71892-3_7.
25. GitHub – OpenNetworkingFoundation/TAPI: ONF Transport API Repository (TAPI), GitHub, 2021. [Online]. Available: <https://github.com/OpenNetworkingFoundation/tapi>.
26. I. Svyd, I. Obod, O. Maltsev, T. Tkachova and G. Zavolodko. Improving Noise Immunity in Identification Friend or Foe Systems // 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), 2019, pp. 73-77, doi: 10.1109/UKRCON.2019.8879812.
27. TAPI Overview – Open Transport Configuration & Control – Confluence, Wiki.opennetworking.org, 2021. [Online]. Available: <https://wiki.opennetworking.org/display/OTCC/TAPI+Overview>.
28. P. Littlewood, F. Masood, E. Follis. Optical transport network. Hannover : Ciena, 2014.

29. TAPI v2.1.3 Reference Implementation Agreement. TR-547. Version 1.0. July 2020, Opennetworking.org, 2018. [Online]. Available: <https://opennetworking.org/wp-content/uploads/2020/08/TR-547-TAPI-v2.1.3-Reference-Implementation-Agreement-1.pdf>.

30. Open Network Operating System (ONOS) SDN Controller for SDN/NFV Solutions, Open Networking Foundation, 2021. [Online]. Available: <https://opennetworking.org/onos/>.

31. Software-Defined Networks: A Systems Approach – Software-Defined Networks: A Systems Approach Version 2.1-dev documentation, Sdn.systemsapproach.org, 2021. [Online]. Available: <https://sdn.systemsapproach.org/index.html>.

32. Chapter 6: Network OS – Software-Defined Networks: A Systems Approach Version 2.1-dev documentation, Sdn.systemsapproach.org, 2021. [Online]. Available: <https://sdn.systemsapproach.org/onos.html>.

33. O. Romanov, N. Korniienko, I. Obod and I. Svyd. Construction of the SDN Control Level Based on ONOS // 2021 IEEE International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), 2021, pp. 127-132, doi: 10.1109/UkrMiCo52950.2021.9716691.

Надійшла до редколегії 12.07.2022

Відомості про авторів:

Романов Олександр Іванович – доктор технічних наук, професор, професор кафедри телекомунікацій, Інститут телекомунікаційних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Україна; email: a_i_romanov@ukr.net; ORCID: <https://orcid.org/0000-0002-8683-3286>

Свид Ірина Вікторівна – кандидат технічних наук, доцент, завідувач кафедри мікропроцесорних технологій і систем, Харківський національний університет радіоелектроніки, Україна; email: iryna.svyd@nure.ua; ORCID: <http://orcid.org/0000-0002-4635-6542>

Корнієнко Надія Ігорівна – студентка кафедри телекомунікацій, Інститут телекомунікаційних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Україна; email: nkornienko2000@ukr.net; ORCID: <https://orcid.org/0000-0002-8402-8603>

Романов Антон Олександрович – аспірант кафедри телекомунікаційних та радіоелектронних систем, Національний Авіаційний Університет, Україна; email: anton3329@gmail.com; ORCID: <https://orcid.org/0000-0002-3425-0441>