

АНАЛІЗ КОНЦЕПЦІЇ ПРОГРАМНО-КОНФІГУРОВАНИХ МЕРЕЖ ТА ПРОТОКОЛУ OPENFLOW

Dakova L.V., Dakov S.Yu. Analysis of the concept of software-defined networks and openflow protocol. Abstract. The article analyzes the concept of software-configurable networks and the OpenFlow protocol, aimed at highlighting the importance of implementing these technologies in modern information and communication systems. The authors put forward the idea of effective management of traffic and network resources through the use of software-controlled approaches, in particular, the use of the OpenFlow concept.

The authors substantiate the relevance of the study in the context of modern requirements for network infrastructure, where the expanding volume of data and diversity of traffic require new methods of management and adaptation. It is pointed out that traditional approaches to network management may be ineffective in the face of increasing complexity and volume of network traffic.

The article proposes a formalized approach to network traffic management, taking into account resource segmentation. The key components of the proposed model, such as resource segments, a mechanism for identifying and analyzing traffic for each segment, and prioritization of traffic based on management efficiency criteria, are highlighted. This approach can help to optimize the management of network resources and improve the overall efficiency of the infrastructure, taking into account the requirements for modern network technologies.

Key words: Analysis, software-configurable networks, OpenFlow protocol, traffic management, resource segmentation, network infrastructure, software-controlled approaches, key components, management efficiency, network traffic, network technologies, information and communication systems.

Дакова Л.В., Даков С.Ю. Аналіз концепції програмно-конфігурованих мереж та протоколу openflow. Анотація. У статті проводиться аналіз концепції програмно-конфігурованих мереж та протоколу OpenFlow, спрямований на висвітлення важливості впровадження цих технологій у сучасній інформаційно-комунікаційній системі. Автор висувують ідею ефективного керування трафіком та ресурсами мережі через використання програмно-керованих підходів, зокрема, застосуванням концепції OpenFlow.

Обґрунтовується актуальність дослідження у контексті сучасних вимог до мережевої інфраструктури, де розширюються обсяги даних та різноманітність трафіку вимагають нових методів керування та адаптації. Вказується на те, що традиційні підходи до управління мережею можуть виявитися неефективними в умовах зростання складності та обсягу мережевого трафіку.

Стаття пропонує формалізований підхід до управління мережевим трафіком, враховуючи сегментацію ресурсів. Висвітлено ключові компоненти запропонованої моделі, такі як сегменти ресурсів, механізм визначення та аналізу трафіку для кожного сегмента, а також пріоритетизація трафіку з урахуванням критеріїв ефективності управління. Цей підхід може сприяти оптимізації управління мережевими ресурсами та покращенню загальної ефективності інфраструктури з урахуванням вимог до сучасних мережевих технологій.

Ключові слова: Аналіз, програмно-конфігурована мережа, протокол OpenFlow, управління трафіком, сегментація ресурсів, інфраструктура мережі, програмно-керовані підходи, ключові компоненти, ефективність управління, мережевий трафік, технології мереж, інформаційно-комунікаційні системи.

Вступ.

У сучасному світі, який характеризується стрімким розвитком технологій та постійним прагненням до оптимізації мережевих інфраструктур, питання управління та конфігурації мережевих пристроїв набувають все більшого значення. Однією із перспективних концепцій в цьому контексті є програмно-конфігуровані мережі (Software-Defined Networking, SDN) та протокол OpenFlow, які визначають новий підхід до управління та контролю над мережевим середовищем. Спрощуючи архітектуру та роботу мережевих пристроїв, програмно-конфігуровані мережі надають можливість централізованого та гнучкого управління, що дозволяє ефективно впроваджувати нові сервіси та забезпечувати високий рівень масштабованості.

Протокол OpenFlow виступає ключовим елементом в реалізації концепції програмно-конфігурованих мереж, забезпечуючи комунікацію між контролером та периферійними пристроями.

Постановка проблеми

Ця стаття присвячена аналізу концепції програмно-конфігурованих мереж та протоколу OpenFlow, розглядаючи їхні переваги, аспекти реалізації, а також виклики, з якими можуть зіткнутися організації при впровадженні цих технологій. Зробимо глибокий занурення у фундаментальні принципи та практичні аспекти цих інноваційних рішень для ефективного управління мережевими інфраструктурами.

Аналіз досліджень і публікацій

Програмно-конфігуровані мережі (SDN) та протокол OpenFlow є ключовими компонентами в розвитку сучасних мережевих технологій. SDN спрямована на створення гнучких та легко керованих мереж шляхом відокремлення управлінської логіки від функцій пересилання даних. Основні елементи SDN включають контролер, який управляє керуючим планом, та пересилаючі пристрої, що виконують команди контролера [1 - 6].

Протокол OpenFlow використовується для забезпечення комунікації між контролером та пересилаючими пристроями [7 - 10]. Він передбачає програмне встановлення правил пересилання (flow entries) на пристроях, які дозволяють контролеру визначати шляхи для пересилання даних. Ключовим елементом є flow table, де визначаються ці правила.

Переваги SDN та OpenFlow включають гнучкість управління мережею та швидку адаптацію до змін. Однак вони також мають недоліки, такі як залежність від працездатності контролера та потреба в стабільній комунікації між контролером та пристроями, які відправляють інформацію.

Застосування SDN та OpenFlow розповсюджуються в області центрів обробки даних для ефективного масштабування та управління трафіком, а також у мережах для гнучкого управління ресурсами та маршрутизації [11 - 13].

Мета дослідження

Метою дослідження є ретельний аналіз концепції програмно-конфігурованих мереж (SDN) та протоколу OpenFlow з метою розкриття їхньої структури, принципів роботи та потенційних вигод для сучасних мережевих інфраструктур. Дослідження спрямоване на визначення ключових аспектів SDN, таких як розділення управлінської логіки та функцій пересилання даних, архітектурні компоненти, такі як SDN-контролер та пристрої, і розгляд принципів взаємодії між ними через протокол OpenFlow.

Виклад основного матеріалу дослідження

Самоорганізуючі мультиагентні моделі управління в мережах NGN. Незважаючи на те, що технологія програмно-конфігурованих мереж на перший погляд є досить новою, вона має витоки у тривалій історії розвитку мережевих технологій. Сучасна концепція програмно-конфігурованих мереж є завершальним етапом довгого процесу зусиль, спрямованих на те, щоб зробити мережі програмованими. Телекомунікаційні мережі завжди відрізнялися великою складністю структури та управління. Для їх побудови використовується різноманітне обладнання, таке як комутатори, маршрутизатори, брандмауери, транслятори мережевих адрес і багато іншого. Традиційно адміністратори мережі налаштовували кожен пристрій окремо, використовуючи різні інтерфейси для конфігурації, що призводило до складнощів [1].

Введення програмно-конфігурованих мереж вносить кардинальні зміни у підхід до проектування та управління мережами. По-перше, концепція SDN відокремлює площину управління мережею від площини передачі даних, дозволяючи централізовано керувати маршрутизацією трафіку. По-друге, SDN об'єднує площину управління, дозволяючи одній

керуючій програмі ефективно керувати багатьма пристроями на площині даних за допомогою стандартизованого інтерфейсу API, такого як OpenFlow.

Незважаючи на те, що концепція SDN набула популярності останнім часом, її ідеї існують вже понад дві десятиліття. До певної міри ці ідеї можна відстежити в розвитку ранніх телефонних мереж, коли управління мережею було відокремлене від мережі каналної комутації. Підходи, що передували SDN, такі як концепція "гнучких комутаторів" для телефонних мереж, демонструють схожість у функціональності та реалізації.

Із зростанням інтересу до програмно-конфігурованих мереж, особлива увага приділяється динамічній віртуалізації ресурсів та оркестрації сервісів. Це дозволяє застосовувати архітектуру SDN у різноманітних областях, таких як підприємства, операторські мережі, центри обробки даних і кампуси, а також адаптувати її до різних умов та вимог користувачів [2].

Традиційні комп'ютерні мережі складаються з взаємопов'язаних мережевих пристроїв, таких як комутатори і маршрутизатори (Рисунок 1). Кожен з цих пристроїв має свій власний механізм передачі на транспортному рівні та власну систему управління, що включає операційну систему та різноманітні додатки. У такій моделі мережеві пристрої мають закриту архітектуру, і відсутність можливості додавання нових функцій. Прив'язка до конкретного виробника мережевого обладнання не гарантує підтримки майбутніх програм і сервісів [3].

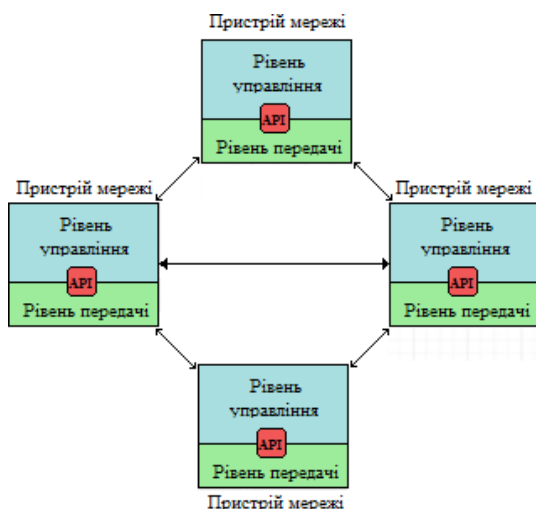


Рисунок 1 Структура традиційної мережі передачі даних

Архітектура програмно-конфігурованих мереж, яка описана в ONF TR-502 [6], базується на розділі функцій управління та передачі трафіку. Вона передбачає централізоване управління, де зовнішнє управління виглядає як єдина система, що забезпечує ефективне використання ресурсів мережі. Також відзначається тим, що сервіси мережі можна програмувати, де інтерфейси між об'єктами SDN розкривають абстракції ресурсів і стану мережі, сприяючи обміну інформацією та управлінню мережею. Ці принципи визначають архітектурний підхід до створення гнучких та продуктивних програмно-конфігурованих мереж [4].

Архітектура програмно-конфігурованих мереж, як видно на рисунку 2.2, включає в себе шість основних компонентів. Перший компонент, розташований на верхньому рівні, відомий як рівень додатків або прикладний рівень, використовується для реалізації різних служб, таких як системи виявлення і запобігання вторгнення, забезпечення якості обслуговування, контроль доступу і проксі-сервер [5].

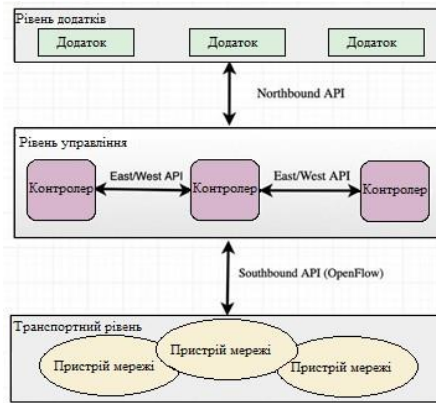


Рисунок 2 Структура мережі SDN

Другий компонент, рівень управління, абстрагує топологію мережі від рівня додатків через northbound API. Контролер SDN, що є ключовим елементом на цьому рівні, координує роботу мережевих пристроїв на рівні передачі даних, використовуючи southbound API, такий як протокол OpenFlow [6]. Тут також існують eastbound і westbound API, які дозволяють обмінюватися інформацією між контролерами мережі. Третій компонент, транспортний рівень, відповідає за обробку і передачу пакетів відповідно до інструкцій, отриманих від рівня управління. Північні та південні API-інтерфейси (Northbound і Southbound interfaces) представляють важливі засоби зв'язку для додатків та контролерів SDN, де північні API дозволяють додаткам використовувати мережеві сервіси, а південні API забезпечують ефективне управління ресурсами мережі. Східні/Західні API-інтерфейси забезпечують комунікацію між різними SDN-контролерами, дозволяючи їм обмінюватися інформацією, пов'язаною з обробкою трафіку на рівні передачі даних.

Ключовим елементом в архітектурі SDN є контролер, який централізує функції мережі і слідкує за її глобальним станом. Цей контролер може бути представлений як апаратно-програмне рішення або програмною реалізацією, причому останній варіант є більш поширеним на ринку. Виділяючи управління мережевим обладнанням і використовуючи програмне забезпечення для цих завдань, контролер спрощує автоматизоване керування мережею і інтеграцію бізнес-додатків [7].

Початкові архітектурні варіанти SDN передбачали єдиний централізований контролер на рівні управління. Згодом з'явилися розподілені архітектури, які використовують кілька контролерів для поліпшення продуктивності і масштабованості мережі.

Основні характеристики контролера програмно-конфігурованих мереж включають продуктивність, час обробки, надійність, ресурсомісткість та масштабованість. Вони визначають ефективність його роботи та можливість обслуговування різноманітних вимог та умов мережі.

Зазвичай SDN-контролер має різні модулі, які виконують різноманітні мережеві завдання. Серед них модулі виявлення каналів, топології, пам'яті, вироблення стратегії, таблиці потоків і управління даними. Ці модулі сприяють виявленню, управлінню та оптимізації мережевих ресурсів.

Для надання послуги маршрутизації використовуються модулі топології і виявлення каналів, які взаємодіють з контролером, створюючи базу даних топології та обчислюючи маршрути в мережі. На сьогоднішній день існує різноманіття відкритих контролерів, що підтримують протокол OpenFlow і відрізняються за мовами програмування, продуктивністю та багатьма іншими.

Протокол OpenFlow. Протокол OpenFlow, розроблений ONF для стандартизації взаємодії контролера та мережевих пристроїв у архітектурі програмно-керованих мереж (ПКМ), представляє собою відкритий стандарт. У вигляді програмного модуля він інтегрується у

апаратне забезпечення Ethernet комутаторів, маршрутизаторів і бездротових вузлів, що розширює їхні можливості для використання в ПКМ.

На сьогоднішній день стандарт OpenFlow широко визнаний і прийнятий більшістю виробників мережевого обладнання, таких як Cisco, Juniper, Brocade, Huawei, Zela та інші. Ці виробники надають комутатори з вбудованою підтримкою OpenFlow, що робить їх готовими для використання в програмно-керованих мережах.

Повідомлення протоколу OpenFlow поділяються на три основні типи:

1. Повідомлення контролер-комутатор: ініціюються контролером та служать для управління комутатором та контролю подій, які відбуваються на ньому. Ці повідомлення включають запити щодо можливостей комутатора, конфігураційні запити, керування станом комутаторів та інші.

2. Повідомлення асинхронного конфігураційного типу: встановлюють фільтри на асинхронні повідомлення від комутаторів за допомогою контролера. Це дозволяє контролеру здійснювати більш гнучке керування подіями в мережі.

Протокол OpenFlow дозволяє ефективну взаємодію контролера та мережевих пристроїв, що робить його ключовим елементом у реалізації концепції програмно-керованих мереж [8].

Перший вид повідомлень - асинхронні повідомлення, ініційовані OpenFlow-комутаторами, призначені для сповіщення контролера про події в мережі, такі як збої, помилки чи зміни стану. Ці повідомлення включають:

1. Повідомлення Packet-in: викликається комутатором і відсилається контролеру, якщо отриманий пакет не має відповідного правила в таблиці комутації. Усі пакети, які пересилаються в віртуальний порт, генерують повідомлення Packet-in на контролер.

2. Повідомлення Flow-Removed: служить для вилучення правил, які не використовуються та є неактивними.

3. Повідомлення Port-status: генерується комутатором та відправляється контролеру при зміні налаштувань порту.

4. Повідомлення Error: використовується контролером для повідомлення про помилки чи збої.

Другий вид повідомлень - симетричні повідомлення, які розсилаються як комутаторами, так і контролером. До цього виду повідомлень відносяться наступні:

1. Повідомлення Hello: служить для обміну між комутатором і контролером при встановленні з'єднання.

2. Повідомлення Echo: має форму запит/відповідь і може ініціюватися як контролером, так і комутатором, з обов'язковою відповіддю. Також може використовуватися для вимірювання затримок чи пропускну здатності з'єднання контролер-комутатор та перевірки ефективності з'єднання.

3. Повідомлення Experimenter: представляє функціональність для проведення експериментів в просторі типів повідомлень OpenFlow.

Порти OpenFlow. Пакети OpenFlow подаються через вхідні порти та обробляються конвеєром, який направляє їх на вихідні порти, використовуючи вихідні дії. Порти OpenFlow є мережевими інтерфейсами для обміну пакетами між обробкою OpenFlow та іншими частинами мережі [9].

Комутатор OpenFlow має три типи портів:

1. Фізичні порти: відповідають апаратним інтерфейсам комутатора.

2. Логічні порти: не відповідають безпосередньо апаратним інтерфейсам комутатора. Обробка, яку вони виконують, залежить від реалізації і має бути прозорою для обробки OpenFlow.

3. Зарезервовані порти: визначають загальні дії пересилки, такі як відправка контролеру, розсилка чи пересилання за допомогою методів, відмінних від OpenFlow. Серед них є обов'язкові порти (ALL, CONTROLLER, TABLE, IN_PORT, ANY, UNSET) і опціональні (LOCAL, NORMAL, FLOOD)

Канал OpenFlow. Канал OpenFlow використовується для обміну повідомленнями між комутатором та контролером OpenFlow. Кожен контролер може управляти кількома каналами

для різних комутаторів, і, відповідно, комутатор може мати канали для декількох контролерів [10].

Зазвичай канал OpenFlow є як одне мережеве з'єднання між комутатором і контролером, використовуючи протокол TLS або TCP для гарантії підтримки з'єднань OpenFlow в різних мережевих умовах [11]. Також можливе складання каналу з декількох допоміжних мережевих з'єднань, заснованих на протоколах TLS, TCP, DTLS і UDP.

Кожне з'єднання обробляється окремо. Якщо з'єднання з певним контролером або комутатором втрачено, це не призводить до припинення з'єднань з іншими. Однак при завершенні або розірванні основного з'єднання усі відповідні допоміжні з'єднання також припиняються [13].

При перерві з'єднання через мережеві умови або таймаут, комутатор або контролер спробує встановити нове з'єднання. Якщо комутатор втрачає контакт з контролерами, він надсилає повідомлення Controller-Status іншим підключеним контролерам, а при відновленні з'єднання відправляє їм оновлене повідомлення про стан контролера [13].

Якщо комутатор втрачає контакт із усіма контролерами, він переходить в "fail secure mode" або "fail standalone mode" відповідно до реалізації та конфігурації.

Висновки і перспективи подальших досліджень

У результаті детального аналізу концепції програмно-конфігурованих мереж та протоколу OpenFlow, виявлено важливі аспекти, спрямовані на вдосконалення та оптимізацію сучасних мережевих інфраструктур. Зроблено основний акцент на тому, як ці технології взаємодіють з актуальними викликами управління трафіком, а також конфігурацією мережевих пристроїв.

Проаналізовано можливості протоколу OpenFlow у вирішенні завдань централізованого керування та підкреслила його роль у підвищенні гнучкості та адаптивності мережі. Розглянуті позитивні аспекти використання цього протоколу, спрямовані на оптимізацію розподілу трафіку та конфігурації мережевих пристроїв, що сприяє поліпшенню продуктивності та забезпеченню ефективного використання ресурсів.

Список використаної літератури:

1. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., ... & Turner, J. (2008). OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2), 69-74.
2. Casado, M., Freedman, M. J., Pettit, J., Luo, J., McKeown, N., & Shenker, S. (2007). Ethane: Taking control of the enterprise. *ACM SIGCOMM Computer Communication Review*, 37(4), 1-12.
3. Kim, H., Feamster, N., & McKeown, N. (2008). Improving network management with software defined networking. *IEEE Communications Magazine*, 51(2), 114-119.
4. Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1), 14-76.
5. Open Networking Foundation. (2012). OpenFlow Switch Specification Version 1.3.0 (Wire Protocol 0x04). Retrieved from <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.3.0.pdf>
6. Kim, J., Shin, S., & Kim, Y. (2013). SDN-based virtual network embedding: A review. *IEEE Communications Surveys & Tutorials*, 15(4), 1842-1858.
7. Reitblatt, M., Foster, N., Rexford, J., & Walker, D. (2013). Abstractions for network update. *ACM SIGPLAN Notices*, 48(4), 333-344.
8. Yu, M., Rexford, J., Freedman, M. J., & Wang, J. (2010). Scalable flow-based networking with DIFANE. *ACM Transactions on Computer Systems (TOCS)*, 28(4), 12.
9. Berde, P., Gerola, M., Vattikonda, A., Hart, J., Higuchi, Y., Kobayashi, M., ... & Parulkar, G. (2014). ONOS: Towards an open, distributed SDN OS. In *Proceedings of the third workshop on Hot topics in software defined networking* (pp. 1-6).

10. Voellmy, A., & Hudak, P. (2011). Nettle: Taking the sting out of programming network routers. In Proceedings of the 6th ACM SIGPLAN workshop on Programming languages and analysis for security (pp. 13-24).
11. Casado, M., & Pettit, J. (2011). OfVis: A tool for visualizing openflow-based network. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking (pp. 97-102).
12. Duan, Q., Guo, C., Wu, H., Zhou, Y., & Shi, L. (2012). A survey on data center networking. IEEE Communications Surveys & Tutorials, 14(4), 1018-1034.
13. Tootoonchian, A., & Ganjali, Y. (2010). HyperFlow: A distributed control plane for OpenFlow. In Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking (pp. 3-10).

Автори статті

Дакова Лариса – кандидат технічних наук, доцент, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

Даков Сергій – кандидат технічних наук, Київський національний університет імені Тараса Шевченка

Authors of the article

Dakova Larysa - Candidate of Science (technic), associate professor, State University of Information and Communication Technologies, Kyiv, Ukraine.

Dakov Serhii – Candidate of Science (technic), Taras Shevchenko National University of Kyiv, Kyiv, Ukraine.