

MININET: СИМУЛЯТОР ДЛЯ СТВОРЕННЯ І ПРОЕКТУВАННЯ ПРОТОТИПІВ МЕРЕЖ SDN

Бурлака Г.Ю.

*Інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Україна
E-mail: fastmotion1111@gmail.com*

MININET: A SIMULATOR FOR EMULATING AND PROTOTYPING SOFTWARE-DEFINED NETWORKS

The concept of software-defined networking (SDN) fundamentally changes the way networks operate and how they are managed. In the rapidly changing modern world, it is the data transmission networks that have been called the "thin link" that limits the growth of application performance as the number of mobile users grows, virtual environments scale, and clusters are formed for Big Data. In SDN networks, traffic switching tasks and control tasks are strictly separated. All control logic is centralized and transferred to the controller.

Останнім часом сильно змінилися концепції обробки і все частіше мережа стає обмежуючим фактором розвитку обчислювальної інфраструктури. Причина полягає в тому, що мережі є статичними, на відміну від серверів, які зобов'язані цим технології віртуалізації. Якщо необхідно, наприклад, внести одну невелику зміну на шляху від однієї віртуальної машини до іншої, то максимум, чого можна домогтися, автоматизації розгортання необхідних налаштувань на обладнанні одного модельного ряду конкретного виробника. Для вирішення цих проблем була розроблена концепція SDN (Software Defined Networks) [1].

Концепція програмно-визначених мереж (SDN) ґрунтовно змінює принципи функціонування мереж і їх управління. У швидко мінливому сучасному світі саме мережі передачі даних були названі «тонким ланкою», яке обмежує зростання продуктивності додатків по мірі зростання кількості мобільних користувачів, масштабування віртуальних середовищ, формування кластерів для Великих Даних. У SDN мережах завдання комутації трафіку і завдання управління строго розділені.

Вся логіка управління централізується і передаються контролеру. Комутатор в концепції SDN - досить примітивний пристрій, який відповідає тільки за перемикання пакетів на підставі дуже простих правил. Контролер SDN управляє всіма комутаторами в мережі і програмує кожен з них для правильної передачі трафіку[2]. Централізація логіки управління дозволяє програмувати мережу як єдине ціле і спростити операційну модель великих корпоративних мереж, які занадто статичні на даний момент і не відповідають сучасному бізнесу, з властивими йому мобільністю користувачів / пристроїв / додатків, розподілом додатків між віртуальними машинами і інтенсивним обміном даними.

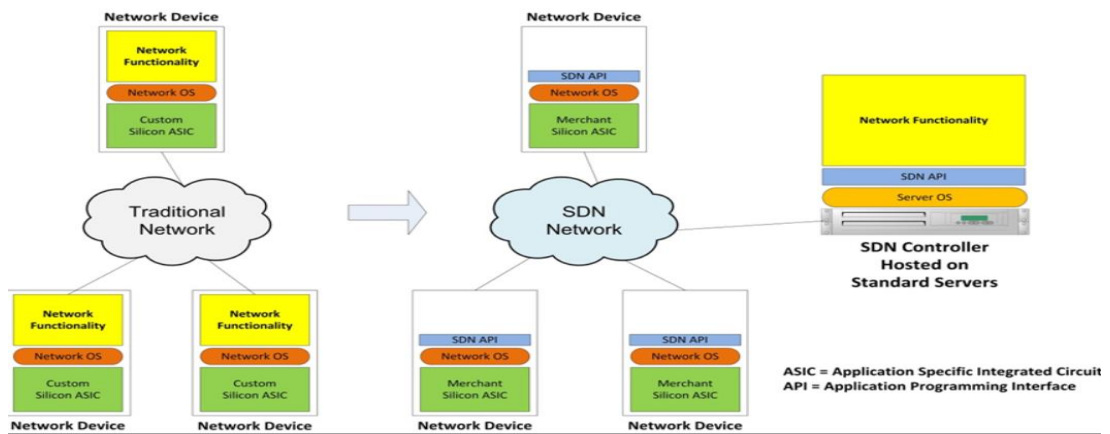


Рис. 1. Перехід від традиційної мережі до SDN.

Завдяки контролеру з розширеними API інтерфейсами, вся мережа стає подібною одному великому логічному комутатору. Протокол OpenFlow - один з найбільш універсальних протоколів комунікації контролерів і комутаторів на сьогоднішній день надає стандартний підхід до програмування таблиць комутації, в яких основним об'єктом є потік даних. Однак в той час як OpenFlow дозволяє контролеру програмувати комутатори, він не визначає, як контролера реагувати і відповідати на виклики, пов'язані з хмарами, BYOD, BYOC, віртуалізацією. Рішення будь-яких проблеми з продуктивністю мережі, та й просто її роботою, покладені на контролер і додатки, які на ньому запуснені. Саме набір таких додатків і відрізняє різні впровадження SDN, а також рішення різних виробників [3].

Для проектування і тестування мереж SDN дуже зручно використовувати симулятор Mininet (рис. 2). Він дозволяє створити мережу віртуальних хостів, комутаторів, контролерів і посилок. Хости Mininet запускають стандартне мережеве програмне забезпечення Linux, а його комутатори підтримують OpenFlow для дуже гнучкою, що налагоджується маршрутизації і програмно визначається мережі. Mininet підтримує дослідження, розробки, навчання, створення прототипів, тестування, налагодження і будь-які інші завдання, які можуть виграти від наявності повної експериментальної мережі на ноутбуці або іншому ПК. Mininet надає простий спосіб отримати правильну поведінку системи (і, наскільки це підтримується вашим обладнанням, продуктивність), а також поекспериментувати з топологіями.

У мережах Mininet працює реальний код. Через цей код ви розробляєте і тестуєте на Mininet, для контролера OpenFlow, модифікованого комутатора або хоста, можете перейти на реальну систему з мінімальними змінами для реального тестування, оцінки продуктивності та розгортання. Важливо, що це означає, що проект, який працює в Mininet, зазвичай може переходити безпосередньо до апаратних комутаторів для пересилання пакетів на лінійній швидкості [4].

Mininet може створювати комутатори OpenFlow в ядрі або призначеному для користувача просторі, контролери для управління комутаторами і хости для зв'язку з модельованою мережі. Mininet з'єднує комутатори та хости за

допомогою віртуальних пар Ethernet (veth). Код Mininet майже повністю написаний на Python, за винятком невеликої утиліти C .

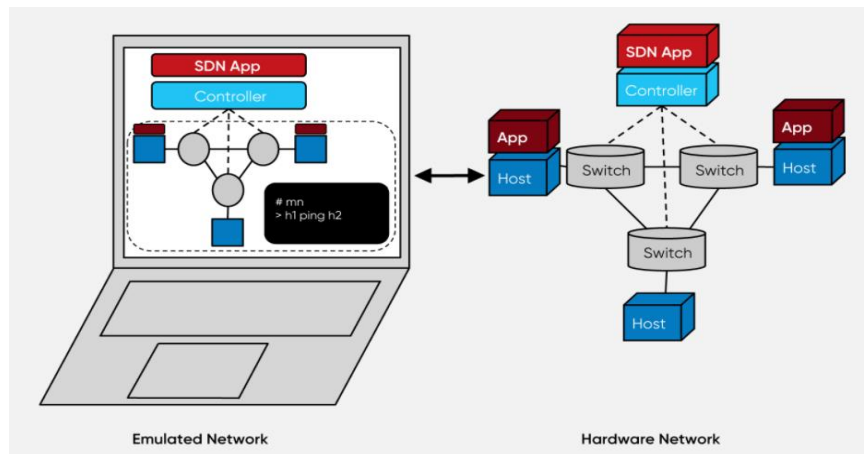


Рис. 2. Структурно - функціональна схема Mininet.

Коли дві програми, такі як клієнт і сервер iperf, обмінюються даними через Mininet, виміряна продуктивність повинна відповідати продуктивності двох (більш повільних) власних машин. Простіше кажучи, віртуальні хости, комутатори, послання і контролери Mininet справжні - вони просто створюються з використанням програмного забезпечення, а не обладнання - і здебільшого їх поведінку схоже на дискретні елементи обладнання [5].

Таким чином, технологія Mininet зручна для створення прототипів і моделювання. Це альтернатива тестуванню реальних мереж, які потребують великих затрат. Створення віртуальної мережі в Mininet дозволяє швидко протестувати системи різної конфігурації

Література

1. Bruno Astuto A. Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, Thierry Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks", Vous Consultez L'Archive Hal, 19 Jan, 2014.
2. McKeown, Nick, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. "OpenFlow: enabling innovation in campus networks." ACM SIGCOMM Computer Communication Review 38, No. 2 (2008): 69–74.
3. Mininet as Software Defined Networking Testing Platform [Електронний ресурс] — Режим доступу: https://www.researchgate.net/publication/287216738_Mininet_as_Software_Defined_Networking_Testing_Platform.
4. Carapinha, J., Feil, P., Weissmann, P., Thorsteinsson, S., Etemoglu, Ç. Ingþórsson, Ó., Çiftçi, S., Melo, M., "Network Virtualization: Opportunities and Challenges for Operators," Future Internet-FIS 2010: 138–147, 2012.
5. Using Mininet for emulation and prototyping Software-Defined Networks [Електронний ресурс] — Режим доступу: https://www.researchgate.net/publication/269292945_Using_Mininet_for_emulation_and_prototyping_Software-Defined_Networks.