

ДОСЛІДЖЕННЯ ПОКАЗНИКІВ ФУНКЦІОНУВАННЯ МЕРЕЖІ SDN З ВИКОРИСТАННЯМ MININET I FLOODLIGHT CONTROLLER

Романов О.І., Марінов А.І., Сколець С.С.

Інститут телекомунікаційних систем КПІ ім. Ігоря Сікорського, Україна
E-mail: a_i_romanov@ukr.net, marinov.anton99@gmail.com, serskols@gmail.com

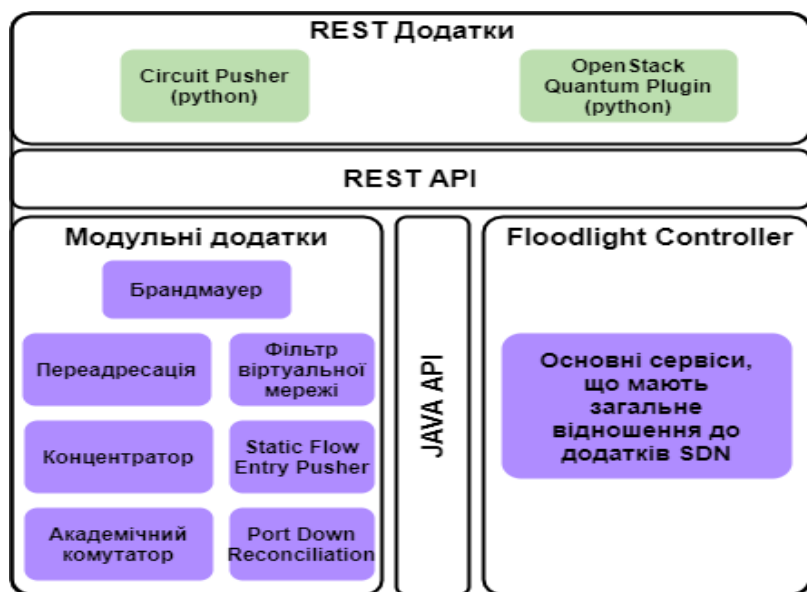
RESEARCH OF SDN PERFORMANCE USING MININET AND FLOODLIGHT CONTROLLER

Comparison of the results gained from the research of bandwidth in the transmission of TCP and UDP traffic, as well as checking for the availability of the balancing function on controller. Discovering and summarizing the results of the RTT research at the different values of delay in transmission channels. Summarizing the results of building a network on a Floodlight controller.

Floodlight Controller розроблений відкритою спільнотою розробників, які підтримують протокол OpenFlow. Моделювання мереж SDN на платформі Mininet з використанням Floodlight Controller має ряд переваг [1]:

- дає можливість легко адаптувати та розробляти програми, які написані на Java;
- включені інтерфейси прикладних програм REST API, які спрощують програмування інтерфейсу;
- веб-сайт Floodlight надає приклади кодування, які допомагають розробникам у створенні продукту;
- працює як з фізичними, так і віртуальними комутаторами, сумісними з OpenFlow;
- може підтримувати мережі на традиційних комутаторах, які не є OpenFlow.

Структура і основні функціональні елементи Floodlight Controller представлені на рис 1.



* Модульні додатки - додатки з вищою пропускну здібністю для комунікації з контролером, такі як Packel's

Рис. 1. Функціональна структура контролера.

Розглянемо призначення основних елементів, REST додатки – програми на будь-якій мові, що використовують сервіси за допомогою REST API, що керуються модулями контролера та модульними додатками [2, 3]. Розглянемо призначення основних елементів, REST додатки – програми на будь-якій мові, що використовують сервіси

за допомогою REST API, що керуються модулями контролера та модульними додатками [2, 3]. Контролер Floodlight (рис.1) сумісний з OpenStack набором

програмних інструментів, які дають можливість створювати та керувати платформами хмарних обчислень для загально користувацьких та приватних хмар. Floodlight можна запустити як мережевий сервер для OpenStack. Плагін Neutron надає модель мережі.

Circuit Pusher з REST API забезпечує управління потоками пакетів на всіх комутаторах на основі IP-адрес із заданим пріоритетом.

Брандмауер - програма для застосування правил ACL для дозволу / заборони трафіку на основі вказаного збігу (відповідності). Переадресація – програма переадресації пакетів Floodlight за замовчуванням. Концентратор – програма-концентратор, яка завжди пересилає всі вхідні пакети на всі інші активні порти. Навчальний комутатор – типовий комутатор рівня L2. Виставляє REST API для таблиці комутаторів (відомі хости: vlan: порт). Фільтр віртуальної мережі – простий додаток для ізоляції мережі на основі MAC. Виставляє набір REST API для додавання, видалення та запитування віртуальних мереж. Static Flow Entry Pusher – додаток для встановлення конкретного входу потоку (співвідношення + дія) на певний комутатор. Виставляє набір REST API для додавання, видалення та запиту записів потоку. Port Down Reconciliation – додаток для узгодження потоків через мережу, коли порт або посилення перестають працювати.

В роботі на базі використання Mininet і Floodlight Controller були досліджені наступні параметри мережі:

- Пропускна здатність каналів мережі при передаванні пакетів UDP та TCP трафіку;
- балансування трафіку Floodlight контролером в режимі перевантаження каналів;
- затримка пакетів RTT при різному навантаженні каналів.

При дослідженні пропускної здатності по каналах велася передача пакетів UDP та TCP з використанням інструментів iperf (для TCP протоколу) та iperfudp (для UDP протоколу).

Таблиця 1. Результати дослідження пропускної здатності

Відрізок передачі	Задане навантаження	Експериментальне навантаження	Розмір переданого трафіку	Середня затримка пакетів	Кількість переданих/втрачених пакетів
Від H1 до H5	6 Мбіт/с	3.7 Мбіт/с	14 МБ	18.621 мс	15201/5238 (34% втрат)
Від H3 до H5	4 Мбіт/с	2.4 Мбіт/с	9.03 МБ	1.787 мс	10178/3741 (37% втрат)
Від H4 до H5	5 Мбіт/с	2.97 Мбіт/с	11 МБ	1.683 мс	12674/4847 (38% втрат)

В цьому експерименті ми використовували 5-й вузол (Host 5) у якості серверу який приймає трафік від трьох клієнтів, якими є 1-й, 3-й та 4-й вузли (відповідно Host 1, Host 3, Host 4). При первинному налаштуванні пропускна здатність усіх каналів в мережі було встановлено 10 Mbit/s. Далі було створено навантаження для кожного клієнта так, щоб було наявне перевантаження. Наприклад, гілка мережі має пропускну здатність 10 Mbit/s, а сумарне навантаження складає 15 Mbit/s. Тобто ціль в тому щоб трафік не пішов по одному найкоротшому шляху, а потрібно було його балансування. В нас частина відправлених пакетів розподілилась на відрізок передачі від sw-1 до sw-4, щоб запобігти появі втрат на відрізку передачі від sw-1 до sw-2.

Отримані результати наведені в таблиці 1. Візуально процес перевантаження зображений на рис. 3.

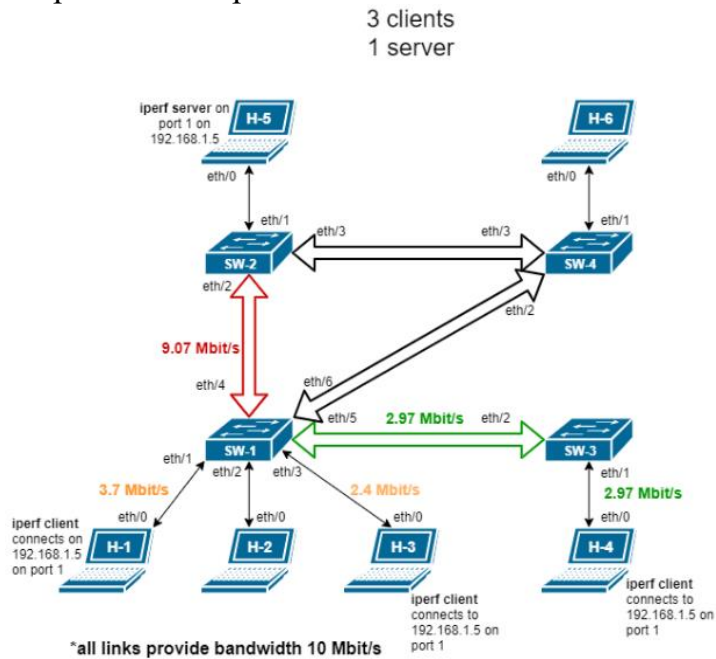


Рис. 3. Процес перенавантаження каналу зв'язку в досліджуваній мережі.

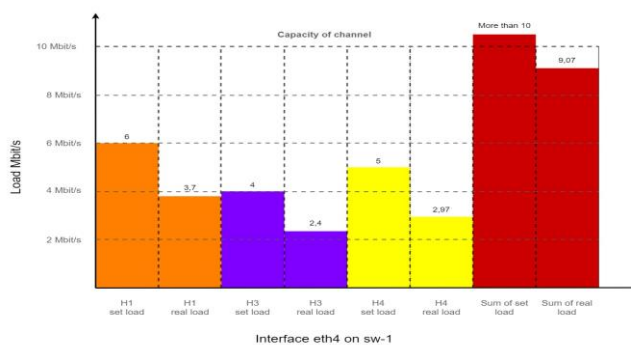


Рис.4. Навантаження на інтерфейсі eth4 комутатора sw-1.

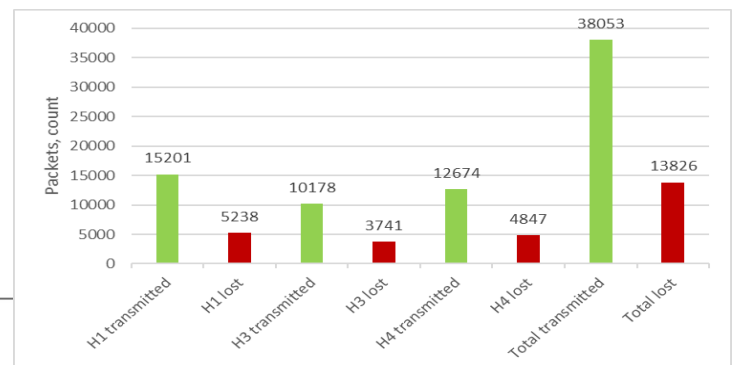


Рис. 5 Статистика передачі пакетів між клієнтами та сервером

За визначення параметру RTT відповідає команда *pingall*. Але для отримання більш розгорнутої статистики використовують команду: *pingallfull*.

Результати дослідження RTT наведені в табл.2.

Таблиця 2. Результати дослідження RTT.

Затримка, мс	Середній RTT, мс	Кількість робочих напрямків зв'язку	Кількість відказаних відрізків
0	0.146	30	0
2.5	22.266	30	0
5	38.578	30	0
50	308.414	30	0
100	909.721	30	0
200	1792.005	30	0
400	3358.825	28	2
450	3693.478	27	3
500	3131.509	18	12

Візуалізація отриманих результатів представлено на рис.6 і рис.7.

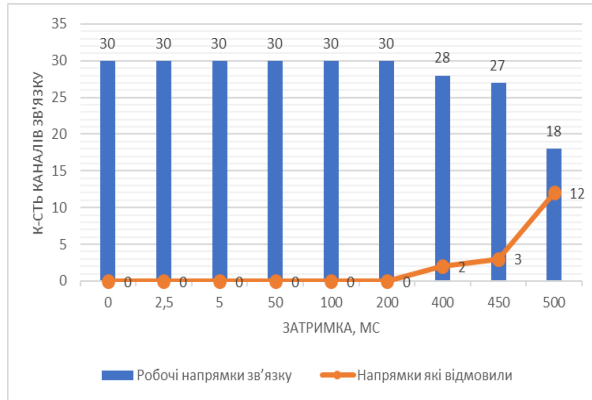


Рис. 6. Залежність кількості робочих і відмовлених напрямків зв'язку при різних затримках в мережі.

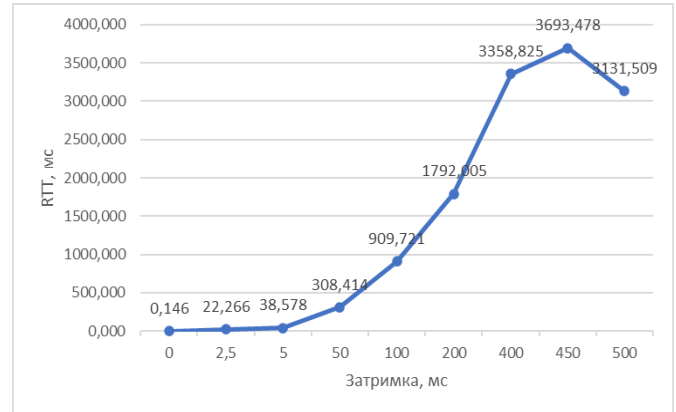


Рис. 7. Залежність зміни RTT при різних затримках в мережі.

Як видно з рис 6, коли передаються пакети зі встановленою затримкою в мережі 400 мс, починають відбуватися перші відключення напрямків зв'язку. А саме відмовляють 2 напрямки з найбільшою кількістю хопів, при 30-ти напрямках зв'язку в мережі (де хоп це «відстань» між двома вузлами в мережі, тобто чим більше хопів — тим складніший шлях маршрутизації і тим «далі» знаходяться вузли один від одного).

Відповідно на рис 7, можна спостерігати спад RTT при затримці 450 мс, що обумовлено відмовою 3-х каналів. При затримці в 500 мс, бачимо набагато більшу кількість відмовлених, а саме 12 напрямків зв'язку.

Також, в результаті проведення дослідів при встановленні всіх вищезазначених затримок (табл. 2), нами були помічені лінії зв'язку з найбільшими затримками: h1-h4, h1-h5, h1-h6, h2-h4, h2-h5, h2-h6, h3-h4, h3-h5, h3-h6, h4-h5, h4-h6, h5-h4, h5-h6, h6-h4.

Таким чином, використання Mininet дозволяє моделювати достатньо складні мережі SDN і отримати важливі показники їх функціонування, такі як пропускна здатність, якість обслуговування, затримка пакетів з використанням різних протоколів, наприклад TCP, UDP та інші. Mininet є у відкритому доступі. Тому вона заслуговує більш детального вивчення.

Література

1. C. Fancy and M. Pushpalatha, "Performance evaluation of SDN controllers POX and floodlight in mininet emulation environment," 2017 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 2017, pp. 695-699, doi: 10.1109/ISS1.2017.8389262;
2. О.І. Романов, М.М. Нестеренко, В.Б. Маньківський, І.О. Сайченко Модель оптимального розподілу навантаження в мережі доступу мобільного оператора//Вісник Черкаського державного технологічного університету № 3 (2020), С.20-29, DOI: <https://doi.org/10.24025/2306-4412.3.2020>
3. Lantz, Bob, Brandon Heller, and Nick McKeown. "A network in a laptop: rapid prototyping for software-defined networks." In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, pp. 19. ACM, 2010;
4. Romanov O.I, Nesterenko M.M., Fesokha N.O., Mankivskyi V.B. Evaluation of productivity virtualization technologies of switching equipment telecommunications networks. Information and Telecommunication Sciences, 2020, Volume 11, Number 1 (20), page 53 – 58.