

**Олещенко Л.М.**

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

**Мовчан К.О.**

Український науково-дослідний інститут спеціальної техніки та судових експертиз  
Служби безпеки України

**Замковий І.Т.**

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

## ПРОГРАМНИЙ МЕТОД ДЛЯ ТЕСТУВАННЯ ТА АДМІНІСТРУВАННЯ МЕРЕЖ SDN

*Щодня відбувається збільшення обсягу мережевого трафіку і все частіше виникає потреба використовувати великомасштабні мережі, які потребують багато часу і ресурсів для їх налаштування та обслуговування. Автоматизація процесу управління мережею допомагає вирішити згадані проблеми. Наявні мережні архітектури орієнтовані на застарілі протоколи, що стають на заваді масштабуванню, швидкому додаванню нових пристроїв та впровадженню інновацій. Саме тому останніми роками зріс попит на нову мережну архітектуру програмно-конфігурованих мереж SDN. Основними перевагами мереж SDN є програмованість, гнучке централізоване керування та можливість масштабування.*

*У цій роботі використовується контролер OpenFlow для взаємодії між мережею та серверною частиною програмного забезпечення. Проаналізовано найбільш популярні контролери: OpenDaylight, Veason та Cisco Open SDN Controller; описано загальну концепцію програмно-конфігурованих мереж і сформульовано основні вимоги до розроблюваного програмного забезпечення. Для серверної частини вибрано мову C#, фреймворк ASP.NET Core, для клієнтської частини вибрано фреймворк Angular з мовою TypeScript. Використано контролер OpenDaylight Project як додаткова технологія для взаємодії, налаштування та роботи з програмно-конфігурованою мережею. Для програмного забезпечення використано клієнт-серверну архітектуру, описано ключові модулі та сутності системи. Реалізовано ключову функціональність та реалізовано обробку повідомлень з програмно-конфігурованої мережі. Описано загальний принцип роботи програмного забезпечення з мережею та наведено приклад роботи адміністратора мережі, змодельованої у середовищі Mininet.*

**Ключові слова:** програмне забезпечення, SDN, OpenFlow, OpenDaylight Project, мережевий пристрій, IP-адресація, Mininet.

**Постановка проблеми.** Натепер мережі SDN (Software-Defined Networking) є одним з найперспективніших та найбільш швидкозростаючих напрямів розвитку мережевих технологій. Відповідно до моделі мережі SDN мережеві пристрої переходять під контроль програмного забезпечення, що дозволяє зробити їх більш інтелектуальними та полегшувати їх управління. Але, незважаючи на стрімкий перехід іноземних компаній на таку технологію, в Україні вона натепер не є досить популярною. Причина цього полягає у відсутності відкритого програмного забезпечення та складності переходу від стандартної мережевої моделі до моделі SDN. Таким чином, метою цього дослідження є створення програмного забезпечення, яке б дозволяло здійснювати швидке нала-

штування та управління мережею SDN, моніторити мережевий трафік та підключати у мережу нових користувачів.

Поточна архітектура SDN базується на протоколі OpenFlow, який відокремлює управління та пересилання таких функцій традиційного перемикачів на окремі логічні суб'єкти [1–2]. Програмна конфігурованість – це підхід до управління мережею, що забезпечує динамічну, програмну конфігурацію мережі для покращення продуктивності та моніторингу, роблячи це більше схожим на хмарні обчислення, ніж на традиційне управління мережею. Натепер апаратна реалізація використання мереж SDN не знайшла широкого вжитку [3–5].

**Огляд наявних комерційних програмних продуктів.** Основними передумовами до вико-

ристання концепції програмно-конфігурованих мереж є зростання трафіку даних і кількості підключених до мережі пристроїв. За результатами досліджень було встановлено, що мережа SDN дозволяє підвищити ефективність мережевого обладнання на 25–30%, знизити витрати на експлуатацію мереж більш ніж на третину, підвищити гнучкість керування мережею за рахунок реалізації відповідних програм, підвищити безпеку, програмно створювати нові сервіси та оперативно завантажувати їх у мережне обладнання. Архітектурна схема мережі SDN містить рівень інфраструктури, керування та додатків (рис. 1).

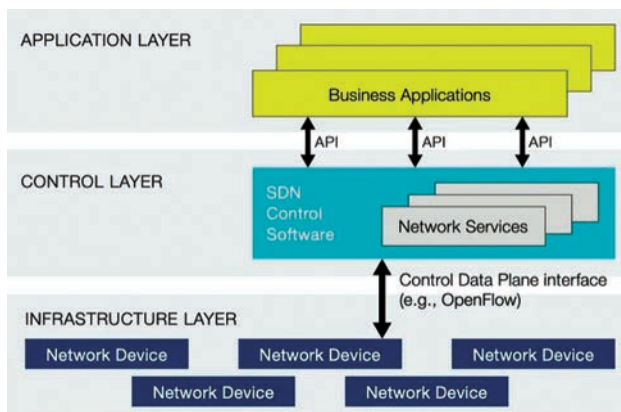


Рис. 1. Архітектурна схема мережі SDN

Керування мережними пристроями відбувається централізовано з консолі або програми на рівні керування.

OpenDaylight Project – це проєкт з відкритим кодом, який створений The Linux Foundation. Основними особливостями контролера є масштабованість та можливість налаштувати архітектуру на мікросервіси, що підвищує продуктивність роботи; контролер підтримує широкий спектр мережних протоколів: OpenFlow, P4 BGP, PCEP, LISP, NETCONF, OVSDB, SNMP; Model Driven Service Abstraction Layer (MD-SAL): рівень абстракції, які дозволяють створити схему бази даних, додатки REST API; кросплатформеність: OpenDaylight працює на багатьох сучасних платформах.

Weason – це швидкий кросплатформний модульний контролер OpenFlow на основі Java, який підтримує як подійну, так і потокову операцію. Ключовими особливостями контролера Weason є стабільність: Weason працює на експериментальному центрі обробки даних із 20 фізичними вимикачами, що має 20 перемикачів, і працює місяцями без простоїв; кросплатформеність: Weason розроблений на Java і працює на багатьох сучасних платформах; open source: Weason має комбіновану

ліцензію GPL v2 та FOSS Стенфордського університету v1.04; динамічність: Weason можна запускати, зупиняти, оновлювати, встановлювати під час виконання, не перериваючи інші незалежні операції; продуктивність: Weason є багатопоточним, що дозволяє розпаралелювати роботу; вебінтерфейс: Weason має додатково вбудований корпоративний вебсервер Jetty та власний UI інтерфейс.

Контролер Cisco Open SDN – це комерційний дистрибутив OpenDaylight, який забезпечує гнучкість бізнесу за допомогою автоматизації мережевої інфраструктури на основі стандартів. Cisco Open SDN абстрагується від складності управління різнорідними мережевими середовищами для поліпшення надання послуг та зменшення експлуатаційних витрат [6].

Такий контролер має відкритий код, а також постійно оновлюється завдяки інноваціям та активній спільноті OpenDaylight. Особливості контролера Cisco Open SDN: кластеризація (можливість високої масштабованості); придатність для обслуговування (Cisco Open SDN дає можливість моніторингу та збору різних метрик); Open Virtual Appliance (спрощена гнучкість встановлення та розгортання); велика кількість інтерфейсів (такий контролер має різні інтерфейси для роботи з мережею та інтеграції додатків).

**Архітектура та програмні модулі програмного забезпечення.** Програмне забезпечення реалізоване у вигляді вебдодатку для взаємодії з мережею. Серверна частина програмного забезпечення виконує такі функції: оброблення запитів, які надходять з клієнтської частини; реалізує роботу з програмно-конфігурованою мережею: зберігає, оброблює та видозмінює дані всередині мережі; забезпечує роботу з OpenFlow контролером, для коректної роботи мережі; проводить налаштування конфігурацій для масштабування мережі; відправка даних на клієнтську частину.

Клієнтська частина програмного забезпечення виконує такі функції, як: взаємодія з функціями системи за допомогою інтерфейса користувача; відображення даних, які надходять із сервера; візуалізація даних про програмно-конфігуровану мережу; відображення даних про трафік всередині мережі; забезпечення захищених запитів до серверної частини; ідентифікація, аутентифікація та авторизація адміністраторів. Основний сценарій використання системи з точки зору адміністратора мережі можна описати таким чином: адміністратор реєструється та авторизується у системі, перевіряє робочий стан мережі, за потреби оновлює дані всередині мережі, переглядає поточний

трафік, опрацьовує нові запити на додавання у мережу, додає новий мережевий пристрій.

Сучасні маршрутизатори виконують два основні завдання: передавання даних – просування пакета від вхідного порту на вихідний порт; керування даними – обробка пакета і ухвалення рішення про те, куди його спрямувати з огляду на поточний стан маршрутизатора. Програмне забезпечення, яке реалізує рівень керування, є досить примітивним порівняно зі складністю мережі, а також є закритим для розробників, дослідників і мережних операторів, що не дозволяє використовувати та налаштовувати мережу відповідно до потреб. Згідно з підходом програмно-конфігурованих мереж запропоновано розмежувати рівень керування та рівень передачі даних. Для ефективної взаємодії між рівнем керування та рівнем передачі даних використовується OpenFlow контролер. Для тестування програмного забезпечення створено локальну мережу за допомогою програми Mininet. Для поєднання програмно-конфігурованої мережі і контролера створено дві віртуальні мережі та налаштовано їх підключення.

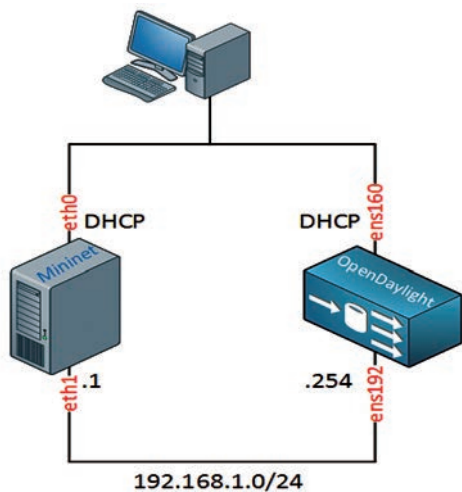


Рис. 2. Схематичне зображення системи з двома віртуальними машинами

Після поєднання мережі і контролера за допомогою протоколу DHCP ми можемо створити мережу відповідно до потреб або технічних вимог. Схематичний рисунок взаємодії мережі і контролера зображено на рис. 3.

Після створення мережі бажаної топології за допомогою контролера з нею можна взаємодіяти, використовуючи мережевий протокол управління NETCONF, який дозволяє встановлювати, маніпулювати та видаляти конфігурації мережевих пристроїв з використанням XML для кодування

даних. NETCONF може бути використаний як альтернатива CLI для налаштування мережевих пристроїв та для отримання статистики з мережевих пристроїв. Контролер OpenDaylight надає RESTCONF API інтерфейс, який дозволяє взаємодіяти через HTTP та отримувати дані з мови моделювання даних YANG, що розроблена для мережевого протоколу NETCONF. Таким чином, формуючи запис на серверній частині, ми можемо звертатися до мережі та проводити маніпуляції, оновлення або видалення конфігурацій мережевих пристроїв, а також моніторити трафік та масштабувати мережу.

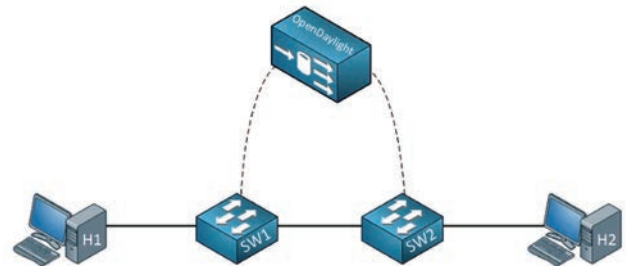


Рис. 3. Взаємодія з мережею через контролер

Змодельована мережа складається з п'яти комутаторів та п'ятнадцяти персональних комп'ютерів. Загальна кількість мережевих пристроїв – 20, що дозволяє створити відносно складну топологію та перевірити швидкість і ефективність такої мережі. Топологія створеної мережі зображена на рис. 4.

Програмне забезпечення містить модулі для роботи з мережею: модуль авторизації та реєстрації, модуль для підключення нових мережних пристроїв, модуль для взаємодії з мережею, перегляду даних про трафік, модуль безпеки.

Сутність мережевого пристрою складається з таких полів: id – унікальний ідентифікатор для кожного елемента мережі, який встановлюється безпосередньо мережею; name – ім'я комутатора, маршрутизатора або комп'ютера кінцевого користувача; ip – IP-адреса пристрою; first-seen – час додавання до мережі; mac – MAC-адреса пристрою; last-seen – час останнього виклику пристрою всередині мережі; attachments-points – список приєднаних мережних пристроїв до цього пристрою.

Поля first-seen та last-seen зберігаються у форматі DateTimeOffset, що являє собою момент часу, який зазвичай виражається у вигляді дати і часу доби щодо часу в форматі UTC та зберігається у наносекундах. У разі додавання нового пристрою до системи слід вказати його ім'я та IP-адресу

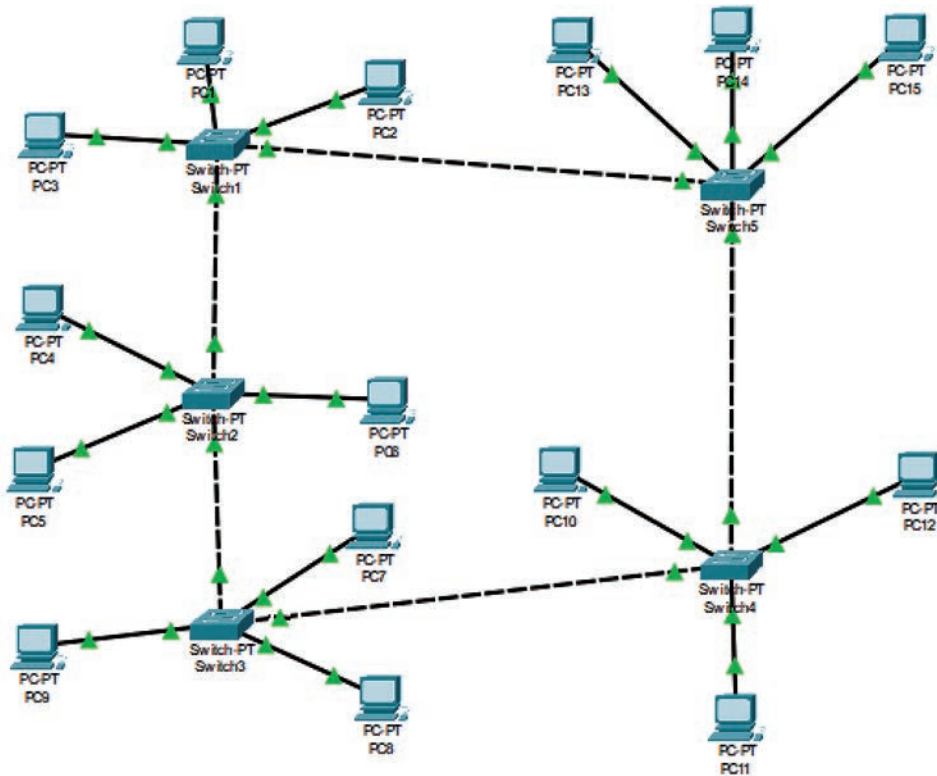


Рис. 4. Топологія мережних пристроїв для програмно-конфігурованої мережі

SDN Software

Home Your Networks About

Welcome, admin, Log Out

### List of available network

Id	Name	Created at	Updated on	Status
ebbe4649-3b14-4e83-8e12-c73747162c93	Network1	2021-06-10T01:10:10.6921783+03:00	2021-06-10T01:10:10.6931426+03:00	ON
6d0f7575-8d12-4ca4-9c1f-ee47cebde26b	test-version	2021-06-10T01:10:10.6933778+03:00	2021-06-10T01:10:10.6933821+03:00	OFF
3607e2ce-be36-419c-b9ff-ee2785a73349	SDN	2021-06-10T01:10:10.6933842+03:00	2021-06-10T01:10:10.6933851+03:00	OFF

Рис. 5. Сторінка вибору мережі

комутатора або роутера, до якого потрібно підключити такий пристрій усередині мережі. Решта полів створюються автоматично всередині мережі.

Сутність мережного пристрою пов'язана з іншою сутністю – attachments. Ця сутність має такі поля: `tr-id` – відображає ім'я приєднаного пристрою; `corresponding-tr` – є поєднанням імені цього пристрою та його MAC-адреси; `active` – булівське поле, що показує, чи є активним такий зв'язок у цей період часу.

Ці сутності зв'язані всередині мережі зв'язком One-to-Many та відображаються одночасно під час запиту на отримання сутності мережного пристрою. Модуль для взаємодії з мережею дозволяє налаштувати та вносити зміни до наявної мережі та адміністратору вносити зміни до окремих компонентів мережі: змінювати підключення між пристроями; встановлювати, налаштувати та змінювати ліцензії для мережі; змінювати конфігураційні файли для пристроїв та мережі;

здійснювати моніторинг мережі. Модуль для перегляду даних про трафік всередині мережі складається з даних про трафік, який проходить через мережу.

Отримана інформація буде відображати пере- хід між мережними пристроями. Наприклад: у разі пінгування з першого комп'ютера до деся- того буде виведений шлях, який проходить між цими двома комп'ютерами. Сутність traffic має такі поля: number – унікальний номер взаємодії; source – IP-адреса пристрою, що посилає дані; destination – IP-адреса пристрою, що отримує дані; time – час виконання. Серверна частина розділена на три ключові модулі: Network Access Layer,

Business Logic Layer та Web API. Модульність програмного забезпечення досягається за рахунок шаблону проектування – впровадження залежност- ей (Dependency injection). Взаємодія користувача з інтерфейсом додатка забезпечена поєднанням шаблонів, засобів Bootstrap та Angular-модулів. У пакеті client-арі розміщено файли та шаблони розроблених вебсторінок. У разі виконання POST- запиту на додавання нового мережного пристрою Адміністратор вводить дані, які проходять валіда- цію на відповідність своїм форматам на клієнт- ській частині, далі надсилаються на серверну час- тину та обробляються, після чого завантажуються у програмно-конфігуровану мережу.

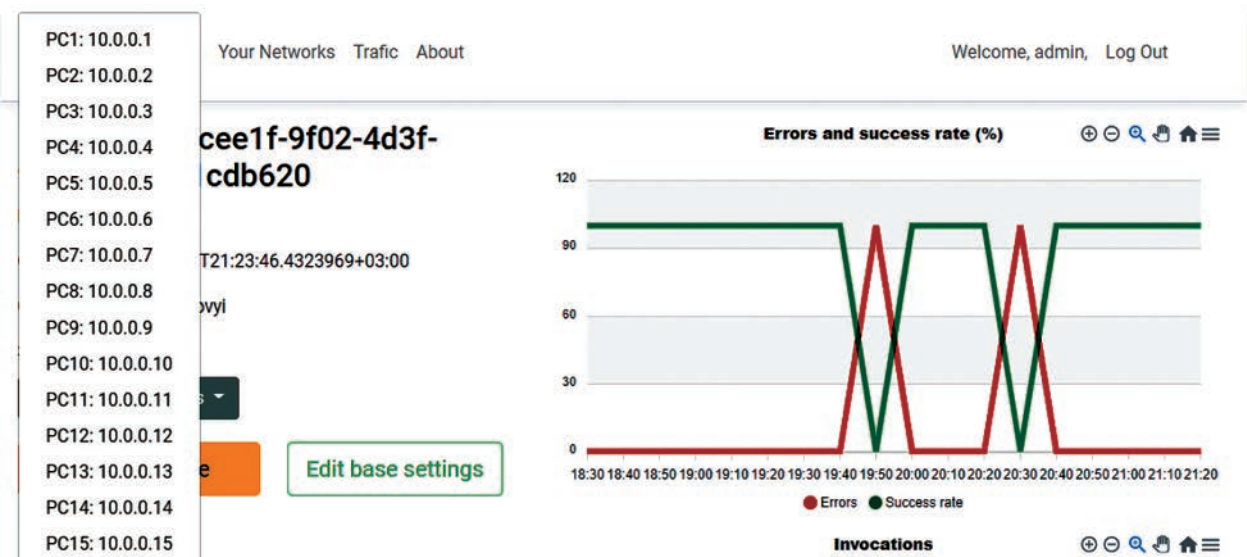


Рис. 6. Перегляд списку підключених пристроїв

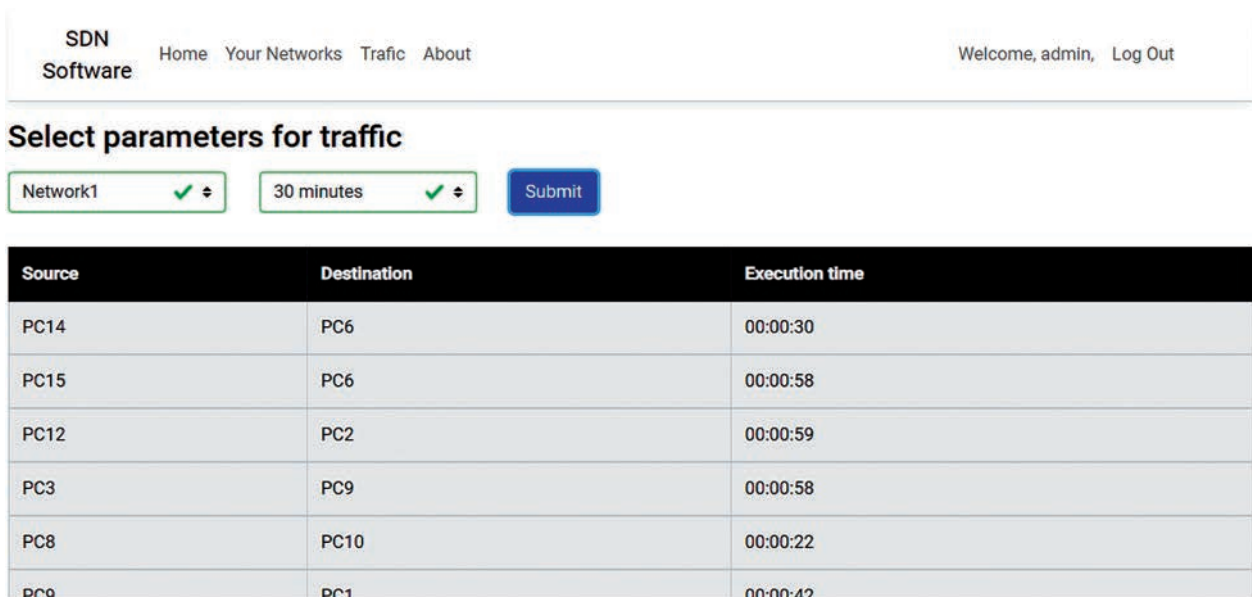


Рис. 7. Сторінка перегляду трафіку

**Висновки.** У цій роботі запропоноване програмне забезпечення для взаємодії між мережею SDN та серверною частиною з використанням OpenFlow контролера. Описано створену топологію для тестування мережі SDN та описано базові модулі програмного забезпечення. Запропонований програмний продукт забезпечує централізоване керування, збір та аналіз трафіку даних, масштабування та внесення змін у мережу. Визначено можливі напрями розширення та покращення запропонованого програмного забезпечення: роз-

роблення механізму автоматичних сповіщень, наприклад, про оновлення мережі; автоматичне оновлення ліцензій всередині мережі; створення можливості для швидкого розгортання вже наявних мереж; створення функціональності для перетворення традиційних мереж на програмно-конфігуровані; зміна способу авторизації адміністраторів з Base Auth на JWT токен; додавання модуля прогнозування трафіку мережі на основі методів машинного навчання; покращення UX/UI клієнтської частини проєкту.

### Список літератури:

1. McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S. and Turner J. "OpenFlow: enabling innovation in campus networks", *SIGCOMM Comput. Commun. Rev.*, 2008.
2. Intel IXP Network Processors. URL: [http://www.intel.com/p/en\\_US/embedded/hwsw/hardware/ixp-4xx](http://www.intel.com/p/en_US/embedded/hwsw/hardware/ixp-4xx).
3. Netronome Flow Processors. URL: <http://www.netronome.com/pages/flow-processors/>.
4. Open vSwitch. URL: <http://openvswitch.org/>.
5. Software-Defined Anything (SDx) Market. URL: <https://www.marketsandmarkets.com/PressReleases/software-defined-anything.asp>
6. Cisco Open SDN Controller. URL: <https://www.cisco.com/c/en/us/products/cloud-systems-management/open-sdn-controller/index.html>

### **Oleshchenko L.M., Movchan K.O., Zamkovyi I.T. SOFTWARE METHOD FOR TESTING AND ADMINISTRATION OF SDN NETWORKS**

*Every day we can see an increase in network traffic and IT companies increasingly need to use large-scale networks that require a lot of time and resources to set up. Automating this process helps to solve these problems. Existing network architectures focus on outdated protocols that interfere with scaling, rapid addition of new devices, and innovation. That is why in recent years the demand for a new network architecture of the software-configured SDN network has increased. The main advantages of SDN networks are programmability and flexible centralized management and scalability.*

*The proposed software uses an OpenFlow controller for interaction between the network and the server part of the program. The most popular controllers are analyzed: OpenDaylight, Beacon and Cisco Open SDN Controller. Problems and shortcomings of traditional networks are analyzed, the general concept of software-configured networks is described and the basic requirements to the software are formulated. C # language, ASP.NET Core framework is chosen for the server part, Angular framework with TypeScript language is chosen for the client part. The OpenDaylight Project controller is used as an additional technology for interaction, configuration and work with a software-configured network. The client-server architecture is used for the software, key modules and essences of the system are described. Implemented key functionality, created a software-configured network and implemented message processing from the network. The general principle of work of the software with a network is described and the example of work of the administrator of the network modelled in the Mininet environment is resulted.*

**Key words:** software, SDN, OpenFlow, OpenDaylight Project, network device, IP addressing, Mininet.