МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ
Циклова комісія (кафедра) комп’ютерної інженерії та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему

АНАЛІЗ АРХІТЕКТУРИ ТА ФУНКЦІОНУВАННЯ ПРОГРАМНО-ВИЗНАЧЕНИХ МЕРЕЖ (SDN)

Виконав: студент групи 1К-21
Спеціальності 123 Комп’ютерна інженерія
Андрій ЖАРКО
Керівник:
Майя ЛЮТА

Черкаси 2025

АНОТАЦІЯ

Кваліфікаційна робота на тему «Аналіз архітектури та функціонування програмно-визначених мереж (SDN)» складається з вступу, основної частини, що містить 3 розділи, висновку та списку використаних джерел. Загальний обсяг роботи – 45 сторінок. У роботі 2 рисунки та 3 таблиці. Перелік використаних ресурсів налічує 21 одиницю.

Програмно-визначені мережі (SDN, Software-Defined Networking) — це сучасний підхід до управління мережевою інфраструктурою, що ґрунтується на відокремленні контрольної площини від площини передачі даних. Такий підхід забезпечує гнучкість, централізоване керування і спрощує впровадження нових мережевих рішень. Архітектура SDN складається з трьох основних рівнів: інфраструктурний рівень (Data Plane), що включає фізичні та віртуальні мережеві пристрої, такі як комутатори, маршрутизатори, точки доступу; контрольний рівень (Control Plane), що представлений програмним контролером — центральним елементом управління мережею; прикладний рівень (Application Plane), що містить мережеві програми, які взаємодіють з контролером через API (інтерфейс прикладного програмування). Ці програми можуть реалізовувати балансування навантаження, захист від атак, маршрутизацію тощо.

SDN є ключовим кроком до побудови адаптивних, керованих і ефективних мереж нового покоління. Завдяки модульності архітектури та програмній гнучкості, програмно-визначені мережі відкривають широкі можливості для автоматизації, безпеки та оптимізації сучасних IT-систем.

Ключові слова: SDN (Software-Defined Networking), програмно-визначені мережі, архітектура SDN, контрольна площина (Control Plane), площина передачі даних (Data Plane), мережевий контролер, OpenFlow, централізоване управління мережею, API (інтерфейс прикладного програмування), програмована мережа, автоматизація мережі, гнучкість мережевої інфраструктури, мережеві протоколи, безпека мережі, маршрутизація трафіку, політика доступу.

ABSTRACT

The qualification work on the topic «Analysis of the architecture and functioning of software-defined networks (SDN)» consists of an introduction, the main part, which contains 3 sections, a conclusion and a list of sources used. The total volume of the work is 45 pages. The work contains 2 figures and 3 tables. The list of resources used has 21 units.

Software-Defined Networking (SDN) is a modern approach to network infrastructure management, which is based on the separation of the control plane from the data plane. This approach provides flexibility, centralized management and simplifies the implementation of new network solutions. The SDN architecture consists of three main levels: the infrastructure layer (Data Plane), which includes physical and virtual network devices, such as switches, routers, access points; the control layer (Control Plane), which is represented by a software controller - the central element of network management; Application Plane, which contains network applications that interact with the controller via API (Application Programming Interface). These applications can implement load balancing, attack protection, routing, etc.

SDN is a key step towards building adaptive, manageable and efficient networks of the next generation. Due to the modularity of the architecture and software flexibility, software-defined networks open up wide opportunities for automation, security and optimization of modern IT systems.

Keywords: SDN (Software-Defined Networking), software-defined networks, SDN architecture, control plane (Control Plane), data plane (Data Plane), network controller, OpenFlow, centralized network management, API (Application Programming Interface), programmable network, network automation, network infrastructure flexibility, network protocols, network security, traffic routing, access policy.

ЗМІСТ

ВСТУП

Програмно-визначені системи (англ. Software-Defined Systems) – це підхід до управління обчислювальними, мережевими чи іншими цифровими ресурсами за допомогою програмного забезпечення, що відокремлює функції управління від фізичного обладнання. Такий підхід значно змінює традиційне уявлення про IT-інфраструктуру та відкриває нові можливості в гнучкості, масштабованості та ефективності.

Значення програмно-визначених мереж полягає у управлінні всією системою здійснюється з єдиного центру, що забезпечує повний контроль і спрощує адміністрування. Адже завдяки абстрагуванню від фізичної інфраструктури, такі системи легко адаптуються до змін – додавання або модифікація компонентів не вимагає складних налаштувань апаратного забезпечення. Оскільки зміни реалізуються на рівні програмного забезпечення, це дозволяє швидше впроваджувати нові сервіси, функціонал або політики без значних витрат.

У сучасних умовах програмно-визначені системи стають основою цифрової трансформації в бізнесі, освіті, охороні здоров’я, телекомунікаціях та багатьох інших сферах, де критично важливо забезпечити швидкість, гнучкість і ефективність управління IT-інфраструктурою.

Виклики та перспективи для використання програмно-визначених мереж

Для використання програмно-визначених мереж постають такі виклики: безпека та захист даних (централізація управління може стати ціллю для атак), високі вимоги до кваліфікації персоналу, інтеграція з існуючими інфраструктурами (складність переходу від традиційних систем), можливі збої в разі помилок у програмному забезпеченні.

Перспективами впровадження та використання програмно-визначених мереж є розширення автоматизації в управлінні ресурсами, широке застосування у хмарних технологіях, IoT, 5G, зниження витрат і підвищення

гнучкості IT-середовища, створення масштабованих, ефективних та інноваційних цифрових платформ.

Актуальність впровадження та використання програмно-визначених мереж полягає в необхідності підвищення гнучкості, масштабованості та ефективності управління сучасними IT-інфраструктурами. У зв'язку зі швидким розвитком хмарних технологій, Інтернету речей та мереж 5G, SDN забезпечує централізований контроль, автоматизацію процесів і оперативне реагування на зміни в мережевому середовищі, що робить її надзвичайно актуальною для підприємств і провайдерів послуг.

Метою даного дослідження є обґрунтування практичних рекомендацій щодо доцільності застосування SDN в різних сферах діяльності на основі огляду успішних випадків використання SDN та дослідження важливих проектів у галузі SDN.

Завданням даної роботи є:

- аналіз теоретичних концепцій та принципів роботи SDN;
- виокремлення архітектури SDN та її складових;
- опис загальної характеристики програмно-визначених мереж;
- порівняння SDN з традиційними мережами;
- виокремлення переваг та недоліків використання SDN.

РОЗДІЛ 1
ЗАГАЛЬНІ ХАРАКТЕРИСТИКИ ПРОГРАМНО-ВИЗНАЧЕНИХ МЕРЕЖ (SDN)

1.1 Основні концепції та принципи роботи SDN

SDN (Software Defined Networking) – це інноваційний підхід до побудови та управління мережами, який радикально змінює традиційну модель мережевої інфраструктури. Основна мета SDN – забезпечити централізоване, програмно-орієнтоване керування мережею, що дозволяє ефективно адаптувати її до змін у вимогах бізнесу або трафіку.

У класичних мережах управління трафіком здійснюється окремими мережевими пристроями (комутаторами, маршрутизаторами), в яких тісно пов’язані площина управління (control plane) та площина передачі даних (data plane). Це створює низку обмежень у масштабованості, гнучкості та швидкості впровадження змін.

SDN розділяє ці площини, дозволяючи керувати всіма пристроями через централізований контролер за допомогою програмного забезпечення.

Для аналізу даного питання доцільно виділити ключові компоненти SDN:

1. Контролер SDN - це "мозок" мережі [15].

Контролер SDN (Software Defined Networking Controller) – це центральний елемент архітектури SDN, який виконує функції управління всією мережею. Його роль можна порівняти з мозком, який координує роботу всіх частин організму – у цьому випадку, мережевих пристроїв.

Контролер SDN – це програмна платформа, яка здійснює централізоване управління мережею, збирає інформацію про її стан, приймає рішення щодо маршрутизації та політик, а також надсилає відповідні інструкції мережевим пристроям (комутаторам, маршрутизаторам тощо). Він оперує площиною управління (control plane), тоді як комутатори працюють у площині передачі (data plane).

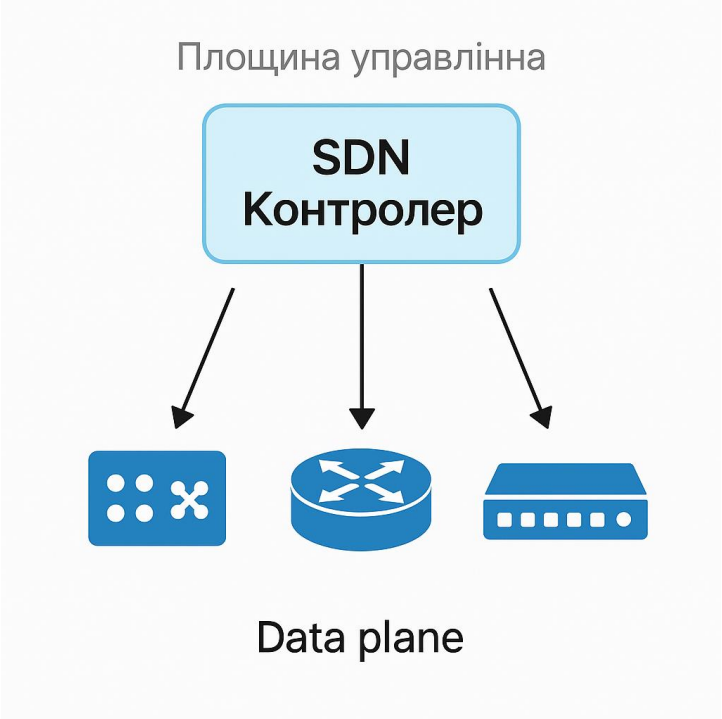

Рисунок 1.1 – Контролер SDN

Основні функції контролера SDN:

- Централізоване управління. Контролер об'єднує усі мережеві пристрої в єдину логічну систему, що дозволяє адміністраторам керувати мережею як цілісною структурою.

- Моніторинг мережі. Контролер постійно отримує інформацію про трафік, стан пристроїв, навантаження каналів тощо, дозволяючи швидко виявляти проблеми або аномалії.

- Маршрутизація та комутація. На основі отриманих даних контролер розраховує найефективніші шляхи передавання даних та надсилає командам комутаторам, які реалізують ці маршрути.

- Застосування політик безпеки. Контролер дозволяє централізовано впроваджувати та змінювати правила безпеки, обмеження доступу, сегментацію мережі тощо.

- Автоматизація. Багато процесів у SDN автоматизовані – наприклад, перенаправлення трафіку у разі перевантаження або аварії.

Контролер використовує два основні типи інтерфейсів:

Southbound API – забезпечує зв’язок із мережевими пристроями (найпоширеніший протокол – OpenFlow) [16].

Northbound API – дозволяє взаємодіяти з прикладними програмами, сервісами керування, оркестраторами. Наприклад, через REST API або SDK інтегруються інструменти безпеки, моніторингу чи аналітики.

Переваги використання контролера SDN:

- Гнучкість – завдяки можливості оперативної зміни конфігурацій без необхідності фізичного втручання.

- Простота адміністрування – вся мережа керується з одного місця.

- Швидке реагування – виявлення проблем і впровадження змін займає мінімум часу.

- Оптимізація ресурсів – трафік, маршрути, політики розподіляються ефективно і в реальному часі.

- Підтримка інновацій – легко впроваджуються нові сервіси, автоматизація, аналітика тощо.

Контролер приймає рішення щодо маршрутизації, політик доступу, пріоритетів трафіку тощо, і надсилає інструкції мережевим пристроям. Він забезпечує централізоване управління всією мережею.

- площина передачі даних (Data Plane). Вона представлена комутаторами або маршрутизаторами, які безпосередньо обробляють трафік згідно з інструкціями від контролера. У SDN ці пристрої працюють як прості "форвардери" (компоненти мережі або програмне забезпечення, що пересилає (форвардить) дані з однієї точки в іншу).

2. Інтерфейси (API). Southbound API – використовується для зв’язку між контролером та мережевими пристроями (наприклад, протокол OpenFlow).

API (Application Programming Interface) – це набір правил, інструкцій та протоколів, який дозволяє різним програмам взаємодіяти одна з одною. Простіше кажучи, API – це посередник, який дозволяє одному програмному компоненту “спілкуватися” з іншим без необхідності розуміти, як той влаштований усередині [17].

Основні функції API:

- Взаємодія між системами. API дозволяє двом незалежним системам або сервісам обмінюватися даними або функціональністю.

- Інкапсуляція логіки. Використання API приховує внутрішню реалізацію процесів. Ви працюєте лише з інтерфейсом, не турбуючись про технічні деталі.

- Стандартизація взаємодії. Завдяки API, різні частини програмного забезпечення можуть взаємодіяти за єдиними правилами, що полегшує інтеграцію.

Типи API:

1. Відкриті (Public/Open API)

Доступні для будь-яких розробників. Використовуються у вебсервісах, наприклад:

- API Google Maps;

- API Twitter;

- API OpenWeather.

2. Закриті (Private/Internal API)

Використовуються лише всередині компанії. Дозволяють організаціям централізовано управляти своїми сервісами.

3. Партнерські (Partner API)

Доступні лише для обраних партнерів. Використовуються для контролю доступу до комерційно важливої інформації чи сервісів.

4. Композитні API (Composite API)

Поєднують декілька запитів в один, щоб оптимізувати взаємодію між сервісами та зменшити кількість викликів.

Види взаємодії (інтерфейсів API) [18]:

- REST (Representational State Transfer) – найпопулярніший тип API, заснований на HTTP-протоколі. Має легку структуру та широко використовується у веброзробці.

- SOAP (Simple Object Access Protocol). Більш формальний протокол, який передає XML-повідомлення. Застосовується там, де важлива безпека та точність.

- GraphQL – API, що дозволяє клієнту самостійно визначати, які саме дані потрібні. Розроблений компанією Facebook.

- gRPC – бінарний протокол від Google, який використовує Protocol Buffers замість JSON або XML. Підходить для високопродуктивних сервісів.

3. Northbound API – дозволяє взаємодіяти між контролером та зовнішніми програмами або системами керування (наприклад, для аналітики або автоматизації).

Northbound API (інтерфейс) – це прикладний програмний інтерфейс, який дозволяє контролеру в архітектурі SDN (Software-Defined Networking) взаємодіяти з прикладними програмами та мережевими сервісами, що знаходяться на вищому рівні. Його основне завдання – забезпечити доступ до функцій контролера з боку зовнішніх додатків, таких як системи управління мережею, аналітичні платформи, засоби моніторингу або навіть спеціальні бізнес-додатки.

У Software-Defined Networking (SDN) управління мережею здійснюється через централізований контролер, який взаємодіє з мережевими пристроями за допомогою API (Application Programming Interface). Це дозволяє програмно змінювати політики маршрутизації, правила доступу, пріоритети трафіку.

Для програмування використовуються такі мови, зокрема:

а) Python – найпопулярніша мова для роботи з SDN API, особливо для написання скриптів, автоматизації, інтеграції з контролерами (як-от ONOS, Ryu, OpenDaylight);

б) Java – часто використовується у великих SDN-контролерах (наприклад, OpenDaylight);

в) Go – застосовується у деяких сучасних SDN-рішеннях, завдяки продуктивності;

г) JavaScript (Node.js) – використовується для веб-інтерфейсів керування SDN через REST API.

1.2 Архітектура SDN та її складові

Основна ідея SDN полягає у централізованому контролі мережі за допомогою програмного забезпечення, що дозволяє зробити мережу більш гнучкою, масштабованою та керованою.

Основні принципи SDN-архітектури [19]:

- Відокремлення контрольного рівня (control plane) від рівня передачі даних (data plane).

Доцільно навести приклади реалізації цього принципу:

1. Комутатор Open vSwitch (OVS) + контролер Ryu або ONOS. OVS виконує роль пристрою передачі даних. Контролер керує правилами пересилання трафіку, оновлюючи таблиці потоків через OpenFlow.

2. Маршрутизатори в дата-центрах під керуванням SDN-контролера. Замість статичної маршрутизації, контролер (наприклад, OpenDaylight) централізовано визначає маршрути залежно від навантаження або політик доступу.

3. Автоматичне сегментування мережі (наприклад, у кампусних мережах). Контролер аналізує тип пристрою або користувача при підключенні й автоматично виділяє VLAN, QoS або політики доступу. Фізичні комутатори лише пересилають пакети згідно з вказівками.

4. Інтеграція з firewall/IDS. Контролер отримує сигнал від системи виявлення загроз і оновлює маршрути або блокує підозрілий трафік на комутаторах. Тобто рішення приймається на контрольному рівні, а дія виконується на рівні передачі.

- Централізоване управління мережею через SDN-контролер.

- Програмованість мережі – мережеві пристрої можуть конфігуруватися за допомогою програмного забезпечення через стандартизовані інтерфейси.

- Використання відкритих API (інтерфейсів програмування додатків) для зв’язку між різними рівнями мережі.

Архітектура SDN умовно поділяється на три рівні:

- Інфраструктурний рівень (Data Plane) – фізичні мережеві пристрої (комутатори, маршрутизатори), які виконують передачу даних.

- Контрольний рівень (Control Plane) – SDN-контролер, який визначає логіку маршрутизації.

- Прикладний рівень (Application Plane) – програми, які керують мережею на високому рівні.

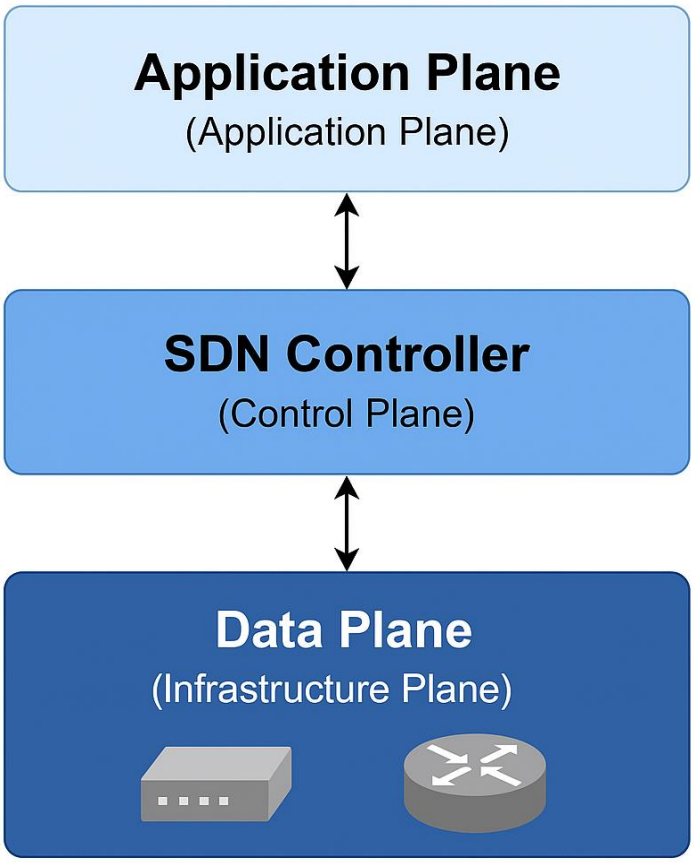

Рисунок 1.2 – Умовний поділ на рівні архітектури SDN

Northbound API з’єднує прикладний рівень з контролером.

Принцип роботи Northbound API:

- Програма (наприклад, система моніторингу) надсилає запит через Northbound API до SDN-контролера.

- Контролер інтерпретує цей запит та надсилає відповідні команди до мережевих пристроїв через Southbound API.

- Відповідь або результат виконання команди повертається до програми через той самий Northbound API.

Основні функції Northbound API:

- Моніторинг мережі: отримання статистики про трафік, топологію, завантаження каналів тощо.

- Управління політиками: визначення пріоритетів трафіку, фільтрація, маршрутизація.

- Забезпечення безпеки: виявлення та блокування аномалій або атак на рівні додатків.

- Оркестрація ресурсів: автоматизація створення, зміни чи видалення мережевих маршрутів або віртуальних сегментів.

- Інтеграція з AI/ML: надання доступу до даних мережі для аналізу або прогнозування навантаження.

Варто зазначити головні принципи роботи SDN [20]:

- Розділення управління та передачі.

Мережеві пристрої не приймають рішення самостійно – вони лише виконують команди, отримані від контролера. Це дозволяє гнучко оновлювати маршрути, політики безпеки, балансування навантаження тощо, не змінюючи фізичну конфігурацію мережі.

- Централізоване управління.

Контролер має повну картину всієї мережі. Це дозволяє швидко реагувати на зміни, виявляти аномалії, оптимізувати маршрутизацію й ефективно розподіляти ресурси.

- Програмована мережа.

SDN надає інтерфейси для створення програм, які можуть змінювати поведінку мережі в реальному часі, адаптуючи її до потреб користувачів або змін навантаження.

У таблиці 1.1 переваги архітектури SDN подано у порівнянні з традиційними (класичними) мережами, де: кожен мережевий пристрій (комутатор, маршрутизатор) має власний control plane; конфігурації виконуються локально і вручну; зміни займають більше часу і вимагають доступу до кожного пристрою окремо; обмежена автоматизація та гнучкість.

Таблиця 1.1 – Переваги архітектури SDN [21]

Перевага	Пояснення
Гнучкість	Швидке впровадження змін без фізичного втручання
Централізація	Повний контроль і видимість мережі в одному місці
Автоматизація	Можливість програмного управління і самонастройки
Масштабованість	Легке додавання нових пристроїв чи сервісів
Безпека	Швидке виявлення загроз і централізована політика контролю доступу

Архітектура SDN має недоліки та виклики:

- Надмірна залежність від контролера – у разі збою можлива втрата керування мережею.

- Проблеми масштабування при великих обсягах трафіку без оптимізації.

- Потреба в переобладнанні мережевої інфраструктури (оновлення ПЗ, підтримка OpenFlow тощо).

- Безпека API – можливість несанкціонованого доступу при поганій реалізації.

Архітектура SDN – це сучасний підхід до управління комп’ютерними мережами, який розділяє логіку управління та фізичну інфраструктуру. Завдяки цьому досягається висока гнучкість, ефективність та адаптивність, що особливо важливо в умовах швидких змін та зростаючих вимог до мережевих сервісів.

Цей підхід ідеально підходить для дата-центрів, корпоративних мереж, провайдерів хмарних послуг та телекомунікаційних компаній.

РОЗДІЛ 2
АНАЛІЗ ПРОГРАМНО-ВИЗНАЧЕНИХ МЕРЕЖ

2.1 Загальні характеристики програмно-визначених мереж

Програмно-визначені мережі (англ. Software-Defined Networking, SDN) – це інноваційний підхід до побудови та управління мережами, який дозволяє централізовано та гнучко контролювати мережеву інфраструктуру за допомогою програмного забезпечення. Основною ідеєю SDN є відділення логіки управління мережею від її апаратної реалізації [1].

Основні характеристики SDN:

1. Централізоване управління мережею

У традиційних мережах кожен пристрій самостійно приймає рішення щодо маршрутизації трафіку. У SDN логіка прийняття рішень переноситься в централізований контролер, який має повну картину стану мережі. Це дозволяє:

- ефективніше використовувати ресурси;
- оперативно реагувати на зміни;
- реалізовувати єдину політику безпеки.

2. Відокремлення рівня управління (control plane) від рівня пересилання (data plane)

У SDN архітектурі:

- Control Plane – приймає рішення про обробку трафіку;
- Data Plane – виконує ці рішення (пересилає пакети).

Це розділення дозволяє централізовано управляти трафіком без зміни логіки на кожному мережевому пристрої.

3. Програмованість мережі.

Однією з головних переваг SDN є можливість динамічно програмувати мережеву інфраструктуру, використовуючи відкриті API. Це означає, що адміністратори можуть:

- автоматизувати конфігурацію. Адміністратору потрібно: встановити або підключитися до SDN-контролера (наприклад, OpenDaylight, Ryu); використати REST API або Python SDK для створення сценаріїв (скриптів), які автоматично конфігурують комутатори чи маршрутизатори; створити шаблони політик та правил у вигляді JSON/XML і запускати їх через API при додаванні нових пристроїв. Результатом роботи буде менше ручної роботи, швидше розгортання;

- створювати нові сервіси. Адміністратору потрібно: розробити програму (на Python, Java, Go), що взаємодіє з контролером через північний API (Northbound API); визначити логіку сервісу (наприклад, створення віртуальних сегментів мережі, моніторинг трафіку, балансування навантаження); реєструвати застосунок у контролері або інтегрувати його з мережею через API. Результатом роботи стане швидке створення нових функцій без зміни фізичної інфраструктури;

- змінювати правила маршрутизації “на льоту” Адміністратору потрібно: отримати поточний стан трафіку або топології через API контролера; визначити нові правила або пріоритети для маршрутизації (наприклад, обхід перевантаженого вузла); через Southbound API (наприклад, OpenFlow) надіслати нові правила на відповідні комутатори. Результатом буде адаптивна мережа, що змінює поведінку в реальному часі.

4. Використання стандартних інтерфейсів API.

- SDN використовує два основні типи інтерфейсів:

- Southbound API – для зв’язку між контролером і мережевими пристроями (приклад: OpenFlow);

- Northbound API – для зв’язку між контролером і мережевими додатками або адміністративними інтерфейсами.

Це сприяє кращій взаємодії між різними компонентами мережі та забезпечує гнучкість.

5. Покращена безпека: централізоване управління в SDN дозволяє більш ефективно реалізовувати політики безпеки, контролювати доступ, відслідковувати аномалії та оперативно реагувати на атаки.

6. Повна видимість і контроль: контролер має повну інформацію про стан мережі, її топологію, завантаженість, кількість активних пристроїв тощо. Це дозволяє:

- точно аналізувати продуктивність;

- оптимізувати маршрути;

- швидко виявляти несправності.

7. Підтримка віртуалізації мереж: SDN дозволяє створювати віртуальні мережі (VLAN, VPN) поверх фізичної інфраструктури, що підвищує ефективність використання ресурсів та гнучкість у наданні сервісів. Варто відмітити основні відмінності підтримки віртуалізації мереж у SDN від традиційного адміністрування маршрутизаторів (див. табл. 2.1).

Таблиця 2.1 – відмінності SDN та традиційного адміністрування маршрутизаторів

Критерій	SDN (програмно-керовані мережі)	Традиційні маршрутизатори
Управління	Централізоване через контролер	Ручне, на кожному пристрої окремо
Конфігурація	Програмна, через API	CLI-конфігурації, часто вручну
Гнучкість змін	Динамічне оновлення “на льоту”	Затримки через необхідність ручних змін
Створення VLAN/VPN	Швидке, автоматизоване	Поступове, вимагає доступу до всіх пристроїв
Масштабованість	Висока, легко додавати нові сегменти	Обмежена, потребує значних зусиль
Можливості сервісів	Підтримка мультиорендності, QoS, політик доступу	Залежить від функціоналу кожного окремого маршрутизатора
Ізоляція трафіку	Гнучка, програмна	Через складну конфігурацію ACL/VLAN
Швидкість впровадження змін	Висока	Низька

8. Гнучкість та масштабованість [2]: завдяки програмованості та централізації SDN-мережі можуть легко масштабуватися під нові потреби

бізнесу або користувачів. Нові пристрої чи сервіси можна інтегрувати без зміни фізичної інфраструктури.

Програмно-визначені мережі – це революційний підхід у сфері телекомунікацій, який дозволяє перейти від статичних і важкокерованих традиційних мереж до динамічних, гнучких і розумних мережевих рішень. Завдяки централізації, програмованості та використанню відкритих стандартів, SDN відкриває нові горизонти для ефективного управління мережевими ресурсами, зменшуючи витрати та підвищуючи рівень безпеки.

Складові SDN можна поділити на три основні рівні, кожен з яких виконує окрему функцію, забезпечуючи розділення логіки управління і пересилки даних, зокрема:

- інфраструктурний рівень (Data Plane), що складається з фізичних або віртуальних пристроїв (комутатори, маршрутизатори, точки доступу), що відповідають за передачу трафіку. Усі рішення щодо маршрутизації тут не приймаються, а виконуються згідно з інструкціями від контролера.

- контрольний рівень (Control Plane). Ядро SDN — це SDN-контролер, який володіє повною інформацією про топологію мережі, стан з'єднань та політики безпеки. Контролер приймає рішення про те, як повинен переміщуватись трафік, і надсилає ці інструкції пристроям передачі.

- прикладний рівень (Application Plane). Тут працюють програми, які керують мережею на високому рівні — балансування навантаження, безпека, моніторинг, автоматичне масштабування. Вони взаємодіють із контролером через Northbound API, формулюючи політики або реагуючи на зміни в мережі. Це може бути як власна розробка, так і стороннє програмне забезпечення (наприклад, для виявлення загроз або оптимізації трафіку).

2.2 Порівняння SDN з традиційними мережами

З розвитком інформаційних технологій потреба в гнучких, масштабованих і ефективно керованих мережах значно зросла. Традиційна мережева

інфраструктура, яка була ефективною протягом тривалого часу, вже не здатна повністю відповідати сучасним вимогам. У відповідь на ці виклики з'явився новий підхід – програмно-визначені мережі (SDN – Software-Defined Networking) [3].

Щоб краще зрозуміти переваги цього підходу, розглянемо ключові відмінності між SDN і традиційними мережами.

1. Централізоване vs децентралізоване управління.

У традиційних мережах кожен маршрутизатор або комутатор приймає рішення на основі локальної інформації, що ускладнює масштабування. У SDN всі ці функції централізуються в контролері, який керує усією мережею як єдиним цілим.

2. Гнучкість та адаптивність.

Зміни в традиційній мережі вимагають фізичного доступу до обладнання та повторного налаштування. У SDN достатньо змінити програмну логіку контролера – зміни негайно застосовуються до всієї мережі.

3. Програмованість та автоматизація.

SDN дозволяє керувати мережею через API, створюючи можливість повної автоматизації мережевих процесів. Це значно зменшує людський фактор і ймовірність помилок.

4. Безпека та моніторинг.

У традиційних мережах політики безпеки встановлюються окремо для кожного пристрою, що створює ризики розбіжностей і помилок. SDN дозволяє централізовано управляти політиками, миттєво застосовуючи зміни до всієї мережі.

Переваги SDN порівняно з традиційними мережами [5]:

- покращене централізоване управління;
- швидке впровадження нових функцій;
- єдина система безпеки;
- оптимізація мережевих ресурсів;
- автоматизація рутинних завдань;

- зменшення витрат на обслуговування.

Таблиця 2.1 – Основні порівняльні аспекти SDN з традиційними мережами [4]

Критерій	Традиційні мережі	SDN (Програмно-визначені мережі)
Архітектура	Децентралізована – кожен пристрій приймає рішення самостійно	Централізована – логіка керування зосереджена в контролері
Управління мережею	Налаштування вручну на кожному пристрої	Центральне управління через програмний контролер
Гнучкість та масштабованість	Обмежена, потребує фізичних змін	Висока, зміни в мережі здійснюються програмно
Розділення Control Plane і Data Plane	Control і Data Plane інтегровані у кожному пристрої	Чітке розділення: контролер – для управління, комутатори – для пересилання
Автоматизація	Мінімальна, здебільшого ручне керування	Підтримка автоматизації за допомогою API та скриптів
Програмованість	Обмежена або відсутня	Повноцінна – використовуються API (Northbound/Southbound)
Політика безпеки	Локальна на кожному пристрої	Централізована, з єдиною політикою безпеки
Моніторинг та аналітика	Обмежена інформація, фрагментована	Повна видимість мережі завдяки контролеру
Інновації та нові сервіси	Впроваджуються повільно, потребують ручних змін	Можливість швидкого створення та розгортання нових сервісів
Вартість обслуговування	Вища – через складність налаштувань і технічну підтримку	Потенційно нижча – завдяки автоматизації та централізації

Програмно-визначені мережі (SDN) демонструють значні переваги над традиційними мережами в умовах сучасного IT-середовища. Вони забезпечують гнучкість, масштабованість, простоту управління та можливість інтеграції з новітніми технологіями, такими як хмарні обчислення, Інтернет речей (IoT) і штучний інтелект.

Для організацій, які прагнуть підвищити ефективність, зменшити витрати та адаптуватися до швидких змін, SDN – це стратегічно вигідний вибір.

2.3 Переваги та недоліки використання SDN

Програмно-визначені мережі (SDN – Software-Defined Networking) стали інноваційною відповіддю на складнощі традиційного мережевого управління. Завдяки логічному розділенню функцій управління та передачі даних, SDN забезпечує гнучке, централізоване та автоматизоване керування мережею. Проте, як і будь-яка технологія, SDN має не лише переваги, а й певні обмеження та виклики [6]. Розгляньмо їх детальніше.

Переваги SDN:

1. Централізоване управління.

Контролер SDN забезпечує єдиний центр прийняття рішень для всієї мережі. Це дає змогу швидко налаштовувати, змінювати або оптимізувати мережу без потреби змінювати конфігурації на кожному окремому пристрої.

Приклад: адміністратор може змінити політику безпеки для всіх вузлів через один інтерфейс

2. Гнучкість і масштабованість.

Завдяки програмній природі SDN, можливо швидко адаптувати мережу до нових потреб, додати або змінити сервіси, розширити пропускну здатність чи інтегрувати нові пристрої без значного втручання в інфраструктуру.

3. Програмованість і автоматизація.

SDN дозволяє використовувати API для інтеграції з іншими системами, створення скриптів, політик і автоматизованих сценаріїв, що значно спрощує обслуговування мережі та зменшує кількість людських помилок.

4. Зниження експлуатаційних витрат.

Автоматизація та централізація сприяють зменшенню потреби в технічному персоналі для обслуговування мережі та зменшують час простоїв.

5. Покращена безпека.

Централізоване управління дозволяє швидко виявляти загрози, відслідковувати аномалії та впроваджувати уніфіковані політики безпеки на всій інфраструктурі.

6. Швидке впровадження інновацій.

SDN полегшує тестування та впровадження нових мережевих сервісів без ризику порушення роботи всієї мережі.

7. Покращене використання ресурсів.

Завдяки динамічному маршрутизаційному контролю SDN забезпечує оптимальне використання мережевих ресурсів, балансування навантаження та уникнення перевантажень.

Недоліки SDN [7]:

1. Залежність від контролера.

У разі збою контролера мережа може частково або повністю втратити керованість. Хоча рішення з резервуванням існують, це все одно залишається критичною точкою.

Контролер у SDN виконує роль «мозку» мережі. Якщо він перестає працювати, то мережа частково або повністю втратить гнучке керування, нові з’єднання можуть не встановлюватись, бо немає кому видавати правила маршрутизації, деякі пристрої продовжать працювати за вже отриманими інструкціями, але зміни або реагування на події будуть неможливими, у мережі, що інтенсивно використовує динамічне керування (QoS, політики безпеки, балансування), простої можуть стати критичними.

Збій контролера можливий за таких умов, зокрема: програмні помилки або збої в ОС, помилки у коді контролера або конфлікти в системі можуть викликати зависання або зупинку процесів, перевантаження, надмірна кількість запитів або недостатні ресурси сервера можуть призвести до відмови, мережева недоступність (якщо з будь-яких причин комутатори втрачають зв’язок із контролером (наприклад, через збій каналу або маршрутизатора), керованість мережі порушується), кібератака (контролер може стати мішенню для DDoS або експлойтів, спрямованих на знищення центрального керування), відсутність

резервування (якщо не налаштовано кластеризацію чи резервний контролер, відмова одного вузла призводить до втрати контролю)

2. Складність впровадження в наявну інфраструктуру.

Переходити від традиційної архітектури до SDN може бути складно та потребувати серйозних змін у структурі мережі, що не завжди прийнятно для великих підприємств. Адже традиційна IT-інфраструктура часто створювалась без урахування можливості повної централізації керування або програмної логіки. Перехід до SDN передбачає перегляд архітектури, що може зачіпати: мережеві протоколи, підтримку OpenFlow чи інших southbound-протоколів, модернізацію обладнання, навички персоналу.

Основні виклики, що можуть виникнути під час такого переходу:

- несумісність обладнання (багато існуючих комутаторів або маршрутизаторів не підтримують OpenFlow або API-інтеграцію, тому їх доведеться оновити або повністю замінити);

- інтеграція з поточними сервісами (служби, що покладаються на статичні маршрути чи ручну конфігурацію, важко адаптувати до динамічної логіки SDN. Потрібна ретельна перевірка сумісності з фаєрволами, VPN, QoS, IDS/IPS);

- недостатній досвід персоналу (IT-відділ може не мати навичок роботи з SDN-контролерами, API, програмуванням мережевої логіки. Потрібні додаткові навчання або залучення зовнішніх консультантів)

- ризики для стабільності (будь-які помилки у новій логіці можуть спричинити мережеві збої або зниження доступності сервісів. Часто доводиться запускати SDN-платформи паралельно з традиційною мережею (гібридна модель), що ускладнює адміністрування);

- вартість і час. Крім вартості обладнання, треба враховувати час на розгортання, тестування, міграцію трафіку, що може бути критично для великих підприємств.

3. Потреба у висококваліфікованих фахівцях.

SDN вимагає від інженерів знань програмування, роботи з API, автоматизації та нових підходів до безпеки, що створює потребу в додатковому навчанні.

4. Можливі загрози безпеці.

Централізована архітектура при погано реалізованих засобах захисту може стати вразливою до атак, таких як DDoS на контролер, перехоплення команд або підробка політик.

У SDN вся логіка управління мережею зосереджена в контролері, тому при неналежному захисті він стає ключовою точкою уразливості. Основні слабкі місця включають:

а) атаки на контролер (DDoS, DoS). Контролер може бути перевантажений фальшивими запитами, особливо якщо немає обмежень частоти або фільтрації. У разі успішної атаки нові правила маршрутизації не видаються, і мережа втрачає керованість.

Приклад поганої реалізації: контролер приймає всі запити без автентифікації та без обмеження частоти (rate limiting). Зловмисник генерує тисячі підроблених потоків — контролер "захлинається" від навантаження.

б) Відсутність шифрування між контролером і пристроями. Якщо комунікація між контролером і комутаторами не зашифрована, зловмисник може перехопити або змінити передані інструкції.

Приклад: OpenFlow-трафік передається по HTTP (без TLS), що дозволяє зловмиснику в локальній мережі підслуховувати або підміняти пакети.

в) недостатній контроль доступу (ACL, RBAC). Якщо не обмежено, хто може взаємодіяти з API контролера, будь-хто з доступом до мережі може відправляти запити, змінювати політики, переглядати топологію.

Приклад: В API не реалізовано рольову модель – будь-який користувач з доступом до API може змінювати правила трафіку, створювати backdoor.

г) вразливості у плагінах або додатках на прикладному рівні. Програми, що працюють через Northbound API, можуть мати SQL-ін’єкції, XSS або логічні помилки, які зловмисник може використати для впливу на роботу мережі.

д) відсутність резервування або кластеризації. Одинокий контролер без резервного вузла є єдиною точкою відмови – це робить мережу нестійкою до як фізичних збоїв, так і атак.

5. Стандартизація та сумісність.

Попри розвиток SDN, ще існує обмежена кількість повністю сумісного обладнання або рішень від різних вендорів, що може створювати залежність від конкретного постачальника.

Хоча концепція SDN базується на відкритих протоколах і стандартах, на практиці її впровадження часто стикається з вузьким вибором обладнання, яке повноцінно підтримує всі функції SDN. Це створює технічні та стратегічні обмеження для організацій. Навіть якщо два пристрої формально підтримують, наприклад, OpenFlow, вони можуть реалізовувати різні версії, мати обмежену функціональність або специфічні розширення. Деякі вендори реалізують SDN-рішення, які працюють лише з власними контролерами чи платформами. Наприклад, обладнання підтримує SDN лише через фірмовий API, який несумісний з іншими системами. На ринку не так багато комутаторів або маршрутизаторів, які одночасно підтримують відкриті протоколи, масштабованість і повну функціональність SDN.

Наслідки може стати:

- залежність від одного вендора (vendor lock-in): якщо рішення базується на власному підході одного постачальника, перехід до іншого — дорогий і складний;

- обмеження в масштабуванні: складно додати нове обладнання без перевірки сумісності або придбання того самого бренду;

- зниження гнучкості: важко інтегрувати рішення з різних джерел або комбінувати їх у гібридних мережах.

Для вирішення цих проблем варто обирати відкриті SDN-контролери (наприклад, OpenDaylight, ONOS) і перевіряти сумісність з конкретним обладнанням, надавати перевагу вендорам, які підтримують open-source або

інтероперабельні протоколи, впроваджувати гібридні моделі поступово, щоб уникнути повної залежності.

6. Витрати на модернізацію.

Хоча експлуатаційні витрати можуть зменшитися, початкові витрати на оновлення інфраструктури (нове обладнання, ліцензії, навчання персоналу) можуть бути суттєвими.

Варто зазначити, що технологія SDN має певні виклики та обмеження [7]:

- Безпека контролера: як єдина точка управління, він може стати ціллю для атак;

- Сумісність: не всі пристрої підтримують сучасні протоколи SDN (наприклад, OpenFlow);

- Складність впровадження: Потрібна певна перебудова інфраструктури та персоналу.

SDN – це потужна технологія, яка відкриває нові горизонти для управління мережами, роблячи їх більш динамічними, ефективними та гнучкими. Однак впровадження SDN несе за собою низку викликів, таких як необхідність перебудови мережевої архітектури, забезпечення надійності контролера та потреба в нових знаннях персоналу.

Тому, перш ніж впроваджувати SDN, організаціям варто ретельно проаналізувати свої потреби, провести тестування та розглядати гібридні підходи – наприклад, інтегрувати SDN поетапно.

РОЗДІЛ 3
ПРИКЛАДИ РЕАЛІЗАЦІЇ SDN

3.1 Огляд успішних випадків використання SDN

Технологія програмно-визначених мереж (SDN) здобула велику популярність у різних галузях завдяки гнучкості, централізованому управлінню та високій масштабованості. Реальні приклади її впровадження у великих компаніях та організаціях демонструють значне підвищення ефективності, безпеки та економії ресурсів. Доцільно розглянути конкретні кейси застосування SDN у таких секторах, як телекомунікації, хмарні сервіси, дата-центри, фінансові організації та науково-дослідницькі мережі [8].

1. Google – B4: SDN у глобальній корпоративній мережі.

Google створила одну з найвідоміших SDN-систем – мережу B4, що з'єднує її дата-центри по всьому світу.

Результатом стало:

- повне використання пропускної здатності між дата-центрами;
- централізоване управління маршрутизацією даних;
- висока надійність завдяки контролю над маршрутом у реальному часі;
- власна SDN-інфраструктура допомогла суттєво зменшити затримки.

Можна з впевненістю зазначити, що SDN дозволив Google збудувати масштабовану, високооптимізовану WAN-мережу.

2. Amazon Web Services (AWS) – гнучке керування віртуальними мережами.

Amazon активно використовує елементи SDN для надання послуг Amazon VPC (Virtual Private Cloud). Це дозволяє користувачам створювати власні віртуальні мережі в хмарі з повним контролем.

Результати:

- миттєве створення, конфігурація та масштабування мережі;
- підтримка безпеки через віртуальні фаєрволи та контроль доступу;

- можливість підключення до локальних мереж через VPN або Direct Connect.

Тобто, SDN у хмарній інфраструктурі забезпечує гнучкість і безпечне підключення клієнтів до ресурсів.

3. GEANT – академічна науково-дослідна мережа Європи.

GEANT – це одна з найбільших мереж, що об’єднує університети та наукові установи Європи. Вона впровадила SDN для оптимізації маршрутизації і управління експериментальними мережами.

Результативність використання даної технології підтверджується такими фактами:

- дослідники можуть створювати віртуальні мережі для експериментів;

- зменшення часу на налаштування з тижнів до хвилин;

- гнучка інтеграція з різними мережевими платформами та протоколами.

SDN відкриває нові можливості для гнучкої роботи наукових спільнот та сприяє інноваціям.

4. Goldman Sachs – Автоматизація в фінансовому секторі.

Один із провідних фінансових гігантів використовує SDN для забезпечення безперебійної та безпечної передачі даних між офісами та дата-центрами.

Результати:

- автоматизоване управління трафіком і політиками безпеки;

- зменшення часу реагування на збої та покращення стійкості мережі;

- підвищення продуктивності за рахунок динамічного маршрутизування трафіку.

SDN дає змогу фінансовим організаціям дотримуватися суворих вимог безпеки та надійності.

5. Walmart – інтелектуальна мережа в ритейлі.

Walmart впровадила SDN для централізованого управління тисячами магазинів і пристроїв в реальному часі.

Результати:

- централізоване оновлення конфігурацій на всіх точках продажу;

- підвищена надійність POS-систем та онлайн-сервісів;

- оптимізація мережевого трафіку в години пік.

SDN допомагає масштабним роздрібним мережам швидко реагувати на зміни та оновлення [9].

Успішне впровадження SDN демонструє такі ключові переваги:

- значне зменшення витрат на обслуговування мереж;

- зростання продуктивності та надійності;

- можливість гнучкого масштабування в режимі реального часу;

- покращення рівня безпеки та керованості;

Сфери, де SDN особливо ефективний:

- хмарні обчислення;

- освітні та наукові мережі;

- фінансові установи;

- телекомунікації;

- ритейл.

3.2 Важливі проекти у галузі SDN

Технологія програмно-визначених мереж (SDN) стає дедалі важливішою в сучасному світі IT-інфраструктур. Вона дозволяє створювати більш гнучкі, масштабовані та керовані мережі, що оптимізують роботу підприємств і покращують їх здатність до реагування на змінювані умови. Розглянемо деякі з найбільш значущих і інноваційних проектів у галузі SDN, які змінили правила гри на ринку і стали еталонами для подальшого розвитку цієї технології [10].

1. OpenFlow – Протокол, який проклав шлях для SDN.

Проект: одним з найважливіших проектів, який лягло в основу SDN, є OpenFlow, який був розроблений в рамках Open Networking Foundation (ONF). Це перший стандартний протокол, що дозволив розділити контрольний і даний шари в мережах.

Опис: OpenFlow дозволяє централізовано управляти мережею, що дозволяє зміщення контролю з апаратного рівня на програмний. Це дає можливість операторам мереж керувати всіма компонентами мережі, такими як комутатори, маршрутизатори та інші пристрої, через програмні контролери.

Результати:

- OpenFlow став основою для розробки різноманітних SDN-платформ;

- його використання допомогло забезпечити більш гнучке і динамічне управління мережами в реальному часі;

- він дозволив реалізувати принципи «Software-Defined Networking», що значно покращило можливості масштабування та оптимізації мереж.

2. Google B4 – Мережа для з'єднання дата-центрів.

Проект: Google створила мережу B4, яка є однією з найбільших і найвідоміших програмно-визначених мереж, використовуваних для з'єднання її дата-центрів по всьому світу.

Опис: Мережа B4 використовує SDN для забезпечення максимальної пропускної здатності та зниження затримок між дата-центрами Google. Всі маршрути в мережі контролюються централізовано через програмний контролер, що дозволяє в реальному часі адаптувати мережу до змін у вимогах та навантаженнях.

Результати:

- значне зниження затримок і покращення пропускної здатності між дата-центрами;

- централізоване управління мережею дозволяє Google ефективно управляти величезною кількістю даних;

- підвищена надійність завдяки автоматичному перерозподілу трафіку в разі збоїв або перевантажень.

3. Facebook – Платформа Wedge та FBOSS.

Проект: Facebook розробив програмно-визначену мережеву платформу Wedge і програмне забезпечення FBOSS, які стали основою для побудови власних дата-центрів і серверних ферм.

Опис: Платформа Wedge є відкритою і використовує SDN для ефективного управління мережею на рівні дата-центрів. FBOSS є програмним забезпеченням, яке забезпечує гнучке керування мережею та взаємодію з апаратним забезпеченням через програмно визначену мережу.

Результати:

- Facebook може швидко масштабувати свою мережу, додаючи нові сервери і з'єднання;

- використання SDN дозволяє зменшити витрати на інфраструктуру;

- платформа допомогла знизити кількість помилок в управлінні мережею та підвищити її ефективність.

4. Cisco ACI (Application Centric Infrastructure).

Проект: Cisco ACI – це одна з найбільш розвинених платформ SDN, розроблена компанією Cisco для автоматизації та програмного управління дата-центрами.

Опис: Cisco ACI об'єднує апаратне забезпечення і програмне забезпечення для забезпечення автоматизації мережевих функцій та бізнес-процесів у дата-центрах. Платформа використовує SDN для централізованого управління політиками і конфігураціями мереж.

Результати:

- підвищена ефективність управління мережею, що дозволяє знижувати витрати на експлуатацію;

- полегшення інтеграції між різними апаратними та програмними компонентами;

- високий рівень автоматизації для швидкого реагування на зміну умов навантаження та вимог.

5. Amazon Web Services (AWS) – Віртуальні приватні мережі (VPC)

Проект: Amazon Web Services (AWS) використовує SDN для надання своїх послуг віртуальних приватних мереж (VPC). Це дозволяє клієнтам створювати персоналізовані, безпечні мережі в хмарі.

Опис: AWS забезпечує своїм користувачам можливість створювати віртуальні приватні мережі, які можуть бути налаштовані для обробки даних з високим рівнем безпеки. SDN дозволяє користувачам реалізовувати маршрутизацію, управління трафіком, а також застосовувати власні мережеві політики.

Результати:

- віртуальні приватні мережі забезпечують високий рівень безпеки та контролю;

- клієнти мають можливість гнучко управляти трафіком і адаптувати мережу під свої потреби;

- відсутність необхідності в апаратному забезпеченні для налаштування мережі значно спрощує її впровадження.

6. SDN для телекомунікаційних компаній – атрибути та впровадження.

Проект: Telenor, один з найбільших постачальників телекомунікаційних послуг у Європі, впровадив SDN для оптимізації своєї мережі та зниження витрат на обслуговування.

Опис: Telenor використовує SDN для централізованого управління мережею, що дозволяє зменшити час на налаштування та обслуговування мережі. Це допомогло значно знизити витрати на інфраструктуру та операційні витрати.

Результати:

- спрощення налаштування і масштабування мережі;

- покращення рівня обслуговування клієнтів через динамічну маршрутизацію;

- зниження операційних витрат завдяки автоматизації процесів [11].

Технологія SDN впроваджена в безлічі різних галузей, що дає змогу значно покращити ефективність, знижуючи витрати та забезпечуючи більшу гнучкість і контроль. Важливі проекти, такі як OpenFlow, Google B4, Facebook Wedge, Cisco ACI та AWS VPC, демонструють різноманітність і потенціал цієї технології, яку можна застосовувати як у великих корпораціях, так і у телекомунікаційних

компаніях або хмарних сервісах. Програмно-визначені мережі не тільки дозволяють більш ефективно управляти мережею, але й сприяють зниженню витрат та покращенню продуктивності компаній у різних секторах.

3.3 Результати досліджень та впровадження SDN

Програмно-визначені мережі (SDN, Software-Defined Networking) є однією з найбільш революційних технологій, яка змінює способи проектування, керування і масштабування мереж інфраструктури. З моменту свого виникнення SDN активно розвивається, і численні дослідження та реальні впровадження підтверджують її численні переваги та можливості для різних сфер, від телекомунікацій до корпоративних і хмарних мереж. Доцільно висвітлити та обґрунтувати результати досліджень та реальні випадки впровадження SDN, а також їхній вплив на мережеву інфраструктуру [12].

1. Оптимізація і гнучкість управління мережею

Одним із найбільших результатів досліджень SDN є значне полегшення управління мережею. Традиційні мережі вимагали для кожного пристрою окремих налаштувань та конфігурацій, що робило процес управління надзвичайно складним, особливо у великих масштабах.

Дослідження, проведене в університетах і наукових інститутах, показало, що завдяки централізованому управлінню в SDN, оператори мереж можуть управляти всіма компонентами мережі з одного контролера. Це дозволяє значно скоротити час на налаштування, а також швидко реагувати на збої або зміни в мережевому середовищі.

Результати впровадження:

- прикладом є дослідження компанії Google у використанні SDN для їхньої мережі між дата-центрами. Google використовує SDN для моніторингу та управління мережею в реальному часі, що дозволяє швидко адаптувати мережу до змін у навантаженні і забезпечувати високу доступність і надійність;

- зниження часу на налаштування мережі;

- централізоване управління всіма пристроями;

- підвищення гнучкості і здатності до масштабування мережі.

2. Зниження витрат на інфраструктуру та експлуатацію.

Одним із основних переваг SDN є зниження капітальних витрат на інфраструктуру. Завдяки тому, що мережеві функції тепер керуються програмно, а не апаратно, компанії можуть використовувати більш дешеві та менш потужні фізичні пристрої для виконання своїх мережевих функцій.

Дослідження, проведене у компанії Facebook, показало, що вони знизили витрати на апаратне забезпечення, впровадивши програмно-визначену мережу у своїх дата-центрах. Це дозволило компанії збільшити кількість серверів без додаткових витрат на купівлю дорогого обладнання.

Результати впровадження:

- зниження витрат на обладнання;

- збільшення ефективності використання існуючих пристроїв;

- зниження витрат на технічне обслуговування завдяки автоматизації мережевих функцій.

3. Покращення безпеки мережі.

SDN надає компаніям нові можливості для покращення безпеки мережі. Завдяки програмному управлінню всіма мережевими пристроями можна забезпечити більш точний контроль за доступом, моніторингом і реагуванням на загрози. Стандартні мережі мають труднощі в реальному часі здійснювати аналіз трафіку та реагувати на аномалії, у той час як SDN дозволяє впроваджувати більш складні та адаптивні стратегії безпеки [13].

У дослідженні, проведеному в галузі фінансових установ, було виявлено, що використання SDN дозволяє значно підвищити захист від DDoS-атак. Завдяки здатності швидко перенаправляти трафік через програмно визначені канали, система здатна виявляти і нейтралізувати загрози в реальному часі.

Результати впровадження:

- покращена безпека завдяки централізованому моніторингу;

- реагування на загрози в реальному часі;

- підвищення стійкості мережі до зовнішніх атак.

4. Підвищення масштабованості та гнучкості мереж.

Однією з основних переваг SDN є можливість масштабувати мережу без необхідності в дорогому та складному апаратному забезпеченні. Впровадження SDN дозволяє компаніям швидко і без труднощів розширювати мережу для підтримки зростаючих вимог бізнесу [14].

Дослідження, проведене в області хмарних обчислень, продемонструвало, що при впровадженні SDN у хмарних мережах значно зросла масштабованість і здатність до автоматичного розподілу ресурсів. Наприклад, у компанії Amazon Web Services (AWS), SDN дозволяє автоматично налаштовувати віртуальні приватні мережі (VPC), що дає можливість швидко масштабувати ресурси для задоволення попиту користувачів.

Результати впровадження:

- швидке та ефективне масштабування мережі;
- можливість автоматизованого налаштування і конфігурації;
- зниження часу на адаптацію мережі до нових вимог.

5. Удосконалення надійності та зниження часу відновлення

Завдяки можливості централізованого управління мережевими ресурсами, SDN дозволяє значно підвищити надійність мереж і знизити час відновлення після збоїв. Це стало можливим завдяки здатності перенаправляти трафік і автоматично переконфігурувати мережу при виявленні збоїв.

В дослідженнях, проведених на великих корпоративних мережах, було продемонстровано, що при використанні SDN знижений час на відновлення збоїв у мережі, оскільки мережеві пристрої можуть швидко перепрограмовуватися для відновлення роботи, замість того, щоб чекати на фізичне втручання техніків.

Результати впровадження:

- значне скорочення часу відновлення після збоїв;
- підвищення надійності мережі;
- зменшення впливу мережевих відмов на бізнес-процеси.

6. Інноваційні впровадження у телекомунікаційній галузі.

У телекомунікаційній індустрії SDN також демонструє великі результати в оптимізації та модернізації мереж. Зокрема, великі телекомунікаційні компанії використовують SDN для створення більш адаптивних і інтелектуальних мереж.

Одним із прикладів є впровадження SDN в мережах Telenor, що дозволило оптимізувати мережеві ресурси, знизити витрати на обслуговування та вдосконалити мережеві сервіси для кінцевих споживачів. Завдяки SDN Telenor змогла автоматизувати багато процесів і забезпечити кращу якість обслуговування.

Результати впровадження:

- покращення управління трафіком і послугами для користувачів;
- зниження витрат на обслуговування мережі;
- підвищення ефективності управління мережею в реальному часі.

Результати досліджень та впровадження SDN демонструють, що ця технологія має величезний потенціал у різних сферах, від корпоративних мереж до телекомунікацій і хмарних обчислень. Вона дозволяє знизити витрати, підвищити гнучкість і ефективність управління, а також значно покращити безпеку та надійність мережі. Прогнозується, що SDN стане основою для більшості майбутніх інфраструктур і продовжить змінювати способи побудови та управління мережами.

SDN – це ключова технологія майбутнього у сфері мереж, що забезпечує високу гнучкість, централізацію, програмну керованість і масштабованість. Її впровадження дозволяє знижувати витрати, підвищувати продуктивність мережі та швидко адаптувати інфраструктуру до змін бізнес-процесів.

Локальні мережі корпусів та навчальних закладів відіграють ключову роль у забезпеченні ефективної цифрової інфраструктури. Їх правильна побудова дозволяє оптимізувати доступ до інформаційних ресурсів, захистити мережевий трафік, а також створити умови для проведення прикладних та наукових досліджень.

Приклади ефективної реалізації мереж у навчальних закладах:

- Локальні сегментовані мережі у корпусах коледжу

Наприклад, у ЧДБК кожен корпус і гуртожиток має окремий мережевий сегмент (VLAN). Це дозволяє розмежувати трафік студентів, викладачів і адміністрації, що підвищує безпеку й контроль доступу.

- Централізоване управління через єдиний дата-центр

Всі корпуси підключені до центрального вузла (Core), що розміщується у головному серверному приміщенні. Це дозволяє адмініструвати мережу з єдиного центру, спрощуючи моніторинг, облік та технічну підтримку.

- Окрема мережа для наукових експериментів

Створення тестових підмереж, ізольованих від основної інфраструктури, дозволяє безпечно проводити дослідження з кібербезпеки, IoT або телемедицини без впливу на навчальний процес.

Рекомендації щодо побудови мереж навчального закладу

- Сегментація трафіку. Використання VLAN для логічного розділення трафіку: адміністративний, навчальний, гостьовий, лабораторний. Це зменшить ризик несанкціонованого доступу та покращить пропускну здатність мережі.

- Резервування каналів зв'язку. Критичні вузли повинні мати резервні підключення (failover), щоб забезпечити безперервність роботи під час збоїв.

- Безпечний Wi-Fi доступ. Для студентів і персоналу рекомендується використовувати WPA3 Enterprise, а для гостей – окрему гостьову мережу з авторизацією через портал.

- Інтеграція з хмарними платформами. Хмарні сервіси (Google Workspace, Microsoft 365, Moodle) дозволяють забезпечити безперебійний доступ до ресурсів навіть за межами кампусу.

- Моніторинг і аналітика трафіку. Запровадження систем типу Zabbix, PRTG або Grafana дає змогу відслідковувати стан мережі та виявляти потенційні загрози або перевантаження.

- Навчально-дослідницька користь. Побудова окремих експериментальних сегментів мережі дозволяє: тестувати нові протоколи (наприклад, IPv6, SDN);

імітувати атаки та перевіряти системи безпеки, працювати зі смарт-пристроями та IoT без ризику для основної мережі.

Грамотне проєктування мережі в освітньому закладі – це не лише питання технічної інфраструктури, а й основа для розвитку цифрових компетентностей, безпеки даних та академічної свободи в дослідженнях. Впровадження сучасних мережевих практик сприяє інноваційності та конкурентоспроможності навчального закладу.

SDN-контролер є головним елементом архітектури програмно-визначених мереж. Від його стабільності, гнучкості та функціональності залежить управління всією мережею, її безперебійна робота, захист від загроз, а також можливості масштабування.

Приклади, які демонструють важливість правильного вибору:

В коледжі було впроваджено SDN на базі легкого контролера Ryu для керування лабораторною мережею. Контролер чудово підходив для освітніх цілей, але коли кількість підключень зросла, він не справлявся з навантаженням. У результаті довелося перейти на більш масштабований контролер ONOS. Вибір контролера має враховувати перспективу зростання і не обмежуватись лише поточними потребами.

Компанія, що займається обробкою великих обсягів даних, інтегрувала OpenDaylight у свою мережу. Завдяки цьому контролеру вдалося централізувати управління, покращити безпеку й автоматизувати маршрутизацію трафіку. Це значно скоротило час реагування на інциденти та підвищило стійкість системи. Потужний контролер дозволяє реалізувати складну політику доступу, автоматизацію та підвищити загальну ефективність IT-інфраструктури.

Дослідницький центр, що працює з мережею IoT-пристроїв, обрав ONOS через його підтримку кластеризації та високої доступності. У разі відмови одного вузла керування, інший одразу брав на себе навантаження. Це дозволило уникнути простоїв і втрати даних. Надійність контролера критична для безперервного функціонування мереж з високими вимогами до доступності.

Рекомендації щодо вибору SDN-контролера:

- Призначення мережі. Обирайте контролер, який відповідає вашому сценарію (освіта, дата-центр, IoT, корпоративна мережа);

- Масштабованість. Перевірте, чи контролер підтримує горизонтальне масштабування або кластеризацію.

- Сумісність з OpenFlow. Контролер має підтримувати останні версії OpenFlow для сучасної взаємодії з пристроями.

- API та розширюваність. Наявність REST API або SDK важлива для інтеграції з іншими сервісами.

- Безпека. Вибирайте рішення, що має вбудовані засоби аутентифікації, шифрування та управління доступом.

- Підтримка та документація. Орієнтуйтесь на контролери з активною спільнотою та хорошою технічною підтримкою.

Обрання надійного SDN-контролера – це не лише питання технічної сумісності, а й стратегічне рішення, яке впливає на майбутній розвиток, продуктивність та безпеку всієї мережі. Для кожного типу середовища потрібен індивідуальний підхід з урахуванням специфіки завдань.

Використання інструментів моніторингу в SDN (Software-Defined Networking) – критично важливий аспект ефективного управління мережею. Ось приклади та рекомендації, які пояснюють, чому варто впроваджувати системи моніторингу у SDN-архітектурах:

Обґрунтування необхідності моніторингу у SDN:

1. Контролер визначає єдиний центр управління. У SDN саме контролер приймає всі рішення щодо маршрутизації та управління трафіком. Якщо він дає збій або працює неефективно, це впливає на всю мережу. Тому моніторинг дозволяє вчасно виявити перевантаження, затримки, або аномальну поведінку.

Приклади використання моніторингу у SDN:

Приклад 1: Корпусна мережа навчального закладу.

Навчальний заклад із великою кількістю підключень почав фіксувати зниження продуктивності під час онлайн-іспитів. Використання інструменту

sFlow-RT допомогло виявити, що під час іспитів виникало перевантаження одного з каналів зв’язку, викликане необмеженим потоковим трафіком.

Результат: Була змінена політика QoS через SDN-контролер, і трафік перенаправлено оптимальнішими маршрутами.

Приклад 2: Хмарна компанія.

Компанія з розміщення даних використовувала OpenDaylight з інтеграцією до Prometheus та Grafana. Завдяки цьому вони виявили затримки у комунікації між серверами, які негативно впливали на SLA. Моніторинг допоміг в реальному часі виявити «вузьке місце», яке не було видно без аналітики.

Результат: Вчасна реакція зменшила ризики порушення договірних зобов’язань перед клієнтами.

Приклад 3: Тестова SDN-мережа у дослідницькому центрі.

Під час тестування нової SDN-архітектури було використано Wireshark та Zabbix для збору трафіку й навантаження на мережеві елементи. Це дозволило відстежити, як оновлення програмної логіки контролера впливає на продуктивність.

Результат: Оптимізація алгоритмів прийняття рішень в контролері та зменшення затримок маршрутизації.

Рекомендації щодо впровадження моніторингу у SDN:

- Використовувати спеціалізовані інструменти. Наприклад: sFlow-RT, Open Network Insight, Grafana, ntopng. Ці інструменти не є виключно SDN-засобами. Вони можуть інтегруватися з SDN-контролерами, щоб розширити функціональність (наприклад, автоматичне реагування на аномалії), але їхнє головне призначення – аналіз та візуалізація мережевого трафіку в будь-якому типі інфраструктури.

- Інтегрувати моніторинг з контролером. Інформація від моніторингу має автоматично надходити до контролера для прийняття рішень у реальному часі.

- Встановлювати пороги сповіщень. Наприклад, при перевищенні навантаження на порт чи затримок понад N мс – надсилається повідомлення адміністратору.

- Використовувати аналітику для оптимізації. Зібрані дані можуть слугувати основою для покращення політик маршрутизації чи балансування навантаження.

- Регулярно тестувати та моделюйте стрес-сценарії. Моніторинг дозволяє перевіряти, як мережа реагує на несподівані навантаження чи збої.

Без належного моніторингу SDN-система ризикує втратити основну свою перевагу – гнучкість і контроль у реальному часі. Інструменти моніторингу не лише виявляють проблеми, а й забезпечують основу для адаптивного управління мережею, що є ключовим для сучасних цифрових інфраструктур.

ВИСНОВКИ

В процесі даного аналізу варто окреслити результати, що мають на меті синтезований опис ключових технологій у програмно-визначених мережах.

Ключовими технологіями у програмно-визначених мережах (SDN) є: централізоване управління мережею за допомогою SDN-контролера, розмежування контрольної та транспортної площин, використання відкритих протоколів, таких як OpenFlow, для взаємодії між пристроями та контролером, а також застосування API (Northbound і Southbound) для гнучкої інтеграції з додатками та мережею. Ці технології забезпечують високу гнучкість, масштабованість і простоту управління.

Програмно-визначені мережі (SDN) знаходять широке застосування в різних сферах завдяки своїй гнучкості, масштабованості та централізованому управлінню. Основні сфери використання включають:

- Центри обробки даних (ЦОД) – SDN дозволяє ефективно керувати ресурсами, автоматизувати налаштування мережі та швидко адаптуватися до змін у навантаженні.

- Хмарні обчислення – забезпечує динамічне розподілення мережевих ресурсів, спрощує керування та зменшує витрати на обслуговування.

- Телекомунікації – оператори зв’язку використовують SDN для оптимізації трафіку, впровадження нових послуг і спрощення інфраструктури.

- Корпоративні мережі – дозволяє централізовано управляти політиками безпеки, покращувати продуктивність і швидко впроваджувати зміни.

- Інтернет речей (IoT) – SDN допомагає керувати великою кількістю пристроїв, забезпечуючи масштабованість і безпеку.

- Мережі корпусів та навчальних закладів – сприяє ефективній організації доступу, ізоляції трафіку та експериментальним дослідженням.

Завдяки цим властивостям, SDN активно впроваджується в інфраструктури нового покоління, підтримуючи цифрову трансформацію бізнесу та IT-середовища.

Впровадження програмно-визначених мереж (SDN) вимагає ретельного планування, підготовки та дотримання низки рекомендацій, щоб забезпечити ефективність і стабільність роботи мережевої інфраструктури.

Варто відзначити ключові поради щодо впровадження SDN:

- Перед початком впровадження важливо чітко визначити, які проблеми має вирішити SDN: оптимізація трафіку, підвищення безпеки, спрощення управління чи зменшення витрат.

- Пілотне впровадження рекомендується почати з малого масштабу – провести пілотний проєкт у межах певного сегмента мережі, щоб оцінити переваги та виявити можливі труднощі без впливу на всю інфраструктуру.

- Підготовка персоналу навчання ІТ-фахівців, мережевих адміністраторів і інженерів новим підходам і технологіям є обов’язковим етапом, оскільки SDN суттєво змінює принципи управління мережею.

- Сумісність з наявною інфраструктурою. Важливо переконатися, що нові SDN-рішення можуть працювати з поточними мережевими пристроями або передбачити їх поступову заміну.

- Вибір надійного SDN-контролера, адже контролер є ядром SDN-мережі, тому потрібно обирати перевірені рішення з широкою підтримкою, хорошою документацією та активною спільнотою.

- Необхідно впровадити надійні механізми автентифікації, шифрування та контролю доступу до SDN-контролера, оскільки він має повний контроль над мережею.

- Система повинна легко адаптуватися до зростання кількості користувачів, сервісів і пристроїв, не втрачаючи продуктивності.

- Потрібно використовувати інструменти моніторингу для оцінки ефективності роботи SDN-мережі та оперативного реагування на потенційні загрози чи несправності.

Дотримання цих рекомендацій дозволяє зробити перехід до SDN безпечним, контрольованим і максимально ефективним для досягнення стратегічних цілей організації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Babayigit B., Ulu B., Abubaker M. Survey Studies of Software-Defined Networking: A Systematic Review and Meta-analysis // *Engineering Journal*. 2023. Vol. 27, No. 10. P. 33–66. URL: https://engj.org/index.php/ej/article/view/4514?utm_source=chatgpt.com

2. Alhilali A.H., Montazerolghaem A. Artificial intelligence based load balancing in SDN: A comprehensive survey. *arXiv preprint arXiv*:2308.02149, 2023. URL: https://arxiv.org/abs/2308.02149?utm_source=chatgpt.com

3. Hamarshe A., Ashqar H.I., Hamarsheh M. Detection of DDoS Attacks in Software Defined Networking Using Machine Learning Models. *arXiv preprint arXiv:*2303.06513, 2023. URL: https://arxiv.org/abs/2303.06513?utm_source=chatgpt.com

4. Nunez A., Ayoka J., Islam M.Z., Ruiz P. A Brief Overview of Software-Defined Networking. *arXiv preprint arXiv*:2302.00165, 2023. URL: https://arxiv.org/abs/2302.00165?utm_source=chatgpt.com

5. Bannour F., Souihi S., Mellouk A. Software-Defined Networking 2: Extending SDN Control to Large-Scale Networks. Hoboken: Wiley, 2023. URL: https://www.barnesandnoble.com/w/software-defined-networking-2-fetia-bannour/1142530507?utm_source=chatgpt.com

6. Sahoo B., Sahoo B., Mishra B.K. Software-Defined Networking for Future Internet Technology: Concepts and Applications. *Boca Raton: CRC Press*, 2023. URL: https://www.routledge.com/Software-Defined-Networking-for-Future-Internet-Technology-Concepts-and-Applications/Sahoo-Sahoo-Mishra/p/book/9781774639702?srsltid=AfmBOopBRjIJxzREUQjHUjH20yd7nKsOHMt8Kj7cb1o-PL5vaGdwLia0&utm_source=chatgpt.com

7. Hazim Alhilali A., Montazerolghaem A. Artificial intelligence based load balancing in SDN: A comprehensive survey. *arXiv preprint arXiv*:2308.02149, 2023. URL: https://arxiv.org/abs/2308.02149?utm_source=chatgpt.com

8. Kreutz, D., Ramos, F. M. V., 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. DOI: 10.1109/JPROC.2014.2371999.

9. Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 16(3), 1617–1634. DOI: 10.1109/SURV.2014.012214.00180.

10. Open Networking Foundation. (2014). SDN architecture overview. ONF TR-521. URL: https://opennetworking.org

11. Kim, H., & Feamster, N. (2013). Improving network management with software defined networking. *IEEE Communications Magazine*, 51(2), 114–119. DOI: 10.1109/MCOM.2013.6461195.

12. Lara, A., Kolasani, A., & Ramamurthy, B. (2014). Network innovation using OpenFlow: A survey. *IEEE Communications Surveys & Tutorials*, 16(1), 493–512. DOI: 10.1109/SURV.2013.081313.00105.

13. Hu, F., Hao, Q., & Bao, K. (2014). A survey on software-defined network and OpenFlow: From concept to implementation. *IEEE Communications Surveys & Tutorials*, 16(4), 2181–2206. DOI: 10.1109/SURV.2014.012214.00180.

14. Tootoonchian, A., & Ganjali, Y. (2010). HyperFlow: A distributed control plane for OpenFlow. *Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking*, 3–3. https://dl.acm.org/doi/10.1145/1855711.1855715.

15. 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. DOI: 10.1145/1355734.1355746.

16. Monsanto, C., Reich, J., Foster, N., Rexford, J., & Walker, D. (2013). Composing software-defined networks. *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, 1–13. URL: https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/monsanto.

17. Bari, M. F., Chowdhury, S. R., Ahmed, R., Boutaba, R., Esteves, R., & Granville, L. Z. (2013). Orchestrating virtualized network functions. *IEEE Transactions on Network and Service Management*, 13(4), 725–739. DOI: 10.1109/TNSM.2016.2597291.

18. Kreutz D., Ramos F.M.V., Verissimo P.E., Rothenberg C.E., Azodolmolky S., Uhlig S. Software-defined networking: A comprehensive survey // *Proceedings of the IEEE*. 2015. Vol. 103, No. 1. P. 14–76. DOI: 10.1109/JPROC.2014.2371999.

19. Nunes B.A.A., Mendonca M., Nguyen X.N., Obraczka K., Turletti T. A survey of software-defined networking: Past, present, and future of programmable networks // *IEEE Communications Surveys & Tutorials*. 2014. Vol. 16, No. 3. P. 1617–1634. DOI: 10.1109/SURV.2014.012214.00180.

20. Open Networking Foundation. SDN architecture overview // ONF TR-521. 2014. URL: https://opennetworking.org.

21. Kim H., Feamster N. Improving network management with software defined networking // *IEEE Communications Magazine*. 2013. Vol. 51, No. 2. P. 114–119. DOI: 10.1109/MCOM.2013.6461195.