DOI: https://doi.org/10.36910/6775-2524-0560-2026-62-41
УДК 004:681.5
Борак Дмитро Юрійович, магістрант
https://orcid.org/0009-0002-3521-8176
Абзятов Андрій Зігмундович, аспірант
https://orcid.org/0009-0007-0375-5141
Ліскевич Ростислав Ігорович, к.т.н., старший викладач
https://orcid.org/0009-0001-1335-4436
Обельовська Квітослава Михайлівна, к.т.н., доцент
https://orcid.org/0000-0002-8714-460X
Національний університет «Львівська політехніка», м. Львів, Україна

МОДЕЛЮВАННЯ ПРОГРАМНО-ВИЗНАЧЕНОЇ МЕРЕЖІ РОЗУМНОГО БУДИНКУ

Борак Д.Ю., Абзятов А.З., Ліскевич Р.І., Обельовська К.М. Моделювання програмно-визначеної мережі розумного будинку. У статті розглянуто проблему забезпечення ефективної роботи мережевої інфраструктури розумного будинку в умовах інтенсивного зростання кількості IoT-пристроїв. Розглянуто два варіанти мережевої інфраструктури: традиційний та з використанням концепції програмно-визначених мереж SDN (Software Defined Networking). Розроблено модель мережі розумного будинку, реалізовану в емуляторі Mininet, для порівняння традиційного підходу та підходу, що передбачає застосування в розумному будинку SDN з використанням контролера Ryu. Проведено експериментальні дослідження ключових характеристик мережі: затримки передавання пакетів, пропускної здатності, рівня втрат пакетів та споживання обчислювальних ресурсів. Для заданих вхідних даних, що були однаковими для обох підходів, результати експериментів показали, що SDN-архітектура забезпечила зменшення середньої затримки приблизно на 10 %, зменшення рівня втрат пакетів (2,3% проти 3%) за незначного зростання використання оперативної пам'яті. Отримані дані підтверджують доцільність застосування SDN у мережах розумного будинку для підвищення гнучкості та продуктивності.
Ключові слова: програмно-визначені мережі, SDN, розумний будинок, Mininet, контролер Ryu, IoT.

Borak D., Abzyatov A., Liskevich R., Obelovska K. Modeling a software-defined networking of a smart home. The article addresses the challenge of ensuring the effective operation of a smart home's network infrastructure in the condition of an intensively increasing number of IoT devices. Two options for the network infrastructure are considered: traditional and using the concept of software-defined networking SDN. A model of a smart home network, implemented in the Mininet emulator, is developed to compare the traditional approach and the approach that involves the use of SDN in a smart home, oriented on the Ryu controller. Experimental studies of key network characteristics are conducted, including packet transmission delay, bandwidth, packet loss level, and resource consumption. For given input data, which were the same for both approaches, the experimental results showed that the SDN architecture provided a reduction in average latency of approximately 10%, a decrease in packet loss (2.3% vs. 3%), with a slight increase in RAM usage. The obtained data confirm the feasibility of using SDN in smart home networks to increase flexibility and performance.
Keywords: software-defined networking, SDN, smart home, Mininet, Ryu controller, IoT.

Постановка проблеми. Сучасний розвиток концепції розумного будинку характеризується стрімким зростанням кількості підключених IoT-пристроїв різного функціонального призначення: датчиків температури, вологості, руху, освітлення, камер відеоспостереження, розумних вимикачів, побутової техніки, систем безпеки тощо. Це створює значне навантаження на мережеву інфраструктуру та висуває жорсткі вимоги до її масштабованості, гнучкості, надійності та енергоефективності.

Традиційні мережі розумних будинків, побудовані на базі децентралізованих протоколів маршрутизації та класичних комутаторів і маршрутизаторів, мають низку суттєвих недоліків у таких умовах:

- складність конфігурації та масштабування при додаванні нових пристроїв;
- необхідність ручного втручання для зміни політик маршрутизації чи пріоритетів трафіку;
- обмежені можливості динамічного перерозподілу ресурсів при зміні топології або пікових навантаженнях;
- низька адаптивність до різнорідності протоколів та вимог різних типів IoT-трафіку (реальний час, періодичне зчитування з датчиків, потокове відео).

Технологія програмно-визначених мереж (Software-Defined Networking, SDN) пропонує принципово інший підхід завдяки відокремленню площини керування (Control Plane) від площини передачі даних (Data Plane) та централізованому управлінню всім мережевим трафіком через програмований контролер. Це дозволяє:

- оперативно застосовувати гнучкі політики маршрутизації та QoS;
- динамічно адаптувати мережу до змін топології чи характеру навантаження;

- спрощувати інтеграцію нових пристроїв та сервісів;
- централізовано моніторити стан мережі та швидко реагувати на збої чи атаки.

Проте, незважаючи на численні теоретичні переваги, практична ефективність SDN у реальних або наближених до реальних сценаріях розумного будинку залишається недостатньо дослідженою. Відсутні систематичні порівняльні дослідження продуктивності традиційних та SDN-мереж в ідентичних умовах типового розумного будинку з урахуванням великої кількості різнорідних IoT-пристроїв. Таким чином, актуальним є моделювання мережі розумного будинку в контрольованому емуляційному середовищі, проведення порівняльного аналізу традиційного та SDN-підходів за ключовими експлуатаційними характеристиками (затримка, пропускна здатність, втрати пакетів, споживання ресурсів) та визначення доцільності впровадження SDN у такі системи.

Аналіз останніх досліджень і публікацій. Мережева інфраструктура розумного будинку є ключовим елементом реалізації концепції Інтернету речей (IoT), забезпечуючи обмін даними між різноманітними пристроями та системами для досягнення комфорту, енергоефективності та безпеки. Характерним для таких мереж є інтеграція сенсорів (температури, вологості, руху, освітлення, газу, диму тощо) та актуаторів, які виконують автоматизовані дії: регулювання клімату, керування освітленням, моніторинг безпеки, управління побутовою технікою [3, 4]. У роботах українських дослідників, зокрема Мацька Ю. та ін. (2025), запропоновано архітектуру IoT-системи для моніторингу ліфтів у багатоповерхових будинках, що включає сенсорні мережі та хмарні сервіси для реального часу обробки даних [3]. Аналогічно, Гринюк С. та ін. (2025) розробили систему автоматичного захисту "Розумний дім" з мобільним керуванням, яка інтегрує датчики та алгоритми реагування на загрози [4]. Ці дослідження підкреслюють актуальність гнучких мережевих рішень для IoT-середовищ, але не дають глибокого аналізу мережевої продуктивності.

Впровадження SDN у мережі розумних будинків розглядається як перспективний напрям для вирішення проблем традиційних архітектур. Систематизований огляд SDN представлено в роботі Hussain M. та ін. (2022), де класифіковано компоненти технології (контролери, протоколи, архітектури), проаналізовано переваги (централізоване керування, програмна гнучкість) та виклики (масштабованість, безпека) у контексті IoT [1]. Автори зазначають, що SDN дозволяє відокремити площину керування від площини даних, що особливо корисно для динамічних IoT-мереж з частою зміною топології. Подібний аналіз подано в огляді Shafiq S. та ін. (2024), де SDN розглядається в інтеграції з розподіленими обчисленнями та блокчейном для мобільних IoT-мереж, з акцентом на виклики безпеки та енергоефективності [2].

Ключовим протоколом SDN є протокол OpenFlow, який стандартизує взаємодію між комутаторами та контролером [5]. Серед контролерів найпоширенішими є Ryu, OpenDaylight (ODL), ONOS, Floodlight та NOX [6, 7]. За результатами порівняльних досліджень контролерів [7], проведених у Mininet, контролер Ryu показав найкращі результати для невеликих топологій (до 50 вузлів) з мінімальною затримкою обробки подій (близько 1–2 мс). У роботі [6] запропоновано багаторівневу SDN-архітектуру для розумних будинків, де два контролери (локальний для IoT-зон та глобальний для координації) забезпечують оптимізацію трафіку та ізоляцію сервісів.

Моделювання SDN у Mininet для виявлення DDoS-атак описано в [8], де з використанням машинного навчання досягнуто точність класифікації трафіку 99,75 %.

Mininet є стандартним інструментом для тестування топологій SDN. Автори роботи [9] дослідили моделі мереж SDN з MiniEdit, визначивши оптимальні топології для різних IoT-застосувань: зіркоподібну для низьконавантажених зон, кільцеву для високої відмовостійкості. В роботі [10] проаналізовано вплив топологій на час побудови мережі в Mininet; показано, що програмний API скорочує час конфігурації на 40–60 % порівняно з ручним налаштуванням. Вивченню продуктивності SDN-мереж з контролером Ryu у Mininet для типових IoT-потоків присвячена робота [11].

Виділення невирішених раніше частин загальної проблеми. Проведений аналіз вказує на актуальність тематики, проте більшість досліджень зосереджена на теоретичних моделях або великих корпоративних мережах, і не ставила своєю задачею проведення систематичного порівняння. Зокрема, відсутні дані про результати порівняння мережевих інфраструктур розумних будинків при традиційному підході та підході з орієнтацією на SDN. Запропонована в цій роботі модель заповнює цю прогалину, розглядає реалістичні сценарії з урахуванням зонування, різнорідності пристроїв та змішаного трафіку, надає дані, що можуть бути використані для практичного впровадження SDN у розумні будинки.

Метою досліджень є розробка моделі мережі розумного будинку, порівняльний аналіз характеристик традиційної та SDN-орієнтованої архітектур в ідентичних умовах та оцінка доцільності використання SDN для підвищення ефективності мереж розумних будинків.

Основна частина дослідження. Експериментальна модель відтворює типовий одноповерховий будинок з господарською будівлею (гаражем) та прибудинковою територією. Для забезпечення бездротового покриття вся територія поділена на чотири зони, кожна з яких обслуговується окремою точкою доступу (у моделі — OpenFlow-комутаторами): s0, s1, s2, s3 (рис. 1).

Рис. 1. Схема розташування маршрутизатора та комутаторів OpenFlow

Опис зон та підключених пристроїв:

- Зона s0 (кабінет, ванна кімната, коридор): точка доступу s0 розташована в кабінеті. Підключені пристрої: локальний домашній сервер, персональний комп'ютер, датчики температури та вологості, розумний вимикач освітлення (кабінет); датчик протікання води, датчик температури та вологості, розумний вимикач освітлення, розумна пральна машина (ванна); розумний вимикач освітлення, датчик відкриття дверей (коридор). Усі пристрої належать до підмережі 10.0.1.0/16.
- Зона s1 (спальня та дитяча кімната): датчики температури, вологості та CO_2, голосовий асистент, розумний вимикач освітлення, смартфон, камера відеоспостереження (спальня, камера — на подвір'ї); датчики температури та CO_2, відеоняня, розумний вимикач (дитяча). Підмережа 10.0.2.0/16.
- Зона s2 (вітальня, кухня, котельня): датчик температури та вологості, голосовий асистент, розумний телевізор, смартфон, розумний вимикач освітлення, камера відеоспостереження

(вітальня); датчики газу, диму та протікання води, розумний чайник, розумна посудомийна машина, голосовий асистент, розумний вимикач (кухня); датчики газу та диму, розумний котел (котельня). Підмережа 10.0.3.0/16.

- Зона s3 (гараж та подвір'я): розумний вимикач, контролер воріт, камера відеоспостереження (гараж); розумний дверний дзвінок, камера відеоспостереження (подвір'я). Підмережа 10.0.4.0/16.

Усі чотири комутатори з'єднані через центральний маршрутизатор r0 (IP-адреса 10.0.0.1), який забезпечує маршрутизацію між зонами та доступ до зовнішньої мережі (Інтернет). Така структура дозволяє моделювати як локальний трафік усередині зон, так і міжзональний та зовнішній обмін даними.

Фізична схема будинку використовується як основа для формування логічної топології. Процес побудови топології у Mininet базується на поетапному створенні мережевих елементів, їх логічному об'єднанні та налаштуванні параметрів з'єднань. Алгоритм створення топології мережі розумного будинку наведено на рис. 2.

Логічна топологія мережі формується програмно в Mininet за допомогою Python-скрипту, алгоритм створення показано на рис. 2.

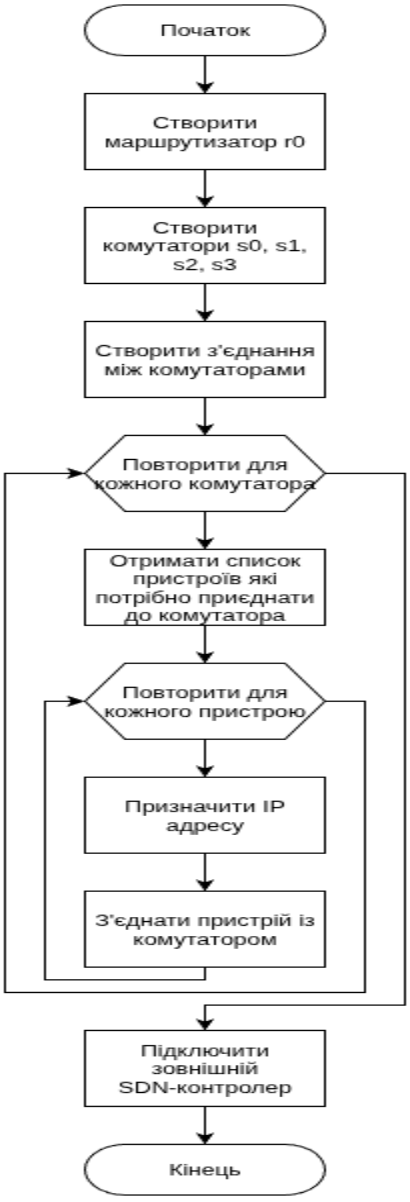

Рис. 2. Алгоритм створення топології мережі розумного будинку

Побудована логічна топологія (рис. 3) включає один маршрутизатор r0, чотири OpenFlow-комутатори s0–s3 та 40 розподілених за зонами хостів (IoT-пристрої та кінцеві вузли).

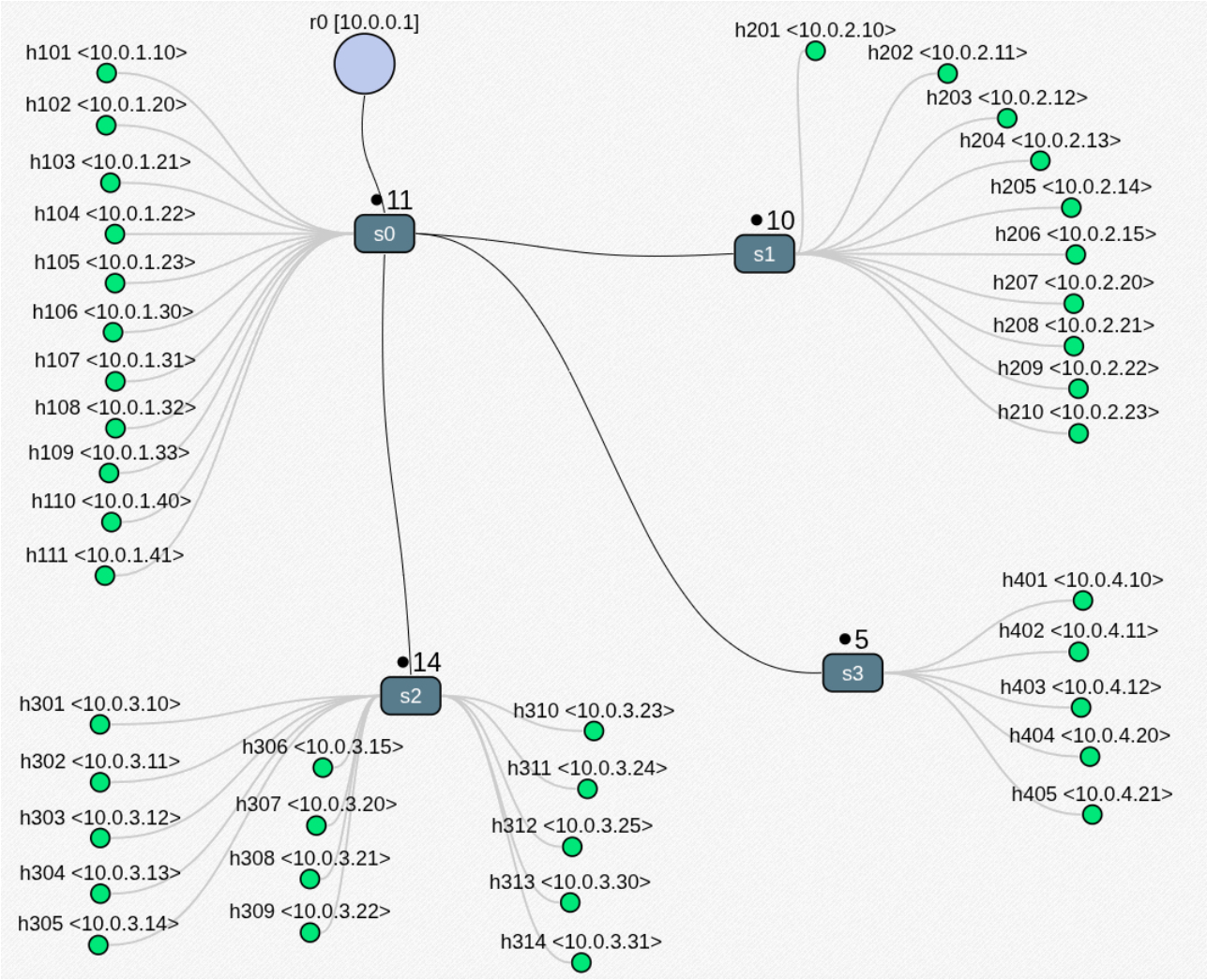

Рис. 3. Логічна топологія мережі

Параметри каналів зв'язку налаштовані для імітації реальних умов: між комутаторами та маршрутизатором пропускна здатність 1000 Мбіт/с, затримка 1 мс, відсутність втрат; між хостами та комутаторами пропускна здатність 100 Мбіт/с, затримка 10 мс, ймовірність втрат пакетів 1 %. Такий підхід забезпечує наближення до дротової (міжзональної) та бездротової (локальної) частин мережі.

Модель має модульну архітектуру:

- Модуль побудови топології — Python-скрипт Mininet, що створює вузли, призначає IP-адреси та параметри каналів.

- Модуль SDN-контролера — програма на базі Ryu, яка обробляє події Packet-In, формує правила потоків та реалізує централізоване керування.
- Модуль генерації трафіку — утиліти ping, iperf та скрипти для імітації IoT-навантаження.

У традиційному варіанті (SmartHomeNoSdn) використовується класична маршрутизація без зовнішнього контролера. У SDN-варіанті (SmartHomeWithSdn) комутатори автоматично підключаються до Ryu через OpenFlow.

Запропонована модель забезпечує реалістичне навантаження (змішаний трафік від датчиків, відео, команд керування) та дозволяє проводити відтворювані експерименти для порівняння двох архітектур.

Експериментальне середовище реалізовано на базі емулятора Mininet версії 2.3.0, який забезпечує віртуалізацію мережевих елементів (хостів, комутаторів, маршрутизаторів) та їхню взаємодію в реальному часі. Mininet використовує Python API для програмного створення топології, призначення параметрів (IP-адреси, затримки, пропускної здатності) та інтеграції з зовнішніми контролерами. Як SDN-контролер обрано Ryu (версія 4.34), який взаємодіє з комутаторами через протокол OpenFlow версії 1.3. Ryu обробляє події Packet-In, формує правила потоків та збирає статистику портів.

Дослідження проводилися на віртуальній машині з Ubuntu 22.04 LTS, процесором Intel Core i7 (4 ядра, 3.6 ГГц), 8 ГБ оперативної пам'яті та 50 ГБ дискового простору. Це забезпечує достатні ресурси для емуляції мережі з 40+ хостами без значних спотворень результатів.

Методика дослідження включає такі етапи:

1. Підготовка середовища: Установка Mininet та Ryu; створення Python-скриптів для топологій SmartHomeNoSdn (традиційна мережа) та SmartHomeWithSdn (SDN-мережа).
2. Запуск топологій: Для традиційної — активація маршрутизатора без зовнішнього контролера; для SDN — запуск Ryu в окремому терміналі, автоматичне підключення комутаторів.
3. Генерація трафіку: Використання утиліт ping (для затримки та втрат пакетів), iperf (для пропускної здатності). Трафік імітує реальні IoT-сценарії: періодичні запити від датчиків (ICMP), потокове відео від камер (TCP/UDP), фонове навантаження від усіх пристроїв.
4. Вимірювання метрик:
    - Затримка: 1000 ICMP-запитів між випадковими хостами (команда ping -c 1000).
    - Пропускна здатність: 10 вимірювань iperf між сервером та камерою (з/без фонового трафіку).
    - Втрати пакетів: Фіксація відсотка недоставлених пакетів під час тестів ping.
    - Споживання ресурсів: Моніторинг процесора та пам'яті за допомогою htop під час активного трафіку.
5. Повторюваність: Кожен тест повторюється 5–10 разів для усереднення результатів та зменшення впливу випадкових факторів.

Експерименти проводилися окремо для двох топологій: традиційної мережі (SmartHomeNoSdn) та програмно-визначеної мережі (SmartHomeWithSdn). Кожен тест повторювався 10 разів для забезпечення статистичної достовірності.

Вимірювання затримки виконувалося за допомогою 1000 ICMP-запитів між випадково обраними вузлами (домашній сервер — камера відеоспостереження). Результати (рис. 4) показали значну перевагу SDN-архітектури. Мінімальна затримка в традиційній мережі становила 44,201 мс, середня — 44,33 мс, максимальна — 45,388 мс. У SDN-мережі відповідні значення: 40,091 мс, 40,161 мс та 41,006 мс. Таким чином, середня затримка зменшилася приблизно на 10 % (4,17 мс).

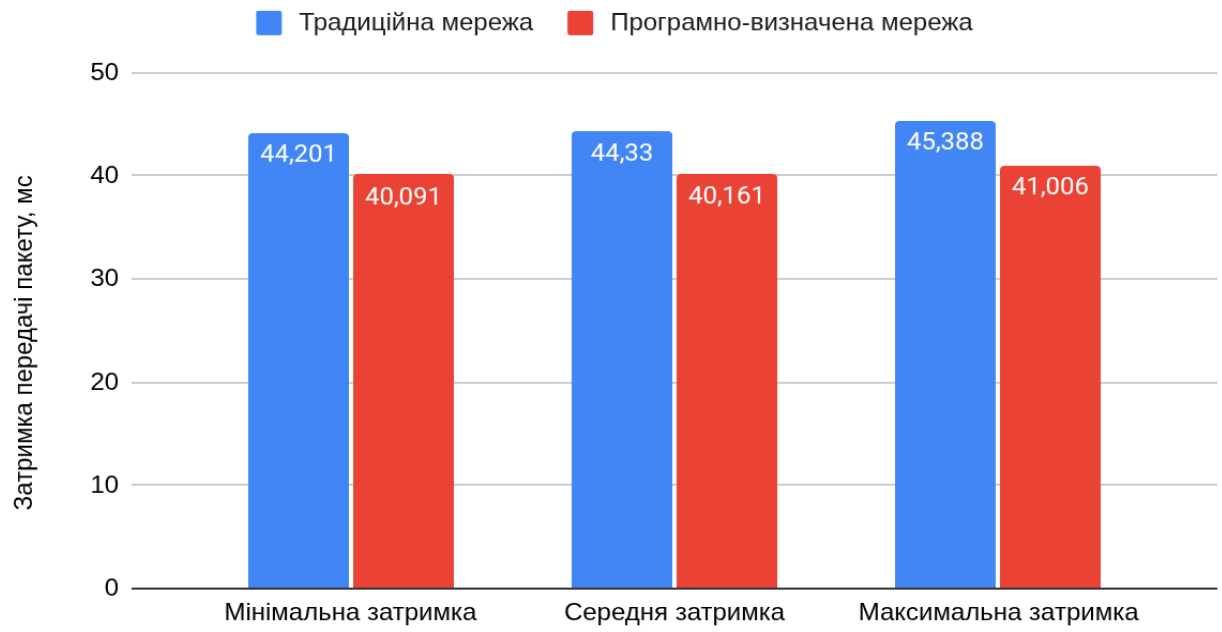

Рис. 4. Порівняння затримок у традиційній та SDN-мережах (мінімальна, середня, максимальна).

Під час тестів ping втрати пакетів у традиційній мережі становили 3 %, у SDN — 2,3 %. Зниження на 0,7 % пояснюється централізованим керуванням потоками, що дозволяє уникати перевантажених лінків.

Вимірювання за допомогою iperf (10 ітерацій) між домашнім сервером та камерою відеоспостереження без фонового трафіку показали середню пропускну здатність 47,31 Мбіт/с у традиційній мережі та 48,45 Мбіт/с у SDN (рис. 5). Перевага SDN становить близько 2,4 %.

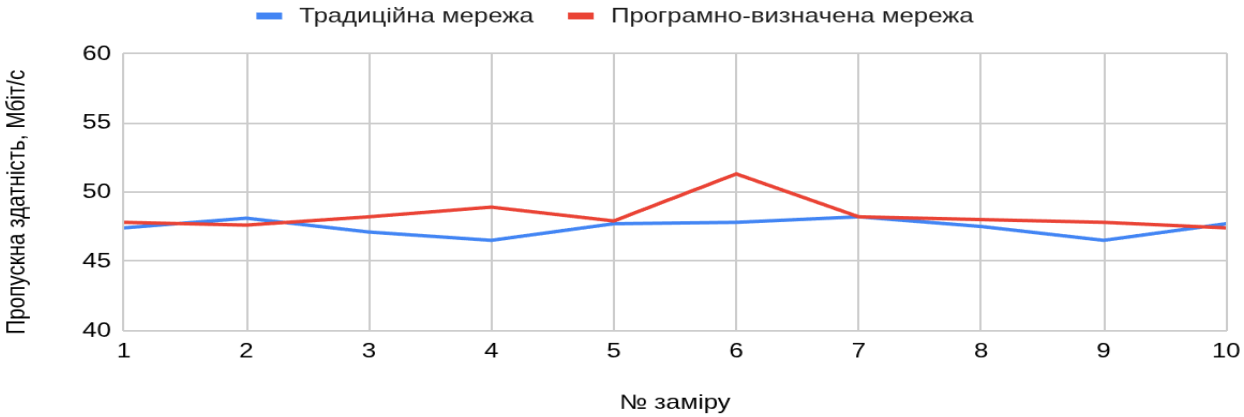

Рис. 5. Пропускна здатність без фонового трафіку.

Під реалістичним навантаженням (одночасна генерація трафіку від усіх IoT-пристроїв) пропускна здатність залишилася практично ідентичною: близько 46–47 Мбіт/с для обох архітектур (рис. 6). Це свідчить про стабільність SDN навіть при високому навантаженні.

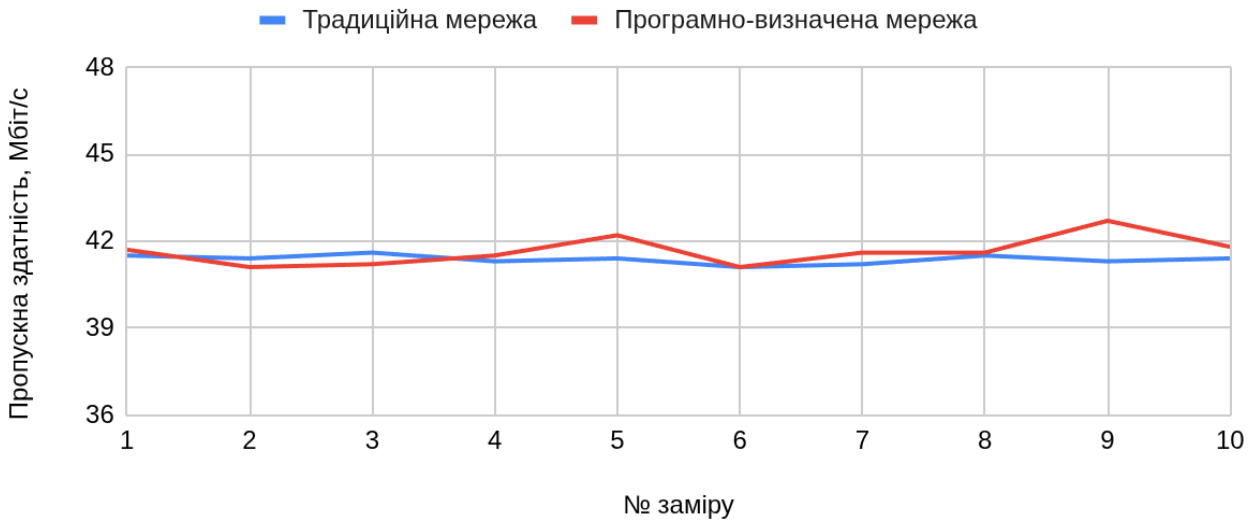

Рис. 6. Пропускна здатність під фоновим навантаженням від IoT-пристроїв.

Моніторинг за допомогою htop під час активного трафіку (рис. 7) показав, що навантаження на процесор коливалося в межах 2–13 % для обох архітектур без суттєвої різниці. Використання оперативної пам'яті в традиційній мережі — 141 МБ, у SDN — 240 МБ (зростання на 99 МБ через процес Ryu).

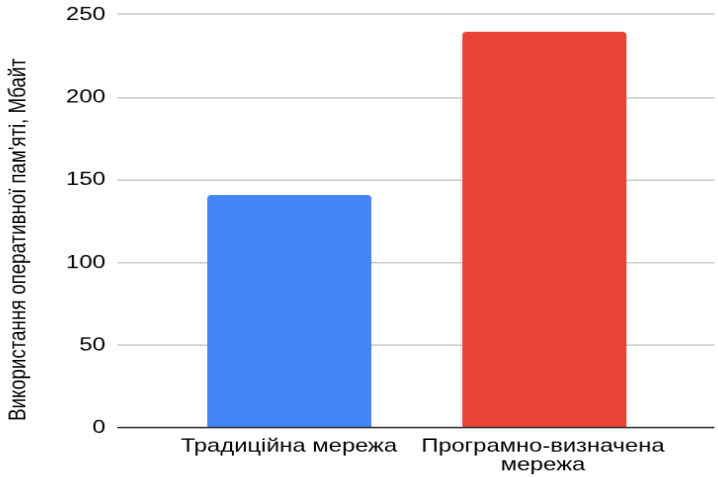

Рис. 7. Використання оперативної пам'яті та процесора під навантаженням.

Отримані експериментальні дані дозволяють оцінити ефективність програмно-визначеної мережі порівняно з традиційною архітектурою в умовах моделювання розумного будинку. Найвиразнішою перевагою SDN є зменшення затримки передавання пакетів приблизно на 10 % (з 44,33 мс до 40,161 мс у середньому). Це пояснюється централізованим керуванням потоками в контролері Ryu, який має глобальний огляд топології та встановлює оптимальні правила маршрутизації без проміжних обчислень на кожному комутаторі. Вища затримка першого пакета (86 мс) є очікуваною особливістю SDN через механізм Packet-In, але після встановлення правил подальша передача відбувається з мінімальними накладними витратами. У традиційній мережі затримки вищі через децентралізовану обробку таблиць маршрутизації на кожному пристрої.

Рівень втрат пакетів також нижчий у SDN (2,3 % проти 3 %). Централізоване керування дозволяє динамічно уникати перевантажених або ненадійних лінків, перерозподіляючи трафік за актуальним станом мережі. У традиційній архітектурі такі адаптації обмежені локальними протоколами.

Щодо пропускної здатності, SDN демонструє незначну перевагу без фонового навантаження (48,45 Мбіт/с проти 47,31 Мбіт/с), що пов'язано з оптимізацією шляхів та пріоритизацією потоків. Під реалістичним IoT-навантаженням різниця нівелюється, оскільки обмежуючим фактором стають параметри каналів доступу до хостів (100 Мбіт/с з затримкою 10 мс), а не міжзональна магістраль.

Це свідчить про те, що SDN зберігає стабільність навіть при високому одночасному трафіку від великої кількості пристроїв.

Споживання обчислювальних ресурсів показало очікуване зростання використання оперативної пам'яті в SDN (240 МБ проти 141 МБ) через процес контролера Ryu. Навантаження на процесор залишилося порівнянним (2–13 %), оскільки основні обчислення в SDN винесено на контролер, а комутатори виконують лише просте пересилання. Загалом додаткові витрати ресурсів є прийнятними з огляду на отримані покращення продуктивності.

Висновки та перспективи подальших досліджень. Розроблено модель для дослідження мережевої інфраструктури розумного будинку, що дозволяє здійснювати порівняльний аналіз традиційної мережі розумного будинку та мережі побудованої за концепцією програмно-визначених мереж. Для моделювання використано симулятор Mininet, в якості контролера SDN використано контролер Ryu. Розроблено методику досліджень для оцінки ефективності мережі як при традиційному застосуванні, так і з імплементацією програмно визначених мереж. Отримані результати показали кращі показники затримки та втрат пакетів за незначного зростання споживання ресурсів для мережі SDN. Зменшення затримки на 10 %, зниження втрат пакетів є суттєвими для застосувань, критичних до реального часу (відеоспостереження, системи безпеки, керування кліматом). Водночас зростання споживання пам'яті вказує на необхідність оптимізації контролера для вбудованих систем з обмеженими ресурсами.

Подальші дослідження можуть бути спрямовані як на модифікацію моделі під конкретні застосування, так і проведення системного дослідження для різних типів контролерів, різних типів пристроїв та режимів їх використання.

Список бібліографічного опису:

1. Hussain, M., Shah, N., Amin, R., Alshamrani, S. S., Alotaibi, A., & Raza, S. M. (2022). Software-defined networking: Categories, analysis, and future directions. Sensors, 22(15), 5551. https://doi.org/10.3390/s22155551
2. Shafiq, S., et al. (2024). A review on software-defined networking for Internet of Things inclusive of distributed computing, blockchain, and mobile network technology: Basics, trends, challenges, and future research potentials. International Journal of Distributed Sensor Networks, 2024. https://doi.org/10.1155/2024/9006405
3. Мацько, Ю., Христинець, Н., & Міскевич, О. (2025). Архітектурна та програмна модель системи Інтернету речей для моніторингу та управління технічним обслуговуванням ліфтів. Комп'ютерно-інтегровані технології: освіта, наука, виробництво, 61, 149–155. https://doi.org/10.36910/6775-2524-0560-2025-61-21
4. Гринюк, С., Поліщук, М., Костючко, С., Конкевич, Л., & Крив'янчук, Н. (2025). Система автоматичного захисту «Розумний дім» з мобільним керуванням. Комп'ютерно-інтегровані технології: освіта, наука, виробництво, 61, 242–248. https://doi.org/10.36910/6775-2524-0560-2025-61-33
5. Suzuki, K., Sonoda, K., Tomizawa, N., Yakuwa, Y., Uchida, T., Higuchi, Y., Tonouchi, T., & Shimonishi, H. (2014). A survey on OpenFlow technologies. IEICE Transactions on Communications, E97.B, 375–386. https://doi.org/10.1587/transcom.E97.B.375
6. Gilani, S. M. M., Usman, M., Daud, S., et al. (2024). SDN-based multi-level framework for smart home services. Multimedia Tools and Applications, 83, 327–347. https://doi.org/10.1007/s11042-023-15678-2
7. Gupta, N., Maashi, M. S., Tanwar, S., Badotra, S., Aljebreen, M., & Bharany, S. (2022). A comparative study of software-defined networking controllers using Mininet. Electronics, 11(17), 2715. https://doi.org/10.3390/electronics11172715
8. Karthika, P., & Karmel, A. (2023). Simulation of SDN in Mininet and detection of DDoS attack using machine learning. Bulletin of Electrical Engineering and Informatics, 12, 1797–1805. https://doi.org/10.11591/eei.v12i3.5232
9. Bhaskaran, S., & Muthuraman, S. (2024). Study on networking models of SDN using Mininet and MiniEdit. In 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) (pp. 1159–1164). https://doi.org/10.1109/IDCIoT59759.2024.10467218
10. Khoa, D. V., & Khanh, T. N. N. (2021). Emulation of software-defined network using Mininet. Dalat University Journal of Science, 11(1), 80–92. https://doi.org/10.37569/DalatUniversity.11.1.657(2021)
11. Романов, О., Сайченко, І., Марінов, А., & Сколець, С. (2021). Дослідження параметрів продуктивності SDN мережі за використанням емулятора мережі Mininet. Інформаційні та телекомунікаційні науки, 12(1), 24–32. https://doi.org/10.20535/2411-2976.12021.24-32

References:

1. Hussain, M., Shah, N., Amin, R., Alshamrani, S. S., Alotaibi, A., & Raza, S. M. (2022). Software-defined networking: Categories, analysis, and future directions. Sensors, 22(15), 5551. https://doi.org/10.3390/s22155551
2. Shafiq, S., et al. (2024). A review on software-defined networking for Internet of Things inclusive of distributed computing, blockchain, and mobile network technology: Basics, trends, challenges, and future research potentials. International Journal of Distributed Sensor Networks, 2024. https://doi.org/10.1155/2024/9006405
3. Matsko Yu., Khrystynets N., Miskevych O. (2025). Architectural and Software Model of an IoT System for Elevator Maintenance Monitoring and Management. Computer-integrated technologies: education, science, production, 61, 149–155. https://doi.org/10.36910/6775-2524-0560-2025-61-21

4. Hrynyuk S., Polishchuk M., Kostyuchko S., Konkevych L., Kryv'yanchuk N. (2025). Smart Home automatic protection system with mobile control. Computer-integrated technologies: education, science, production, 61, 242–248. https://doi.org/10.36910/6775-2524-0560-2025-61-33
5. Suzuki, K., Sonoda, K., Tomizawa, N., Yakuwa, Y., Uchida, T., Higuchi, Y., Tonouchi, T., & Shimonishi, H. (2014). A survey on OpenFlow technologies. IEICE Transactions on Communications, E97.B, 375–386. https://doi.org/10.1587/transcom.E97.B.375
6. Gilani, S. M. M., Usman, M., Daud, S., et al. (2024). SDN-based multi-level framework for smart home services. Multimedia Tools and Applications, 83, 327–347. https://doi.org/10.1007/s11042-023-15678-2
7. Gupta, N., Maashi, M. S., Tanwar, S., Badotra, S., Aljebreen, M., & Bharany, S. (2022). A comparative study of software-defined networking controllers using Mininet. Electronics, 11(17), 2715. https://doi.org/10.3390/electronics11172715
8. Karthika, P., & Karmel, A. (2023). Simulation of SDN in Mininet and detection of DDoS attack using machine learning. Bulletin of Electrical Engineering and Informatics, 12, 1797–1805. https://doi.org/10.11591/eei.v12i3.5232
9. Bhaskaran, S., & Muthuraman, S. (2024). Study on networking models of SDN using Mininet and MiniEdit. In 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) (pp. 1159–1164). https://doi.org/10.1109/IDCIoT59759.2024.10467218
10. Khoa, D. V., & Khanh, T. N. N. (2021). Emulation of software-defined network using Mininet. Dalat University Journal of Science, 11(1), 80–92. https://doi.org/10.37569/DalatUniversity.11.1.657(2021)
11. Oleksandr I. Romanov, Ivan O. Saychenko, Anton I. Marinov, Serhii S. Skolets (2021). Research of SDN network performance parameter using the Mininet network emulator. Information and telecommunication sciences, 12(1), 24–32. https://doi.org/10.20535/2411-2976.12021.24-32

Історія статті:
Отримано: 20.01.2026 Доопрацьовано: 05.02.2026 Прийнято до друку: 23.03.2026 Опубліковано: 29.03.2026