

Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

КВАЛІФІКАЦІЙНА РОБОТА

за освітнім ступенем «бакалавр»

на тему:

**«Симуляція IoT-мережі з використанням
протоколу MQTT у Cisco Packet Tracer»**

Виконала:

студентка IV-го курсу

групи ПЗ-41

спеціальності 121 «Інженерія
програмного забезпечення»

Бель Ангеліна Олександрівна

Керівник:

к.т.н., доцент Шинкарчук Н. В.

Рівне – 2026

АНОТАЦІЯ ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ

Бель А.О. Симуляція IoT-мережі з використанням протоколу MQTT у Cisco Packet Tracer. Кваліфікаційна робота на здобуття ступеня «бакалавр» за спеціальністю 121 «Інженерія програмного забезпечення» – Рівненський державний гуманітарний університет. Рівне, 2026. 48 с.

Кваліфікаційна робота присвячена дослідженню та розробці моделі IoT-мережі з використанням протоколу MQTT у середовищі Cisco Packet Tracer. Основна увага приділена проєктуванню системи «Розумного будинку», у якій реалізовано взаємодію між датчиками, виконавчими пристроями, ноутбуком та іншими IoT-компонентами, а також автоматичне керування пристроями залежно від показників датчиків. У межах роботи створено модель локальної IoT-мережі, що забезпечує обмін даними між пристроями, моніторинг стану системи та виконання автоматизованих сценаріїв керування.

У роботі розглянуто теоретичні засади функціонування Інтернету речей, особливості побудови IoT-мереж та сучасні технології передачі даних. Проведено аналіз принципів роботи протоколу MQTT, його моделі publish/subscribe та переваг використання у системах із обмеженими ресурсами. Також досліджено можливості програмування одноплатного комп'ютера в Cisco Packet Tracer (SBC) за допомогою мови Python для реалізації обробки даних і керування виконавчими пристроями.

Практична частина роботи включає створення мережевої інфраструктури в Cisco Packet Tracer, налаштування MQTT-брокера на ноутбуці, підключення IoT-пристроїв до мережі та реалізацію системи автоматичного контролю температури. У процесі моделювання було налаштовано обмін повідомленнями між пристроями, автоматичне керування системами охолодження та нагрівання, а також моніторинг стану пристроїв у режимі реального часу.

Ключові слова: Інтернет речей (IoT), MQTT, Cisco Packet Tracer, симуляція мережі, Розумний будинок, MQTT-брокер, одноплатний комп'ютер (SBC), автоматизація, Python.

ABSTRACT OF THE BACHELOR THESIS

Bel, A.O. Simulation of an IoT network using the MQTT protocol in Cisco Packet Tracer. Qualification thesis for the degree of Bachelor in the specialty 121 "Software Engineering" – Rivne State Humanitarian University. Rivne, 2026. 48 p.

This thesis is devoted to the research and development of an IoT network model using the MQTT protocol in the Cisco Packet Tracer environment. The main focus is on designing a "Smart Home" system that implements interaction between sensors, actuators, a laptop, and other IoT components, as well as automatic control of devices based on sensor readings. As part of this work, a model of a local IoT network was created that enables data exchange between devices, system status monitoring, and the execution of automated control scenarios.

This paper examines the theoretical foundations of the Internet of Things, the characteristics of IoT network architecture, and modern data transmission technologies. It analyzes the operating principles of the MQTT protocol, its publish/subscribe model, and the advantages of using it in resource-constrained systems. It also explores the possibilities of programming a single-board computer in Cisco Packet Tracer (SBC) using the Python language to implement data processing and control of actuators.

The practical part of the work includes creating a network infrastructure in Cisco Packet Tracer, configuring an MQTT broker on a laptop, connecting IoT devices to the network, and implementing an automatic temperature control system. During the simulation, message exchange between devices, automatic control of cooling and heating systems, and real-time monitoring of device status were configured.

Keywords: Internet of Things (IoT), MQTT, Cisco Packet Tracer, network simulation, Smart Home, MQTT broker, single-board computer (SBC), automation, Python.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ОСНОВИ ПОБУДОВИ ІОТ-МЕРЕЖ І ЗАСАДИ ОБМІНУ ПОВІДОМЛЕННЯМИ В ІОТ-СИСТЕМАХ.....	5
1.1. Поняття та концепція Інтернету речей	5
1.2. Архітектура та основні компоненти IoT-систем	7
1.3. Технології та протоколи передачі даних у IoT-мережах	10
РОЗДІЛ 2. ПРОТОКОЛ І АРХІТЕКТУРА MQTT. ВИКОРИСТАННЯ MQTT В ІОТ-ПРОЄКТАХ.....	14
2.1. Призначення та принцип роботи протоколу MQTT.....	14
2.2. Архітектура MQTT та модель publish-subscribe	17
2.3. Використання MQTT у сучасних IoT-системах.....	20
РОЗДІЛ 3. ОГЛЯД CISCO PACKET TRACERЯК СЕРЕДОВИЩА МОДЕЛЮВАННЯ MQTT-ВЗАЄМОДІЇ ІОТ-ПРИСТРОЇВ.....	23
3.1. Призначення та функціональні можливості Cisco Packet Tracer	23
3.2. Підтримка IoT-технологій у Cisco Packet Tracer	25
3.3. Інструменти моделювання IoT-мереж та MQTT-взаємодії	28
РОЗДІЛ 4. ПРОЄКТУВАННЯ, РЕАЛІЗАЦІЯ ТА СИМУЛЯЦІЯ MQTT- ВЗАЄМОДІЇ В CISCO PACKET TRACER.....	32
4.1. Проєктування моделі IoT-мережі	32
4.2. Налаштування MQTT-брокера та IoT-пристроїв.....	35
4.3. Реалізація обміну повідомленнями та аналіз результатів симуляції.....	40
ВИСНОВКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	44
ДОДАТОК А.....	46

ВСТУП

Актуальність роботи. Швидкий розвиток концепції Інтернету речей (IoT) та зростаючий попит на автоматизацію життєвого простору зумовлюють необхідність впровадження ефективних та масштабованих протоколів передачі даних. Протокол MQTT став стандартом для децентралізованих систем завдяки своїй енергоефективності, мінімальній надлишковості даних та здатності стабільно працювати в умовах обмежених мережевих ресурсів. Використання середовища моделювання Cisco Packet Tracer дозволяє проводити детальне проектування та тестування архітектури «Розумного будинку» без значних фінансових витрат на фізичне обладнання, що є критично важливим для оптимізації логіки керування на базі одноплатних комп'ютерів.

Робота в середовищі Cisco Packet Tracer сприяє розвитку навичок побудови складних топологій мереж, програмуванню взаємодії пристроїв на мові Python та розумінню принципів роботи MQTT-брокерів у віртуалізованому середовищі.

Мета дослідження: розробити та реалізувати симуляційну модель IoT-мережі «Розумний будинок» з використанням протоколу MQTT у середовищі Cisco Packet Tracer.

Об'єктом дослідження є процеси передачі даних та керування пристроями в мережах Інтернету речей.

Інструментом дослідження є середовище Cisco Packet Tracer забезпечує можливості для віртуального моделювання IoT-систем і мереж. Воно дозволяє відтворювати взаємодію пристроїв та процеси обміну даними в IoT-мережі з використанням MQTT.

Предметом дослідження є методи побудови архітектури та програмної реалізації логіки взаємодії пристроїв IoT за допомогою протоколу MQTT у віртуальному середовищі.

Завдання дослідження:

1. Провести аналіз теоретичних засад функціонування технологій IoT та особливостей протоколу MQTT.

2. Дослідити можливості середовища Cisco Packet Tracer для моделювання інтелектуальних систем.
3. Розробити архітектуру IoT-мережі, що включає MQTT-брокер, MQTT-клієнт, мережеві шлюзи та кінцеві пристрої.
4. Програмно реалізувати логіку взаємодії пристроїв на мові Python та провести тестування працездатності симуляції.

Апробація результатів кваліфікаційної роботи. Результати виконання кваліфікаційної роботи, окремі її аспекти та одержані узагальнення і висновки були оприлюднені на I Міжнародній науково-практичній конференції «Сучасні інформаційні технології: від теорії до практики» (м. Луцьк, 2026 р.), звітній науковій конференції викладачів, співробітників та здобувачів вищої освіти Рівненського державного гуманітарного університету за 2025 рік (м. Рівне, 2026 р.).

Публікації. Результати, які були отримані в ході кваліфікаційного дослідження опубліковано у вигляді тез доповідей у збірнику матеріалів I Міжнародної науково-практичної конференції «Сучасні інформаційні технології: від теорії до практики» (м. Луцьк, 2026 р.) (Додаток А).

Структура роботи. Робота складається зі вступу, чотирьох розділів, висновків та списку використаних джерел. У першому розділі здійснено теоретичний огляд концепції Інтернету речей, розглянуто особливості функціонування IoT-мереж, сучасні технології та протоколи передачі даних. Другий розділ присвячено аналізу архітектури системи «Розумний будинок», принципів MQTT-взаємодії та особливостей використання протоколу MQTT у системах автоматизації. Третій розділ містить огляд середовища Cisco Packet Tracer, його функціональних можливостей для моделювання IoT-мереж. У четвертому розділі описано проектування та реалізацію моделі IoT-мережі, налаштування MQTT-брокера, підключення IoT-пристроїв і роботу з одноплатним комп'ютером в Cisco Packet Tracer, а також реалізацію програмної логіки мовою Python і результати симуляції MQTT-взаємодії між пристроями. Список літератури містить шістнадцять джерел.

РОЗДІЛ 1

ОСНОВИ ПОБУДОВИ ІОТ-МЕРЕЖ І ЗАСАДИ ОБМІНУ ПОВІДОМЛЕННЯМИ В ІОТ-СИСТЕМАХ

1.1. Поняття та концепція Інтернету речей

Концепція Інтернету речей (Internet of Things, IoT) є одним із ключових напрямів розвитку сучасних інформаційно-комунікаційних технологій. Вона передбачає об'єднання фізичних об'єктів, оснащених сенсорами, виконавчими пристроями та вбудованими обчислювальними модулями, у єдину мережу з можливістю збору, передавання та обробки даних із мінімальною участю людини [1]. Завдяки цьому IoT-системи забезпечують автоматизацію процесів, підвищення ефективності керування ресурсами та швидке реагування на зміни параметрів навколишнього середовища.

Важливою особливістю концепції Інтернету речей є інтеграція фізичного та цифрового середовища. Завдяки використанню сенсорних пристроїв та мережевих технологій фізичні об'єкти отримують можливість передавати інформацію про свій стан або про параметри навколишнього середовища до інформаційних систем. У результаті формується єдиний інформаційний простір, у якому дані можуть аналізуватися та використовуватися для автоматичного прийняття рішень або підтримки управлінських процесів [2].

З розвитком інформаційних технологій кількість підключених до мережі пристроїв постійно зростає. За оцінками міжнародних дослідницьких організацій, у найближчі роки кількість IoT-пристроїв у світі становитиме десятки мільярдів. Це створює нові можливості для розвитку цифрових сервісів, автоматизації виробничих процесів та створення інтелектуальних систем управління інфраструктурою міст, підприємств і житлових об'єктів [3].

Інтернет речей застосовується у безлічі галузей, охоплюючи як повсякденні домашні гаджети, так і масштабні інфраструктурні рішення для міст та підприємств. Найбільш поширеними напрямками впровадження цієї технології є

автоматизація житла, логістика й транспорт, енергетичний сектор, медицина, а також аграрна сфера. Ключовою вимогою до таких мереж є забезпечення безперервної та швидкої комунікації між апаратними компонентами та сервісами в режимі реального часу.

Одним із найбільш перспективних напрямів розвитку IoT є створення систем «розумних міст» (Smart City). У них датчики та мережеві пристрої використовуються для моніторингу транспортних потоків, управління енергоспоживанням, контролю якості повітря, організації систем освітлення та гарантування безпеки. Використання технологій Інтернету речей у міській інфраструктурі сприяє підвищенню ефективності використання ресурсів і покращенню якості життя населення [4].

Для забезпечення ефективної роботи таких систем необхідно організувати взаємодію великої кількості різноманітних пристроїв. До складу IoT-мереж можуть входити сенсорні модулі, контролери, мережеве обладнання, серверні платформи та програмні сервіси. Кожен із цих елементів виконує власні функції, але разом вони формують єдину систему збору, передавання та обробки інформації. Саме тому при проєктуванні IoT-рішень важливу роль відіграє визначення архітектури системи та принципів взаємодії її компонентів.

Технічна архітектура IoT-рішень є комплексною багаторівневою структурою. Вона забезпечує тісну інтеграцію апаратних засобів та програмного забезпечення. Для підвищення ефективності проєктування систему прийнято розділяти на три функціональні рівні. Такий підхід гарантує гнучкість системи та її швидке масштабування.

Сенсорний рівень – фундамент всієї системи, що безпосередньо взаємодіє з фізичним світом. Головним завданням тут є безперервний збір метрик навколишнього середовища. До складу рівня входять різноманітні датчики: від сенсорів температури до детекторів руху. Також тут функціонують виконавчі пристрої (актуатори), які виконують отримані команди. Апаратне забезпечення на цьому етапі часто має жорсткі ліміти обчислювальної потужності. Через обмежений енергоресурс пристроїв виникає потреба у використанні

спеціалізованих протоколів. Останні повинні мати мінімальне навантаження та бути енергоефективними.

Мережевий рівень – виконує функцію сполучної ланки в архітектурі. Він відповідає за безпечне та стабільне транспортування даних між пристроями та серверами. Інфраструктура рівня охоплює шлюзи, маршрутизатори та різні комунікаційні канали. Тут реалізується логіка передачі пакетів інформації у реальному часі. Мережеві компоненти забезпечують координацію роботи всіх вузлів системи. Саме на цьому етапі досягається висока надійність зв'язку у територіально розподілених мережах. Це дозволяє системі працювати як єдиний цілісний організм.

Рівень додатку призначений для зберігання, аналізу та візуалізації даних, отриманих із сенсорного рівня. До нього належать серверні платформи, бази даних, веб-інтерфейси та аналітичні модулі, які реалізують сценарії автоматизації та взаємодії з користувачем. Функціональні можливості прикладного рівня визначають практичну цінність IoT-системи та ефективність її використання в конкретній предметній області [3, 4].

Мережева архітектура IoT вирізняється високим ступенем децентралізації та об'єднує величезну кількість взаємопов'язаних вузлів. У подібному середовищі, де комунікація часто відбувається в умовах нестабільного з'єднання та апаратних обмежень пристроїв, критично важливо забезпечити ефективну маршрутизацію та передачу даних. Саме це зумовлює потребу в спеціалізованих інструментах взаємодії та масовому переході на легковагові, оптимізовані протоколи обміну повідомленнями (такі як MQTT чи CoAP).

1.2. Архітектура та основні компоненти IoT-систем

Функціонування Інтернету речей неможливе без використання ефективних мережевих технологій, які забезпечують передавання даних між фізичними пристроями, проміжними вузлами обробки та прикладними сервісами. На відміну від традиційних комп'ютерних мереж, IoT-системи характеризуються великою

кількістю пристроїв із обмеженими ресурсами – низькою обчислювальною потужністю, мінімальним енергоспоживанням та невеликою пропускнуою здатністю каналів зв'язку [1].

Однією з характерних особливостей IoT-систем є різноманітність пристроїв, що беруть участь у процесі обміну даними. До складу таких систем можуть входити сенсорні вузли, виконавчі пристрої, контролери, шлюзи та серверні платформи. Сенсорні вузли виконують функцію збору інформації про параметри навколишнього середовища, зокрема температуру, вологість, освітленість, рух або рівень шуму. Виконавчі пристрої, своєю чергою, реагують на отримані команди та здійснюють певні дії, наприклад вмикання освітлення, регулювання температури або керування обладнанням.

Важливу роль у таких системах відіграють також IoT-шлюзи (gateway), які забезпечують взаємодію між локальними мережами пристроїв та глобальною мережею Інтернет. Їх значення полягає не лише у передачі даних, але й у виконанні функцій попередньої обробки, агрегації та фільтрації інформації. Таким чином, шлюзи дозволяють зменшити обсяг даних, що передаються до хмарних платформ, та оптимізувати використання мережевих ресурсів.

Подальший аналіз IoT-архітектури показує, що вибір мережевих технологій є одним із ключових факторів, який визначає ефективність усієї системи. Залежно від сфери застосування та масштабу системи, в IoT використовуються різні мережеві технології. Для локального з'єднання пристроїв поширеними є Wi-Fi, Bluetooth Low Energy, ZigBee та Z-Wave. Вони забезпечують відносно невелике енергоспоживання і достатню швидкість передачі даних на коротких відстанях. У випадках необхідності передавання даних на значні відстані застосовуються мобільні мережі (3G, 4G, 5G), а також спеціалізовані технології LPWAN – LoRaWAN і NB-IoT. Вибір конкретної технології визначається вимогами до енергоспоживання, дальності зв'язку, швидкості передавання даних та надійності з'єднання [2].

Для IoT-систем важливе значення мають протоколи прикладного рівня, які забезпечують обмін даними між пристроями та сервером. Найчастіше для цього

використовуються HTTP, CoAP, AMQP та MQTT. Кожен із цих протоколів має свої особливості та сферу застосування. Для середовища Інтернету речей важливими є невеликий обсяг службових даних, можливість роботи при нестабільному з'єднанні та швидка передача повідомлень.

У мережевій взаємодії IoT-пристроїв використовуються різні моделі обміну повідомленнями. Найбільш поширеними є модель «клієнт – сервер» та модель «publish/subscribe» (публікація – підписка). У першій пристрій-клієнт надсилає запит до сервера, після чого отримує відповідь. Такий підхід широко використовується у веб-технологіях і реалізований, зокрема, у протоколі HTTP. Проте у великих IoT-системах ця модель може створювати значне навантаження на сервер і вимагати постійних запитів від пристроїв. Альтернативою є модель publish/subscribe, у якій пристрої не взаємодіють безпосередньо один з одним. Замість цього вони передають повідомлення до спеціального проміжного вузла – брокера повідомлень. Пристрої-публікатори надсилають дані на певну тему (topic), а пристрої-підписники отримують повідомлення, підписавшись на відповідні теми. Такий підхід дозволяє значно підвищити масштабованість системи та зменшити навантаження на мережу.

Протокол HTTP широко застосовується у веб-системах, однак його використання в IoT обмежене через значний обсяг службових заголовків та орієнтацію на модель «клієнт – сервер». Для пристроїв із низькою потужністю більш доцільним є використання легковагових протоколів, таких як CoAP або MQTT, які оптимізовані для передавання невеликих повідомлень і підтримки великої кількості підключених вузлів [3].

Сучасні тенденції розвитку IoT також пов'язані з впровадженням концепції кордонних обчислень (Edge Computing), коли частина обробки даних здійснюється безпосередньо поблизу джерела їх генерації. Такий підхід дозволяє зменшити затримки передачі, знизити навантаження на центральні сервери та підвищити стійкість системи до збоїв у мережі [4].

З огляду на специфіку середовища Інтернету речей до протоколів обміну повідомленнями висувається низка вимог. Насамперед вони повинні забезпечувати

мінімальний обсяг службових даних, оскільки багато IoT-пристроїв мають обмежену пропускну здатність каналів зв'язку. Важливою характеристикою є також підтримка роботи в умовах нестабільних або переривчастих мережових з'єднань. Крім того, протоколи повинні підтримувати механізми підтвердження доставки повідомлень, можливість асинхронної передачі даних, а також ефективне управління великою кількістю підключених пристроїв. У сучасних IoT-системах важливим є і питання безпеки передавання даних, зокрема підтримка механізмів автентифікації та шифрування інформації.

Отже, мережеві технології та протоколи є критичними компонентами IoT-інфраструктури. Правильний вибір технології зв'язку та протоколу обміну даними безпосередньо впливає на ефективність, масштабованість і надійність системи. З огляду на специфіку IoT-середовища особливу увагу приділяють легковаговим протоколам обміну повідомленнями, зокрема MQTT.

1.3. Технології та протоколи передачі даних у IoT-мережах

Ефективність функціонування систем Інтернету речей значною мірою залежить від того, як організовано обмін даними між пристроями, мережевими вузлами та хмарними платформами. На відміну від традиційних комп'ютерних мереж, IoT-середовище характеризується великою кількістю пристроїв з обмеженими обчислювальними ресурсами, низьким енергоспоживанням і невеликою пропускну здатністю каналів зв'язку. Через це у таких системах доцільним є використання спеціалізованих технологій зв'язку та протоколів передачі даних, які оптимізовані для роботи в умовах обмежених ресурсів.

Однією з важливих особливостей передачі даних у мережах Інтернету речей є необхідність забезпечення взаємодії великої кількості розподілених пристроїв. При цьому пристрої можуть працювати в умовах нестабільних мережових з'єднань, а обсяг повідомлень, що передаються, зазвичай є невеликим. Тому протоколи, що використовуються в IoT-системах, повинні забезпечувати ефективну передачу

коротких повідомлень, мінімальний обсяг службових даних та можливість роботи в мережах із низькою пропускнуою здатністю [2].

Для організації зв'язку в системах Інтернету речей застосовуються різні мережеві технології, такі як Wi-Fi, Bluetooth Low Energy, ZigBee та Z-Wave. Використовують їх залежно від умов експлуатації системи, необхідної дальності передачі даних та кількості підключених пристроїв.

Однією з найбільш поширених є Wi-Fi, яка забезпечує достатньо високу швидкість передачі інформації та дозволяє інтегрувати IoT-пристрої у звичайні локальні мережі. Ця технологія часто використовується у системах «розумного будинку», відеоспостереженні та побутових IoT-пристроях, однак має відносно високе енергоспоживання.

Для пристроїв, що працюють від батареї, більш доцільним є використання Bluetooth Low Energy (BLE). Ця технологія забезпечує обмін даними на невеликій відстані при значно меншому енергоспоживанні, тому активно застосовується у фітнес-браслетах, безпроводних навушниках, медичних сенсорах та мобільних пристроях. Також у сенсорних мережах широко використовуються технології ZigBee та Z-Wave, які дозволяють створювати мережі з великої кількості пристроїв та підтримують передачу даних між вузлами без значного навантаження на мережу [3].

Для передачі інформації на значні відстані використовуються мобільні мережі та спеціалізовані технології LPWAN (Low Power Wide Area Network), зокрема LoRaWAN та NB-IoT. Такі технології дозволяють передавати невеликі обсяги даних на великі відстані при мінімальному енергоспоживанні, що робить їх придатними для використання в системах моніторингу навколишнього середовища, управління міською інфраструктурою та інших проєктах «розумного міста» [4].

Передача даних у IoT-середовищі базується на використанні стандартних мережевих протоколів Інтернету. Для транспортування повідомлень між вузлами мережі найчастіше використовуються протоколи TCP та UDP. Протокол TCP забезпечує надійну доставку даних завдяки механізмам контролю помилок,

підтвердження отримання пакетів і повторної передачі втрачених повідомлень. Такий підхід є важливим у системах, де втрата інформації є критичною. Водночас використання TCP створює додаткове навантаження на мережу через значний обсяг службових даних. Протокол UDP працює без встановлення постійного з'єднання між пристроями та не потребує підтвердження доставки пакетів. Це дозволяє значно зменшити затримки передачі даних і скоротити службовий трафік. Саме тому UDP часто використовується у системах, де швидкість передачі важливіша за абсолютну надійність доставки повідомлень, наприклад у сенсорних мережах та системах телеметрії.

На прикладному рівні використовуються спеціалізовані протоколи обміну повідомленнями між IoT-пристроями та серверними платформами. Одним із найпоширеніших є протокол HTTP, який широко використовується у веб-технологіях. Однак для IoT-середовища його застосування є не завжди ефективним через значний обсяг службових заголовків і орієнтацію на модель взаємодії «клієнт – сервер». Для пристроїв із обмеженими ресурсами часто застосовується протокол CoAP (Constrained Application Protocol), який був спеціально розроблений для сенсорних мереж. CoAP використовує транспортний протокол UDP та має компактний формат повідомлень, що дозволяє зменшити навантаження на мережу та забезпечити ефективну передачу даних навіть у мережах з обмеженою пропускною здатністю [5, 6].

Одним із найбільш популярних протоколів у системах Інтернету речей є MQTT (Message Queuing Telemetry Transport). Його основною особливістю є використання моделі publish/subscribe, яка забезпечує обмін повідомленнями через окремий сервер – брокер (рис. 1.1.). Такий підхід дозволяє організувати взаємодію між великою кількістю пристроїв без необхідності прямого з'єднання між ними. MQTT характеризується компактною структурою повідомлень та мінімальним обсягом службових даних, завдяки чому протокол ефективно працює навіть у мережах із низькою пропускною здатністю або нестабільним підключенням. Додатковою перевагою є підтримка різних рівнів QoS (Quality of Service, QoS), що

дає можливість налаштовувати рівень надійності доставки повідомлень залежно від потреб конкретної IoT-системи.

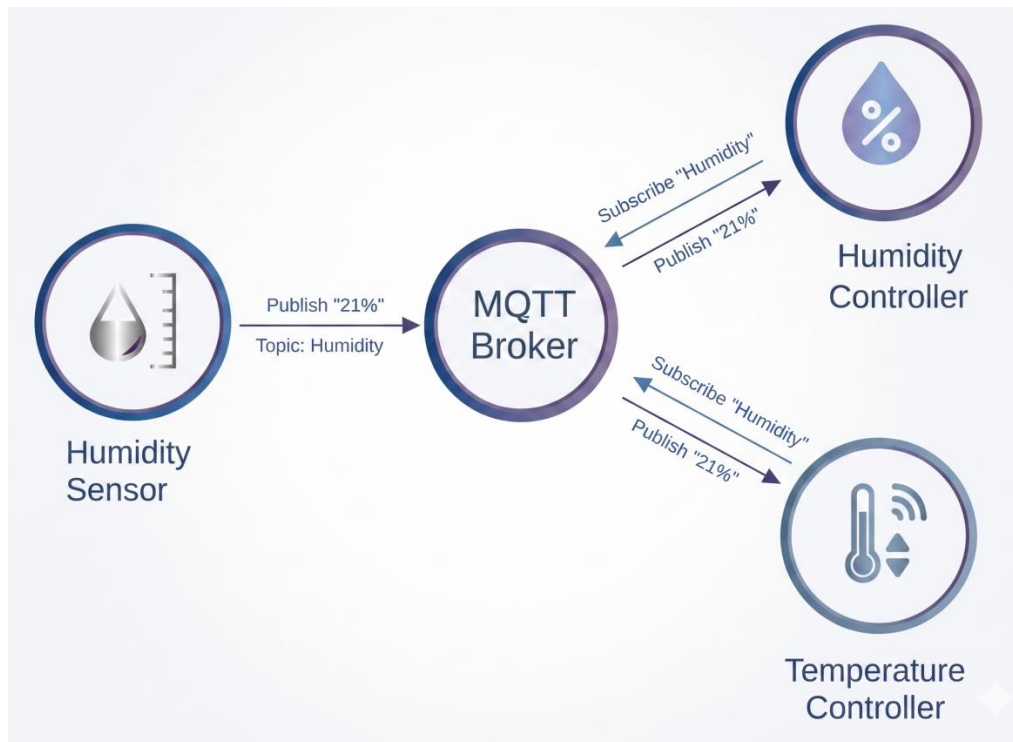


Рисунок 1.1. Схема роботи MQTT-брокера

У сучасних IoT-системах важливу роль відіграє також питання безпеки передачі даних. Оскільки пристрої часто працюють через відкриті мережі Інтернет, виникає необхідність захисту інформації від несанкціонованого доступу та перехоплення. Для цього використовуються механізми шифрування даних, автентифікації користувачів і захищені канали зв'язку на основі протоколів TLS та SSL. Реалізація таких механізмів дозволяє підвищити рівень безпеки IoT-мереж та забезпечити надійний обмін даними між пристроями.

Отже, передача даних у системах Інтернету речей здійснюється за допомогою різноманітних мережевих технологій і протоколів, які забезпечують взаємодію великої кількості пристроїв у розподіленому середовищі. Вибір конкретної технології залежить від вимог до дальності зв'язку, енергоспоживання, швидкості передачі даних та надійності мережі. Серед протоколів обміну повідомленнями особливу роль відіграє MQTT, який завдяки своїй ефективності та простоті став одним із найбільш поширених рішень у IoT-системах.

РОЗДІЛ 2

ПРОТОКОЛ І АРХІТЕКТУРА MQTT.

ВИКОРИСТАННЯ MQTT В ІОТ-ПРОЄКТАХ

2.1. Призначення та принцип роботи протоколу MQTT

З розвитком Інтернету речей кількість підключених пристроїв у мережах постійно зростає. Багато таких пристроїв працюють із дуже обмеженими обчислювальними ресурсами. Вони мають невелику пам'ять і низьку продуктивність. Також вони часто працюють у мережах із нестабільним або повільним з'єднанням. Через це використання класичних мережевих протоколів не завжди є ефективним або доцільним. Для забезпечення швидкого та зручного обміну повідомленнями в IoT-системах широко використовується протокол MQTT (Message Queuing Telemetry Transport).

MQTT є легким протоколом передачі повідомлень, який призначений для роботи в мережах із низькою пропускну здатністю. Його основне завдання полягає в організації обміну невеликими повідомленнями між пристроями через центральний сервер. Протокол характеризується простотою реалізації, невеликим обсягом службових даних та низьким навантаженням на мережу. Завдяки цьому MQTT активно застосовується у системах «розумного будинку», промисловій автоматизації, моніторингових системах та інших IoT-рішеннях.

Протокол MQTT був розроблений наприкінці 1990-х років компаніями IBM та Arcom для використання у системах дистанційного моніторингу. Основною задачею було створити механізм передавання телеметричних даних із віддалених пристроїв через нестабільні супутникові канали зв'язку. Саме тому в основу протоколу були закладені принципи мінімального використання мережевих ресурсів, компактності повідомлень та високої надійності доставки інформації. Згодом MQTT став відкритим стандартом і отримав широке поширення у сфері IoT-технологій [3].

Однією з головних характеристик MQTT є його легковаговість. Заголовки повідомлень у цьому протоколі мають дуже малий розмір, що дозволяє суттєво зменшити обсяг даних, які передаються. Це особливо важливо для пристроїв, які працюють від акумуляторів або використовують бездротові канали зв'язку з обмеженою пропускнуою здатністю. Завдяки цьому MQTT може ефективно використовуватися навіть у мережах із низькою швидкістю передачі даних або нестабільним інтернет-з'єднанням [4].

У цьому протоколі обмін повідомленнями організовано таким чином, що пристрої не взаємодіють безпосередньо один з одним. Уся передача даних відбувається через центральний сервер, який називається брокером. Брокер приймає повідомлення від клієнтів, обробляє їх і розсилає тим пристроям, які підписані на відповідні канали. Такий принцип роботи спрощує взаємодію між великою кількістю пристроїв. Він також підвищує масштабованість системи та робить її більш гнучкою.

Клієнтами в системі MQTT можуть бути різні типи пристроїв або програмного забезпечення. Це можуть бути датчики температури, контролери освітлення, сервери обробки даних, мобільні застосунки або інші компоненти інформаційної системи. Кожен клієнт може виконувати роль як відправника повідомлень, так і їх отримувача. Для цього використовується механізм підписки на певні інформаційні канали, які в MQTT називаються топіками [2].

Топік є логічним каналом, через який здійснюється передача повідомлень. Клієнт може публікувати інформацію в певний топік або підписуватися на нього, щоб отримувати повідомлення від інших клієнтів. Наприклад, датчик температури може передавати дані в топік `home/temperature`, а мобільний застосунок користувача може бути підписаний на цей топік для відображення поточної температури у приміщенні. Таким чином, брокер автоматично передає всі нові повідомлення від датчика до клієнтів, що підписані на відповідний канал [3].

Ще однією важливою особливістю протоколу MQTT є підтримка різних рівнів якості доставки повідомлень (Quality of Service, QoS). Цей механізм дозволяє

визначити, наскільки надійно має бути доставлене повідомлення від відправника до отримувача. У протоколі MQTT передбачено три основні рівні QoS.

Перший рівень QoS (QoS 0) передбачає передачу повідомлення без підтвердження доставки. У цьому випадку повідомлення відправляється один раз і може бути втрачено у разі проблем із мережею. Такий режим використовується тоді, коли швидкість передачі важливіша за абсолютну надійність доставки даних.

Другий рівень (QoS 1) гарантує, що повідомлення буде доставлене принаймні один раз. Для цього використовується механізм підтвердження отримання повідомлення. Якщо підтвердження не надходить, повідомлення може бути повторно передано. У результаті отримувач може отримати одне і те саме повідомлення кілька разів, але ймовірність його втрати значно зменшується.

Третій рівень (QoS 2) забезпечує найвищу надійність передачі даних. У цьому випадку застосовується складніший механізм обміну службовими повідомленнями, який гарантує, що повідомлення буде доставлено рівно один раз. Такий режим використовується в системах, де втрата або дублювання повідомлень є неприпустимими, наприклад у промислових системах [4].

Для підтримки стабільної роботи мережі MQTT використовує також механізм підтримки з'єднання (Keep Alive). Клієнти періодично надсилають брокеру спеціальні сигнали, які повідомляють про те, що з'єднання залишається активним. Якщо брокер протягом певного часу не отримує таких сигналів, він може вважати клієнта відключеним та завершити з'єднання. Це дозволяє своєчасно виявляти проблеми з мережею та підтримувати актуальний стан системи.

Ще одним важливим механізмом MQTT є так звані збережені повідомлення (Retained Messages). Якщо повідомлення публікується з відповідною ознакою, брокер зберігає його і передає новим клієнтам одразу після підписки на відповідний топік. Завдяки цьому нові пристрої або користувачі можуть миттєво отримати актуальний стан системи без необхідності чекати на появу нового повідомлення.

Окрім цього, протокол MQTT підтримує механізм Last Will and Testament (LWT), який дозволяє автоматично повідомляти інші пристрої про аварійне відключення клієнта. Під час підключення клієнт може передати брокеру

спеціальне повідомлення, яке буде опубліковане у випадку його несподіваного відключення. Такий механізм широко використовується для моніторингу стану пристроїв у розподілених системах.

Завдяки описаним особливостям MQTT забезпечує ефективний і надійний обмін даними між великою кількістю пристроїв у мережах Інтернету речей. Простота реалізації, невелике навантаження на мережу та підтримка різних рівнів надійності доставки роблять цей протокол одним із найбільш популярних рішень для побудови сучасних IoT-систем.

Отже, протокол MQTT є спеціалізованим мережевим протоколом, призначеним для організації ефективного обміну повідомленнями між пристроями в умовах обмежених мережесих ресурсів. Його архітектура дозволяє забезпечити масштабовану та гнучку взаємодію між численними компонентами системи.

2.2. Архітектура MQTT та модель publish-subscribe

Однією з ключових особливостей протоколу MQTT є його архітектура, яка базується на використанні моделі обміну повідомленнями publish-subscribe. Такий підхід суттєво відрізняється від традиційної клієнт-серверної взаємодії, що використовується у багатьох мережесих протоколах. Завдяки цій моделі MQTT забезпечує гнучку, масштабовану та ефективну систему передачі даних між великою кількістю пристроїв у розподілених мережах, що є особливо важливим для систем Інтернету речей [7].

У традиційній клієнт-серверній архітектурі один пристрій безпосередньо звертається до іншого для отримання або передавання інформації. Такий підхід може бути неефективним у середовищі, де одночасно працює велика кількість пристроїв, оскільки кожен із них повинен знати адресу інших учасників мережі. У системах IoT кількість підключених пристроїв може бути дуже великою, тому використання прямої взаємодії між ними ускладнює організацію мережі та знижує її масштабованість.

Для розв'язання цієї проблеми у протоколі MQTT використовується посередницька архітектура, у якій центральним елементом виступає сервер-посередник, що називається брокером. Усі клієнти підключаються до брокера, а обмін повідомленнями між ними відбувається саме через цей сервер. Такий підхід дозволяє значно спростити структуру взаємодії між пристроями та підвищує ефективність функціонування системи [8].

Архітектура MQTT складається з трьох основних компонентів: клієнтів, брокера та топіків (каналів повідомлень). Кожен із цих елементів виконує певну функцію в процесі обміну інформацією між пристроями.

У системі MQTT клієнтом називають будь-який пристрій або програму, які можуть підключатися до брокера та виконувати обмін повідомленнями. Клієнтами можуть бути датчики, контролери, сервери обробки даних або мобільні застосунки. Основна функція клієнтів полягає в обміні повідомленнями через брокера. Вони можуть як надсилати інформацію, так і отримувати її від інших пристроїв. У більшості випадків один клієнт одночасно виконує обидві функції

Брокер є центральним елементом архітектури MQTT. Він відповідає за приймання повідомлень від клієнтів, їх обробку та подальше розповсюдження серед інших клієнтів, які підписані на відповідні канали. Крім того, брокер керує підключеннями клієнтів, контролює стан мережі та забезпечує коректну доставку повідомлень у системі [9].

Ще одним важливим елементом архітектури MQTT є топіки. Топік – це логічний канал, через який передаються повідомлення між клієнтами. Коли клієнт надсилає повідомлення брокеру, він вказує назву топіка, до якого належить це повідомлення. Інші клієнти можуть підписатися на цей топік, щоб отримувати відповідні повідомлення.

Назви топіків зазвичай мають ієрархічну структуру. Наприклад, у системі розумного будинку можуть використовуватися такі топіки:

home/livingroom/temperature;

home/livingroom/light;

home/kitchen/humidity;

Завдяки такій структурі можна легко організувати обмін даними між різними групами пристроїв у системі.

У протоколі MQTT використовується модель publish-subscribe для обміну повідомленнями між клієнтами. У цій моделі одні пристрої передають інформацію, а інші її отримують. Клієнт, який надсилає повідомлення, називається publisher. Клієнт, який отримує повідомлення після підписки на топик, називається subscriber. Передача даних між ними відбувається через брокер. Саме брокер приймає повідомлення та надсилає їх усім підписаним клієнтам. Такий підхід спрощує взаємодію між пристроями в системі.

Процес обміну повідомленнями у такій системі відбувається наступним чином. Спочатку клієнт підключається до брокера та підписується на певний топик. Після цього інший клієнт може опублікувати повідомлення у цей топик. Брокер отримує повідомлення та автоматично передає його всім клієнтам, які підписані на відповідний канал. При цьому відправник повідомлення не знає, які саме пристрої отримають інформацію.

Такий механізм забезпечує слабке зв'язування компонентів системи, що є важливою перевагою для IoT-середовища. Пристрої можуть працювати незалежно один від одного, що спрощує розширення системи та підключення нових пристроїв.

Важливою перевагою архітектури MQTT є її висока масштабованість. Використання брокера дозволяє системі одночасно працювати з великою кількістю клієнтів. Завдяки цьому MQTT можна застосовувати у великих мережах Інтернету речей. Протокол забезпечує стабільний обмін повідомленнями між багатьма пристроями та програмними компонентами. Це робить MQTT ефективним рішенням для розподілених IoT-систем.

Перед початком обміну повідомленнями клієнт встановлює з'єднання з брокером та ініціалізує сеанс взаємодії. Після встановлення з'єднання клієнт може підписуватися на топики та передавати повідомлення іншим учасникам системи через брокер.

Таким чином, архітектура MQTT базується на використанні брокера як центрального елемента системи та моделі publish-subscribe для організації обміну

повідомленнями між клієнтами. Такий підхід забезпечує гнучку, масштабовану та ефективну взаємодію між великою кількістю пристроїв у мережах Інтернету речей.

2.3. Використання MQTT у сучасних IoT-системах

Інтернет речей охоплює велику кількість пристроїв, які взаємодіють між собою через мережу для збору, передавання та обробки даних. Такі пристрої можуть працювати автономно та виконувати різні функції, зокрема вимірювання параметрів середовища, керування обладнанням або передачу інформації до централізованих систем. У таких системах важливим є забезпечення ефективного та надійного обміну інформацією між датчиками, контролерами, серверами обробки даних і користувацькими застосунками.

Одним із найбільш поширених протоколів, що використовується для організації такої взаємодії, є MQTT. Його особливістю є легкість, мінімальні вимоги до ресурсів та можливість ефективної роботи навіть у нестабільних або обмежених мережах. Завдяки цьому MQTT широко застосовується в IoT-системах різного масштабу – від невеликих локальних рішень до великих розподілених платформ. Використання цього протоколу дозволяє будувати гнучкі та масштабовані системи, які легко адаптуються до збільшення кількості пристроїв [10].

Серед найпоширеніших напрямів використання MQTT виділяють системи розумного будинку. У таких системах різні пристрої підключаються до мережі та постійно обмінюються даними. Це можуть бути датчики температури, вологості, руху або освітлення, а також виконавчі пристрої, такі як розумні вимикачі, термостати або системи опалення. Наприклад, датчик температури передає дані до системи керування, яка аналізує отриману інформацію та автоматично регулює мікроклімат у приміщенні. Уся взаємодія між пристроями відбувається через MQTT-брокер, що значно спрощує архітектуру системи та зменшує складність її налаштування.

MQTT також широко використовується у промислових IoT-системах. У таких системах датчики та контролери контролюють роботу обладнання та виробничі процеси. Зібрані дані передаються до серверів або хмарних платформ для подальшого аналізу. Це дозволяє швидко реагувати на зміни в роботі системи. Використання MQTT допомагає зменшити навантаження на мережу та забезпечує стабільну передачу даних між пристроями [11].

Окрім цього, протокол активно застосовується у хмарних IoT-платформах. Пристрої передають дані до хмарних сервісів, де вони зберігаються, обробляються та використовуються для подальшої аналітики. Завдяки моделі publish-subscribe різні компоненти системи можуть взаємодіяти між собою без прямого зв'язку. Одні пристрої публікують дані у певні топіки, а інші отримують їх шляхом підписки на ці канали. Такий підхід забезпечує гнучкість архітектури та спрощує масштабування системи при збільшенні кількості пристроїв.

Ще однією важливою перевагою MQTT є його ефективність у мережах із обмеженою пропускнуою здатністю. Багато IoT-пристроїв працюють у бездротових або мобільних мережах, де швидкість передачі даних може бути обмеженою. Протокол використовує компактні повідомлення, що зменшує навантаження на мережу та підвищує стабільність обміну інформацією.

Важливою перевагою також є проста інтеграція MQTT з різними програмними платформами. Для нього існує велика кількість бібліотек, які підтримують різні мови програмування та операційні системи. Це дозволяє використовувати протокол у веб-застосунках, мобільних додатках і хмарних системах, створюючи комплексні IoT-рішення.

У навчальних та дослідницьких проєктах MQTT також активно використовується разом із середовищами моделювання мереж. Такі інструменти дозволяють створювати віртуальні IoT-системи та досліджувати їх роботу без використання реального обладнання.

У середовищах моделювання можна відтворювати роботу датчиків, контролерів і серверів, які обмінюються даними через MQTT. Це дозволяє

аналізувати процес передачі повідомлень і взаємодію компонентів системи в різних сценаріях.

Такий підхід значно спрощує навчання та дослідження IoT-технологій. Він допомагає краще зрозуміти принципи побудови мереж, роботу брокера та логіку обміну повідомленнями між пристроями.

Завдяки своїй простоті, ефективності та гнучкості MQTT став одним із ключових протоколів у сфері Інтернету речей. Його використовують у різних галузях, зокрема в розумних будинках, промислових системах, міській інфраструктурі та хмарних IoT-платформах.

РОЗДІЛ 3

ОГЛЯД CISCO PACKET TRACER ЯК СЕРЕДОВИЩА МОДЕЛЮВАННЯ MQTT-ВЗАЄМОДІЇ ІОТ-ПРИСТРОЇВ

3.1. Призначення та функціональні можливості Cisco Packet Tracer

У процесі дослідження та проєктування мереж Інтернету речей важливу роль відіграють засоби моделювання, які дозволяють відтворити роботу системи без необхідності використання реального обладнання. Одним із найбільш поширених програмних середовищ для моделювання комп'ютерних мереж та IoT-систем є Cisco Packet Tracer. Це програмний інструмент, розроблений компанією Cisco в межах освітньої ініціативи Cisco Networking Academy, який використовується для навчання, дослідження та експериментального проєктування мережевих рішень [12].

Основне призначення Cisco Packet Tracer полягає в моделюванні роботи комп'ютерних мереж різного рівня складності. Це можуть бути локальні, глобальні або бездротові мережі. Середовище дозволяє створювати мережеві топології та налаштовувати обладнання. Також у ньому можна аналізувати передачу пакетів і перевіряти роботу мережевих протоколів. Важливою перевагою є візуалізація процесів передачі даних. Це допомагає краще зрозуміти принципи роботи мережевих систем.

Інтерфейс Cisco Packet Tracer складається з робочої області, панелі пристроїв, панелі з'єднань та інструментів керування симуляцією. Робоча область використовується для побудови мережевої топології шляхом розміщення пристроїв і встановлення між ними зв'язків (рис. 3.1.). Панель пристроїв містить різні типи обладнання, до прикладу, маршрутизатори, комутатори, сервери, персональні комп'ютери та IoT-пристрої. Панель з'єднань дозволяє обирати тип кабелю або бездротового з'єднання для встановлення зв'язку між елементами мережі.

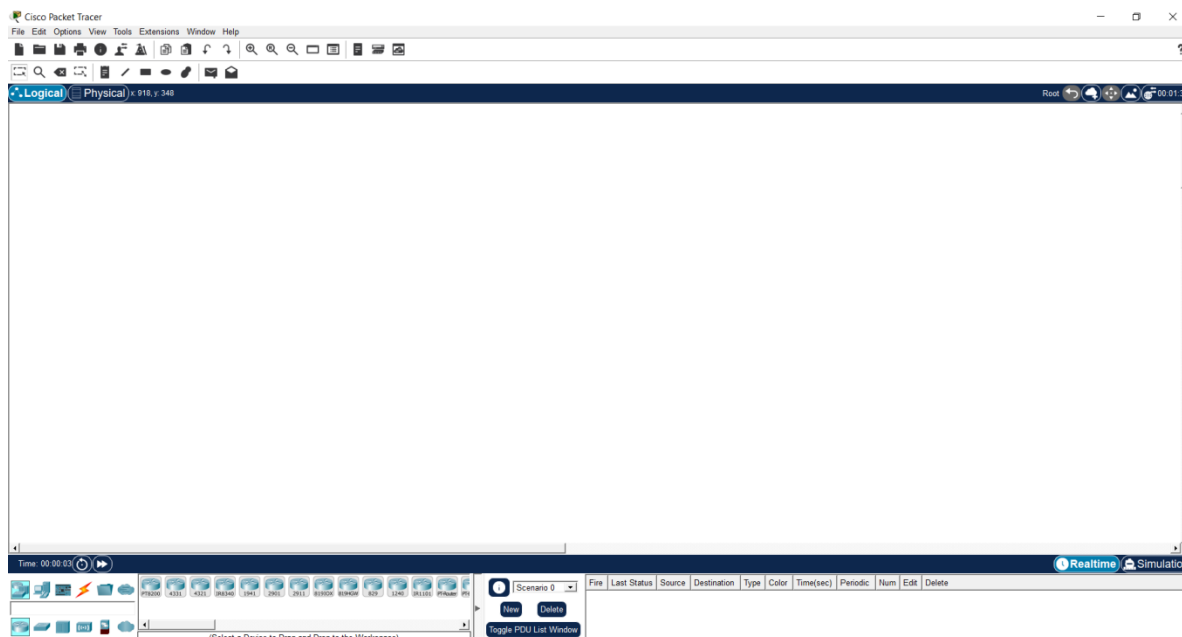


Рисунок 3.1. Робоча область Cisco Packet Tracer

Cisco Packet Tracer підтримує два основні режими роботи: Real-Time та Simulation. Це одна з його ключових можливостей. Режим реального часу забезпечує роботу мережі без затримок. Він використовується для швидкої перевірки налаштувань і працездатності конфігурації. Режим симуляції дозволяє глибше аналізувати роботу мережі. Користувач може спостерігати за передачею пакетів і відстежувати їх рух між пристроями. Також є можливість досліджувати роботу протоколів на різних рівнях моделі OSI

Програмне середовище також підтримує налаштування мережевих параметрів пристроїв, включаючи IP-адресацію, маршрутизацію, налаштування протоколів передачі даних та параметрів безпеки. Це дозволяє моделювати як прості мережі, так і складні багаторівневі системи з великою кількістю вузлів. Крім того, Packet Tracer надає можливість використовувати вбудовані інструменти для генерації трафіку, що дає змогу досліджувати поведінку мережі під навантаженням [13].

Важливою перевагою Cisco Packet Tracer є його орієнтованість на моделювання сучасних технологій, зокрема Інтернету речей. У середовищі реалізована підтримка різних типів IoT-пристроїв, таких як сенсори, виконавчі механізми, розумні пристрої та шлюзи. Це дозволяє створювати моделі реальних IoT-систем, наприклад «розумного будинку» або системи моніторингу

навколишнього середовища, без необхідності використання фізичних пристроїв [10].

Ще однією важливою функціональною можливістю є підтримка сценарного програмування та автоматизації. Користувач може налаштовувати поведінку IoT-пристроїв, задавати умови їхньої роботи та моделювати взаємодію між ними. Це особливо важливо при дослідженні протоколів обміну повідомленнями, таких як MQTT, оскільки дозволяє відтворити реальні сценарії передачі даних між пристроями.

Окремо варто відзначити можливість навчального використання середовища. Cisco Packet Tracer дозволяє поступово вивчати принципи побудови мереж від простих топологій до складних розподілених систем. Завдяки цьому користувач може поетапно розвивати навички налаштування мережевого обладнання та аналізу трафіку.

Крім того, середовище підтримує інтерактивну взаємодію між пристроями, що дозволяє моделювати поведінку системи в реальному часі. Це дає змогу перевіряти різні сценарії роботи мережі та оцінювати їх ефективність ще на етапі проєктування.

Таким чином, Cisco Packet Tracer є потужним інструментом для моделювання мережевих та IoT-систем, який поєднує простоту використання з широкими функціональними можливостями. Його використання дозволяє ефективно досліджувати принципи побудови мереж, аналізувати роботу протоколів передачі даних та моделювати взаємодію IoT-пристроїв у різних умовах. Це робить його доцільним вибором для виконання практичної частини даної кваліфікаційної роботи.

3.2. Підтримка IoT-технологій у Cisco Packet Tracer

Cisco Packet Tracer дозволяє моделювати роботу великої кількості розумних пристроїв у єдиній мережі, що стало необхідним через стрімкий розвиток технологій IoT. Завдяки цьому користувач отримує можливість створювати та

досліджувати IoT-мережі без використання реального обладнання, що є особливо важливим у навчальних і дослідницьких проєктах.

Підтримка IoT-технологій у Cisco Packet Tracer реалізована через набір вбудованих пристроїв, сенсорів, виконавчих механізмів та мережевих компонентів. У програмному середовищі доступні різні типи IoT-пристроїв, серед яких датчики температури, вологості, освітленості, руху, диму та рівня води. Крім сенсорів, Packet Tracer містить виконавчі пристрої, наприклад лампи, вентилятори, двері, сигналізації та системи кондиціонування (рис. 3.2.). Це дозволяє моделювати реальні сценарії автоматизації, характерні для систем «розумного будинку» або промислового моніторингу [11].



Рисунок 3.2. Панель IoT-пристроїв у Cisco Packet Tracer

У Cisco Packet Tracer IoT-пристрої можуть працювати через кабельне або бездротове підключення. Для створення бездротової мережі використовуються точки доступу, Wi-Fi маршрутизатори та шлюзи Home Gateway. Основну роль у такій системі виконує Home Gateway, який об'єднує всі пристрої в єдину мережу. Через нього сенсори та інші IoT-пристрої отримують доступ до локальної мережі й можуть взаємодіяти із серверними платформами. Шлюз також дозволяє контролювати роботу пристроїв, отримувати інформацію про їхній стан і забезпечує обмін даними між компонентами системи. Завдяки цьому підтримується стабільне функціонування IoT-мережі та передавання інформації між її елементами.

Особливістю Cisco Packet Tracer є підтримка різних способів підключення IoT-пристроїв. У межах моделювання можна використовувати Ethernet-з'єднання, Wi-Fi або бездротові IoT-протоколи. Це дозволяє досліджувати особливості роботи різних мережевих технологій та оцінювати їхню ефективність залежно від типу системи. Наприклад, для сенсорів із низьким енергоспоживанням доцільним є використання бездротових підключень, тоді як сервери або центральні вузли

системи можуть використовувати дротові канали зв'язку для забезпечення стабільності передачі даних [5].

Кожен IoT-пристрій у Packet Tracer має власні параметри конфігурації. Користувач може змінювати мережеві налаштування, задавати IP-адреси, налаштовувати бездротове підключення та параметри взаємодії з іншими пристроями. Крім того, середовище підтримує зміну станів сенсорів у реальному часі. Наприклад, можна вручну змінити температуру або рівень освітленості та спостерігати, як система реагує на ці зміни. моделювати поведінку мережі двома шляхами: пасивним, через коригування зовнішніх параметрів середовища (температури, освітленості), та активним, шляхом безпосередньої взаємодії з пристроями.

Важливою перевагою Packet Tracer є можливість створення комплексних IoT-сценаріїв. Пристрої можуть взаємодіяти між собою за певними правилами або умовами. Наприклад, при спрацюванні датчика руху система може автоматично вмикати освітлення. Аналогічно, датчик температури може запускати систему охолодження після перевищення встановленого значення (рис. 3.3.). Завдяки цьому середовище дозволяє моделювати не лише окремі пристрої, а й повноцінні системи автоматизованого управління [10].

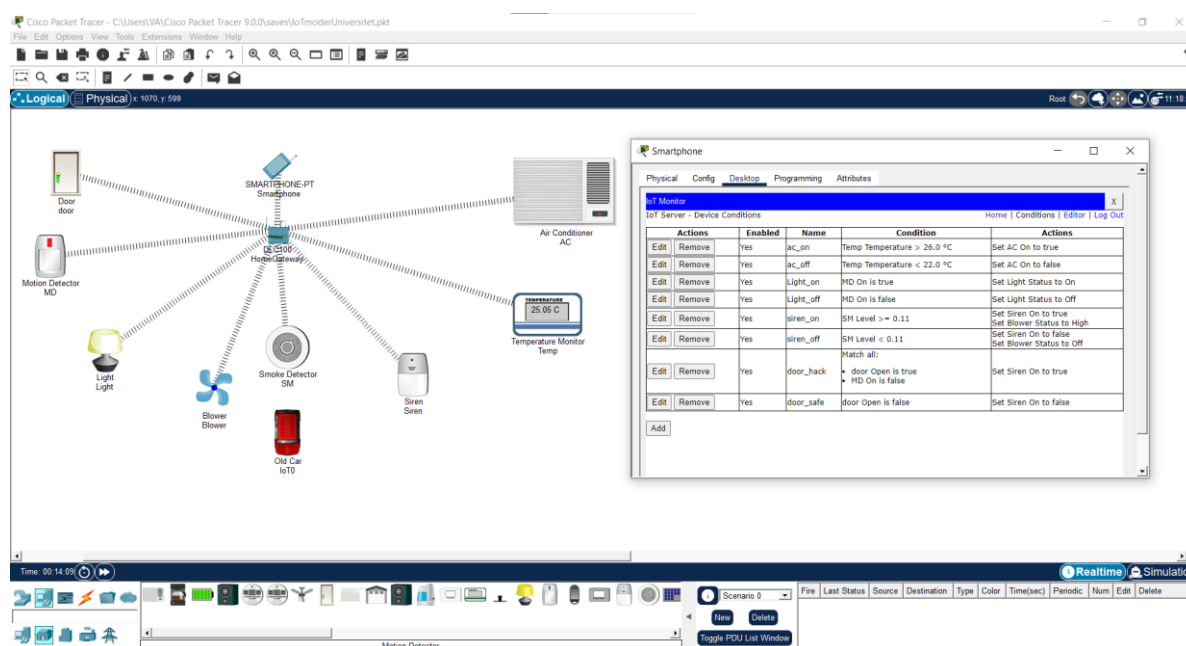


Рисунок 3.3. Приклад IoT-мережі в Cisco Packet Tracer

У цій програмі значна увага приділяється підтримці IoT-серверів і серверних сервісів. У середовищі доступний IoT Server, який використовується для реєстрації пристроїв, передавання повідомлень і централізованого керування мережею. За допомогою сервера можна створювати облікові записи користувачів, відстежувати стан підключених пристроїв і налаштовувати взаємодію між елементами системи. Такий підхід робить модель більш наближеною до реальних IoT-платформ, що застосовуються у сучасних практичних рішеннях

Таким чином, Cisco Packet Tracer забезпечує широкі можливості для моделювання IoT-технологій та взаємодії розумних пристроїв. Підтримка сенсорів, виконавчих механізмів, шлюзів і серверних сервісів дозволяє створювати комплексні IoT-системи та досліджувати їхню роботу в умовах, наближених до реальних. Це робить Packet Tracer ефективним інструментом для аналізу мережевої взаємодії та подальшого моделювання MQTT-обміну повідомленнями між IoT-пристроями [14].

3.3. Інструменти моделювання IoT-мереж та MQTT-взаємодії

Однією з найважливіших переваг середовища моделювання Cisco Packet Tracer є підтримка технологій Інтернету речей (IoT), що дозволяє створювати не лише класичні комп'ютерні мережі, а й складні системи взаємодії між інтелектуальними пристроями. Завдяки такому функціоналу користувач отримує можливість моделювати реальні сценарії автоматизації, аналізувати роботу сенсорних систем та досліджувати принципи обміну повідомленнями між пристроями у цифровому середовищі. Це особливо важливо в умовах стрімкого розвитку IoT-технологій, які активно впроваджуються у промисловість, транспорт, медицину, енергетику та побутові системи.

Середовище Cisco Packet Tracer забезпечує комплексний підхід до створення IoT-інфраструктури, поєднуючи мережеві компоненти, серверні служби та інтелектуальні пристрої в єдину систему. Користувач може будувати топології різної складності, використовуючи маршрутизатори, комутатори, сервери, точки

бездротового доступу, Home Gateway та численні IoT-пристрої. До таких пристроїв належать датчики температури, вологості, освітлення, руху, диму, а також виконавчі механізми – лампи, вентилятори, системи кондиціонування, електронні замки та інші елементи автоматизації. Наявність великої кількості готових компонентів дозволяє моделювати як невеликі побутові системи, так і масштабні промислові IoT-мережі.

Важливою особливістю Packet Tracer є можливість використання різних способів підключення пристроїв. Для передавання даних можуть застосовуватися як дротові Ethernet-з'єднання, так і бездротові технології, зокрема Wi-Fi. Це дає змогу створювати моделі «розумного будинку», систем екологічного моніторингу, автоматизованих складів або виробничих комплексів. Завдяки підтримці бездротової взаємодії можна досліджувати стабільність мережевого з'єднання, радіус дії сигналу, затримки передавання повідомлень та вплив навантаження на роботу IoT-системи.

Під час створення IoT-мережі значну роль відіграє етап логічного проектування. На цьому етапі визначаються функції окремих пристроїв, структура взаємодії між ними та принципи обробки інформації. Для сучасних IoT-систем характерною є інтеграція великої кількості сенсорів, які постійно генерують дані, тому ефективна організація обміну повідомленнями має критичне значення. У Packet Tracer користувач може реалізувати як централізовану модель керування, де всі дані проходять через сервер або брокер повідомлень, так і децентралізовані схеми взаємодії. Такий підхід дозволяє оцінити переваги та недоліки різних архітектур з точки зору масштабованості, швидкодії та надійності системи.

Одним із ключових інструментів аналізу мережевих процесів у Cisco Packet Tracer є режим Simulation. Його використання дає можливість у реальному часі спостерігати за процесом передавання пакетів між пристроями, аналізувати маршрути руху даних та відстежувати роботу мережевих протоколів. Для IoT-систем це особливо важливо, оскільки дозволяє детально дослідити механізми взаємодії між сенсорами, сервером та кінцевими пристроями. Користувач може бачити, як формується повідомлення, через які вузли воно проходить і яким чином

реагує система після отримання даних. Такий рівень візуалізації значно спрощує розуміння принципів функціонування мережі та допомагає виявляти помилки конфігурації.

Особливе місце в моделюванні IoT-систем займає протокол MQTT, який реалізує модель взаємодії publish/subscribe. У Cisco Packet Tracer функції MQTT-брокера виконують IoT Server або Registration Server. Саме через них здійснюється обмін повідомленнями між IoT-пристроями. Сенсори публікують дані у відповідні теми (topics), а інші пристрої або сервіси, які підписані на ці теми, отримують повідомлення автоматично. Така архітектура є ефективною для IoT-систем, оскільки забезпечує низьке навантаження на мережу та дозволяє підтримувати обмін даними навіть за нестабільного з'єднання. Переваги використання MQTT у порівнянні з традиційними підходами до передачі даних розглядаються у матеріалі [15].

Практичне моделювання MQTT-взаємодії в Packet Tracer дозволяє створювати сценарії автоматизації різного рівня складності. Наприклад, датчик температури може постійно надсилати інформацію на MQTT-брокер, а система кондиціонування – аналізувати отримані значення та автоматично змінювати режим роботи. Аналогічно можна реалізувати сценарії автоматичного освітлення, контролю вологості, сигналізації або систем безпеки. Подібні моделі демонструють, як IoT-технології забезпечують автономну взаємодію між пристроями без прямого втручання користувача.

Додатковою перевагою Cisco Packet Tracer є підтримка програмування поведінки IoT-пристроїв. Користувач може застосовувати вбудовані сценарії або створювати власну логіку роботи за допомогою JavaScript та Python-подібних скриптів. Це суттєво розширює можливості моделювання, оскільки дозволяє реалізовувати складні алгоритми автоматизації, умови реагування на події та взаємодію між декількома сенсорами одночасно. Наприклад, можна створити систему, у якій датчик руху активує освітлення лише за умови низького рівня освітленості або автоматично надсилає повідомлення про тривогу на сервер моніторингу.

Не менш важливим аспектом є дослідження роботи бездротових IoT-мереж. У Packet Tracer IoT-пристрої можуть взаємодіяти через Home Gateway або точки бездротового доступу, що дозволяє моделювати реальні умови експлуатації систем. Під час експериментів користувач може аналізувати вплив відстані між пристроями, навантаження на мережу та кількості підключених сенсорів на швидкість і стабільність обміну повідомленнями. Такі дослідження допомагають оцінити ефективність мережевої архітектури та виявити потенційні проблеми ще на етапі проєктування системи [16].

Окремої уваги заслуговують можливості Packet Tracer у сфері мережевої безпеки. Сучасні IoT-системи часто працюють із критично важливими даними, тому питання захисту мережі є надзвичайно актуальним. У середовищі підтримуються базові механізми автентифікації, контроль доступу до серверів та перевірка підключення пристроїв до мережі. Це дозволяє досліджувати, як різні параметри конфігурації впливають на рівень безпеки та стабільність роботи системи. Аналіз таких аспектів є важливим для розуміння практичних викликів, які виникають під час впровадження IoT-рішень у реальних умовах.

Загалом Cisco Packet Tracer виступає ефективним середовищем для вивчення принципів функціонування IoT-мереж і протоколу MQTT. Поєднання мережевих інструментів, режиму симуляції, підтримки серверів, сенсорів та виконавчих механізмів дозволяє створювати комплексні сценарії автоматизації та аналізувати їхню роботу в умовах, наближених до реальних. Завдяки цьому Packet Tracer є корисним засобом як для навчання, так і для практичного дослідження сучасних IoT-технологій [15].

РОЗДІЛ 4

ПРОЄКТУВАННЯ, РЕАЛІЗАЦІЯ ТА СИМУЛЯЦІЯ MQTT-ВЗАЄМОДІЇ В CISCO PACKET TRACER

4.1. Проєктування моделі IoT-мережі

Практична частина кваліфікаційної роботи полягала у створенні моделі IoT-мережі в середовищі Cisco Packet Tracer для дослідження взаємодії пристроїв із використанням протоколу MQTT. Основною метою проєктування було створення мережі типу «розумний будинок», у якій пристрої здійснюють автоматичний обмін повідомленнями через MQTT-брокер [10, 12].

На початковому етапі проєктування було визначено структуру мережі, перелік необхідних пристроїв та принцип їх взаємодії. Для реалізації моделі на робочу область Cisco Packet Tracer було додано мережеві та IoT-компоненти, необхідні для побудови локальної системи автоматизації. До складу створеної мережі увійшли Home Gateway, ноутбук для керування системою, SBC-контролер, LCD-монітор, температурний датчик, система охолодження, нагрівальний елемент, а також допоміжні пристрої для тестування та перевірки роботи мережі (рис. 4.1.).

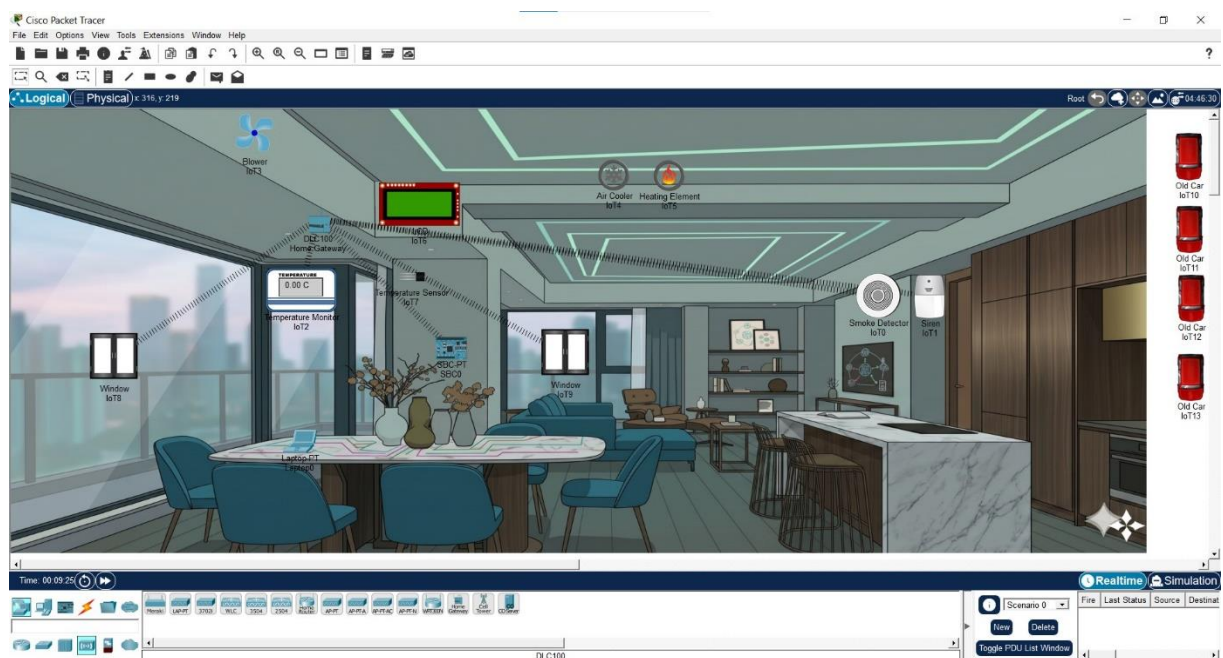


Рисунок 4.1. Загальна структура IoT-мережі в Cisco Packet Tracer

Центральним елементом створеної мережі став Home Gateway, який виконував функції бездротової точки доступу та локального IoT-сервера. Саме через нього здійснювалося підключення всіх бездротових пристроїв до локальної мережі, а також забезпечувалася їх взаємодія в межах єдиної інфраструктури. Використання Home Gateway дозволило спростити процес реєстрації IoT-пристроїв та централізувати керування ними.

У процесі налаштування Home Gateway було змінено стандартні параметри пристрою (рис. 4.2.). Зокрема, встановлено власну назву бездротової мережі (SSID), задано пароль доступу та активовано параметри захисту мережі. Це дозволило ідентифікувати створену мережу серед інших доступних бездротових підключень і забезпечити контроль доступу до неї [12]. Крім того, зміна стандартних параметрів дозволила уникнути конфліктів із типовими конфігураціями та забезпечила більш реалістичні умови моделювання IoT-середовища.

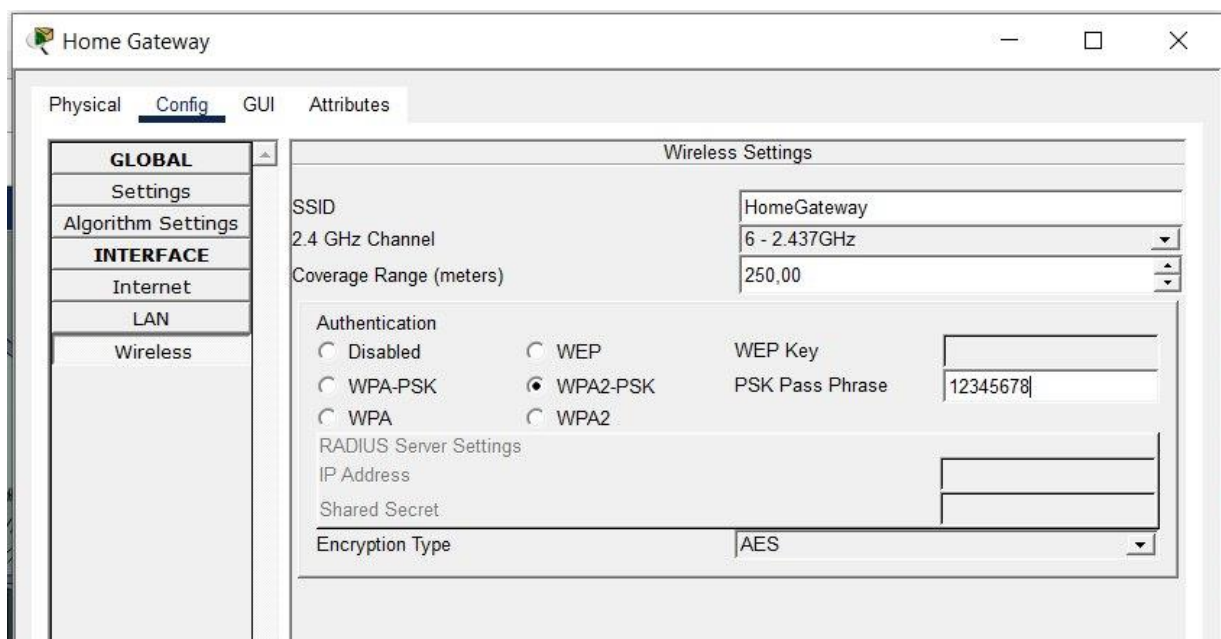


Рисунок 4.2. Налаштування параметрів Home Gateway

Для забезпечення підключення ноутбука до бездротової мережі було змінено його апаратну конфігурацію шляхом заміни стандартного мережевого компонента на бездротовий адаптер (рис. 4.3.). Така модифікація дала змогу використовувати Wi-Fi-з'єднання для взаємодії з Home Gateway та подальшого налаштування MQTT-брокера.

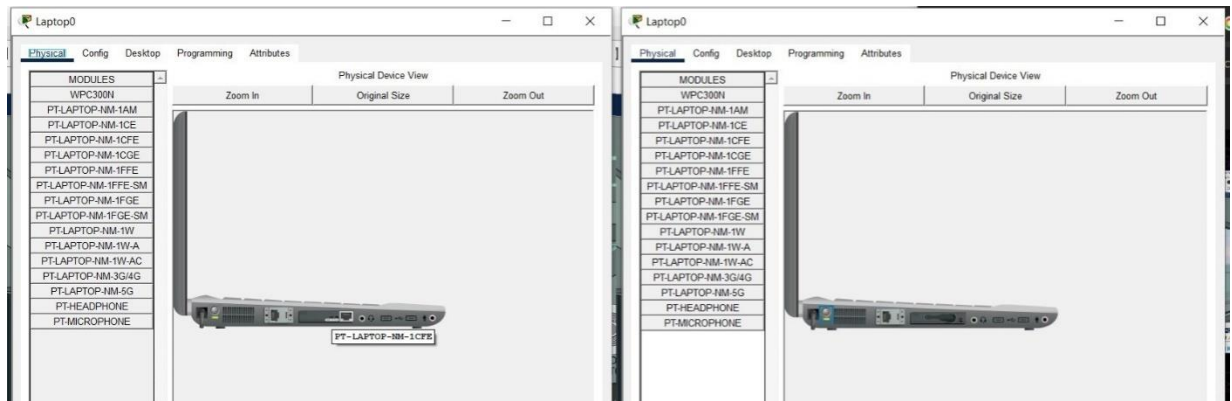


Рисунок 4.3. Зміна фізичних параметрів Laptop

Після встановлення бездротового адаптера ноутбук було підключено до створеної мережі із використанням заданих параметрів автентифікації. Підключення виконувалося шляхом вибору необхідної бездротової мережі та введення пароля доступу. Після успішної авторизації ноутбук отримав IP-адресу від Home Gateway і став повноцінним вузлом локальної мережі (рис. 4.4.).

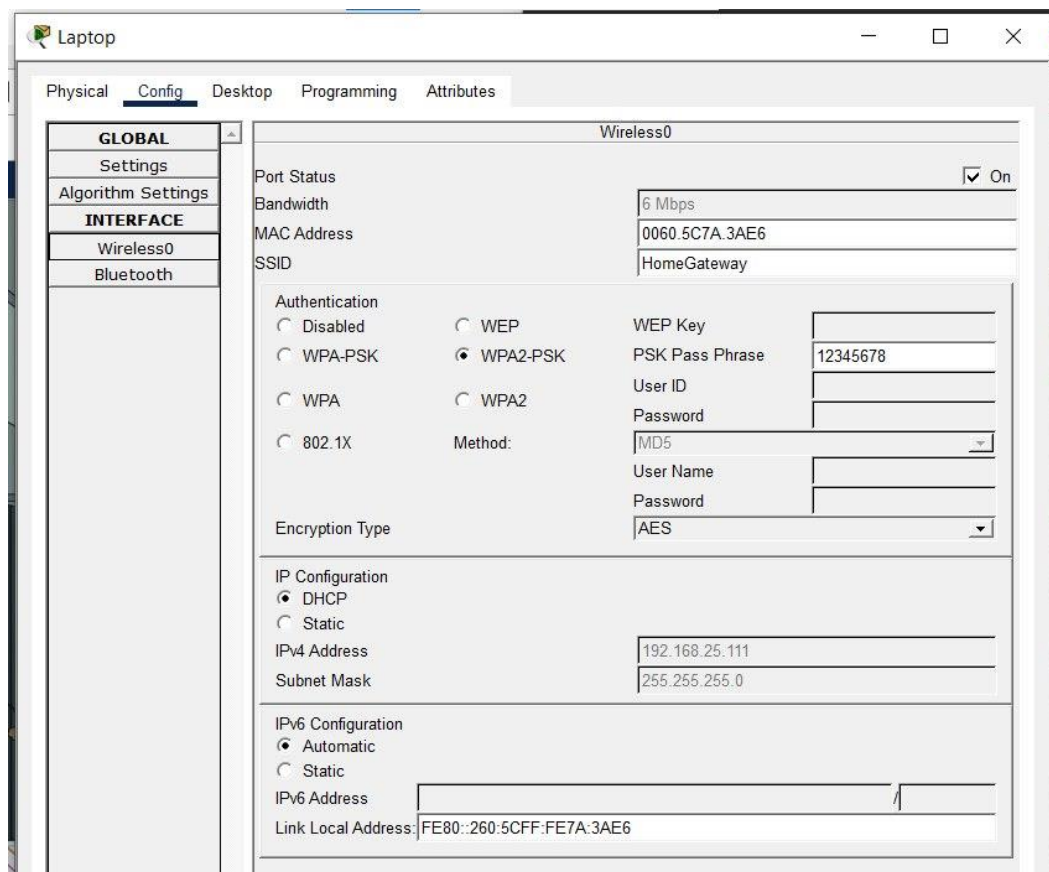


Рисунок 4.4. Підключення ноутбука до бездротової мережі

На наступному етапі в налаштуваннях Home Gateway було активовано IoT Server, який забезпечував реєстрацію та підключення IoT-пристроїв у межах

локальної мережі. Завдяки цьому пристрої отримали можливість автоматично виявлятися системою та взаємодіяти між собою через мережеву інфраструктуру. Для організації MQTT-взаємодії окремо використовувався ноутбук, на якому надалі було налаштовано MQTT-брокер [7].

Після налаштування центрального вузла до мережі було підключено інші IoT-пристрої. Кожен із них проходив процедуру реєстрації через Home Gateway, після чого ставав доступним у системі керування. Такий підхід дозволив реалізувати централізовану модель підключення та забезпечив стабільний обмін даними між усіма компонентами мережі.

Окрему частину моделі становив SBC-контролер, до якого через спеціальні дротові з'єднання було підключено LCD-монітор, температурний сенсор, систему охолодження та нагрівальний елемент. Використання SBC-контролера дозволило реалізувати локальну логіку обробки даних, а також виконання автоматизованого керування виконавчими пристроями залежно від показників температури.

Підключення датчиків і виконавчих елементів через GPIO-інтерфейси дозволило максимально наблизити модель до принципів роботи реальних IoT-систем. Температурний датчик передавав поточні значення температури на контролер, який аналізував отримані дані та формував відповідні команди для систем охолодження або нагрівання.

Отже, на етапі проєктування було сформовано структуру IoT-мережі, яка забезпечує можливість подальшого налаштування MQTT-взаємодії між сенсорами, контролером та виконавчими пристроями. Створена модель стала основою для реалізації програмної логіки, проведення симуляції та дослідження процесу обміну повідомленнями між компонентами IoT-системи.

4.2. Налаштування MQTT-брокера та IoT-пристроїв

Після завершення проєктування мережевої структури було виконано налаштування MQTT-брокера та конфігурацію IoT-пристроїв для забезпечення коректної взаємодії між усіма компонентами системи.

Для організації обміну повідомленнями MQTT-брокер було налаштовано на ноутбучі, який був підключений до локальної бездротової мережі через Home Gateway. У параметрах ноутбука було активовано службу MQTT Broker та створено обліковий запис користувача із логіном і паролем (рис. 4.5.) для автентифікації клієнтів [7].

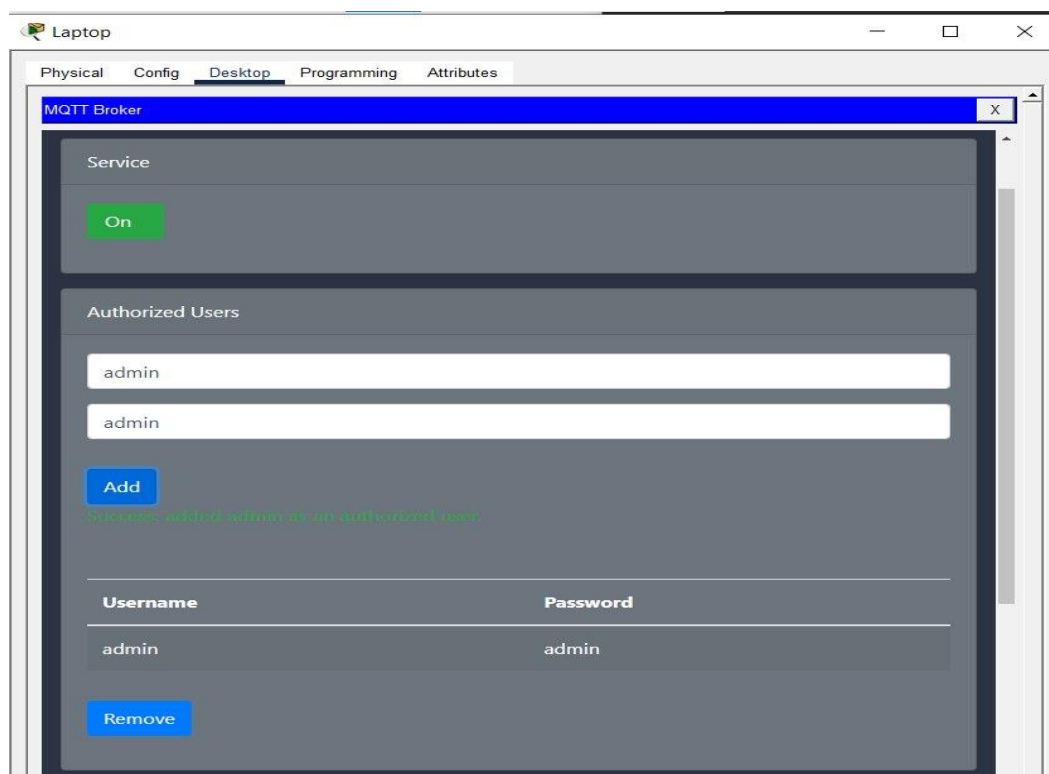


Рисунок 4.5. Налаштування MQTT-брокера на ноутбучі

У процесі конфігурації брокера було перевірено коректність роботи мережевих параметрів, а також доступність пристрою в локальній мережі. Додатково було визначено параметри підключення клієнтів, що забезпечило контроль доступу до системи та підвищило надійність обміну повідомленнями між компонентами мережі.

Home Gateway у створеній системі виконував функції бездротової точки доступу та IoT Server. Через нього здійснювалося підключення IoT-пристроїв до локальної мережі, їх реєстрація в системі та подальший обмін даними. Завдяки використанню IoT Server усі пристрої стали доступними для моніторингу та автоматизації через IoT Monitor.

Після налаштування MQTT-брокера було визначено IP-адресу ноутбука, яка використовувалася SBC-контролером для підключення до брокера повідомлень.

Вказана IP-адреса стала основним параметром для встановлення MQTT-з'єднання між контролером і сервером повідомлень.

Наступним етапом стало налаштування SBC-контролера. До нього було підключено LCD-дисплей, температурний сенсор, систему охолодження та нагрівальний елемент через спеціальні дротові інтерфейси Cisco Packet Tracer (рис. 4.6.) [13].

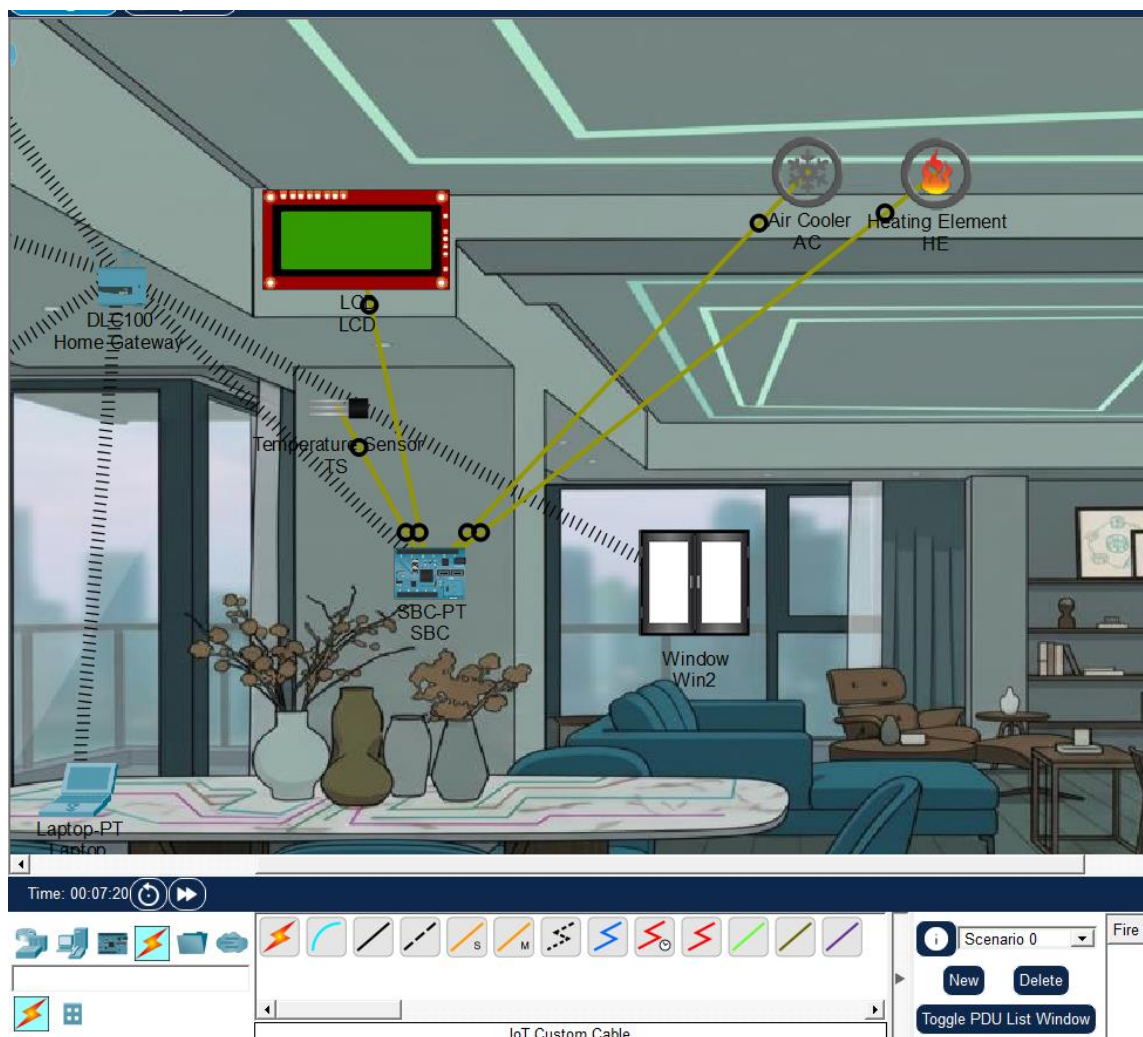


Рисунок 4.6. Підключення компонентів до SBC-контролера

Після фізичного підключення компонентів було виконано налаштування їх програмної взаємодії. Для реалізації MQTT-взаємодії на SBC було встановлено MQTT Client. У його параметрах було вказано IP-адресу брокера, логін та пароль користувача, створеного на Laptop (рис. 4.7.). Після введення параметрів було виконано підключення контролера до MQTT-брокера, у результаті чого SBC отримав можливість передавати та приймати MQTT-повідомлення.

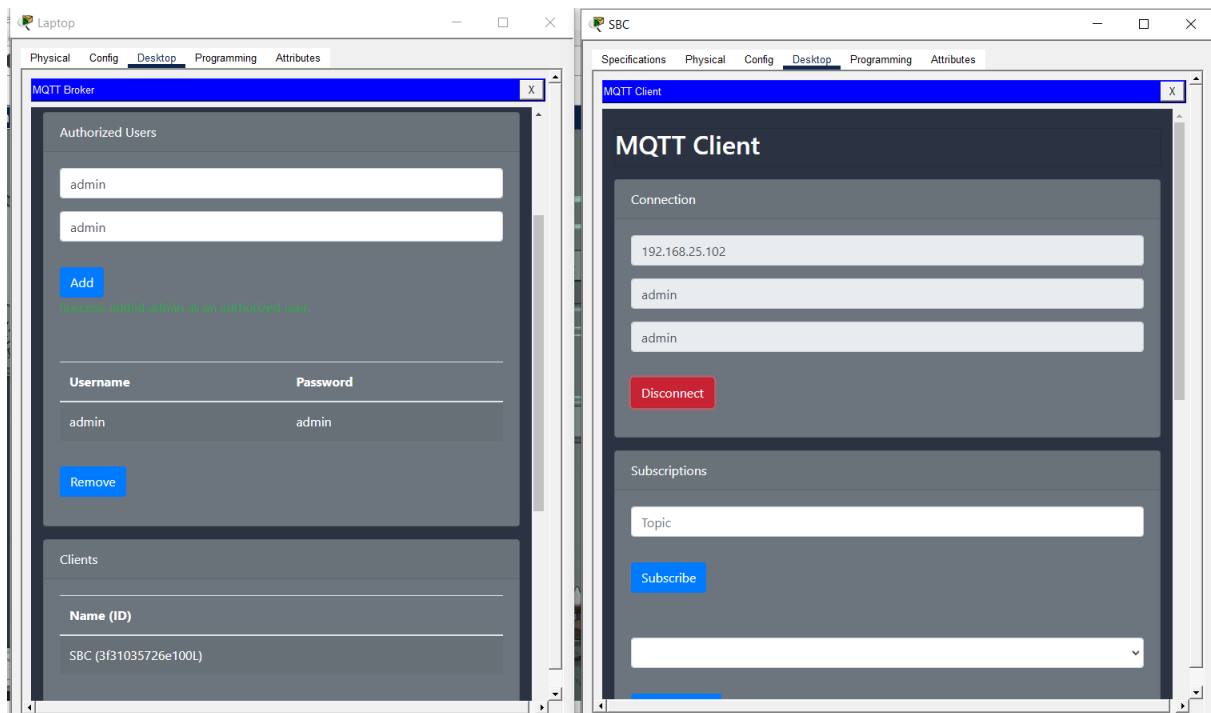


Рисунок 4.7. Налаштування MQTT-клієнта на SBC

Після успішного встановлення MQTT-клієнта на SBC автоматично завантажився базовий програмний шаблон для роботи з MQTT-протоколом. Надалі цей шаблон було доповнено необхідною логікою для реалізації системи автоматичного керування температурою.

У процесі налаштування системи було визначено GPIO-порти для підключення температурного датчика, системи охолодження, нагрівання та LCD-дисплея. Також було задано MQTT-теми для передавання показників температури та стану кондиціонування, налаштовано параметри підключення до брокера повідомлень і реалізовано алгоритм автоматичного керування виконавчими пристроями залежно від значення температури.

Фрагмент реалізованого програмного коду, наведено нижче:

```
# Номери GPIO-портів SBC-контролера
PIN_TEMP_SENSOR = 0
PIN_HEATING = 1
PIN_COOLING = 2
PIN_LCD = 3

# MQTT-топіки для обміну повідомленнями
TOPIC_TEMP = "home/temperature"
TOPIC_AC_STATUS = "home/ac_status"

# Параметри підключення до MQTT-брокера
broker_ip = "192.168.25.108"
username = "admin"
```

```

password = "admin"

# Підключення до MQTT-брокера
mqttclient.connect(broker_ip, username, password)

# Налаштування режимів роботи GPIO-портів
pinMode(PIN_TEMP_SENSOR, IN)
pinMode(PIN_HEATING, OUT)
pinMode(PIN_COOLING, OUT)
pinMode(PIN_LCD, OUT)

# Основний цикл роботи системи
while True:
# Зчитування температури та передавання даних через MQTT
    temp = digitalRead(PIN_TEMP_SENSOR)
    payload = json.dumps({ "temp": temp })
    mqttclient.publish(TOPIC_TEMP, payload, "0")
    print("Published Temperature:", temp)

# Увімкнення охолодження при високій температурі
if temp > 650:
    digitalWrite(PIN_COOLING, HIGH)
    digitalWrite(PIN_HEATING, LOW)
    customWrite(PIN_LCD, "AC-ON")
    mqttclient.publish(TOPIC_AC_STATUS, "on", "0")
# Увімкнення нагрівання при низькій температурі
elif temp < 510:
    digitalWrite(PIN_HEATING, HIGH)
    digitalWrite(PIN_COOLING, LOW)
    customWrite(PIN_LCD, "AC-OFF")
    mqttclient.publish(TOPIC_AC_STATUS, "off", "0")

```

Наведений алгоритм забезпечував циклічне зчитування температури, передавання показників через MQTT-брокер та автоматичне керування виконавчими пристроями. При перевищенні встановленого порогу активувалася система охолодження, а при зниженні температури – нагрівальний елемент. Одночасно інформація про поточний стан системи виводилася на LCD-дисплей.

Окремо було налаштовано інші IoT-пристрої через Home Gateway для їх коректної реєстрації в мережі. Це дозволило забезпечити їх відображення в системі керування та доступність для автоматизації через IoT Monitor.

Після завершення базових налаштувань через додаток IoT Monitor на ноутбучі було створено сценарії автоматичної взаємодії між пристроями. Було визначено умови автоматичного увімкнення системи охолодження при перевищенні температурного порогу та активації нагрівального елемента при зниженні температури нижче встановленого значення. Використання таких

сценаріїв дозволило реалізувати базові механізми автоматизації, характерні для систем «розумного будинку».

Таким чином, виконане налаштування MQTT-брокера та IoT-пристроїв забезпечило готовність системи до повноцінної симуляції обміну повідомленнями, а також підтвердило можливість реалізації автоматизованої IoT-системи в середовищі Cisco Packet Tracer.

4.3. Реалізація обміну повідомленнями та аналіз результатів симуляції

Після завершення налаштування всіх компонентів було проведено симуляцію роботи створеної IoT-мережі для перевірки коректності MQTT-взаємодії між пристроями та оцінки стабільності обміну повідомленнями в межах локальної мережі.

Під час роботи системи температурний датчик безперервно передавав поточні значення на SBC-контролер. Контролер обробляв отримані дані відповідно до реалізованого програмного алгоритму та передавав повідомлення до MQTT-брокера, налаштованого на ноутбучі. Передавання даних здійснювалося циклічно через визначені MQTT-теми, що дозволяло організувати структурований обмін повідомленнями між компонентами системи.

У разі перевищення встановленого температурного значення контролер автоматично формував повідомлення для активації системи охолодження. При цьому система нагрівання вимикалася, що дозволяло уникнути одночасної роботи двох виконавчих механізмів. Якщо температура знижувалася нижче допустимого рівня, надсилалася команда на ввімкнення нагрівального елемента та деактивацію охолодження.

Такий підхід забезпечував автоматичне підтримання заданого температурного режиму без необхідності ручного втручання користувача. Реалізований алгоритм продемонстрував принцип роботи IoT-систем автоматизації, у яких сенсори, контролери та виконавчі пристрої взаємодіють через MQTT-протокол.

LCD-монітор, підключений до SBC, відображав поточні показники температури та стан активних пристроїв. Це дозволяло в режимі реального часу контролювати процес виконання команд, спостерігати за зміною параметрів системи та перевіряти коректність роботи програмного алгоритму.

Через додаток IoT Monitor на ноутбучі здійснювався візуальний контроль роботи мережі. Усі зміни станів пристроїв відображалися в інтерфейсі системи, що дозволяло перевіряти правильність виконання сценаріїв автоматизації та аналізувати поведінку мережі під час симуляції (рис. 4.8.) [12].

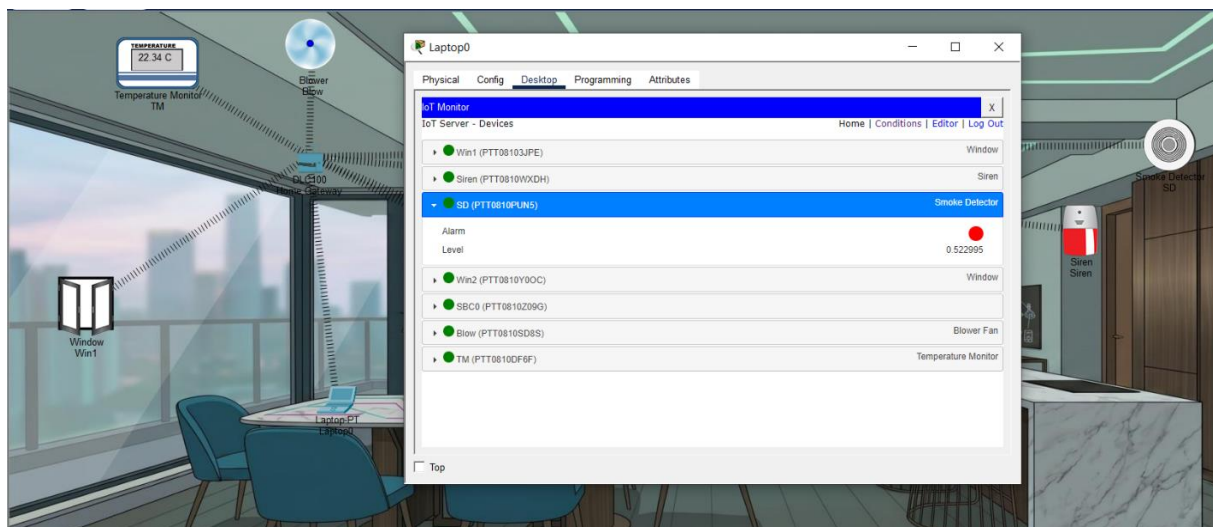


Рисунок 4.8. Моніторинг IoT-пристроїв у IoT Monitor

Отримані результати показали, що реалізована модель коректно функціонує в межах локальної IoT-мережі та забезпечує стабільний обмін повідомленнями між підключеними пристроями. Це підтверджує ефективність використання MQTT для організації взаємодії компонентів IoT-систем у середовищі Cisco Packet Tracer.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було досліджено особливості побудови IoT-мереж та принципи організації обміну повідомленнями в системах Інтернету речей. Проведено аналіз архітектури IoT-систем, основних компонентів мережі, мережевих технологій і протоколів передачі даних, що використовуються для взаємодії між пристроями. Особливу увагу приділено протоколу MQTT як одному з найбільш поширених легковагових протоколів обміну повідомленнями в IoT-середовищі.

У роботі розглянуто принципи функціонування MQTT, модель publish/subscribe та особливості роботи MQTT-брокера. Визначено, що використання MQTT дозволяє організувати зручний обмін повідомленнями між IoT-пристроями через брокер повідомлень. Під час симуляції було перевірено роботу системи при передачі повідомлень між пристроями, а також реакцію мережі у випадку вимкнення MQTT-брокера.

Також було проведено аналіз середовища Cisco Packet Tracer як інструмента моделювання IoT-мереж. Досліджено його функціональні можливості, підтримку IoT-пристроїв та MQTT-взаємодії. Визначено, що Cisco Packet Tracer дозволяє створювати моделі IoT-систем, налаштовувати мережеву взаємодію між пристроями та виконувати симуляцію обміну повідомленнями в режимі реального часу.

Практична частина роботи полягала у проектуванні та реалізації моделі IoT-мережі типу «розумний будинок» у середовищі Cisco Packet Tracer. У створеній мережі було використано Home Gateway, SBC-контролер, сенсор температури, LCD-монітор, охолоджуючий та нагрівальний елементи, датчик диму, сирену та інші IoT-пристрої. Для організації взаємодії між компонентами системи було налаштовано MQTT-брокер та реалізовано передачу повідомлень між пристроями.

У процесі симуляції перевірено роботу системи автоматичного контролю температури та елементів безпеки. Результати моделювання підтвердили коректність MQTT-взаємодії між пристроями та ефективність використання моделі

publish/subscribe для організації обміну даними в IoT-мережі. Система успішно реагувала на зміну параметрів середовища та автоматично керувала виконавчими пристроями.

У ході виконання роботи, було створено та досліджено модель IoT-мережі з використанням протоколу MQTT у Cisco Packet Tracer. Отримані результати підтверджують доцільність використання MQTT у системах Інтернету речей та можливість ефективного застосування Cisco Packet Tracer для моделювання IoT-взаємодії між пристроями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жураковський Б. Ю., Зенів І. О. Технології інтернету речей. Навчальний посібник: навч. посіб. Для студ. спеціальності 126 «Інформаційні системи та технології», спеціалізація «Інформаційне забезпечення робототехнічних систем» Київ: КПІ ім. Ігоря Сікорського, 2021. 271 с.
2. Atzori L., Iera A., Morabito G. The Internet of Things: A survey. Computer Networks. 2010. Vol. 54, no. 15. P. 2787–2805. URL: <https://doi.org/10.1016/j.comnet.2010.05.010> (дата звернення: 25.02.2026).
3. Internet of Things (IoT): A vision, architectural elements, and future directions. / J.Gubbi et al. Future Generation Computer Systems. 2013. Vol. 29, no. 7. P. 1645–1660.
4. Шпак О., Федорка П., Пригара М. Розумні міста та Інтернет речей: вплив розробок у сфері ІТ на розвиток міст і покращення якості життя. Сучасний стан наукових досліджень та технологій в промисловості. 2022, № 2(10).
5. Азаров О. Д., Захарченко С. М., Кадук О. В. та ін. Комп'ютерні мережі : підручник. Вінниця : ВНТУ, 2020. 378 с.
6. Kurose J. F., Ross K. W. Computer Networking: A Top-Down Approach. – 8th ed. Pearson, 2021. 775 p.
7. ISO/IEC 20922:2016. Information technology – Message Queuing Telemetry Transport (MQTT). Geneva : ISO, 2016. 80 p.
8. Hunkeler U., Truong H. L., Stanford-Clark A. MQTT-S – A publish/subscribe protocol for Wireless Sensor Networks. Middleware and Workshops (COMSWARE '08), Bangalore, India, 6–10 January 2008. 2008. URL: <https://doi.org/10.1109/comswa.2008.4554519> (дата звернення: 10.03.2026).
9. Light R. Mosquitto: server and client implementation of the MQTT protocol. Journal of Open Source Software. 2017. Vol. 2, no. 13. P. 265. URL: <https://doi.org/10.21105/joss.00265> (дата звернення: 13.03.2026).
10. Пулеко І. В., Єфіменко А. А. Методика моделювання IoT проектів «розумного» будинку в Cisco Packet Tracer. URL:

- <https://conf.ztu.edu.ua/wp-content/uploads/2019/06/43-1.pdf> (дата звернення: 29.03.2026).
- 11.Іванов П. В. Моделювання пристроїв інтернету речей засобами Cisco Packet Tracer – журнал «Фахова передвища освіта». URL: <https://osvitaftp.com.ua/2025/01/30/modeliuvannia-prystroiv-internetu-rechej-zasobamy-cisco-packet-tracer/> (дата звернення: 27.03.2026).
 - 12.Introduction to IoT and Packet Tracer / Cisco Networking Academy. URL: <https://www.netacad.com/courses/exploring-iot-cisco-packet-tracer?courseLang=en-US> (дата звернення: 06.04.2026).
 - 13.Cisco Packet Tracer – IoT Documentation. URL: <https://www.cisco.com> (дата звернення: 14.04.2026).
 - 14.MQTT Essentials – Introduction to MQTT. 90 p. URL: <https://www.hivemq.com/mqtt-essentials/> (дата звернення: 20.04.2026).
 - 15.MQTT vs HTTP for IoT Applications. URL: <https://developer.ibm.com/articles/mqtt-vs-http/> (дата звернення: 22.04.2026).
 - 16.Internet of Things (IoT) Overview – Cisco Systems. URL: <https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html> (дата звернення: 23.04.2026).

ДОДАТОК А

МОДЕЛЮВАННЯ ІОТ-СИСТЕМ У ВІРТУАЛЬНОМУ МЕРЕЖЕВОМУ СЕРЕДОВИЩІ

Бель Ангеліна Олександрівна

Рівненський державний гуманітарний університет, бакалавр, bel.anhelina.ipz@rshu.edu.ua

Шинкарчук Назар Володимирович

Рівненський державний гуманітарний університет, доцент кафедри інформаційних технологій та моделювання, кандидат технічних наук, доцент, nazar.shynkarchuk@rshu.edu.ua

MODELING IOT SYSTEMS IN A VIRTUAL NETWORK ENVIRONMENT

Bel Anhelina Oleksandrivna

Rivne State University of the Humanities, Bachelor, bel.anhelina.ipz@rshu.edu.ua

Shynkarchuk Nazar Volodymyrovych

Rivne State University of the Humanities, Associate Professor of the Department of Information Technologies and Modeling, Candidate of Technical Sciences, Associate Professor, nazar.shynkarchuk@rshu.edu.ua

This paper examines approaches to the design and modeling of Internet of Things (IoT) networks in a virtual environment Cisco Packet Tracer. It analyzes the architecture of IoT systems, networking technologies, and the specific features of using a modeling environment to create intelligent solutions. An example of an IoT system implementation is presented, and its effectiveness is evaluated.

Keywords: Internet of Things, modeling, networks, Cisco Packet Tracer.

Сучасний розвиток інформаційних технологій, зумовлює активне впровадження та використання технології Інтернет речей (IoT), яка передбачає інтеграцію фізичних об'єктів у цифрове середовище. Основна ідея такого підходу полягає у створенні мережі пристроїв, здатних самостійно збирати, передавати та обробляти дані без постійного втручання людини. Вперше цю концепцію було сформульовано Кевіном Ештоном у 1999 році, який розглядав Інтернет речей як систему взаємопов'язаних об'єктів, що обмінюються інформацією через мережу [1].

Подальший розвиток IoT привів до формування цілісного наукового підходу до побудови таких систем. Зокрема, дослідники визначають Інтернет речей як багаторівневу архітектуру, яка включає сенсорний, мережевий та прикладний рівні [2]. На сенсорному рівні здійснюється збір інформації про стан середовища, мережевий рівень забезпечує передачу даних, а прикладний – аналіз для прийняття рішень. Така структура дозволяє забезпечити ефективну взаємодію між компонентами системи і створює основу для автоматизації процесів.

З огляду на складність IoT-інфраструктури, важливим є не лише теоретичне розуміння її принципів, але й можливість їх практичної реалізації. Саме тому особливого значення набувають інструменти віртуального моделювання IoT-систем. Одним із таких інструментів є Cisco Packet Tracer (рис. 1.), який широко використовується для навчального моделювання мережевих взаємодій та дозволяє відтворювати сценарії роботи IoT-пристроїв [3].

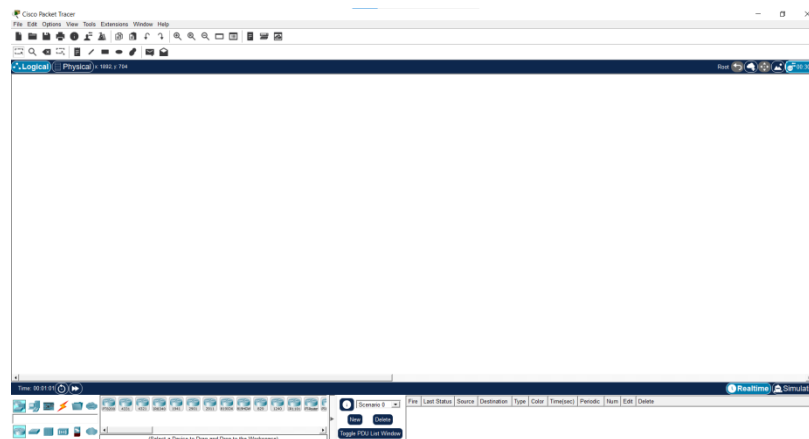


Рисунок 1. Робоче вікно Cisco Packet Tracer

Використання середовищ моделювання створює умови для переходу від теоретичних уявлень до практичного застосування Інтернету речей. У цьому контексті важливими є дослідження, спрямовані на адаптацію IoT-рішень до реальних умов та їх масштабування, що підтверджується сучасними науковими підходами до впровадження таких систем [4].

У межах даного дослідження, в Cisco Packet Tracer, було розроблено модель інтелектуального середовища, що імітує роботу мережі пристроїв Інтернету речей з автоматизованим керуванням. Побудована система включає сенсорні пристрої, які здійснюють моніторинг параметрів навколишнього середовища, зокрема температури, наявності руху та диму. Зібрані дані передаються через мережеву інфраструктуру.

Ключовою особливістю моделі є реалізація подієво-орієнтованого керування, відповідно до якого система реагує на зміну умов у режимі реального часу. Наприклад, виявлення руху призводить до автоматичного вмикання освітлення, що забезпечує раціональне використання енергоресурсів. Зміна температури активує систему кондиціонування, а фіксація диму запускає сигналізацію, що моделює механізми безпеки.

Таким чином, спроектована модель демонструє не лише технічну сторону реалізації технології Інтернет речей, але й її практичну доцільність. Вона наочно ілюструє, як взаємодія між сенсорами та виконавчими пристроями дозволяє автоматизувати процеси та підвищити ефективність управління середовищем.

Важливим результатом дослідження є підтвердження того, що використання середовищ моделювання дозволяє значно спростити процес проектування IoT-систем (рис. 2.). Це дає можливість тестувати різні сценарії роботи, виявляти потенційні проблеми та оптимізувати систему ще до її фізичної реалізації. Такий підхід є особливо актуальним у навчальному процесі, оскільки сприяє формуванню практичних навичок без необхідності використання дороговартісного обладнання.

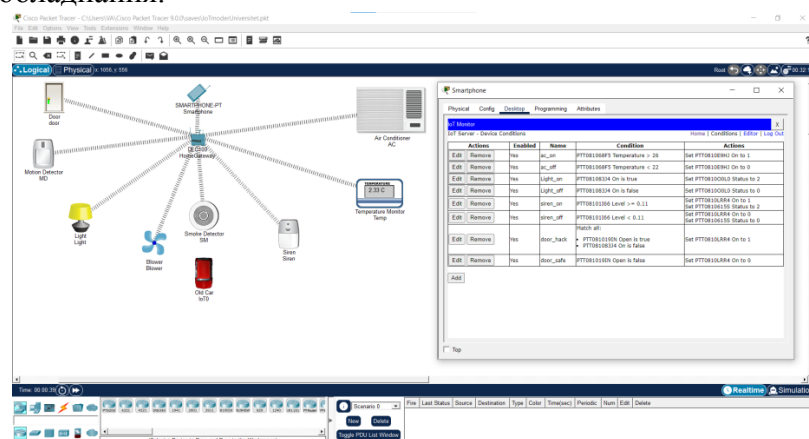


Рисунок 2. Моделювання IoT-мережі у Cisco Packet Tracer

Візуальне представлення моделі дозволяє краще зрозуміти принципи організації IoT-систем та наочно демонструє результати виконаного проекту.

Отже, поєднання теоретичних основ Інтернету речей із їх практичною реалізацією у середовищі моделювання забезпечує цілісне розуміння принципів побудови сучасних інформаційних систем. Отримані результати можуть бути використані як у навчальному процесі, так і при розробці реальних рішень Інтернету речей.

Список використаних джерел

1. That “Internet of Things” Thing. RFID Journal. URL: <https://www.rfidjournal.com/expert-views/that-internet-of-things-thing/73881/> (дата звернення 16.04.2026).
2. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29 (2013), P. 1645-1660.
3. Cisco Packet Tracer. URL: <https://www.netacad.com/cisco-packet-tracer> (дата звернення: 22.04.2026).
4. Vermesan O., Friess P. Internet of Things: From Research and Innovation to Market Deployment. River Publishers. 2014, 373 p.