

ДИПЛОМНА РОБОТА

Другий (Магістерський)

Освітній рівень

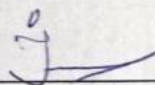
Галузь знань 17 Електроніка та телекомунікації
Шифр і назва спеціальності

Спеціальність 172 Телекомунікації та радіотехніка
Шифр і назва спеціальності

на тему «Метод моніторингу параметрів IoT-мережі на основі
аналізу фазових портретів»

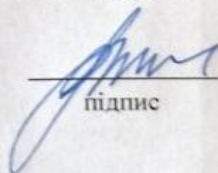
ДРТР. 2021075.01.04 ПЗ

Виконав: студент 2 курсу, група ТР_м-21-1


підпис

В.В. Іваненко
Ініціали, прізвище

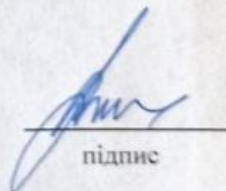
Керівник: докт. техн. наук, проф.


підпис

С.К. Підченко
Ініціали, прізвище

До захисту допускаю:

Зав. кафедри: д-р техн. наук, проф.


підпис

С.К. Підченко
Ініціали, прізвище

13 12 2022 р.

Хмельницький, 2022

Хмельницький національний університет

Факультет інформаційних технологій
 Кафедра телекомунікацій, медійних та інтелектуальних технологій
 Освітній рівень другий (магістерський)
 Галузь знань 17 – Електроніка та телекомунікації
 Спеціальність 172 – Телекомунікації та радіотехніка
 Освітня-професійна програма Телекомунікації та радіотехніка

ЗАТВЕРДЖУЮ

Зав. кафедрою ТМІТ

« 05 » 09 2022р.

**ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ**

Іваненку Валерію Вікторовичу

1 Тема роботи: *«Метод моніторингу параметрів IoT-мережі на основі аналізу фазових портретів»*

керівник роботи Підченко Сергій Костянтинович, д.т.н, професор

Затверджено наказом по університету від «1» липня 2022 р. № 83

2 Строк подання студентом роботи на кафедру: 25.11.2022 р.

3 Вихідні дані (характеристика об'єкта, умов дослідження та ін.)

Мета роботи – підвищення ефективності моніторингу параметрів мережі пристроїв IoT за рахунок використання графоаналітичного методу аналізу фазових портретів, отриманих в результаті реконструкції часових рядів зміни пропускнуої спроможності каналу передачі даних

Об'єкт дослідження – процес функціонування та міжмережевої комунікації пристроїв Інтернету речей у складі локальних комп'ютерних мереж.

Предмет дослідження – метод моніторингу параметрів IoT-мережі на основі аналізу фазових портретів.

4 Зміст пояснювальної записки (перелік питань, що їх належить розробити)

1 Аналітичний огляд архітектури інтернету речей

2 Математична модель методу моніторингу параметрів мережі пристроїв Інтернету речей на основі аналізу фазових портретів

3 Модель циклу моніторингу та керування параметрів мережі пристроїв Інтернету речей

4 Встановлення та налаштування програмного забезпечення для Web-орієнтованого інтерфейсу користувача – адміністратора локальної мережі пристроїв Інтернету речей

Завдання отримав  Іваненко В.В.


підпис

Науковий керівник  Підченко С.К.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів (розділів) дипломної роботи	Строк виконання етапів дипломної роботи	Примітка
1.	Аналіз завдання на ДП	15.09.2022р.	Виконано
2.	Вступ. Розробка 1-го розділу	25.09.2022р.	Виконано
3.	Розробка 2-го розділу	15.10.2022р.	Виконано
4.	Розробка 3-го розділу	25.10.2022р.	Виконано
5.	Розробка 4-го розділу та висновки	15.11.2022р.	Виконано
6.	Корекція зауважень керівника роботи	20.11.2022р.	Виконано
7.	Підготовка доповіді та презентаційних матеріалів	25.11.2022р.	Виконано

Студент


підпис

Іваненко В.В.

Ініціали, прізвище

Керівник роботи


Підпис

Підченко С.К.

Ініціали, прізвище

ЗМІСТ

Перелік умовних скорочень	6
Вступ.....	7
1 Аналітичний огляд архітектури Інтернету речей	10
1.1 Основні поняття та історія створення Інтернету речей	10
1.2 Огляд архітектури Інтернету речей.....	13
1.3 Розширення простору розумних пристроїв IoT для задоволення потреб користувачів.....	17
Висновки до першого розділу.....	21
2 Математична модель методу моніторингу параметрів мережі пристроїв Інтернету речей на основі аналізу фазових портретів	22
2.1 Метод фазової площини	22
2.2 Моделювання локальної IoT мережі в програмному симуляторі Cisco Packet Tracer.....	23
2.3 Аналітичний розв’язок логістичного рівняння.....	27
Висновки до другого розділу	29
3 Модель циклу моніторингу та керування параметрів мережі пристроїв Інтернету речей	30
3.1 Моніторинг та керування Інтернетом речей	30
3.2 Цикли вимірювання, контролю, аналізу та оптимізації Інтернету речей	35
3.3 Обробка подій в Інтернеті речей	41
Висновки до третього розділу.....	45
4 Встановлення та налаштування програмного забезпечення для web-орієнтованого інтерфейсу користувача – адміністратора локальної мережі пристроїв Інтернету речей	46
4.1 Встановлення та налаштування програмного засобу Virtual Box.....	46
4.2 Встановлення та налаштування гостьової машини на Ubuntu Server	49

4.3 Встановлення системи управління Web-контентом WordPress та розширення SCADA MQTT	56
Висновки до четвертого розділу.....	63
Загальні висновки до роботи.....	65
Перелік джерел посилання	66
Додаток А. Конфігураційні файли Apache2 та PHP.....	71

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

МК – Мікроконтролер

ДС – Динамічна система

ОС – Операційна система

ПЗ – Програмне забезпечення

ВМ – Віртуальна машина

ІоТ – Internet of Things

МТС – Machine Type Communications

М2М – Machine-to-Machine

WWAN – Wireless Wide Area Networks

WLAN – Wireless Local Area Networks

WPAN – Wireless Personal Area Networks

LTE – Long Term Evolution

WSN – Wireless Sensor Network

MAC – Medium Access Control

WSN – Wireless Sensor Network

OSS – Operation Support System

BSS – Business Support System

BRM – Business Rule Management

BPM – Business Process Management

CPU – Central Processor Unit

ВСТУП

Актуальність теми дослідження. Для ефективного проектування, розробки та впровадження інтелектуальних пристроїв Інтернету речей (Internet of Things, IoT) необхідно вирішити ряд інженерно-технічних задач щодо оптимізації роботи та забезпечення відмовостійкості пристроїв, особливо в умовах нестабільного підключення, хакерських атак, зовнішнього джерела завад тощо. На етапі, який передувє безпосередній розробці та апаратно-програмній реалізації IoT-мереж – етапі моделювання та проектування, критично важливим є якісна та кількісна оцінка параметрів мережі, в першу чергу завантаження та пропускнуої здатності міжсегментних ліній передачі, що дозволить в майбутньому уникнути непередбачуваних відмов та зменшити витрати, пов'язані із експлуатацією.

З іншого боку, вже реалізована і функціонуюча система IoT пристроїв, яка на даний момент дозволяє доволі ефективно вирішувати поставлені на етапі проектування технологічні задачі, може бути схильна до непередбачуваних відмов та дестабілізації функціонування, отже, описані підходи також актуальні для модернізації існуючих рішень та систем, які вже введені в експлуатацію.

Мета і задачі дослідження. Метою роботи є підвищення ефективності моніторингу параметрів мережі пристроїв IoT за рахунок використання графоаналітичного методу аналізу фазових портретів, отриманих в результаті реконструкції часових рядів зміни пропускнуої спроможності каналу передачі даних

Для досягнення поставленої мети в роботі сформульовані та вирішені наступні *задачі*:

- аналітичний огляд концепцій Інтернету речей та його архітектури;
- аналіз архітектурних рівнів Інтернету речей з позиції розробки функцій моніторингу мережі;

- застосування методу аналізу фазових портретів для моніторингу та контролю завантаження каналу локальної IoT-мережі.

- моделювання нестабільного режиму роботи локальної IoT мережі, викликаного нестабільними режимами роботи кінцевих пристроїв;

- встановлення та налаштування прототипу централізованого сервера для керування та контролю мережі IoT-пристроїв.

Об'єкт дослідження: процес функціонування та міжмережевої комунікації пристроїв Інтернету речей у складі локальних комп'ютерних мереж.

Предмет дослідження: метод моніторингу параметрів IoT-мережі на основі аналізу фазових портретів.

Методи дослідження: для вирішення поставлених задач були використані методи математичного моделювання, математичного аналізу, системного аналізу, алгоритмізації та програмування.

Наукова новизна отриманих результатів: отримав подальшого розвитку графоаналітичний метод моніторингу параметрів мережі пристроїв Інтернету речей на основі аналізу фазових портретів завантаженості каналу передачі даних, які були побудовані на основі часових рядів миттєвих значень швидкості прийому та передачі даних між сервером та кінцевими пристроями мережі.

Практичне значення отриманих результатів:

- засобами програмного симулятора Cisco Packet Tracer була розроблена імітаційна модель локальної мережі, яка складається із двох сегментів із різними режимами роботи кінцевих пристроїв;

- на основі логістичного відображення була розроблена модель нестабільності мережевих підключень кінцевих пристроїв, що призводить до нерівномірного неперіодичного режиму завантаженості мережі, яких подібний до стану динамічного хаосу;

- налаштовано сервер збору та відображення даних параметрів мережі, який працює під управлінням операційної системи Ubuntu Linux в середовищі віртуалізації Virtual Box.

Апробація результатів дослідження. За результатами магістерської роботи, відповідно до теми дослідження, були підготовлені тези, які були опубліковані в збірнику тез доповідей XIV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук», яка проходила в м. Хмельницькому 18-19 листопада 2022 року.

1 АНАЛІТИЧНИЙ ОГЛЯД АРХІТЕКТУРИ ІНТЕРНЕТУ РЕЧЕЙ

1.1 Основні поняття та історія створення Інтернету речей

Інтернет речей (Internet of Things, IoT) — це мережа фізичних об'єктів, які мають вбудовані технології, що дозволяють здійснювати взаємодію з зовнішнім середовищем, передавати відомості про свій стан і приймати дані ззовні. Складовою частиною Інтернету речей є Індустріальний (або Промисловий) Інтернет речей (Industrial Internet of Things, IIoT).

Концепція Інтернету речей полягає у підключення будь-якого пристрою (якщо він має перемикач включення / вимикання) до Інтернету та інших під'єднаних пристроїв. Таким чином, IoT представляє собою величезну базу даних, в якій, інтелектуальні пристрої обмінюються інформацією, що була отримана з датчиків, а також електронних пристроїв, системної мережі та апаратно-програмного забезпечення. Зв'язок між пристроями може відбуватися між різними фізичними об'єктами, наприклад, в офісі (корпоративний сегмент, бізнес), вдома (системи «розумний будинок»), чи на підприємстві (промисловий Інтернет речей).

Термін «Інтернет речей» вперше був запропонований в 1999 році Кевіном Ештоном, одним з трьох засновників Центру автоматичної ідентифікації Массачусетського університету (США) [1].

Одним з перших підключених до мережі пристроїв став апарат з продажу напою Coca-Cola, встановлений в Університеті Карнегі – Меллон у 1982 році. Апарат мав можливість передавати на сервер дані про кількість пляшок напою, що містились в ньому, та про свій стан в цілому.

Періодом активних обговорень мереж, які змогли б забезпечити міжмашинну взаємодію стали 1990-ті роки. Наприклад, керівник дослідницьких робіт в Xerox

PARC (дослідному центрі компанії Xerox) Марк Вейзер запропонував концепцію комп'ютеризації, що передбачала масове впровадження комп'ютерів і організацію зв'язку між ними, завдяки якій машини могли б самостійно вирішували повсякденні задачі користувача.

Вчений Білл Джой в рамках виступу на Міжнародному економічному форумі в Давосі в 1999 році запропонував ідею «Шести типів Web» - шести типів Інтернету майбутнього. У ній він досить точно спрогнозував появу бездротових мобільних мереж, інтелектуальних голосових помічників і комунікацій між пристроями (за його термінологією такий зв'язок називалася Device-to-device). В тому ж році були спроби створення перших IoT-проектів.

Технологія Інтернету речей (Internet of Things, IoT) на сьогоднішній день є новою ключовою технологією майбутнього, суть якої полягає в об'єднанні величезної кількості окремих вузлів (речей), таких як сенсори, приводи та інші розумні об'єктів, що підключені до Інтернету. Ці сенсори та пристрої керування (наприклад, камери спостереження, побутова техніка та сенсори моніторингу навколишнього середовища) зазвичай оснащені різними типами мікроконтролерів (МК), приймачів та протоколів для передачі даних вимірювальних сигналів та сигналів керування [2-5]. Такі вузли (сенсори, або виконавчі пристрої) з'єднуються один з одним для передачі отриманих даних на централізовані сервери, де інформація збирається та зберігається і робиться доступною для конкретних користувачів з відповідними правами доступу.

Передача даних від одного вузла (сенсора або приводу) до іншого вузла чи сервера IoT здійснюється за допомогою нової парадигми зв'язку, яка називається Machine Type Communications (MTC) або Machine-to-Machine (M2M).

Канал зв'язку між пристроями IoT і сервером IoT реалізується у вигляді бездротового радіоканалу, що спрощуватиме процеси установки і розгортання.

Пристрої IoT використовують технології бездротового радіодоступу, такі як бездротові глобальні мережі (WWAN) і бездротові локальні мережі (WLAN) для зв'язку з серверами [6]. У деяких випадках, внаслідок деяких обмежень, пристрої IoT можуть спочатку зв'язуватися з проміжними суб'єктами, які називаються шлюзами IoT або M2M-шлюзами через бездротові персональні мережі (WPAN) або WLAN. Шлюзи, в свою чергу, пересилають дані з цих пристроїв до серверів IoT і діють як перекладач між пристроями IoT та серверами [7-10].

Зв'язок між пристроями IoT та іншими шлюзами або серверами IoT може бути забезпечений за допомогою різних видів бездротових технологій, таких як 3GPP Long Term Evolution (LTE) та LTE-Advanced, WiFi, ZigBee та Bluetooth або інших стандартних бездротових технологій [11,12].

Мережеві характеристики IoT з використанням цих бездротових технологій досить сильно відрізняються від характеристик для традиційних дротових або бездротових мереж, оскільки кількість пристроїв, що беруть участь у зв'язку дуже велика. Крім того, трафік на один пристрій IoT, як правило, не такий великий, оскільки кожен пристрій IoT сприймає і передає невеликий обсяг даних на відповідний IoT-сервер, хоча дані, що генеруються з величезної кількості об'єктів, можуть в сукупності мати певний вплив на продуктивність мережі. Крім того, мережі IoT повинні працювати стабільно і стійко протягом більш тривалого періоду без будь-якої необхідності втручання людини [13].

Інший аспект полягає в тому, що шлюзи можуть включати в себе кілька радіоінтерфейсів для забезпечення універсальності цільовими пристроїв, та оптимізації таких параметрів як пропускну здатність, затримка і енергоефективність [15].

Пристрої IoT в таких мережах, як правило, будуть працювати на основі акумуляторних джерел живлення, а отже, енергоефективність, природно, має

першочергове значення в управлінні пристроями. Розглядаючи конкретну область бездротових сенсорних мереж (WSN), енергоефективність сенсорних вузлів, що працюють від батарей, і продовження терміну служби представляли питання, які досліджувались протягом тривалого часу [15,16], де протоколи рівня керування доступом до середовища (MAC) зосереджені на регулюванні робочого циклу сенсорних вузлів, а протоколи рівня маршрутизації призначені для агрегації даних та передачі "багато до одного". Аналогічно, оскільки пристрої IoT також працюють від батареї, слід враховувати споживання батареї під час розгортання мережі IoT [17]. Однак, характеристики мережі IoT та сценарії розгортання є більш складними, ніж традиційні WSN в різних аспектах, наприклад, обсяг пристроїв IoT, двонаправлений трафік між пристроями IoT та серверами, різноманітні дані для зондування та активації, використання різноманітного бездротового радіодоступу гетерогенних технологій бездротового радіодоступу, наявність шлюзів IoT тощо. З цієї причини деякі традиційні стратегії енергозбереження WSN, такі як агрегація та кластеризація однорідних даних не можуть бути безпосередньо застосовані до IoT. Проводяться широкі дослідження з енергозбереження для пристроїв IoT з обмеженим ресурсом батареї з різних аспектів, таких як стандартизація, академічні дослідження і розвиток промисловості.

1.2 Огляд архітектури Інтернету речей

Інтернет речей складається з набору різних інфокомунікаційних технологій, що забезпечують функціонування Інтернету речей, і його архітектура показує, як ці технології пов'язані один з одним.

Спрощено система IoT складається з чотирьох основних компонентів, як показано на рисунку.

Річ (Thing) – пристрій з мережевим інтерфейсом, до складу якого входять сенсори збору даних та механізмів керування.

Шлюз (Gateway) – пристрій (точка) доступу до Інтернету.

Хмара (Cloud) – сукупність серверів в мережі Інтернет, що об'єднані в кластери з єдиним інтерфейсом доступу для програм для збереження та обробки даних.

Інтерфейси людини-користувача (Human-interfaces) – програмне забезпечення високого рівня для взаємодії із кінцевими користувачами системи. Це можуть бути мобільні програми керування та візуалізації даних, Web-орієнтовані додатки чи настільні додатки для операційних систем Windows, Linux тощо.

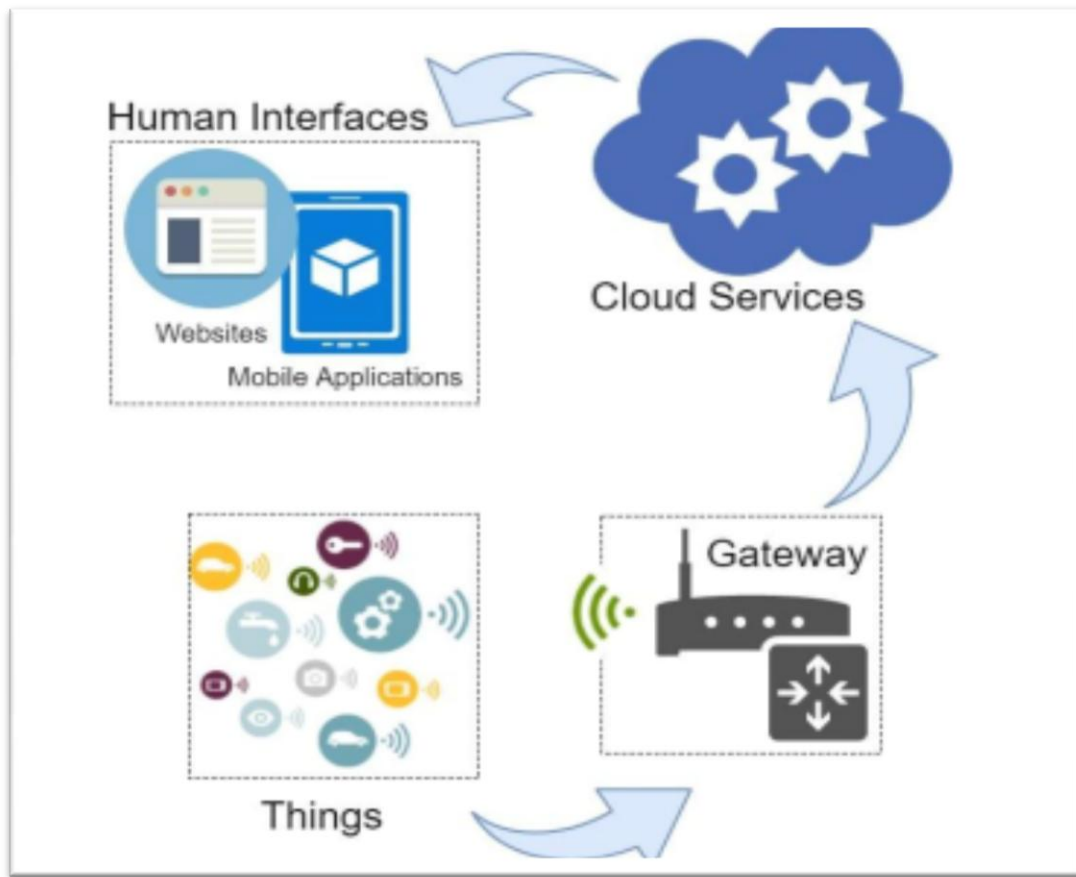


Рисунок 1.1 – Складові Інтернету речей

Архітектура IoT включає чотири функціональні рівні:

- 1) рівень сенсорів та сенсорних мереж
- 2) рівень шлюзів та мереж
- 3) сервісний рівень
- 4) рівень додатків

Найнижчий рівень архітектури IoT складається з «розумних» (smart) об'єктів – речей, інтегрованих із сенсорами (датчиками). Сенсори реалізують поєднання фізичного та віртуального (цифрового) середовищ, забезпечуючи збирання та обробку інформації в реальному масштаб часу. Мініатюризація, що призвела до зменшення фізичних розмірів апаратних сенсорів, дозволила інтегрувати їх безпосередньо в об'єкти фізичного середовища. Сенсори класифікуються відповідно до їх призначення, наприклад, сенсори навколишнього середовища, сенсори для тіла, сенсори для побутової техніки, сенсори для побутової техніки транспортних засобів тощо.

Більшість сенсорів потребує з'єднання з агрегатором сенсорів (шлюзом), який може бути реалізовано бути реалізований з використанням локальних обчислювальних мереж (LAN, Local Area Network), таких як Ethernet та Wi-Fi або персональних мереж (PAN, Personal Area Network), таких як ZigBee, Bluetooth та ультраширокопasmугового бездротового зв'язку на малих відстанях (UWB, Ultra-Wide Band).

Для сенсорів, які не вимагають підключення до агрегатора, їх зв'язок із серверами/додатками може надаватися з використанням глобальних бездротових мереж WAN, таких як GSM, GPRS та LTE. Сенсори, які характеризуються низьким енергоспоживанням та низькою швидкістю передачі даних, утворюють широко відомі бездротові сенсорні мережі (WSN, Wireless Sensor Network). WSN набирають

все більшої популярності, оскільки вони можуть утримувати набагато більше сенсорів з підтримкою роботи від батарей та охоплюють великі площі.

Великий обсяг даних, створюваних на першому рівні IoT численними мініатюрними сенсорами, вимагає надійної та високопродуктивної дротової або бездротової мережевої інфраструктури в якості транспортного середовища. Існуючі мережі зв'язки, що використовують різні протоколи, можуть бути використані для підтримки міжмашинних комунікацій M2M та їх додатків. Для реалізації широкого спектру послуг та додатків в IoT необхідно забезпечити спільну роботу безлічі мереж різних технологій та протоколів доступу в гетерогенній конфігурації. Ці мережі повинні забезпечувати необхідні значення якості передачі інформації, і насамперед із затримки, пропускнуої спроможності та безпеки. Цей рівень складається з конвергентної мережної інфраструктури, що створюється шляхом інтеграції різнорідних мереж у єдину мережеву платформу. Конвергентний абстрактний мережевий рівень у IoT дозволяє через відповідні шлюзи кільком користувачам використовувати ресурси однієї мережі незалежно та спільно без зниження рівня конфіденційності, безпеки та продуктивності.

Сервісний рівень містить набір інформаційних послуг, покликаних автоматизувати технологічні та бізнес операції в IoT: підтримки операційної та бізнес діяльності (OSS/BSS, Operation Support System/Business Support System), різної аналітичної обробки інформації (наприклад, статистичного та інтелектуального аналізу даних та текстів) зберігання даних, забезпечення інформаційної безпеки, управління бізнес-правилами (BRM, Business Rule Management), управління бізнес-процесами (BPM, Business Process Management) тощо.

На четвертому рівні архітектури IoT існують різні типи додатків для відповідних промислових секторів та сфер діяльності (енергетика, транспорт,

торгівля, медицина, освіта тощо). Програми можуть бути «вертикальними», коли вони є специфічними для конкретної галузі промисловості, а також «горизонтальними», (наприклад, керування автопарком, відстеження активів фірми тощо), які можуть використовуватись у різних секторах економіки

1.3 Розширення простору розумних пристроїв IoT для задоволення потреб користувачів

Перше десятиліття XXI століття було дуже плідним на дослідження, спрямовані на втілення в життя бачення Вейзера [18] щодо розповсюджених обчислень (Pervasive Computing). Ці ранні роботи привели до ряду захоплюючих результатів досліджень, які сприяли появі все більшої кількості малих за розміром пристроїв, підключених один до одного і до Інтернету. Різні технології були задіяні для покращення комунікаційних можливостей інтелектуальних об'єктів, наприклад, RFID/NFC-мітки, бездротові вбудовані сенсори і виконавчі пристрої, протоколи зв'язку з низьким енергоспоживанням тощо. Всі ці технології об'єднані концепцією, яка охоплює багато дослідницьких напрямків: Інтернет речей, який був задуманий з багатообіцяючою ідеєю поліпшення зв'язку обмежених і вбудованих пристроїв з Інтернетом.

Спочатку застосування Інтернету речей (IoT) було зосереджено на управлінні інформацією за допомогою RFID-міток [19, 20], таких як відстеження товарів, управління предметами повсякденного вжитку, автоматичні платежі на ринках та військове застосування. Еволюція мікроелектромеханічних (MEMS) технологій сприяла еволюції Інтернету речей за рахунок мініатюризації апаратних компонентів, а саме бездротових приймачів, сенсорів, актуаторів, мікроконтролерів тощо. Ці досягнення мотивували багато пропозицій з удосконалення програмного

забезпечення вбудованих пристроїв в декількох аспектах, наприклад, протоколи, операційні системи, архітектури тощо, які мали важливе значення для оптимізації взаємодії "машина-машина" (M2M), забезпечуючи набагато більш активний характер додатків на основі IoT. У цьому сенсі сучасна парадигма IoT полягає у створення розумних просторів, які складаються з тисяч "розумних" речей з автономними можливостями.

В той час як впровадження Інтернету речей зростає в геометричній прогресії, Інтернет послуг вже став реальністю. Одним із актуальних напрямків майбутнього Інтернету є зосереджений метод інтеграції реальних послуг, що надаються Інтернетом речей, в поєднання традиційних послуг. Таких новий підхід в Інтернеті буде розглядатися з двох точок зору, що визначають два виміри взаємодії, горизонтальний і вертикальний підходи, що передбачають взаємодію "машина-машина" і "людина-машина" (H2M), відповідно. Другий підхід також передбачає взаємодію між машинами і процесами, які можуть бути запущені на зовнішніх об'єктах, підключених до інших мереж. Останнім часом виникло занепокоєння щодо того, як оптимізувати управління такими вертикальними взаємодіями, які генеруватимуть величезні обсяги інформації з точки зору гетерогенності, багаторазового використання, масштабованості або доступності.

Основною проблемою, яка перешкоджає вирішенню цих питань, є широке використання спеціальних і монолітних конструкцій, які ускладнюють конвергенцію різних додатків за допомогою загального і відкритого рішення. Для того, щоб зробити здійсненними і багаторазово використовуваними технології на основі IoT, необхідна відкрита інфраструктура, яка дозволяє отримати доступ до гетерогенних датчиків і виконавчих пристроїв, що працюють за стандартами Web та Інтернету. Майбутні додатки для мереж на основі IoT вимагатимуть значного поліпшення повторного використання ресурсів розгортання, щоб бути доступними

для багатьох високорівневих об'єктів високого рівня, наприклад, додатків для смартфонів, веб-сайтів, експертних систем тощо.

Змістовні пропозиції щодо вирішення вищезазначених проблем виникли за концепції відкритої платформи. Ця концепція включає в себе набір конкретних питань, які можуть значно полегшити швидке прототипування і розгортання великих мереж на основі Інтернету речей, сприяти швидкому створенню прототипів і розгортанню великих мереж на основі IoT. Хоча її основною особливістю є відкритість інтерфейсів (тобто добре задокументовані і загальнодоступні інтерфейсів), вона включає в себе інші ідеї, які сприяють повторному використанню, розширенню і повторному впровадженню функціональних можливостей, для яких платформа не була спочатку спроектована.

Важливо підкреслити, що під цим поняттям не обов'язково мається на увазі відкритий вихідний код результатів, а набір загальнодоступних операцій, які визначені в API (Application Programming Interface – інтерфейс прикладного програмування). Кілька реалізацій одного і того ж API можуть бути запропоновані для використання для різних технологічних платформ (наприклад, різних мобільних операційних систем) або мов програмування (наприклад, JAVA, C, C++ тощо). Нова тенденція у відкритих платформах є надання деяких цікавих інструментів для інтеграції мереж IoT з відомими веб-технологіями за допомогою категорії сервісів хмарних обчислень: "Платформа як послуга" (Platform as a Service, PaaS). Модель PaaS дозволяє споживачам послуг легко розробляти та розгортати послуги за допомогою інструментів робочого столу та/або бібліотек, що надаються постачальником, допомагає споживачеві витратити менше ресурсів на управління та обслуговування великих інфраструктур апаратного та програмного забезпечення. Рішення на основі PaaS мають як спільні риси, так і специфічні характеристики, що відрізняють їх одне від одного. Спільними рисами є, серед

іншого, надання середовища розгортання, інформаційної панелі моніторингу, а також комунікаційних механізмів для обміну інформацією між платформою та клієнтом. Обчислювальні ресурси, що передаються клієнтським обліковим записам, зазвичай залежать від масштабованості та якості обслуговування, необхідних для додатків, що використовують платформи. Поєднання відкритої платформи з парадигмою PaaS набуло популярності серед дослідників, які зосереджені на інтеграції IoT в Інтернет. PaaS тісно пов'язана з Інтернетом речей та Вебом Речей (Web of things, WoT). Взагалі кажучи, WoT намагається подолати розрив між вбудованими пристроями і Інтернетом, щоб інтегрувати дуже різні і ізольовані мережі IoT з середовищами користувачів за допомогою веб-технологій, а також один з одним. Для досягнення основної мети WoT необхідно адаптувати традиційні веб-технології для створення відкритих інтерфейсів, форматів представлення, протоколів виявлення і зв'язку або механізмів пошуку інформації, які полегшують інтеграцію об'єктів реального світу, в тому числі в IoT. Наприклад, спільним аспектом цих підходів є використання архітектурного стилю Representational State Transfer (REST) [21] та JavaScript Object Notation (JSON) для форматів представлення. У підходах, заснованих на REST, об'єкти реального світу ідентифікуються за допомогою унікальних ідентифікаторів ресурсів (URI), а пов'язана з ними інформація може бути доступна за допомогою виклику методів HTTP. Ці особливості (серед іншого) полегшують розробку на основі IoT, спрямованих на збір, обробку або агрегування великих обсягів даних з датчиків та інших віртуальних джерел даних. Розробка підходів за цими принципами має мотивувати появу відкритих платформ, орієнтованих на абстрагування кожного платформ, орієнтованих на абстрагування кожної концепції, пов'язаної з IoT, з метою побудови стовпів для майбутнього WoT [22].

Висновки до першого розділу

В першому розділі магістерської роботи було зроблено наступне:

1. Проведено огляд основних концепцій, визначень та історії виникнення та розвитку концепції Інтернету речей.
2. Проведено аналітичний огляд архітектури Інтернету речей, яка в загальному складається з чотирьох основних рівнів: сенсорний, шлюзовий (мережевих), сервісний, прикладний (рівень додатків).
3. Розглянуто тенденції розвитку Інтернету речей, розширення ринку надання інформаційних послуг в глобальній мережі Інтернет та загальне зростання популярності концепцій Інтернету речей.

2 МАТЕМАТИЧНА МОДЕЛЬ МЕТОДУ МОНІТОИНГУ ПАРАМЕТРІВ МЕРЕЖІ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ НА ОСНОВІ АНАЛІЗУ ФАЗОВИХ ПОРТРЕТІВ

2.1 Метод фазової площини

Для довільної динамічної системи (ДС) із двома змінними стану система рівнянь матиме вигляд:

$$\begin{cases} du/dt = P(u, v) \\ dv/dt = Q(u, v) \end{cases}, \quad (2.1)$$

де $u(t), v(t)$ – фазові змінні;

P, Q – нелінійні функції двох змінних, отримані в результаті поліноміальної апроксимації вихідного часового ряду.

Система (2.1) описує динаміку нелінійної ДС на заданому проміжку часу (за умови існування та однозначності рішень функцій правих частин системи P, Q).

Геометрично стан (поточну фазу) системи (2.1) можна зобразити на фазовій площині в декартових координатах, де миттєвому стану системи буде поставлена у відповідність точка $M(u_k, v_k)$. Таким чином, еволюція системи представлена деякою фазовою траєкторією, яку описує точка M , напрямок та швидкість руху якої визначається вектором фазових швидкостей.

Для системи (1) рівняння фазових траєкторій матиме вигляд:

$$v(u) = \int \frac{Q}{P} du + C, \quad (2.2)$$

де C – постійна інтегрування, що визначається на основі початкових значення координат v_0 та u_0 для моменту часу $t = 0$.

У разі неможливості аналітичного розв'язування системи (2.1) та виразу (2.2), фазовий портрет системи може бути побудовано за допомогою ітеративного відображення, отриманого для системи (2.1) шляхом заміни диференціалів функцій скінченними різницями, або за допомогою чисельного розв'язку системи (2.1), наприклад, методом Рунге-Кутти.

2.2 Моделювання локальної IoT мережі в програмному симуляторі Cisco Packet Tracer

Проведемо дослідження на прикладі локальної мережі, схема якої зображена на Рисунку 2.1. Дана комп'ютерна мережа складаються із трьох сегментів (А, В і С), маршрутизатора 1804 та трьох комутаторів 2960 [22, 23].

Розглянемо завантаження каналу передачі даних між сервером, який знаходиться в сегменті А, та усіма пристроями, розміщеними в сегментах В та С (Рисунок 1).

Позначимо $x_{B,rx}$ та $x_{B,tx}$ – миттєві значення швидкості прийому та передачі даних (Mbit/s) між сервером та кінцевими пристроями сегменту В. Аналогічно $x_{C,rx}$ та $x_{C,tx}$ – відповідні значення для пристроїв сегменту С.

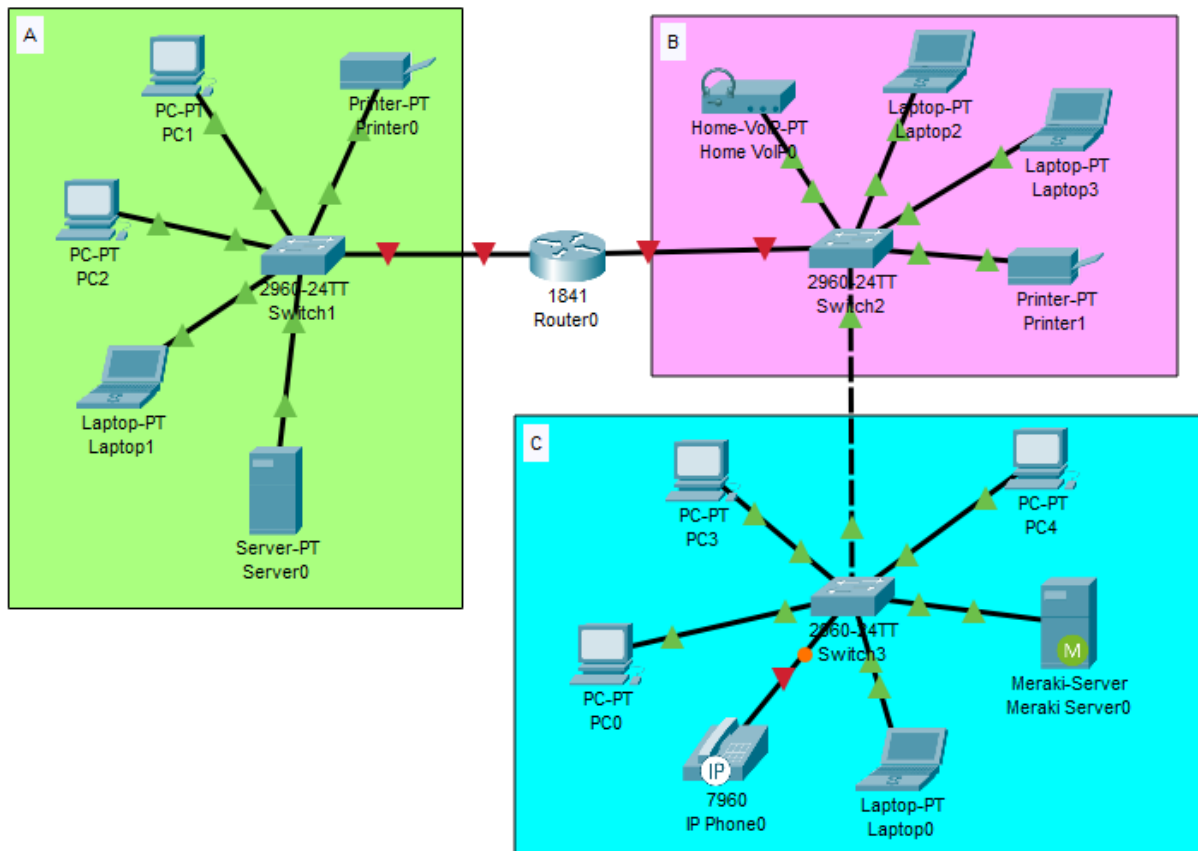


Рисунок 2.1 – Схема досліджуваної мережі

Нехай завантаження каналу пристроями сегменту мережі В носить детермінований і постійних характер, тому для деякого інтервалу часу Δt можна вважати:

$$\begin{aligned} x_{B,rx} &= X_{rx} = const \\ x_{B,tx} &= X_{tx} = const \end{aligned} \quad (2.3)$$

В той же час, номенклатура та режими роботи кінцевих пристроїв сегменту мережі С обумовлюють складний неперіодичний характер завантаження мережі, що можна порівняти із режимами детермінованих хаотичних коливань.

Розглянемо логістичне відображення в якості математичної моделі джерела таких коливань:

$$x_{n+1} = rx_n(1 - x_n), \quad (2.4)$$

де x_n – фазова змінна,
 r – біфуркаційний параметр.

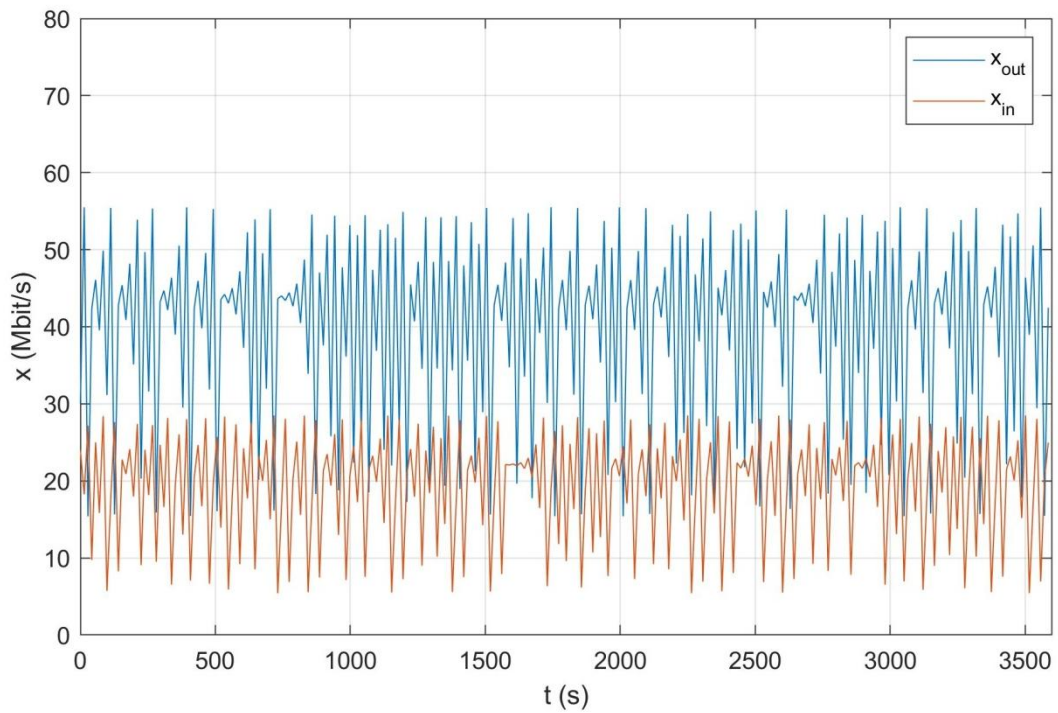
Отже, зображені на Рисунку 2.2,а часові ряди для вхідного та вихідного трафіку у випадку складного, неперіодичного характеру функціонування номенклатури пристроїв сегменту С мережі (див. Рисунок 2.1) відповідають хаотичному режиму роботи модельної системи генератора, створеного на базі логістичного відображення на основі виразу (2.4). Якісно це можна оцінити із форми «дивного атрактора» на фазовій площині (Рисунок 2.2,б-с).

В якості моделі хаотичного процесу можуть бути використані й інші нелінійні ДС, наприклад, система Лоренца, Реслера, Чена [24].

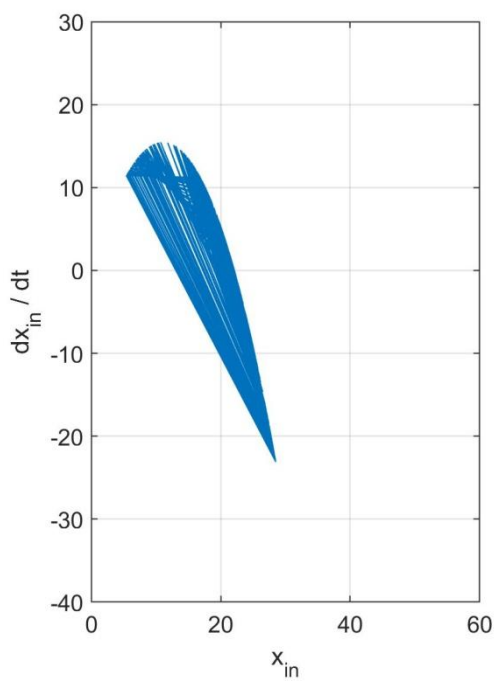
Моделювання таких систем базується на використанні чисельних методів розв'язку систем диференціальних рівнянь.

Візуальне середовище моделювання MATLAB/Simulink представляє собою альтернативне середовище, яке дозволяє візуально проектувати досліджувану НС у вигляді сигналів та блоків.

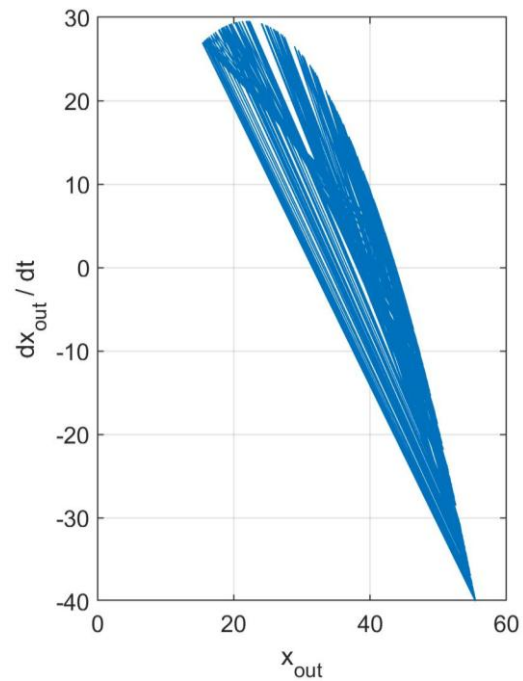
Запропонований підхід може бути застосований для інтелектуального моніторингу параметрів локальних мереж Інтернету речей у вигляді самостійного моніторингового програмного засобу, який працює на локальному сервері чи хмарному середовищі, або у складі існуючої системи в якості інтегрованого модуля.



(a)



(б)



(в)

Рисунок 2.2 – Результат моделювання: часові ряди (а), фазові портрети для вхідного (б) та вихідного (в) трафіку для сегменту мережі С

2.3 Аналітичний розв'язок логістичного рівняння

Логістичне рівняння, яке інколи також називається моделлю Верхульста або логістичною кривою зростання [25], - це модель зростання населення, вперше опублікована П'єром Верхульстом (1845, 1847). Дана модель є неперервною в часі, але також широко використовується модифікація з дискретним часом у вигляді квадратичного рекурентне рівняння, відоме як логістичне відображення (2.4).

Неперервна форма рівняння логістичної моделі описується диференціальним рівнянням:

$$\frac{dN}{dt} = \frac{rN(K - N)}{K}, \quad (2.5)$$

де r – параметр, який визначає швидкість максимального зростання популяції, K – пропускна здатність (максимальна стійка популяція).

Розділивши обидві частини рівняння на K та позначивши $x := N/K$, отримаємо логістичне рівняння у такому вигляді:

$$\frac{dx}{dt} = rx(1 - x) \quad (2.6)$$

Розв'язок рівняння (2.6) має вигляд:

$$x(t) = \frac{1}{1 + \left(\frac{1}{x_0} - 1\right)e^{-rt}} \quad (2.7)$$

Хоча зазвичай параметр r приймає додатні значення, на Рисунку 2.3 показані графіки розв'язку логістичного рівняння для різних додатних та від'ємних значень параметру r за початкових умов $x_0 = x(t=0)$ в діапазоні від 0.00 до 1.00 з кроком 0.05.

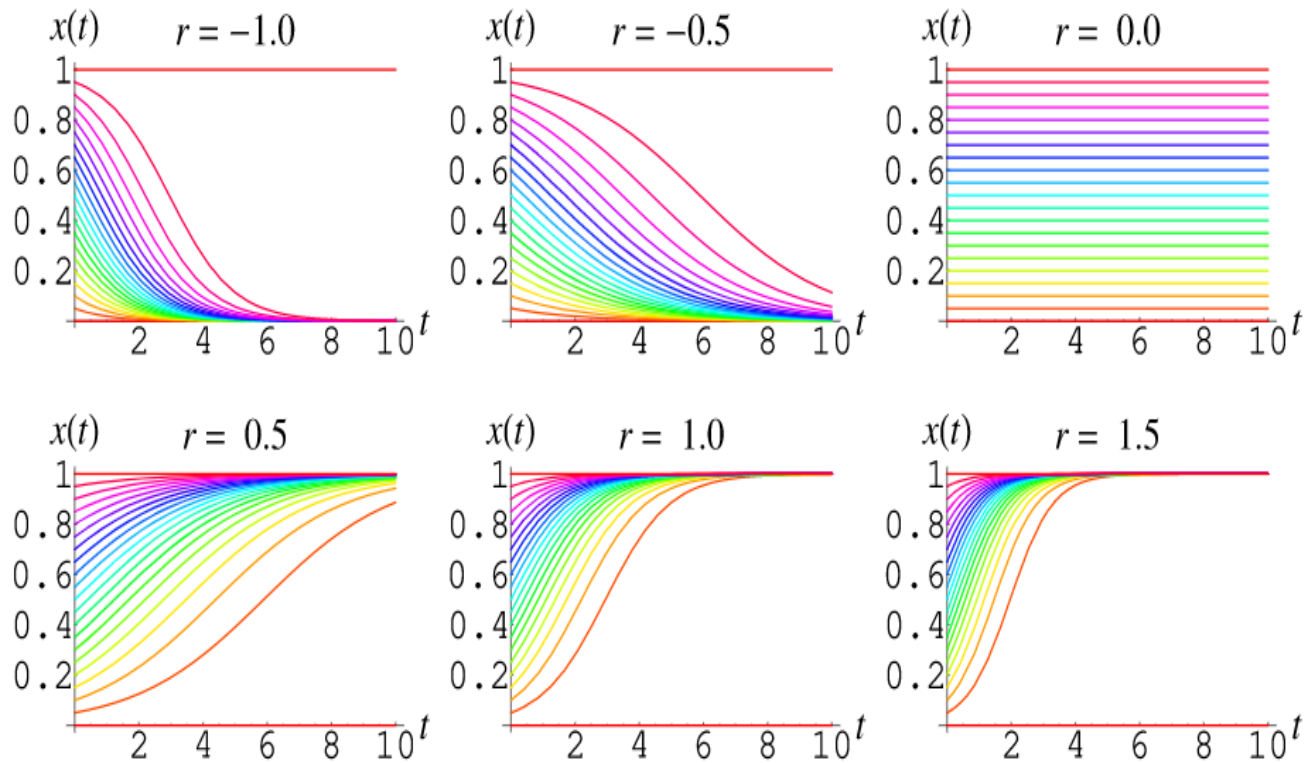


Рисунок 2.3 – Графіки розв'язку логістичного рівняння

Дискретна модель логістичного рівняння (2.6) відповідає логістичному відображенню (2.4).

Функції $x = \frac{a}{1 + bq^t}$, яка отримана з кривої (2.7), називається логістичної функцією.

Висновки до другого розділу

В другому розділі магістерської роботи було зроблено наступне:

1. Розроблено та описано метод фазової площини для аналізу фазових портретів, отриманих в результаті реконструкції динамічної системи на основі поліноміальної апроксимації часових рядів.
2. Запропонована модель нерівномірного завантаження каналу локальної IoT мережі, що викликана хаотичним режимом роботи та підключення кінцевих пристроїв, де в якості моделі динамічної системи використовується логістичне відображення.
3. Наведено аналітичних розв'язок неперервного логістичного рівняння, як більш загальної моделі деякого хаотичного процесу.

3 МОДЕЛЬ ЦИКЛУ МОНІТОРИНГУ ТА КЕРУВАННЯ ПАРАМЕТРІВ МЕРЕЖІ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ

3.1 Моніторинг та керування Інтернетом речей

Проведено аналіз загальних характеристик про архітектуру та компоненти циклу моніторингу та управління процесами Інтернету речей (згідно Microsoft) [26].

Відповідно до визначення, яке дає Microsoft [26], цикл керування і моніторингу IoT – механізм спостереження, що гарантує функціонування системи межах робочих порогових значень параметрів.

Структурна схема моделі циклу моніторингу та керування IoT показана на Рисунку 3.1.

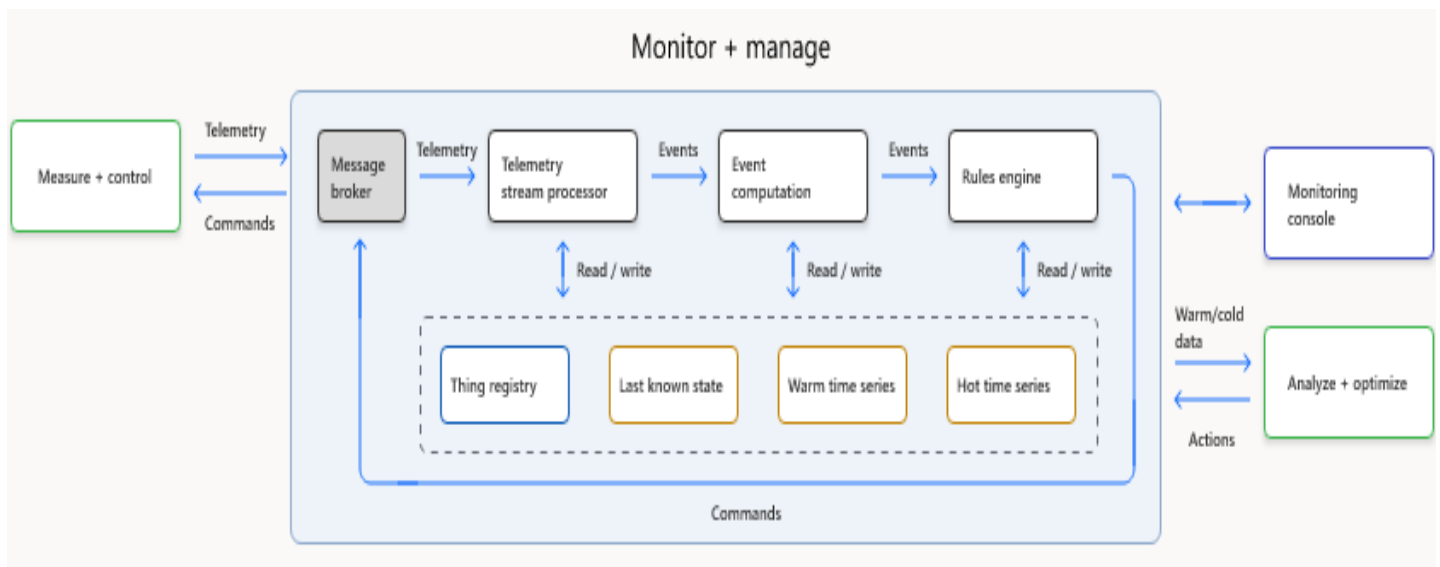


Рисунок 3.1 – Структурна схема моделі циклу моніторингом та керування IoT від Microsoft

Для досягнення цільового функціонального стану в даному прийнятному діапазоні, а також з метою його підтримки, декілька пристроїв у системі повинні працювати спільно.

З метою визначення поточного стану цикл управління і моніторингу відстежує і зіставляє тенденції сигналів активної телеметрії з декількох пристроїв. Логіка поєднує ці тенденції з "теплим" журналом часових рядів (нещодавно отриманих) і сигналами корпоративних систем для отримання нових цінних відомостей. Потім цикл моніторингу та управління надсилає аналітичні відомості через обробник правил генеруючи відповідні команди виконання.

Характеристики циклу моніторингу та керування:

1. Можливість віддаленого та близького розміщення по відношенню до фізичних пристроїв IoT. Об'єкти, які можуть керуватись лише віддалено в силу своїх експлуатаційних особливостей, наприклад нафто- і газопроводи, силові трансформатори, інтелектуальні дверні дзвінки, небезпечні середовища і засоби відстеження ресурсів, не можуть підтримувати цю інфраструктуру. Для таких систем цикли моніторингу та управління працюють з віддаленими об'єктами, такими як загальнодоступні або приватні хмари. У переробних галузях, таких як нафтоперероблення і виробництво хімічних продуктів, цикли моніторингу та управління можуть бути розгорнуті ближче до пристроїв фізичних пристроїв.

2. Цикли моніторингу і контролю напряму залежать від циклів вимірювання та контролю [26] на рівні пристроїв, що зв'язані із основними процесами моніторингу та керування.

3. Можливість інтеграції із іншими корпоративними системами, такими як планування ресурсів підприємства, керування взаємодією із клієнтами (CRM-системи), керування життєвим циклом продукту та системи з підтримки

контекстуальних операцій. Цикли моніторингу та управління не залежать від цих систем.

4. Цикли моніторингу та керування приймають потоки даних телеметрії від сенсорів та приймають участь у визначенні останнього відомого стану пристрою, отриманні активного «кешу» та «теплого» журналу часових рядів, а також у збиранні зведених даних.

5. Створення команд стеження на пристроях з метою отримання інформації про поточний стан, який відрізняється від нормального, з метою його виправлення.

6. Обчислення стану залежних пристроїв та надання стрічок подій зовнішнім пристроям.

7. Інтеграція з пристроями та корпоративними системами через мережеві протоколи HTTP, MQTT і AMQP.

8. Час циклу може становити декілька секунд в залежності від сценарію Інтернету речей. При використанні мережевих протоколів, які не враховують час, таких як MQTT, HTTP, AMQP може виникнути відхилення чи коливання затримки мережевих пакетів [26].

До складу циклів моніторингу та керування входять такі компоненти:

1. Брокери повідомлень, що опрацьовують дані телеметрії та посилають команди пристроям.

2. Реєстр пристроїв, або система запису та єдине джерело достовірних даних для усіх пристроїв IoT. Призначений для зберігання метаданих про пристрої та їхню взаємодію між собою. Обробник потоку даних телеметрії використовує відомості з реєстру для розуміння структури повідомлень телеметрії, а також для аналізу та виконання логіки потокової обробки. Брокер повідомлень користується реєстром для перевірки запитів на під'єднання пристроїв і ухвалення рішень про маршрутизацію повідомлень. Логіка обробки подій використовує метадані

сутності, щоб переконатися, що логіка вхідних і вихідних даних, а також логіка обробки відповідають структурним і семантичним відношенням і взаємодіям сутності.

3. Обробник потоку телеметричних даних отримує дані від системи телеметрії та на їхній основі визначає стан окремих пристроїв та набору пристроїв, а також виявляє помилки і відхилення в роботі. Цей модуль визначає стан помилки та відправляє скомпоновані чи не скомпоновані помилки відповідним обробниками помилок, а також в «гаряче» та «тепле» сховище для подальшої обробки та зберігання записів.

4. Журнал часових рядів – високошвидкісне сховище даних, що знаходиться в пам'яті або на віддаленому кеші. Це сховище надає інформацію щодо останній відомих метрик пристрою і набору точок даних для виявлення тенденцій у квазі-реальному часі.

5. «Теплий» журнал часових рядів зберігає точки дані за декілька тижнів, що допомагає у співставленні тенденцій, які були визначені в квазі-реальному часі, із довгостроковими тенденціями з метою визначення та виявлення можливих відхилень від очікуваного стану. Цикли моніторингу і управління також можуть використовувати індексоване сховище для попереднього обчислення тенденції.

6. Обчислення подій визначає події, які потребують реакції – оброблення. Для цього відомості про події обробника потоку об'єднуються із відомостями про останній відомий стан елементів системи, тенденціями, отриманими в квазі-реальному часі від «теплого» журналу часових рядів, а також дані самого журналу часових рядів.

7. Обробник правил використовує та обробляє бізнес-події, налаштовуючи потрібні стани пристрою за допомогою відповідних команд. Він також може публікувати події та повідомлення в консоль моніторингу.

8. Консоль моніторингу забезпечує візуальне відображення та втручання людини у разі необхідності.

До головних функцій циклу моніторингу та керування належать спостереження за фізичною системою та керування пристроями Інтернет речей, що підключені до мережі. Цикл гарантує що стан системи знаходиться в межах допустимих значень робочих параметрів та посилає команди керування пристроям.

Потенціальні приклади використання циклів моніторингу та керування [26]:

- інтелектуальне збирання сміття: вантажівка прямує за оптимальним маршрутом, який проходить через заповнені контейнери;

- інтелектуальне студентське містечко: у разі виявлення пожеж у кількох будівлях видаються сповіщення про евакуацію студентського містечка;

- розподіл електроживлення: за наявності прогнозу про сильний вітер і дощ завчасно вимикається живлення в кількох міських кварталах.

- стеження за газопроводом: газоперекачувальну станцію вимикають у разі виявлення падіння тиску в кількох сегментах віддаленого газопроводу.

- інтелектуальні вимірювальні прилади: дають змогу відстежувати енергоспоживання з урахуванням прогнозу погоди, щоб автоматично піднімати задане значення терморегулятора в будинках у рамках програми надання знижок ощадливим споживачам.

- вітрова електростанція: при виявленні падіння коефіцієнта потужності вітрової електростанції планується обстеження вітроустановок, які можуть бути несправними.

- переробні галузі: стеження та управління процесом крекінгу нафти на нафтопереробному заводі або виготовленням фарб і наливних хімічних продуктів.

- дискретне виробництво: стеження та контроль зони, де здійснюється перевірка та пакування дрібних виробів.

3.2 Цикли вимірювання, контролю, аналізу та оптимізації Інтернету речей

Модель циклу вимірювання і контролю [27] обмежується одним рівнем абстракції, що складається із пристроїв, сенсорів, виконавчих механізмів (актуаторів) та контролерів. Ці механізми можуть бути також інтегровані із циклами моніторингу і керування, а також із циклами аналізу та оптимізації [26-28].

Розглянемо загальні відомості про щодо архітектури, основних характеристик та компонентів цього типу циклу IoT, що описано в специфікації від Microsoft [27].

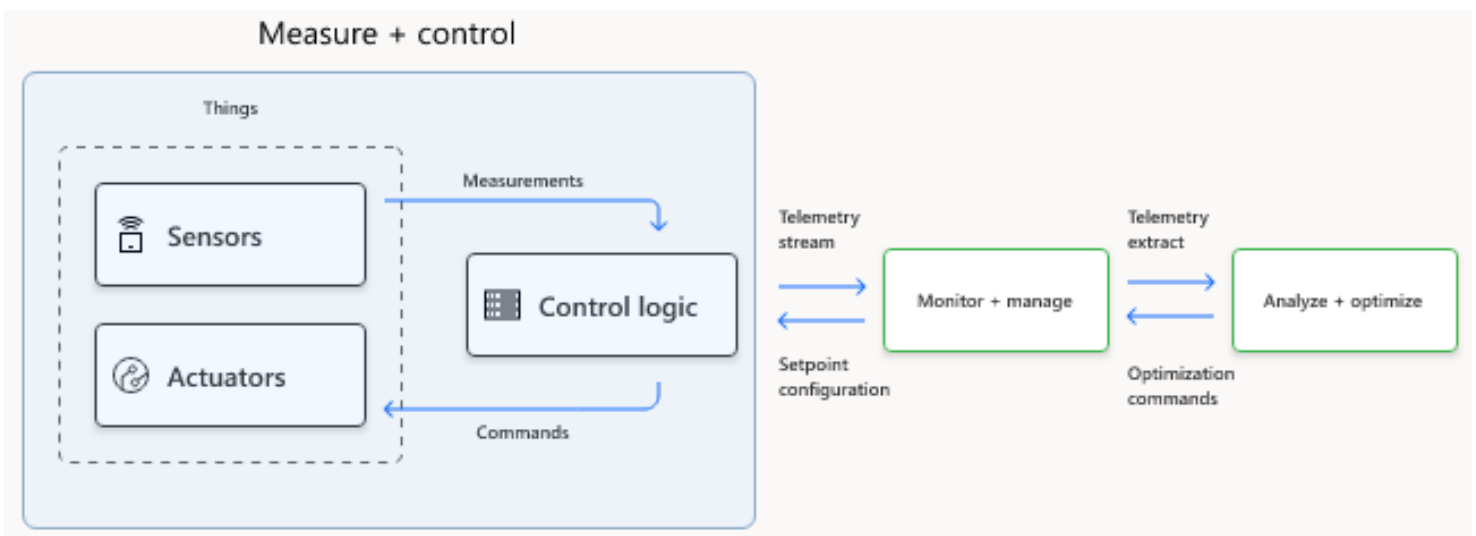


Рисунок 3.2 – Архітектура циклу вимірювання та контролю

Далі наведені основні характеристик циклу вимірювання та контролю відносяться:

- функції виконуються поблизу пристроїв або всередині них;

- використовує час циклу в залежності від сценарію Інтернету речей. Всередині мережі, де облік часу враховується, дане значення може досягати декількох мілісекунд [27];

- цикл приймає вхідні дані вимірювань з урахуванням конфігурації заданих значень, інформації про останні відомі вимірювання з датчиків і невеликого журналу часових рядів для кожного вимірювання;

- приймає команди з циклів моніторингу та управління, щоб коригувати конфігурацію заданих значень, і примусові команди, щоб керувати виконавчими компонентами;

- реалізовує виконання команд виконавчих компонентів, щоб підтримувати стан пристрою наближеним до конфігурації заданих значень;

- не залежить від будь-яких зовнішніх систем, щоб забезпечувати короткий час циклу та операційну автономність;

- може передавати дані телеметрії для використання в диспетчерських системах;

- використовує протоколи промислової мережі, таких як Modbus, RS485, EtherCAT і SERCOS, для передавання метрик із датчиків і команд для виконавчих компонентів;

- може інтегруватися з диспетчерськими системами, наприклад циклами моніторингу та управління, використовуючи протоколи Центру Інтернету речей, такі як HTTP, MQTT і AMQP.

Пристрій Інтернету речей складається з датчиків, виконавчих компонентів і контролера [27]. Усі три типи компонентів працюють спільно, щоб забезпечувати належну роботу пристрою в його середовищі. Прикладами таких пристроїв є вітрові турбіни, пральні машини, автономні системи і резервуари для зберігання газу [27]. До прикладів значень, що задаються, належать кількість обертів на

хвилину для вітряної турбіни, значення безпечної температури та тиску для газового резервуара, безпечна відстань автоматичного відкриття для інтелектуального дверного замка [27].

До основних компонентів циклу вимірювання та контролю відносяться [27]:

- сенсори;
- виконавчі компоненти;
- логіка контролю.

Сенсорні пристрої та вимірювальні перетворювачі постійно вимірюють поточні умови пристрою і повідомляють їх контролеру. Прикладами вимірювань можуть слугувати температура і тиск у резервуарі зберігання природного газу, температура і вологість, які контролює система розумного будинку, а також кількість обертів на хвилину ротора у вітряній турбіні і потужність виробництва енергії. Частота вибірки конкретного сенсора залежить від роботи пристрою. Для пристроїв з повільною зміною стану, наприклад для газових резервуарів великого об'єму, потрібна вибірка з низькою частотою, тоді як для пристроїв зі швидкою зміною стану, таких як вітрові турбіни, потрібна висока частота вибірки.

Виконавчі компоненти, або актуатори, – це фізичні складові, які змінюють стан пристрою. До прикладів таких пристроїв належить впускний клапан резервуара для природного газу, гальмівна система, що уповільнює обертання ротора у вітровій турбіні, або замок на входних дверях, який блокується, якщо власник поза домом. Контролер керує виконавчими компонентами на основі даних вимірювань із датчика та зовнішніх джерел. У деяких пристроях є тільки функції визначення і немає виконавчих компонентів, тому потрібно компенсувати відповідну частину циклу.

Логіка контролю, яка є основою алгоритму програми керування IoT, зберігає пристрій у необхідному діапазоні прийнятних значень. Поточний стан

обчислюється на основі даних вимірювань сенсорів. Якщо поточний стан відрізняється від цільового, контролер виконує коригувальні дії, надсилаючи команди виконавчим компонентам. Прикладами коригувальних дій можуть бути закриття клапана резервуара для природного газу, увімкнення системи обігріву в будинку або гальмування ротора у вітровій турбіні. За необхідності контролер також може передавати дані телеметрії та приймати зовнішні команди з циклу моніторингу та управління.

Цикл вимірювання та контролю в IoT має на меті підтримку роботи пристрою в діапазоні конфігурації прийнятних заданих значень за допомогою процесу контролю в реальному часі із замкнутим циклом. Пристрій може належати до більшої фізичної системи під керуванням програмного забезпечення з одним або кількома мережевими пристроями, які входять до його складу.

Для пристрою Інтернету речей, схильного до впливу зовнішніх подій, потрібен процес контролю із замкнутим циклом, щоб забезпечити його стан відповідно до конфігурації заданих значень. Логіка управління в циклі вимірювання та контролю передбачає відстеження пристрою на основі метрик датчиків і вживає коригувальні дії за допомогою виконавчого компонента.

Приклади можливих варіантів використання ідеально підходять для енергетики та засобів моніторингу навколишнього середовища. Приклади використання циклів вимірювання та контролю [27]:

- інтелектуальна мишоловка – закриття пастки активує подію, коли сенсори виявляють мишу;
- сенсори задимлення – активується розбризкування в разі виявлення диму кількома сенсорами одночасно;
- трансформатор напруги – вимикається напругу живлення пристрою, якщо прогнозується сильна гроза;

- монітор газопроводу – відкривається клапан, щоб компенсувати зниження тиску;

- домашній термостат – збільшується подача газу в системі обігріву, якщо виміряна температура нижча за задане значення;

- вітрова турбіна – здійснюється гальмування, щоб зменшити швидкість обертання ротора, якщо її значення наближається до попереджувальної порогової кількості обертів на хвилину;

- сонячні панелі – коригується кут положення панелі в міру зміни положення сонця відносно горизонту, щоб максимально збільшити виробництво енергії.

На Рисунку 3.3 зображено архітектуру системи аналізу та оптимізації [27].

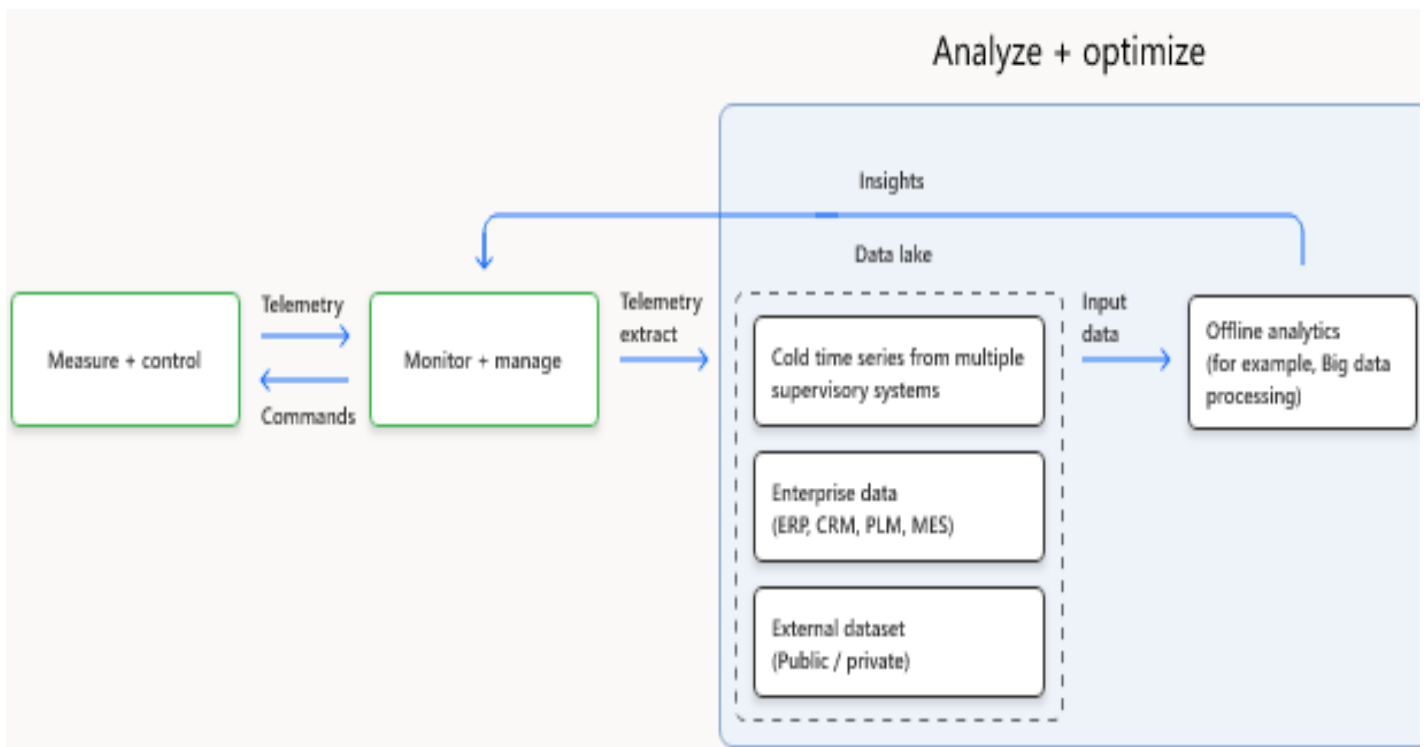


Рисунок 3.3 – Архітектура циклу аналізу та оптимізації

Цикл аналізу та оптимізації IoT [27, 28] дозволяє програмувати та застосовувати аналітичні алгоритми та засоби задля оптимізації діяльності системи щодо одного або декількох компонентів, що знаходяться під зовнішнім впливом програмного забезпечення в даному контексті. Даний цикл використовує в якості джерела вихідних даних сигнали телеметрії, достовірність та наявність яких гарантується системою моніторингу та управління, яка була розглянута вище, уточнюючи їх та формуючи базис даних для інтелектуального аналізу.

Відповідно до задач циклу аналізу та оптимізації [27] дані з різних джерел IoT, підприємств, приватних і загальнодоступних ресурсів надходять у хмарні сховища даних. В автономній аналітиці такі сховища даних використовуються для виявлення прихованих трендів і отримання аналітичних відомостей щодо оптимізації діяльності компанії. Аналітичні дані щодо оптимізації процесів автономної аналітики надходять назад до установок Інтернету речей через цикли відстеження та управління, а також цикли вимірювання та управління [26-28].

Цикл аналізу та оптимізації виконується асинхронно, через це жорсткі умови та терміни щодо аналізу даних або надсилання сигналів оптимізації на пристрої не встановлюють. Під час виконання пакетних завдань ці цикли залежать від журналу даних телеметрії та журналу корпоративних операційних даних.

До числа системних залежностей входить кілька систем передачі даних через сховище даних, зокрема системи Інтернету речей і канали корпоративних систем. Для інтеграції з системами контролю та іншими корпоративними системами цикл оптимізації в основному використовує протоколи веб-служб [28].

Основні компоненти системи керування та оптимізації [28]:

- банк даних – велике сховище, оптимізоване для зниження витрат на споживання протягом більш тривалих періодів часу. Приклад такого сховища даних – сховище HDFS у контексті обробки MapReduce [28]. Сховище даних

відкладає структуру даних на час обробки, тому добре підходить для зберігання як структурованих, так і неструктурованих даних;

- «холодні» дані часового ряду - необроблені або оброблені дані телеметрії, що є важливими для автономної аналітики та часто надходять з декількох систем Інтернету речей. Завдання аналітики продовжують уточнювати і об'єднують ці дані з корпоративними і зовнішніми наборами даних;

- корпоративні дані, створені корпоративними системами, наприклад, управління життєвим циклом продукту, ланцюжок поставок, фінанси, продажі, виробництво і розповсюдження, управління відносинами з клієнтами. Корпоративні дані, об'єднані із зовнішніми наборами даних, наприклад, погода, можуть поміщати дані телеметрії Інтернету речей у контекст сфери діяльності для створення несуперечливих аналітичних відомостей;

- автономна аналітика для обробки великих даних у пакетному режимі. Завдання Spark і обробка MapReduce Hadoop - це лише пара прикладів. Потім процеси циклу відстеження та управління і циклу вимірювання та контролю застосовують аналітичні дані, отримані від циклів аналізу та оптимізації на пристроях Інтернету речей.

3.3 Обробка подій в Інтернеті речей

В складі мереж пристрої IoT надсилають сповіщення про події події (сповіщення, підтвердження, дані телеметрії) у програму збору даних та аналітики. Програмні засоби можуть знадобитися певні підмножини подій для обробки або зберігання в різних кінцевих точках. Ці події також необхідно спрямувати в різні служби для подальшого опрацювання. Коли рішення IoT горизонтально масштабується, також змінюються кількість пристроїв, обсяг подій, різноманітність

подій і служб, що використовують рішення. Для реалізації цього шаблону потрібен гнучкий, масштабований і надійний метод маршрутизації подій [29].

Архітектура маршрутизації подій зображена на Рисунку 3.4 [29].

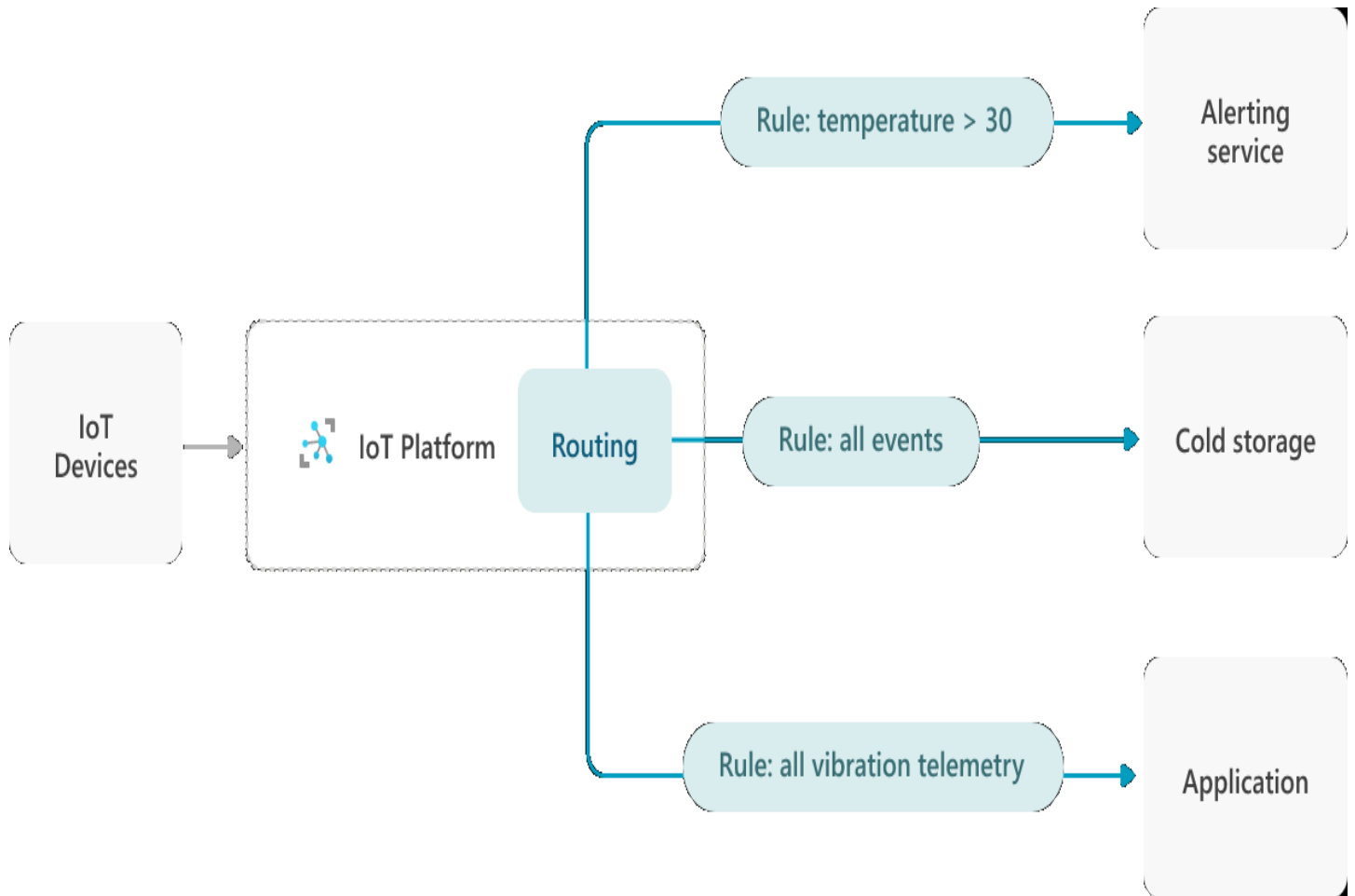


Рисунок 3.4 – Архітектура системи маршрутизації подій

На базі платформи IoT можна створювати набір правил для детальної маршрутизації подій. Архітектура, схема якої зображена на Рисунку 3.4, дозволяє

створення групи правил, які будуть застосовуватись до всіх вхідних подій в напрямку до цільового кінцевого пристрою.

Проілюструємо деякі можливі потенційні варіанти використання системи на прикладі системи роздрібного магазину, яка відстежує холодильники для заморожених продуктів відділу харчування [29]:

- коли температура в холодильниках падає нижче встановленого порогового значення, надсилається сигнал події. Можна створити правило маршрутизації з правилом порога для надсилання конкретних подій у систему сповіщень.

- команда обробки та аналізу даних створює модель виявлення аномалій, щоб виявляти проблеми з холодильниками, перш ніж вони зламаються. Правило маршрутизації повідомлень може надсилати всі необроблені дані телеметрії в спеціальний обліковий запис зберігання, щоб команда обробки та аналізу даних могла використовувати їх для навчання і моделювання.

Такий типовий сценарій може бути застосований до систем, пов'язаних з різними галузями промисловості та економіки, наприклад, галузей роздрібної торгівлі, енергетики, навколишнього середовища тощо [29].

Розглянемо деякі важливі характеристики при використанні даного архітектурного шаблону [29]:

1. Пропускна здатність кінцевих пристроїв. Кінцеві точки, які отримують події, повинні обробляти вхідний трафік подій, відправлених через маршрутизацію. Необхідно переконатися, що для служб кінцевих точок налаштована можливість приймати і зберігати дані до їх використання.

2. Формат подій. З метою забезпечення масштабованості та гнучкості маршрутизації, події повинні мати загальний формат для взаємодії між протоколами.

3. Обробка подій. Якщо подія відповідає кільком маршрутам, що вказують на одну кінцеву точку, вона має відправитися в цю точку тільки один раз. Також важливо забезпечити впорядкування повідомлень у таких ситуаціях.

4. Дублювання подій. Для обробки дублікатів повідомлень рекомендується використовувати унікальний ідентифікатор у властивостях програми в повідомленні в точці походження, яка зазвичай є пристроєм або модулем. Потім служба, що використовує повідомлення, може обробляти їхні дублікати за допомогою цього ідентифікатора.

6. Наявність резервного маршруту. Події, які не відповідають правилам, повинні поміщатися в резервний маршрут, щоб їх можна було правильно обробити без втрат.

7. Не пов'язані із телеметрією події. У застосуваннях IoT можуть мати місце різні типи подій, наприклад зміни стану пристрою або події його життєвого циклу. Маршрут подій повинен захоплювати і застосовувати правила до таких подій, щоб забезпечити автоматизацію і моніторинг.

Отже, згідно рекомендацій Microsoft [29], даний шаблон слід використовувати в таких випадках:

- під час надсилання повідомлень телеметрії пристроїв, подій їхнього життєвого циклу або зміни двійника пристрою в конкретні кінцеві точки, визначені правилами;

- під час фільтрації подій шляхом застосування конкретних правил.

В той же час, таке рішення не рекомендується у разі маршрутизації на основі складного аналізу даних часового ряду в режимі реального часу. Наприклад, під час порівняння середніх даних телеметрії за 15 хвилин. Якщо потрібен аналіз даних у режимі реального часу, використовуйте службу аналітики в режимі реального часу для даних критичного шляху [29].

Висновки до третього розділу

В третьому розділі магістерської роботи було зроблено наступне:

1. Проведено аналіз погромних циклів від Microsoft щодо моніторингу і керування Інтернетом речей. Розглянуто потенційні варіанти використання цього шаблону та відповідні рекомендації.
2. Проведено аналіз структури та шаблонів циклів вимірювання, контролю, аналізу та оптимізації Інтернету речей.
3. Розглянуто шаблон обробки подів в Інтернеті речей, його основні функції та архітектуру. Проаналізовано рекомендації Microsoft щодо доцільності застосування цього рішення при проектуванні IoT-системи.

4 ВСТАНОВЛЕННЯ ТА НАЛАШТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ WEB-ОРІЄНТОВАНОГО ІНТЕРФЕЙСУ КОРИСТУВАЧА – АДМІНІСТРАТОРА ЛОКАЛЬНОЇ МЕРЕЖІ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ

4.1 Встановлення та налаштування програмного засобу Virtual Box

Традиційного для встановлення та тестування деякої операційної системи (ОС) потрібно встановити її на фізичну машину. З метою тестування та оцінки якості того чи іншого програмного забезпечення (ПЗ), яке не може бути встановлене основну робочу ОС, альтернативним рішенням є використання засобів віртуалізації, таких як наприклад Oracle VirtualBox [30]. Це дозволить уникнути необхідності щодо переналаштування програмного забезпечення на основній фізичній машині. Програма VirtualBox призначений для запуску віртуальних машин на основній фізичній машині без повторного встановлення ОС, під управлінням якої працює фізична машина. Також перевагою VirtualBox є те, що цей продукт можна встановити безкоштовно.

Віртуальна машина (ВМ) працює так само, як і фізична. ОС і додатки, встановлені всередині ВМ, функціонують аналогічно тому, як би вони працювали на реальній фізичній машині, оскільки для запуску ВМ на VirtualBox використовується емульоване апаратне забезпечення. Віртуальні машини ізольовані одна від одної та від хостової операційної системи. Таким чином, користувач може виконувати свої тести в ізольованих віртуальних машинах без будь-яких ризиків пошкодити ОС хоста або інші ВМ.

ПЗ VirtualBox підтримує широкий набір хостових і гостьових ОС. Хостовою ОС називається така ОС, що встановлена на фізичній машині, на якій в свою чергу

встановлений VirtualBox. Гостьова ОС - це така ОС, що встановлена на ВМ, яка працює всередині VirtualBox.

ПЗ VirtualBox можна встановити на Windows, Linux, macOS, Solaris і FreeBSD. На VirtualBox можна запускати ВМ з Windows, Linux, macOS, Solaris, FreeBSD, Novell Netware та іншими операційними системами [30,31].

Процедура встановлення ПЗ VirtualBox не відрізняється особливим рівнем складності і може бути виконана на ОС Windows або Linux в якості хостової ОС.

Для повноцінного функціонування необхідно увімкнути функції апаратної віртуалізації CPU (Central Processor Unit) фізичної машини, наприклад, Intel VT-X або AMD-V в UEFI/BIOS фізичного комп'ютера [30]. У разі, якщо процесор фізичної машини не підтримує жодну з цих функцій, в якості гостьових ОС можуть бути встановлені лише 32-бітні версії ОС.

Варто зауважити, ПЗ Hyper-V від Microsoft може блокувати розширення апаратної віртуалізації, які використовує VirtualBox для запуску віртуальних машин, тому рекомендується видалити Hyper-V для попередження можливих помилок [30].

На сьогодні переважна більшість сучасних процесорів підтримують апаратну віртуалізації [31].

Інсталяційний файл Virtual Box може бути завантажений із офіційного сайту, на якому також розміщена офіційна документація до даного програмного продукту. Доступні версії для хостових ОС Windows, Linux, Mac та інших ОС [31].

На Рисунку 4.1 зображено головне вікно інтерфейсу користувача ПЗ Virtual Box.

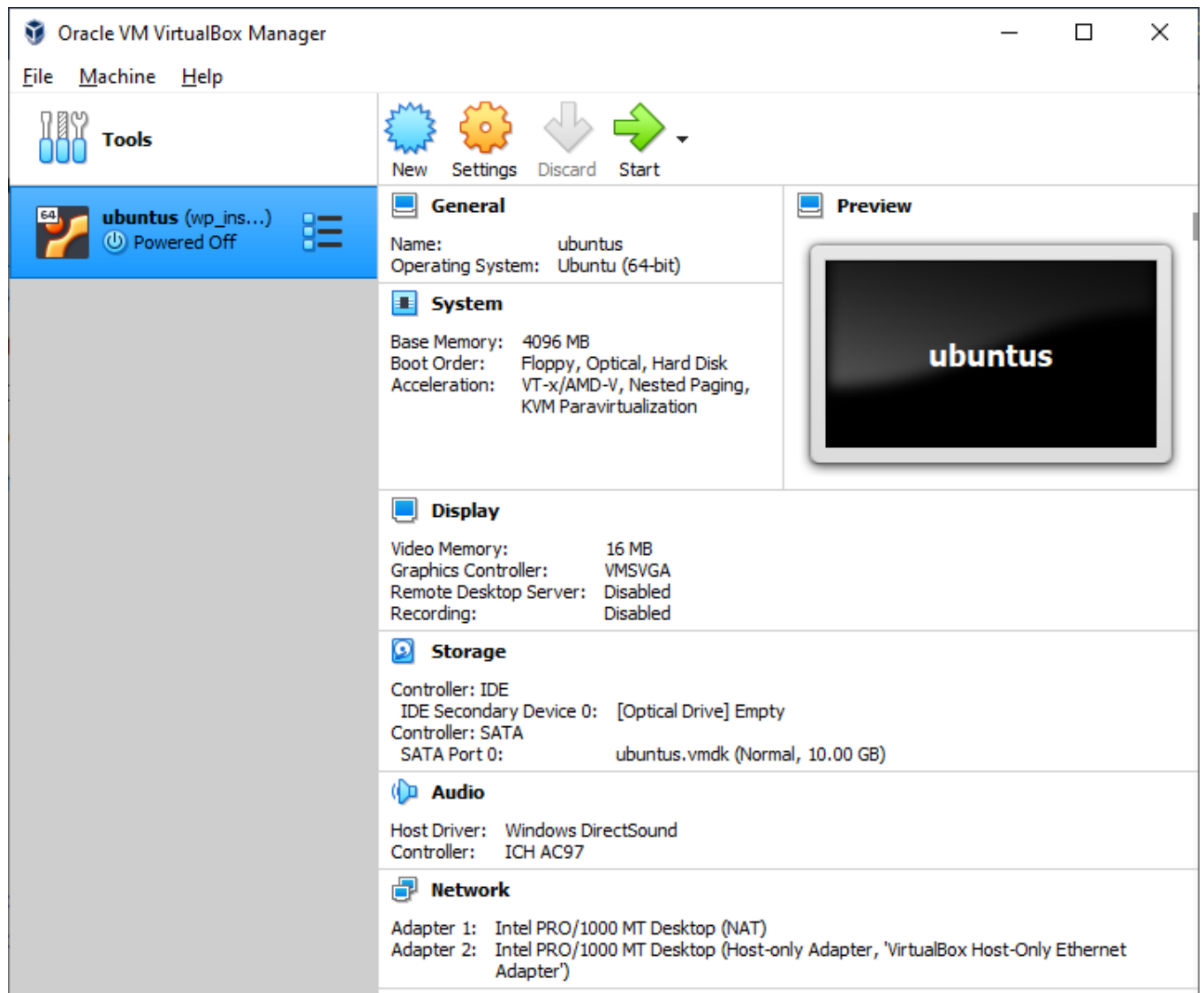


Рисунок 4.1 – Головне вікно Virtual Box

Процес інсталяції ПЗ VirtualBox для ОС Windows складається з таких кроків:

1. Запуск інсталятор VirtualBox – супроводжується запуском програми майстра установки, що має GUI (графічний інтерфейс користувача).
2. Вибір способу установки функцій за допомогою відповідного каталогу вибору компонентів до встановлення – можна залишити значення за

замовчуванням. Вибір опцій щодо створення ярликів доступу на робочому столі та меню «Пуск», а також створення асоціації файлів дисків віртуальних машин.

3. Підтвердження установки мережевих інтерфейсів VirtualBox.

4. Запуск процесу встановлення.

Після завершення установки можна встановити прапорець для запуску VirtualBox після установки.

4.2 Встановлення та налаштування гостьової машини на Ubuntu Server

По завершенню встановлення ПЗ VirtualBox, засобами графічного інтерфейсу користувача можна перейти до меню створення VM.

В якості сервера-контролера для системи моніторингу IoT-мережі буде використано ОС Ubuntu (серверну версію), інсталяційний .iso-образ якої можна завантажити із офіційного сайту [32].

Основні параметри віртуальної машини, на базі якої буде розгорнуто цільовий сервер, наведені в Таблиці 4.1.

Таблиці 4.1 – основні характеристики віртуальної машини

Характеристика	Значення
Версія ОС	Ubuntu Server 20.04
Обсяг ОЗП	4 Гб
Кількість логічних процесорів	4
Обсяг дискового простору	16 Гб
Кількість мережевих адаптерів	2 (Nat + VB HostOnly network)
Формат файлу диску	.vmdk (VM Ware – сумісний)

На Рисунку 4.2 Показано діалогове вікно програми-майстра для створення нової ВМ засобами ПЗ VirtualBox.

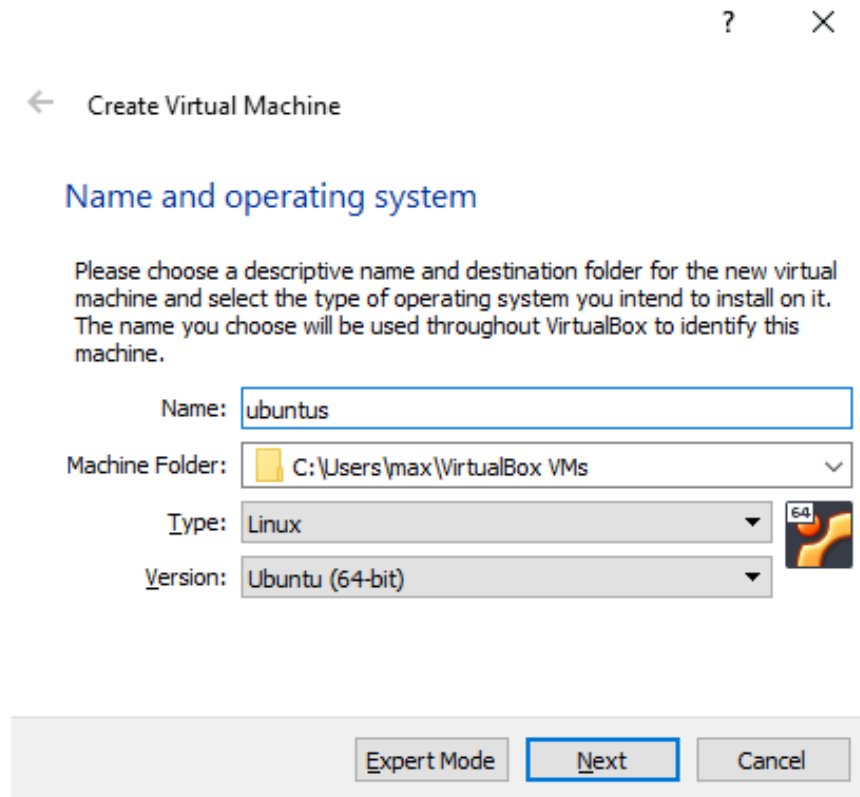


Рисунок 4.2 – Створення нової віртуальної машини

Серверна ВМ під управлінням ОС Ubuntu Server 20.04 була успішно створена в середовищі Virtual Box. Технічні параметри ВМ відповідають тим, що були заявлені в Таблиці 4.1, – це мінімально необхідна конфігурація, яка дозволить ефективно тестувати і впроваджувати розроблювану IoT систему.

На Рисунку 4.3 зображено вікно налаштування параметрів системи ВМ, а саме таких параметрів як обсяг оперативної пам'яті (ОЗП), кількість логічних процесорів.

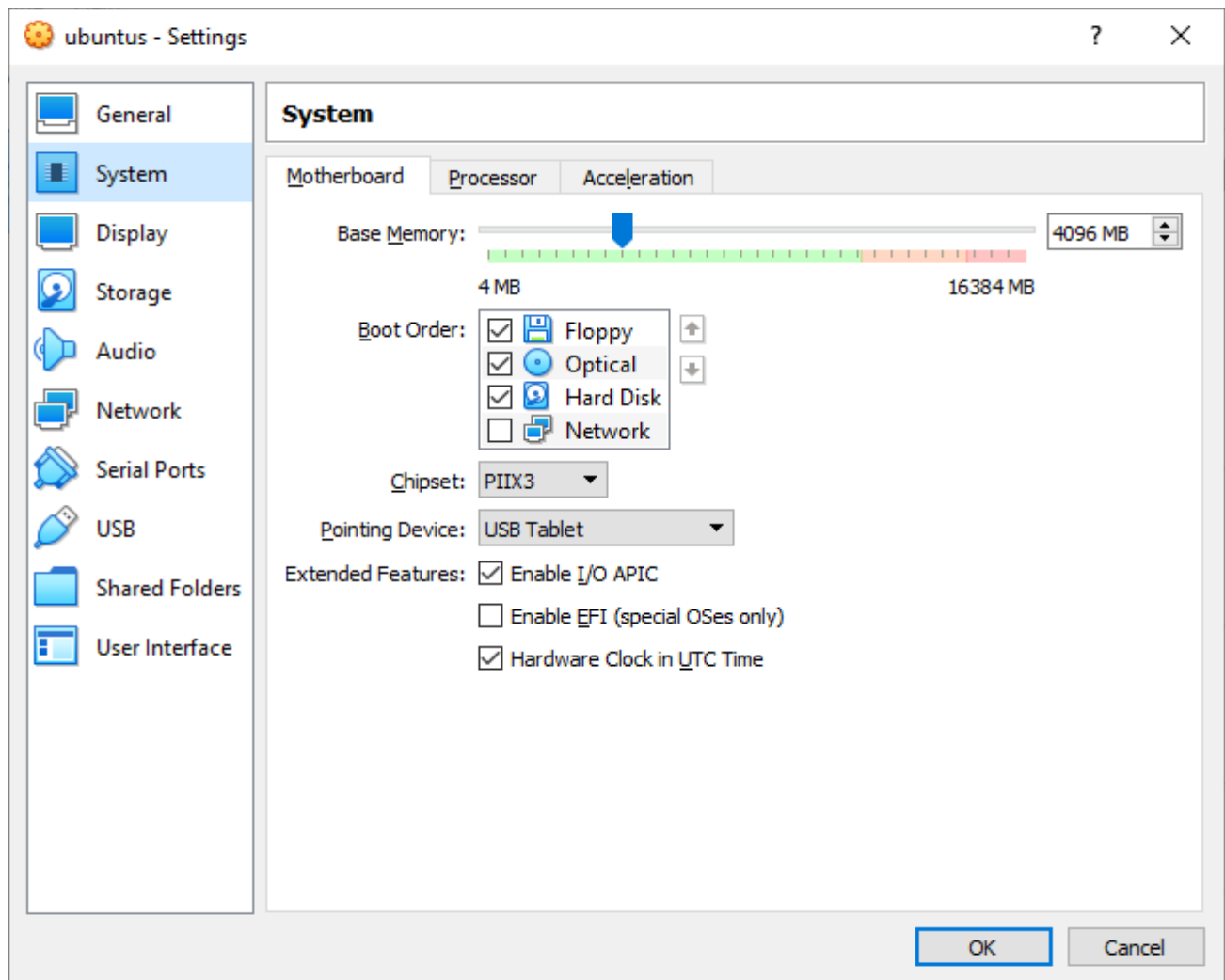


Рисунок 4.3 – Вікно системних налаштувань VM

Для забезпечення мережевого підключення до VM із хостової Windows-системи, на віртуальній машині було налаштовано додатковий мережевий адаптер – VM HostOnly Network [32], який забезпечує організацію віртуальної локальної мережі, яка дозволить помістити хостову та гостьові ОС в єдиний сегмент мережі.

На Рисунок 4.4 зображено вікно налаштувань мережевих інтерфейсів VM.

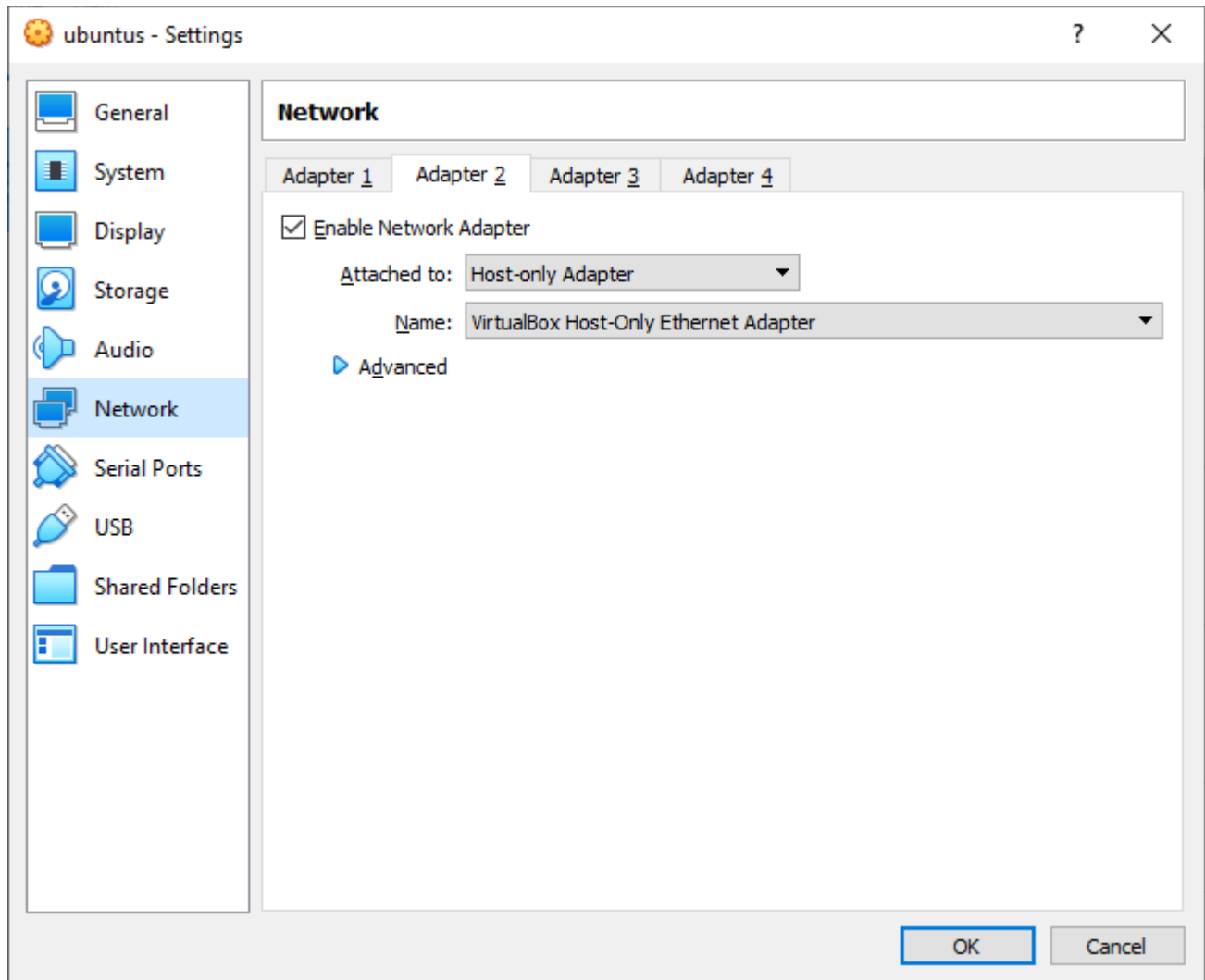


Рисунок 4.4 – Вікно налаштування мережевих інтерфейсів VM

Перед стартом системи, для неї необхідно виділити місце під файлову систему та Boot. Для забезпечення сумісності із серверними програмами віртуалізації, такими як VM Wave ESXI, було обрано відповідний формат файлового образу диску - .vmdk.

На рисунку 4.5 показано вікно вибору формату диску майстра встановлення та налаштування VM.

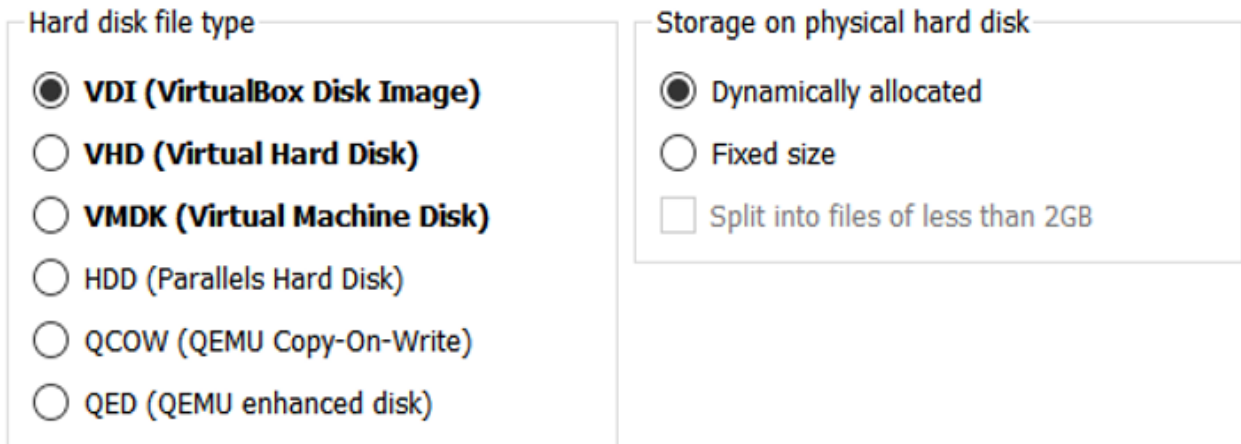


Рисунок 4.5 – Налаштування диску для системи

Для коректної роботи та забезпечення стабільності функціонування ВМ, було виділено 15 Гб дискового простору, якого вистачить для встановлення системи та усього необхідного ПЗ для роботи.

На Рисунку 4.6 зображено вікно налаштувань віртуальної файлової системи створеної ВМ.

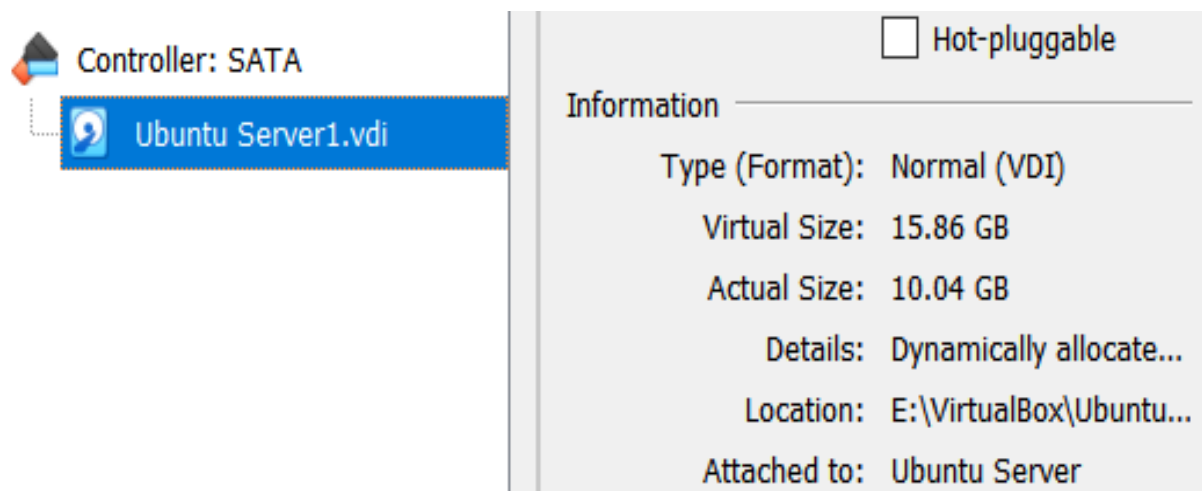
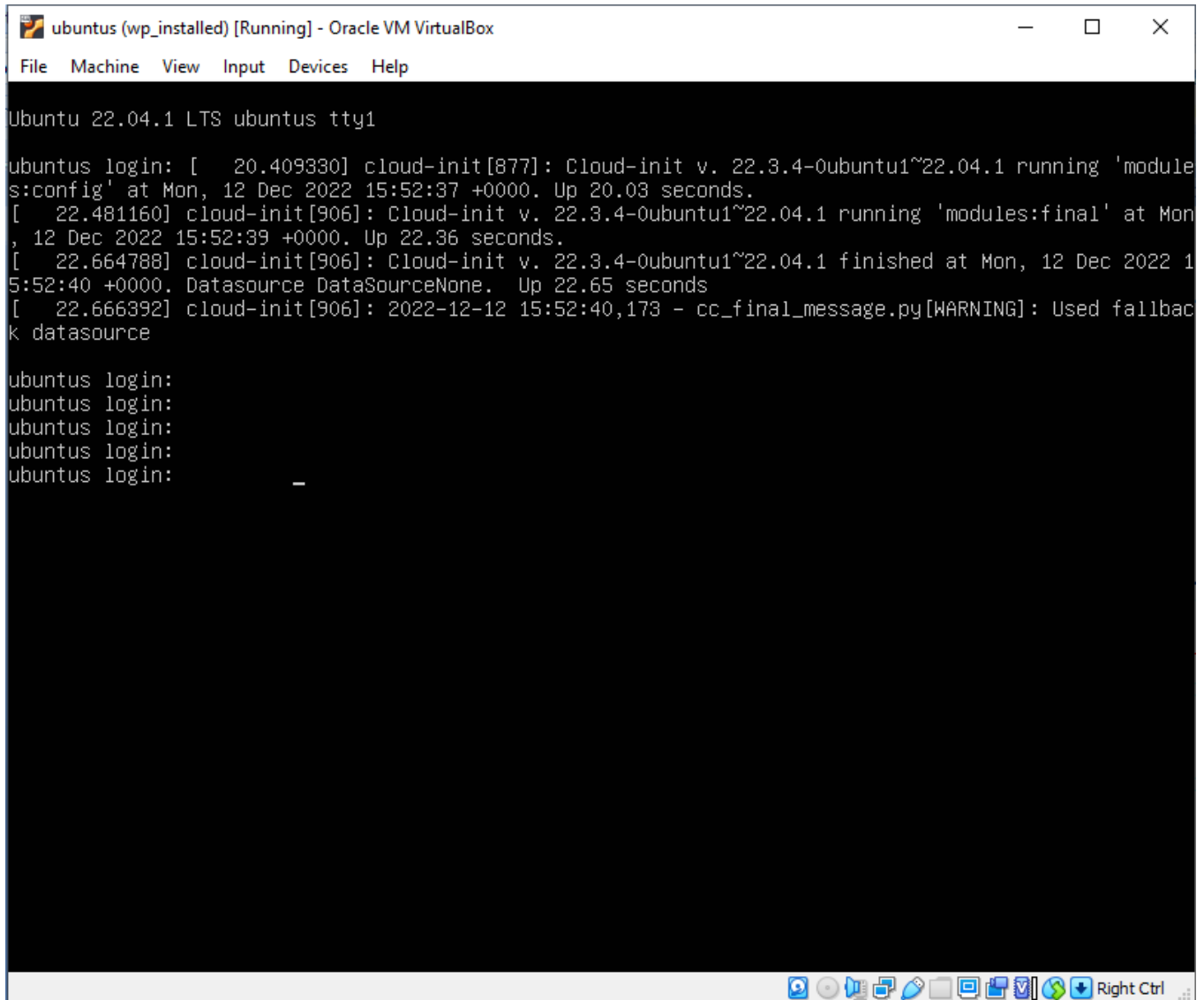


Рисунок 4.6 – Налаштування віртуальна файлова система

Встановлена та успішно запущена ВМ під управлінням Ubuntu Server показана на скріншоті 4.7.



```
ubuntu (wp_installed) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Ubuntu 22.04.1 LTS ubuntu tty1
ubuntu login: [ 20.409330] cloud-init[877]: Cloud-init v. 22.3.4-0ubuntu1~22.04.1 running 'modules:config' at Mon, 12 Dec 2022 15:52:37 +0000. Up 20.03 seconds.
[ 22.481160] cloud-init[906]: Cloud-init v. 22.3.4-0ubuntu1~22.04.1 running 'modules:final' at Mon, 12 Dec 2022 15:52:39 +0000. Up 22.36 seconds.
[ 22.664788] cloud-init[906]: Cloud-init v. 22.3.4-0ubuntu1~22.04.1 finished at Mon, 12 Dec 2022 15:52:40 +0000. Datasource DataSourceNone. Up 22.65 seconds
[ 22.666392] cloud-init[906]: 2022-12-12 15:52:40,173 - cc_final_message.py[WARNING]: Used fallback datasource
ubuntu login:
ubuntu login:
ubuntu login:
ubuntu login:
ubuntu login:
-
```

Рисунок 4.7 – Запущена ВМ на Ubuntu Server

Для серверних рішень встановлення оточення робочого столу не є обов'язковим та потребує виділення додаткових ресурсів, тому в даному випадку було обмежено інтерфейсом командного рядка.

Реалізація віддаленого підключення до серверу можлива за допомогою SSH-клієнту, наприклад програми Putty для Windows, запущеної на машині, що знаходиться в одному сегменті локальної IoT мережі (в даному випадку на хостовій ОС Windows 10).

Приклад підключення через протокол SSH показано на Рисунку 4.8.

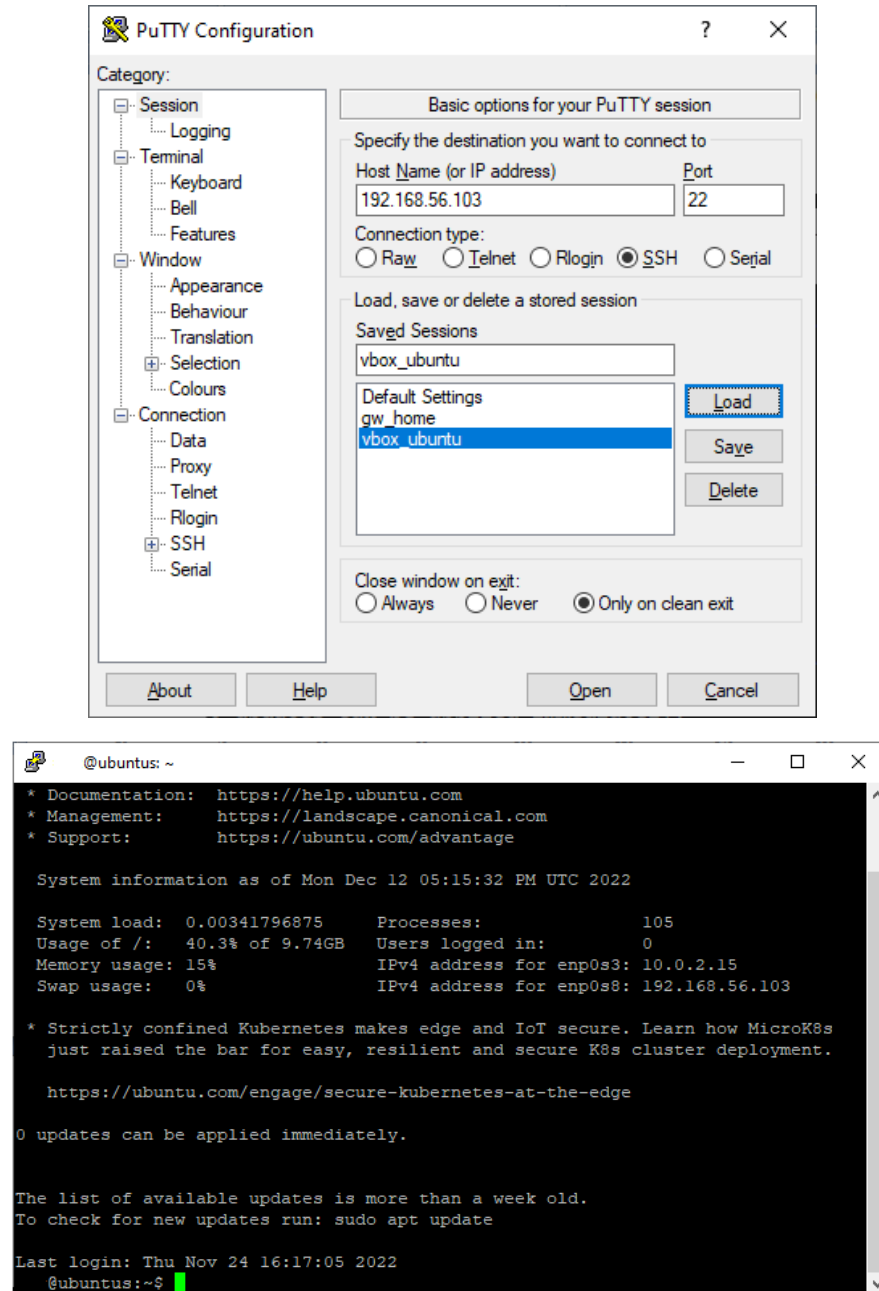


Рисунок 4.8 – Віддалене підключення по SSH

4.3 Встановлення системи управління Web-контентом WordPress та розширення SCADA MQTT

Система управління Web-контентом WordPress є потужним рішенням для розробки web-сайтів та web-додатків різного призначення. Програмна платформа WordPress підтримує широкий набір розширень (плагінів), в тому числі таких, що дозволяють використовувати її для відображення даних сенсорів для Інтернету речей [33].

Програмне забезпечення WordPress основане на стеку LAMP (Linux, Apache, MySQL, PHP), який є популярним набором технологій в індустрії web-розробки [33].

Процедура встановлення WordPress була виконана на VM Ubuntu 20.04, опис процесу встановлення та налаштування якої було наведено в попередньому підрозділі.

Процес встановлення складається із таких етапів:

- 1) авторизація користувача із правами sudo;
- 2) налаштувати стеку LAMP;
- 3) встановлення SSL-сертифікату;
- 4) завантаження, розпакування та налаштування WordPress;
- 5) авторизація в панелі користувача та встановлення базових розширень;
- 6) встановлення плагінів для роботи з IoT.

Встановлення основних залежностей виконується за допомогою такої команди:

```
sudo apt update  
sudo apt install apache2 \
```

```
ghostscript \  
libapache2-mod-php \  
mysql-server \  
php \  
php-bcmath \  
php-curl \  
php-imagick \  
php-intl \  
php-json \  
php-mbstring \  
php-mysql \  
php-xml \  
php-zip
```

В результаті виконання цієї команди буде встановлено HTTP-сервер Apache та інтерпретатор PHP.

Для встановлення останньої версії ПЗ WordPress її необхідно завантажити із офіційного сайту [34]. Такий підхід є бажаним, оскільки забезпечує встановлення останньої версії програми.

Для цього необхідно створити директорію та завантажити файли за допомогою таких команд:

```
sudo mkdir -p /srv/www  
sudo chown www-data: /srv/www  
curl https://wordpress.org/latest.tar.gz | sudo -u www-data tar zx -  
C /srv/www
```

Далі необхідно провести налаштування Apache та бази даних MySQL. Відповідні конфігураційні файли знаходяться в Додатку А.

Після успішного встановлення та налаштування стає доступною панель користувача.

Перед першим входом необхідно створити надійний пароль (Рисунок 4.9).

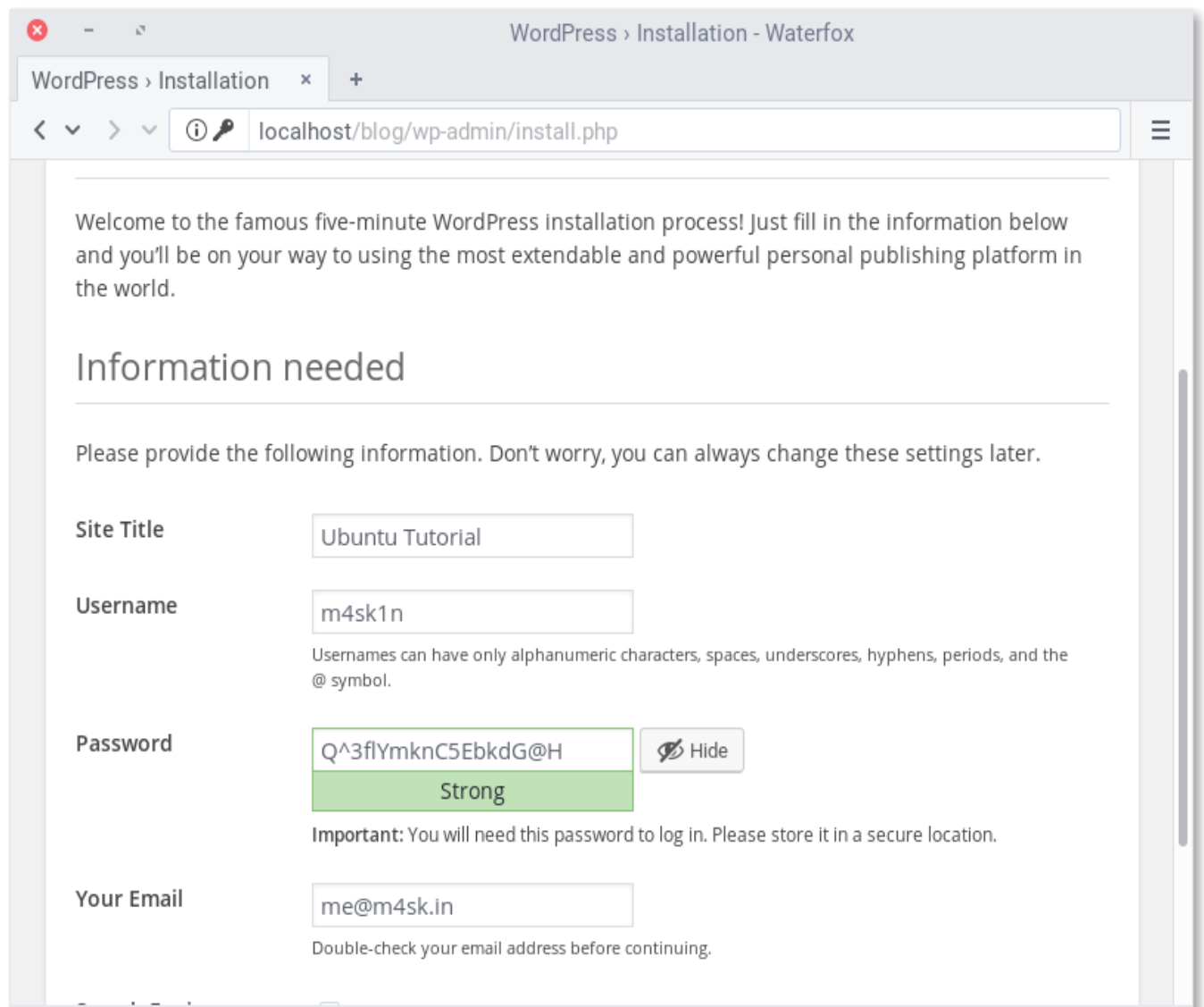


Рисунок 4.9 – Створення паролю для панелі адміністрування WordPress

На Рисунку 4.10 зображено інтерфейс панелі адміністрування WordPress.

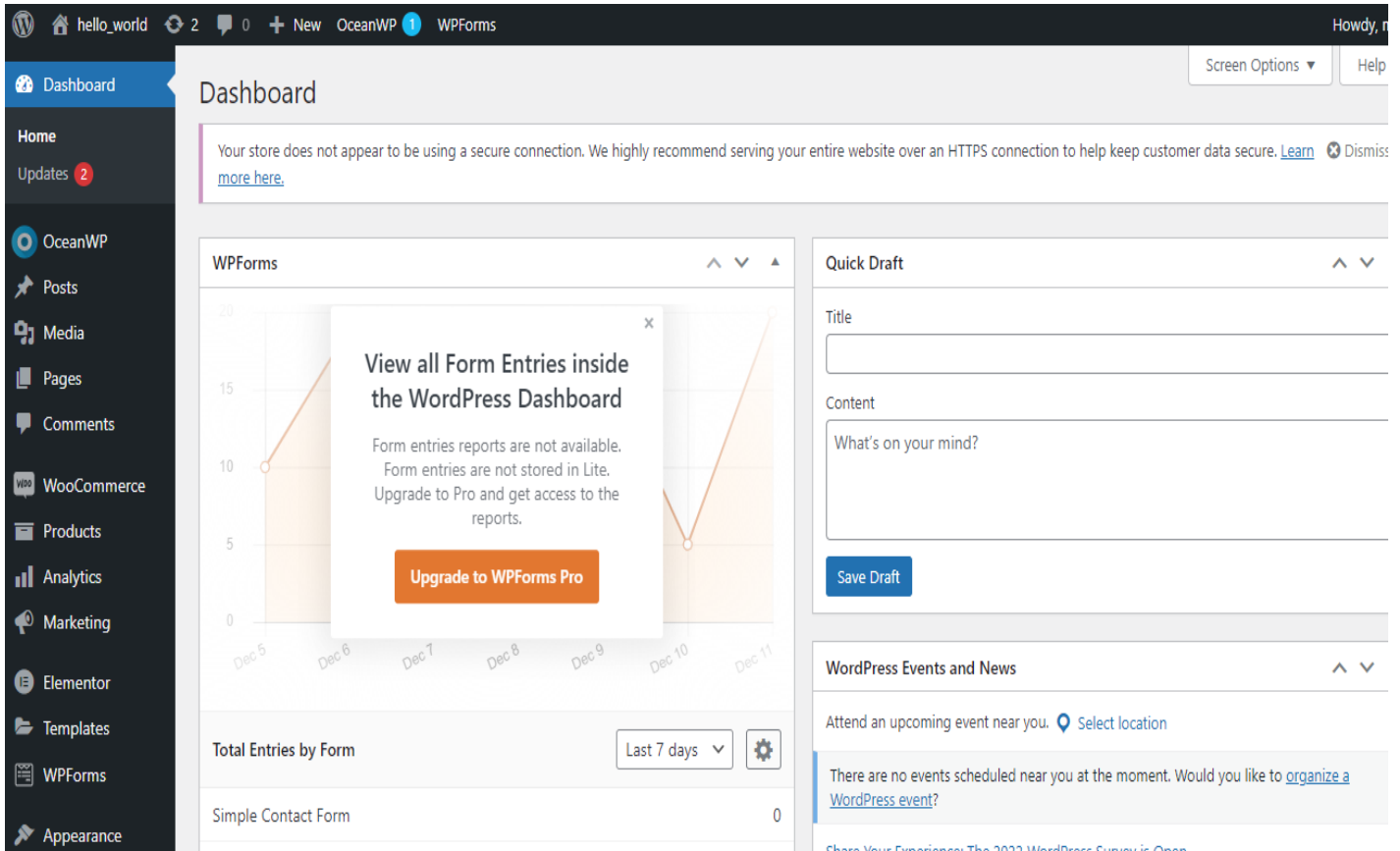


Рисунок 4.10 – Головна сторінка панелі адміністрування WordPress

Для реалізації взаємодії системи WordPress з Інтернетом речей, існують ряд розширень плагінів, кожен із яких виконує свій набір функцій.

В проєкті було використано:

- WP-MQTT [35];
- DIOT SCADA with MQTT [36];
- AnyViz [37];
- LED-SITE-INDICATOR [38].

Розширення WP-MQTT – це плагін WordPress, який використовує протокол MQTT для між машинної (M2M) комунікації в Інтернеті речей. Даний протокол є надзвичайно легким транспортним протоколом для публікації / підписки на повідомлення [36].

Розширення WP-MQTT визначає заздалегідь ряд подій – достатньо лише поставити відповідну галочку та заповнити тему і текст повідомлення:

- перегляд сторінки
- Вхід користувача
- невдалий вхід користувача
- опубліковано повідомлення
- сторінка опублікована
- новий коментар

Інші події можна додати за допомогою розділу «користувацькі події» Це дозволяє використовувати більш гнучко налаштувати дане розширення.

На Рисунку 4.11 зображено інтерфейс налаштування розширення WP-MQTT.

Розширення DIOT (що розшифровується як Decoupled IoT), має свої функції SCADA, розділені на хост і вузол для гнучкості і масштабованості, які відповідають концепції IoT. Цей плагін функціонує як хост SCADA для роботи з Інтернет пристроєм або системою, що розглядається як вузол SCADA. Також працює за протоколом IoT – MQTT. Користувачу необхідно ввести дані брокера/сервера MQTT в конфігурацію та підписатися на потрібну тему для відображення на будь-якій потрібній веб-сторінці або повідомленні [36].

На Рисунку 4.12 показано приклад використання плагіну DIOT SCADA для відображення деякого часового ряду вимірювальної величини в реальному часі.

Help ▾

WP-MQTT settings

Broker settings

Please provide information about the MQTT broker you want to connect to:

Broker URL

Broker Port

QoS

Client ID

Username

Password

Common events

Select common WordPress events that should trigger MQTT messages

Pageview

User login

Failed user login

Post published

Page published

New comment

Custom events

Trigger MQTT message using WordPress hooks. Please don't enable this unless you're familiar with WordPress's filter/action hook system.

Enable custom events I know what I'm doing, enable custom events

Custom events

<input type="checkbox"/>	<input type="text" value="heartbeat_tick"/>	<input type="text" value="roytanck/heartbeat"/>	<input type="text" value="tick"/>	<input type="button" value="✕"/>
<input checked="" type="checkbox"/>	<input type="text" value="spam_comment"/>	<input type="text" value="roytanck/comment"/>	<input type="text" value="A comment was marked as spam."/>	<input type="button" value="✕"/>

Рисунок 4.11 – Інтерфейс налаштування розширення WP-MQTT

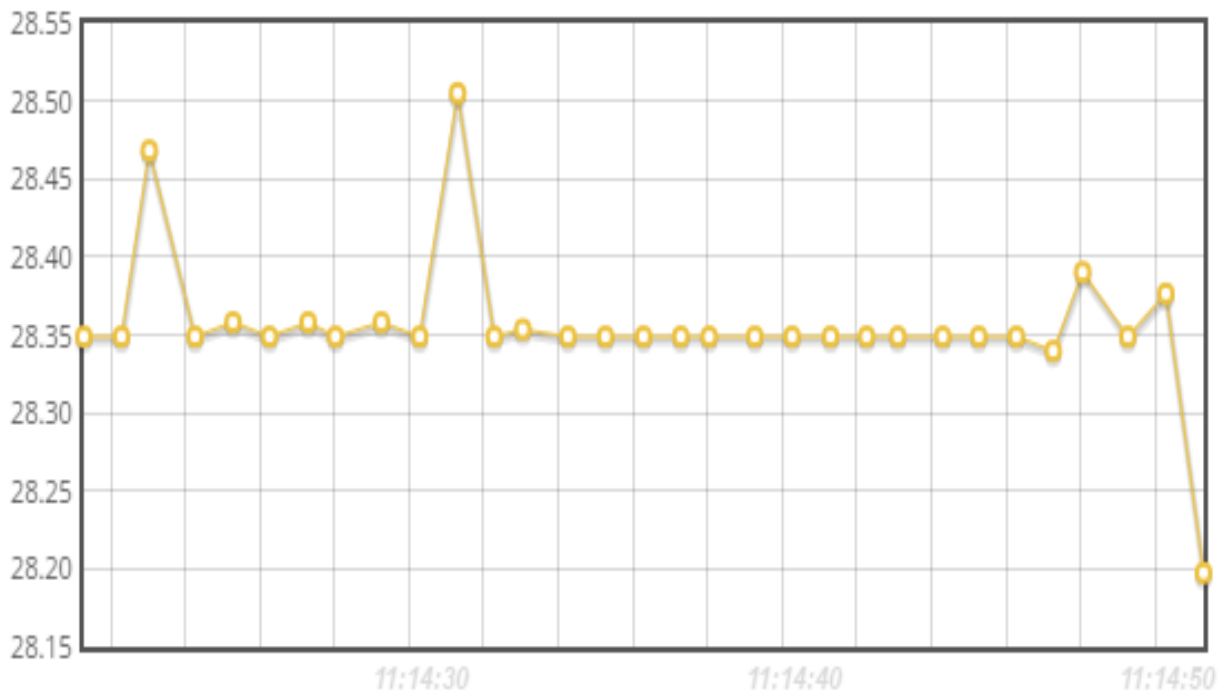


Рисунок 4.12 – Відображення часового ряду в реальному часі за допомогою плагіну
DIOT SCADA

Плагін AnyViz – це хмарне рішення, робота якого основана на підключенні до хмари AnyViz через REST API. Для роботи спочатку необхідно створити токен REST API для проекту AnyViz і ввести його в налаштуваннях плагіна [37].

Архітектура запропонованої системи моніторингу параметрів пристроїв Інтернету речей передбачає наявність централізованого сервера збору та візуалізації даних, який виконуватиме такі функції:

- збір та візуалізації даних, отриманих від пристроїв (речей), які оснащені сенсорами та виконують функції вимірювання та контролю фізичних величин;
- моніторинг параметрів локальної мережі пристроїв Інтернету речей з метою контролю мережевого трафіку та моніторингу підключення усіх пристроїв в реальному часі;
- забезпечення Web-орієнтованого інтерфейсу для користувачів та системних адміністраторів із реалізацією функції розмежування доступу;
- резервне копіювання бази даних та конфігураційних файлів системи.

Висновки до четвертого розділу

В четвертому розділі магістерської роботи було зроблено наступне:

1. Встановлено та налаштовано середовище віртуалізації VirtualBox, на базі якого було розгорнуто віртуальну машину під управлінням ОС Ubuntu Server.
2. Встановлено та налаштовано сервер для керування та моніторингу IoT мережею на базі ОС Ubuntu Server.
3. Розроблено Web-інтерфейс на базі системи WordPress та наботу розширень для Інтернету речей.

В якості серверної операційної системи було обрано Ubuntu Server версії 20.04. На слайді наведені деякі скріншоти процесу встановлення та налаштування цієї системи, а також HTTP-сервера Apache2. Робота проводилась в середовищі віртуалізації VirtualBox.

Система WordPress – це програмне забезпечення, яке автоматизує розробку Web-орієнтованих додатків та інтерфейсів користувача складних програмних систем для Linux-подібних операційних систем. Для роботи цієї програми

необхідно додатково встановити сервер баз даних MySQL та інтерпретатор PHP. На слайді наведені деякі скріншоти процесу встановлення програми WordPress та її компонентів на Ubuntu Linux.

ЗАГАЛЬНІ ВИСНОВКИ ДО РОБОТИ

1. В роботі проведено аналіз сучасної технології Інтернету речей, її архітектури та структурних рівнів організації з позиції розробки функцій моніторингу мережі.

2. Показна необхідність та доцільність застосування методів інтелектуального аналізу параметрів мереж пристроїв Інтернету речей на етапі проектування, а також в рамках модернізації існуючих проектних рішень. Запропоновано метод підвищення ефективності моніторингу за допомогою використання графоаналітичного методу аналізу фазового портрету, отриманого в результаті реконструкції часового ряду зміни пропускну́ї спроможності каналу передачі даних.

3. Отримав подальшого розвитку графоаналітичний метод моніторингу параметрів мережі пристроїв Інтернету речей на основі аналізу фазових портретів завантаженості каналу передачі даних, які були побудовані на основі часових рядів миттєвих значень швидкості прийому та передачі даних між сервером та кінцевими пристроями мережі.

4. Запропонований метод може бути застосований для інтелектуального моніторингу параметрів локальних мереж Інтернету речей у вигляді самостійного моніторингового програмного засобу, який працює на локальному сервері чи хмарному середовищі, або у складі існуючої системи в якості інтегрованого модуля

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kevin Ashton Invents the Term "The Internet of Things" [Electronic resource]. – Access mode: <https://www.historyofinformation.com/detail.php?id=3411> – 1.11.2022.
2. Abbas, Zeeshan, and Wonyong Yoon. 2015. "A Survey on Energy Conserving Mechanisms for the Internet of Things: Wireless Networking Aspects" *Sensors* 15, no. 10: 24818-24847.
3. Zanella, A.; Bui, N.; Castellani, A. Internet of Things for Smart Cities. *IEEE Int. Things J.* 2014, 1, 22–32.
4. Jin, J.; Gubbi, J.; Marusic, S.; Palaniswami, M. An Information Framework for Creating a Smart City through Internet of Things. *IEEE Int. Things J.* 2014, 1, 112–121.
5. Stankovic, J.A. Research Directions for the Internet of Things. *IEEE Int. Things J.* 2014, 1, 3–9
6. Taleb, T.; Kunz, A. Machine Type Communications in 3GPP Networks: Potential, Challenges, and Solutions. *IEEE Commun. Mag.* 2012, 50, 178–184.
7. Datta, S.K.; Bonnet, C.; Nikaiein, N. An IoT Gateway Centric Architecture to Provide Novel M2M Services. In *Proceedings of the IEEE World Forum Internet of Things (WF-IoT)*, Seoul, Korea, 6–8 March 2014; pp. 514–519
8. Huang, J.-D.; Hsieh, H.-C. Design of Gateway for Monitoring System in IoT Networks. In *Proceedings of the IEEE Green Computing and Communications (GreenCom)*, Beijing, China, 20–23 August 2013; pp. 1876–1880.
9. Guoqiang, S.; Yanming, C.; Chao, Z.; Yanxu, Z. Design and Implementation of a Smart IoT Gateway. In *Proceedings of the IEEE Green Computing and Communications (GreenCom)*, Beijing, China, 20–23 August 2013; pp. 720–723.

10. Song, G.; Zhou, Y.; Zhang, W.; Song, A. A Multi-interface Gateway Architecture for Home Automation Networks. *IEEE Trans. Consum. Electron.* 2008, 54, 1110–1113.
11. Petrov, V.; Andree, S.; Koucheryavy, Y. An Applicability Assessment of IEEE 802.11 Technology for Machine-Type Communications. In *Proceedings of the 11th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Ayia Napa, Cyprus, 19–22 June 2012; pp. 24–31.
12. Costantino, L.; Buonaccorsi, N.; Cicconetti, C.; Mambrini, R. Performance Analysis of an LTE Gateway for the IoT. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, San Francisco, CA, USA, 25–28 June 2012; pp. 1–6.
13. Kim, J.; Lee, J.; Kim, J.; Yun, J. M2M Service Platforms: Survey, Issues, and Enabling Technologies. *IEEE Commun. Surv. Tutor.* 2014, 16, 61–76.
14. Pering, T.; Agarwal, Y.; Gupta, R.; Want, R. Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*, Uppsala, Sweden, 19–22 June 2006; pp. 220–232.
15. Akyildiz, I.F. Wireless Sensor Networks: A Survey. *Comput. Netw.* 2002, 38, 393–422.
16. Sendra, S.; Lloret, J.; Garcia, M.; Toledo, J.F. Power Saving and Energy Optimization Technique for Wireless Sensor Networks (Invited Paper). *J. Commun.* **2011**, 6, 439–459.
17. Hazmi, A.; Rinne, J.; Valkama, M. Feasibility Study of IEEE 802.11ah Radio Technology for IoT and M2M use Cases. In *Proceedings of the GC'12 Workshop: Second International Workshop on Machine-to-Machine Communications “Key” to the Future Internet of Things*, Anaheim, CA, USA, 3–7 December 2012; pp. 1687–1692.

18. Weiser, M.: The computer for the 21st century. *IEEE Pervasive Comput.* 99(1), 19–25 (2002).

19. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. *Comput. Netw.* 54, 2805 (2010).

20. Besbes, M.A., Hamam, H.: An intelligent RFID checkout for stores. In: 2011 International Conference on Microelectronics (ICM), pp. 1–12 (2011).

21. Fielding, R.T., Taylor, R.N.: Principled design of the modern web architecture. In: Proceedings of the 2000 International Conference on Software Engineering, pp. 407–416 (2000).

22. Комп'ютерні мережі - Частина 1 навчальний посібник: навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення» та 126 «Інформаційні системи та технології» / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 336 с.

23. M. L. Rajaram, E. Koungianos, S. P. Mohanty and U. Choppali, "Wireless sensor network simulation frameworks: A tutorial review: MATLAB/Simulink bests the rest," in *IEEE Consumer Electronics Magazine*, vol. 5, no. 2, pp. 63-69, April 2016, doi: 10.1109/MCE.2016.2519051.

24. 1. Прикладне застосування теорії хаотичних систем у телекомунікаціях: монографія / [Ю.Я. Бобало, С.Д. Галюк, М.М. Климаш, Р.Л. Політанський]; Нац. ун-т «Львів. політехніка». – Львів: Коло, 2015. – 178 с.

25. Logistic Equation [Electronic resource]. – Access mode: <https://mathworld.wolfram.com/LogisticEquation.html>– 1.11.2022.

26. IoT monitor and manage loops [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/azure/architecture/example-scenario/iot/monitor-manage-loop> – 1.11.2022.

27. IoT measure and control loops [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/azure/architecture/example-scenario/iot/measure-control-loop> – 1.11.2022.
28. IoT analyze and optimize loops [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/azure/architecture/example-scenario/iot/analyze-optimize-loop> – 1.11.2022.
29. IoT event routing [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/azure/architecture/example-scenario/iot/event-routing> – 1.11.2022.
30. VirtualBox [Electronic resource]. – Access mode: <https://www.virtualbox.org/> – 1.11.2022.
31. How to Use VirtualBox: Quick Overview [Electronic resource]. – Access mode: <https://www.nakivo.com/blog/use-virtualbox-quick-overview/> – 1.11.2022.
32. Get Ubuntu Server Option 1: Manual server installation [Electronic resource]. – Access mode: <https://ubuntu.com/download/server> – 1.11.2022.
33. How to Install WordPress with LAMP on Ubuntu 20.04 [Electronic resource]. – Access mode: <https://www.cloudsigma.com/how-to-install-wordpress-with-lamp-on-ubuntu-20-04/> – 1.11.2022.
34. WordPress: Publish your passion [Electronic resource]. – Access mode: <https://wordpress.org/> – 1.11.2022.
35. WP-MQTT [Electronic resource]. – Access mode: <https://wordpress.org/plugins/wp-mqtt/> – 1.11.2022.
36. DIOT SCADA with MQTT [Electronic resource]. – Access mode: <https://wordpress.org/plugins/ecava-diot-scada/> – 1.11.2022.

37. AnyViz [Electronic resource]. – Access mode:
<https://wordpress.org/plugins/anyviz/> – 1.11.2022.

38. LED-SITE-INDICATOR [Electronic resource]. – Access mode:
<https://wordpress.org/plugins/led-site-indicator/> – 1.11.2022.

ДОДАТОК А

Конфігураційні файли Apache2 та PHP

Конфігураційний файл /etc/apache2/sites-available/wordpress.conf:

```
<VirtualHost *:80>
  DocumentRoot /srv/www/wordpress
  <Directory /srv/www/wordpress>
    Options FollowSymLinks
    AllowOverride Limit Options FileInfo
    DirectoryIndex index.php
    Require all granted
  </Directory>
  <Directory /srv/www/wordpress/wp-content>
    Options FollowSymLinks
    Require all granted
  </Directory>
</VirtualHost>
```

Конфігурація бази даних MySQL:

```
$ sudo mysql -u root
```

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 7

Server version: 5.7.20-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> CREATE DATABASE wordpress;
```

```
Query OK, 1 row affected (0,00 sec)
```

```
mysql> CREATE USER wordpress@localhost IDENTIFIED BY '<your-  
password>';
```

```
Query OK, 1 row affected (0,00 sec)
```

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER  
-> ON wordpress.*  
-> TO wordpress@localhost;
```

```
Query OK, 1 row affected (0,00 sec)
```

```
mysql> FLUSH PRIVILEGES;
```

```
Query OK, 1 row affected (0,00 sec)
```

```
mysql> quit
```

```
Bye
```

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра телекомунікацій, медійних та інтелектуальних технологій

ДИПЛОМНА РОБОТА

*«Метод моніторингу параметрів IoT-мережі на основі
аналізу фазових портретів»*

Спеціальність 172 – «Телекомунікації та радіотехніка»

Виконав: студент 2 курсу, група ТРМ-21-1

В.В. Іваненко

Керівник: д-р. техн. наук, проф.

С.К. Підченко

Хмельницький, 2022

ВСТУП

Актуальність теми дослідження.

Для ефективного проектування, розробки та впровадження інтелектуальних пристроїв Інтернету речей (Internet of Things, IoT) необхідно вирішити ряд інженерно-технічних задач щодо оптимізації роботи та забезпечення відмовостійкості пристроїв, особливо в умовах нестабільного підключення, хакерських атак, зовнішнього джерела завад тощо. На етапі, який передуює безпосередній розробці та апаратно-програмній реалізації IoT-мереж – етапі моделювання та проектування, критично важливим є якісна та кількісна оцінка параметрів мережі, в першу чергу завантаження та пропускної здатності міжсегментних ліній передачі, що дозволить в майбутньому уникнути непередбачуваних відмов та зменшити витрати, пов'язані із експлуатацією.

З іншого боку, вже реалізована і функціонуюча система IoT пристроїв, яка на даний момент дозволяє доволі ефективно вирішувати поставлені на етапі проектування технологічні задачі, може бути схильна до непередбачуваних відмов та дестабілізації функціонування, отже, описані підходи також актуальні для модернізації існуючих рішень та систем, які вже введені в експлуатацію.

Мета і задачі дослідження. Метою роботи є підвищення ефективності моніторингу параметрів мережі пристроїв IoT за рахунок використання графоаналітичного методу аналізу фазових портретів, отриманих в результаті реконструкції часових рядів зміни пропускної спроможності каналу передачі даних

Для досягнення поставленої мети в роботі сформульовані та вирішені наступні задачі:

- аналітичний огляд концепцій Інтернету речей та його архітектури;
- аналіз архітектурних рівнів Інтернету речей з позиції розробки функцій моніторингу мережі;
- застосування методу аналізу фазових портретів для моніторингу та контролю завантаження каналу локальної IoT-мережі.
- моделювання нестабільного режиму роботи локальної IoT мережі, викликаного нестабільними режимами роботи кінцевих пристроїв;
- встановлення та налаштування прототипу централізованого сервера для керування та контролю мережі IoT-пристроїв.

Об'єкт дослідження: процес функціонування та міжмережевої комунікації пристроїв Інтернету речей у складі локальних комп'ютерних мереж.

Предмет дослідження: метод моніторингу параметрів IoT-мережі на основі аналізу фазових портретів.

Методи дослідження: для вирішення поставлених задач були використані методи математичного моделювання, математичного аналізу, системного аналізу, алгоритмізації та програмування.

Наукова новизна отриманих результатів: отримав подальшого розвитку графоаналітичний метод моніторингу параметрів мережі пристроїв Інтернету речей на основі аналізу фазових портретів завантаженості каналу передачі даних, які були побудовані на основі часових рядів миттєвих значень швидкості прийому та передачі даних між сервером та кінцевими пристроями мережі.

Практичне значення отриманих результатів:

-засобами програмного симулятора Cisco Packet Tracer була розроблена імітаційна модель локальної мережі, яка складається із двох сегментів із різними режимами роботи кінцевих пристроїв;

-на основі логістичного відображення була розроблена модель нестабільності мережевих підключень кінцевих пристроїв, що призводить до нерівномірного неперіодичного режиму завантаженості мережі, яких подібний до стану динамічного хаосу;

-налаштовано сервер збору та відображення даних параметрів мережі, який працює під управлінням операційної системи Ubuntu Linux в середовищі віртуалізації Virtual Box.

Апробація результатів дослідження. За результатами магістерської роботи, відповідно до теми дослідження, були підготовлені тези, які були опубліковані в збірнику тез доповідей XIV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук», яка проходила в м. Хмельницькому 18-19 листопада 2022 року.

1 АНАЛІТИЧНИЙ ОГЛЯД АРХІТЕКТУРИ ІНТЕРНЕТУ РЕЧЕЙ

Основні поняття Інтернету речей

Інтернет речей (Internet of Things, IoT) — це мережа фізичних об'єктів, які мають вбудовані технології, що дозволяють здійснювати взаємодію з зовнішнім середовищем, передавати відомості про свій стан і приймати дані ззовні. Складовою частиною Інтернету речей є Індустріальний (або Промисловий) Інтернет речей (Industrial Internet of Things, IIoT).

Концепція Інтернету речей полягає у підключення будь-якого пристрою (якщо він має перемикач включення / вимикання) до Інтернету та інших під'єднаних пристроїв. Таким чином, IoT представляє собою величезну базу даних, в якій, інтелектуальні пристрої обмінюються інформацією, що була отримана з датчиків, а також електронних пристроїв, системної мережі та апаратно-програмного забезпечення. Зв'язок між пристроями може відбуватися між різними фізичними об'єктами, наприклад, в офісі (корпоративний сегмент, бізнес), вдома (системи «розумний будинок»), чи на підприємстві (промисловий Інтернет речей).

Історія Інтернету речей

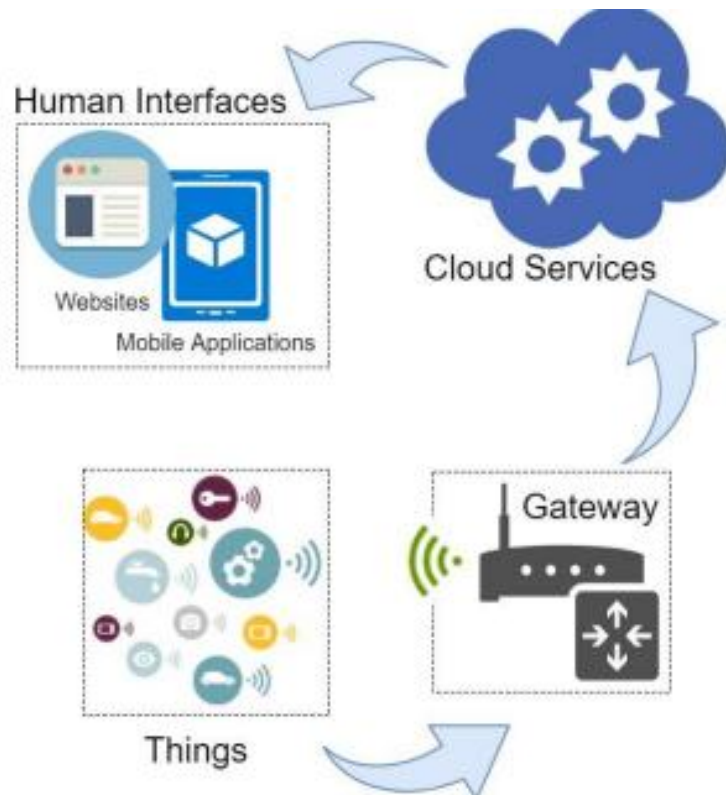
Термін «Інтернет речей» вперше був запропонований в 1999 році Кевіном Ештоном, одним з трьох засновників Центру автоматичної ідентифікації Массачусетського університету (США).

Одним з перших підключених до мережі пристроїв став апарат з продажу напою Coca-Cola, встановлений в Університеті Карнегі – Меллон у 1982 році. Апарат мав можливість передавати на сервер дані про кількість пляшок напою, що містились в ньому, та про свій стан в цілому.

Періодом активних обговорень мереж, які змогли б забезпечити міжмашинну взаємодії стали 1990-ті роки. Наприклад, керівник дослідницьких робіт в Херох PARC (дослідному центрі компанії Херох) Марк Вейзер запропонував концепцію комп'ютеризації, що передбачала масове впровадження комп'ютерів і організацію зв'язку між ними, завдяки якій машини могли б самостійно вирішували повсякденні задачі користувача.

Вчений Білл Джой в рамках виступу на Міжнародному економічному форумі в Давосі в 1999 році запропонував ідею «Шести типів Web» - шести типів Інтернету майбутнього. У ній він досить точно спрогнозував появу бездротових мобільних мереж, інтелектуальних голосових помічників і комунікацій між пристроями (за його термінологією такий зв'язок називалася Device-to-device). В тому ж році були спроби створення перших IoT-проектів.

Огляд архітектури Інтернету речей



Інтернет речей складається з набору різних інфокомунікаційних технологій, що забезпечують функціонування Інтернету речей, і його архітектура показує, як ці технології пов'язані один з одним.

Спрощено система IoT складається з чотирьох основних компонентів, як показано на рисунку.

Річ (Thing) – пристрій з мережевим інтерфейсом, до складу якого входять сенсори збору даних та механізмів керування.

Шлюз (Gateway) – пристрій (точка) доступу до Інтернету.

Хмара (Cloud) – сукупність серверів в мережі Інтернет, що об'єднані в кластери з єдиним інтерфейсом доступу для програм для збереження та обробки даних.

Інтерфейси людини-користувача (Human-interfaces) – програмне забезпечення високого рівня для взаємодії із кінцевими користувачами системи. Це можуть бути мобільні програми керування та візуалізації даних, Web-орієнтовані додатки чи настільні додатки для операційних систем Windows, Linux тощо.

Архітектура IoT включає чотири функціональні рівні:

1 Рівень сенсорів та сенсорних мереж

Найнижчий рівень архітектури IoT складається з «розумних» (smart) об'єктів – речей, інтегрованих із сенсорами (датчиками). Сенсори реалізують поєднання фізичного та віртуального (цифрового) середовищ, забезпечуючи збирання та обробку інформації в реальному масштаб часу. Мініатюризація, що призвела до зменшення фізичних розмірів апаратних сенсорів, дозволила інтегрувати їх безпосередньо в об'єкти фізичного середовища. Сенсори класифікуються відповідно до їх призначення, наприклад, сенсори навколишнього середовища, сенсори для тіла, сенсори для побутової техніки, сенсори для побутової техніки транспортних засобів тощо.

Більшість сенсорів потребує з'єднання з агрегатором сенсорів (шлюзом), який може бути реалізовано бути реалізований з використанням локальних обчислювальних мереж (LAN, Local Area Network), таких як Ethernet та Wi-Fi або персональних мереж (PAN, Personal Area Network), таких як ZigBee, Bluetooth та ультраширококуткового бездротового зв'язку на малих відстанях (UWB, Ultra-Wide Band).

Для сенсорів, які не вимагають підключення до агрегатора, їх зв'язок із серверами/додатками може надаватися з використанням глобальних бездротових мереж WAN, таких як GSM, GPRS та LTE. Сенсори, які характеризуються низьким енергоспоживанням та низькою швидкістю передачі даних, утворюють широко відомі бездротові сенсорні мережі (WSN, Wireless Sensor Network). WSN набирають все більшої популярності, оскільки вони можуть утримувати набагато більше сенсорів з підтримкою роботи від батарей та охоплюють великі площі.

2 Рівень шлюзів та мереж

Великий обсяг даних, створюваних на першому рівні IoT численними мініатюрними сенсорами, вимагає надійної та високопродуктивної дротової або бездротової мережевої інфраструктури в якості транспортного

середовища. Існуючі мережі зв'язки, що використовують різні протоколи, можуть бути використані для підтримки міжмашинних комунікацій M2M та їх додатків. Для реалізації широкого спектру послуг та додатків в IoT необхідно забезпечити спільну роботу безлічі мереж різних технологій та протоколів доступу в гетерогенній конфігурації. Ці мережі повинні забезпечувати необхідні значення якості передачі інформації, і насамперед із затримки, пропускнуєї спроможності та безпеки. Цей рівень складається з конвергентної мережної інфраструктури, що створюється шляхом інтеграції різнорідних мереж у єдину мережеву платформу. Конвергентний абстрактний мережевий рівень у IoT дозволяє через відповідні шлюзи кільком користувачам використовувати ресурси однієї мережі незалежно та спільно без зниження рівня конфіденційності, безпеки та продуктивності.

3 Сервісний рівень

Сервісний рівень містить набір інформаційних послуг, покликаних автоматизувати технологічні та бізнес операції в IoT: підтримки операційної та бізнес діяльності (OSS/BSS, Operation Support System/Business Support System), різної аналітичної обробки інформації (наприклад, статистичного та інтелектуального аналізу даних та текстів) зберігання даних, забезпечення інформаційної безпеки, управління бізнес-правилами (BRM, Business Rule Management), управління бізнес-процесами (BPM, Business Process Management) тощо.

4 Рівень додатків

На четвертому рівні архітектури IoT існують різні типи додатків для відповідних промислових секторів та сфер діяльності (енергетика, транспорт, торгівля, медицина, освіта тощо). Програми можуть бути «вертикальними», коли вони є специфічними для конкретної галузі промисловості, а також «горизонтальними», (наприклад, керування автопарком, відстеження активів фірми тощо), які можуть використовуватись у різних секторах економіки.

2 МАТЕМАТИЧНА МОДЕЛЬ МЕТОДУ МОНІТОИНГУ ПАРАМЕТРІВ МЕРЕЖІ ПРИБРОЇВ ІНТЕРНЕТУ РЕЧЕЙ НА ОСНОВІ АНАЛІЗУ ФАЗОВИХ ПОРТРЕТІВ

Для довільної динамічної системи (ДС) із двома змінними стану система рівнянь матиме вигляд:

$$\begin{cases} du/dt = P(u, v) \\ dv/dt = Q(u, v) \end{cases}, \quad (1)$$

де $u(t), v(t)$ – фазові змінні; P, Q – нелінійні функції двох змінних, отримані в результаті поліноміальної апроксимації вихідного часового ряду.

Система (1) описує динаміку нелінійної ДС на заданому проміжку часу (за умови існування та однозначності рішень функцій правих частин системи P, Q). Геометрично стан (поточну фазу) системи (1) можна зобразити на фазовій площині в декартових координатах, де миттєвому стану системи буде поставлена у відповідність точка $M(u_k, v_k)$. Таким чином, еволюція системи представлена деякою фазовою траєкторією, яку описує точка M , напрямок та швидкість руху якої визначається вектором фазових швидкостей.

Для системи (1) рівняння фазових траєкторій матиме вигляд:

$$v(u) = \int \frac{Q}{P} du + C, \quad (2)$$

де C – постійна інтегрування, що визначається на основі початкових значення координат v_0 та u_0 для моменту часу $t = 0$.

У разі неможливості аналітичного розв'язування системи (1) та виразу (2), фазовий портрет системи може бути побудовано за допомогою ітеративного відображення, отриманого для системи (1) шляхом заміни диференціалів функцій скінченними різницями, або за допомогою чисельного розв'язку системи (1), наприклад, методом Рунге-Кутти.

Проведемо дослідження на прикладі локальної мережі, схема якої зображена на рисунку 1. Дана комп'ютерна мережа складаються із трьох сегментів (А, В і С), маршрутизатора 1804 та трьох комутаторів Cisco 2960.

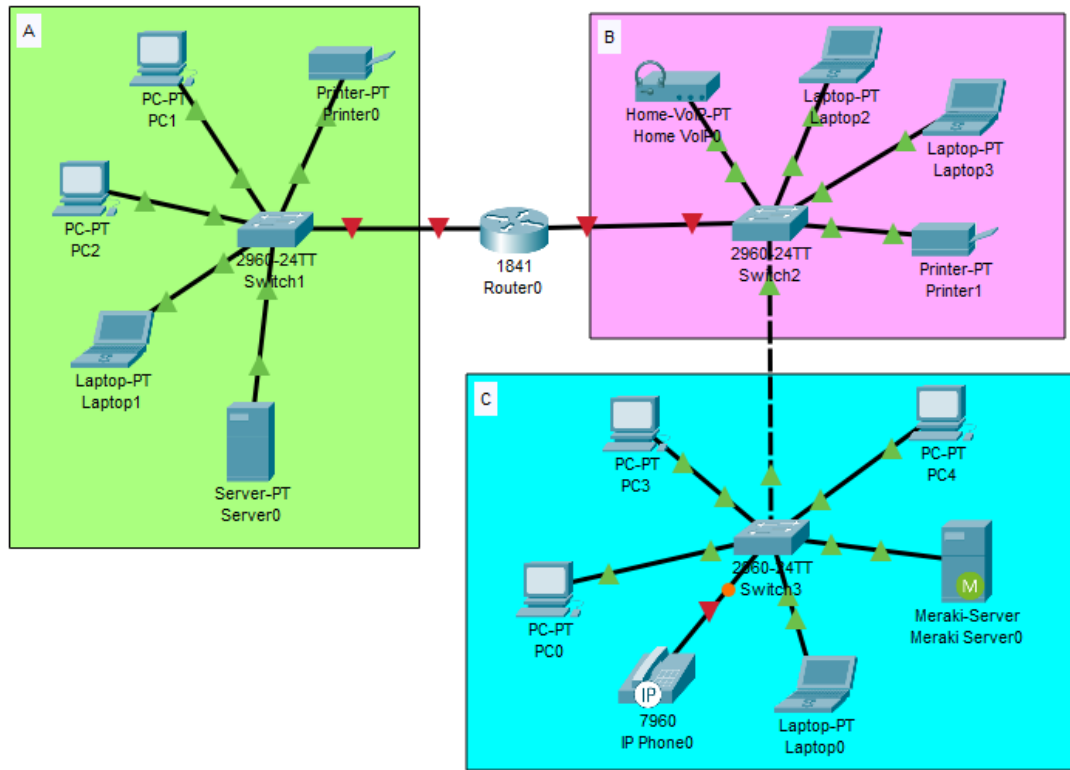


Схема досліджуваної мережі

В той же час, номенклатура та режими роботи кінцевих пристроїв сегменту мережі С обумовлюють складний неперіодичний характер завантаження мережі, що можна порівняти із режимами детермінованих хаотичних коливань.

Розглянемо логістичне відображення в якості математичної моделі джерела таких коливань:

$$x_{n+1} = rx_n(1 - x_n), \quad (4)$$

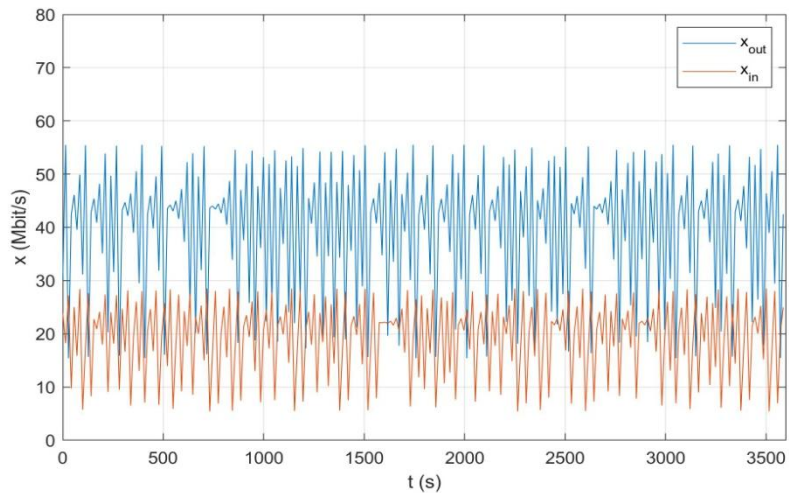
де x_n – фазова змінна, r – біфуркаційний параметр.

Розглянемо завантаження каналу передачі даних між сервером, який знаходиться в сегменті А, та усіма пристроями, розміщеними в сегментах В та С (рисунок 1).

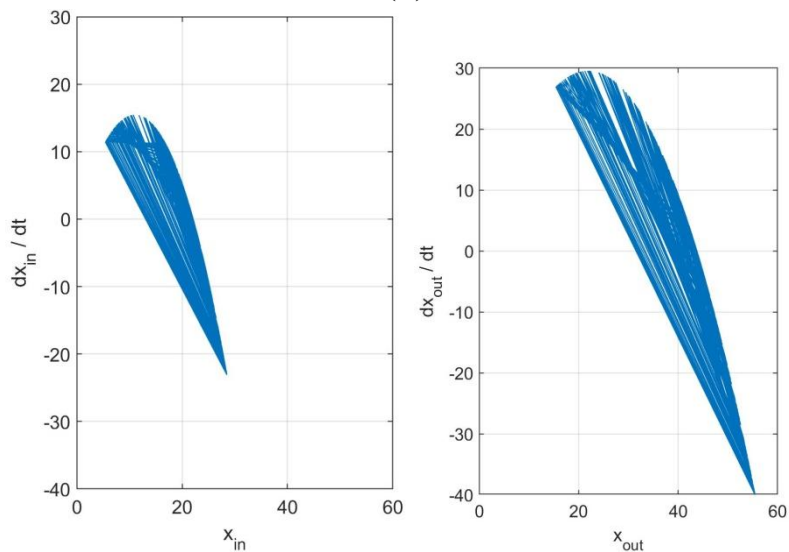
Позначимо $x_{B,rx}$ та $x_{B,tx}$ – миттєві значення швидкості прийому та передачі даних (Mbit/s) між сервером та кінцевими пристроями сегменту В. Аналогічно $x_{C,rx}$ та $x_{C,tx}$ – відповідні значення для пристроїв сегменту С.

Нехай завантаження каналу пристроями сегменту мережі В носить детермінований і постійний характер, тому для деякого інтервалу часу Δt можна вважати:

$$\begin{aligned} x_{B,rx} &= X_{rx} = const \\ x_{B,tx} &= X_{tx} = const \end{aligned} \quad (3)$$



(a)



(б)

(в)

Результат моделювання:
 часові ряди (а), фазові портрети для вхідного
 (б) та вихідного (в) трафіку для сегменту
 мережі С

Результати моделювання трафіку локальної IoT мережі, що складається із двох сегментів, зображені на рисунку.

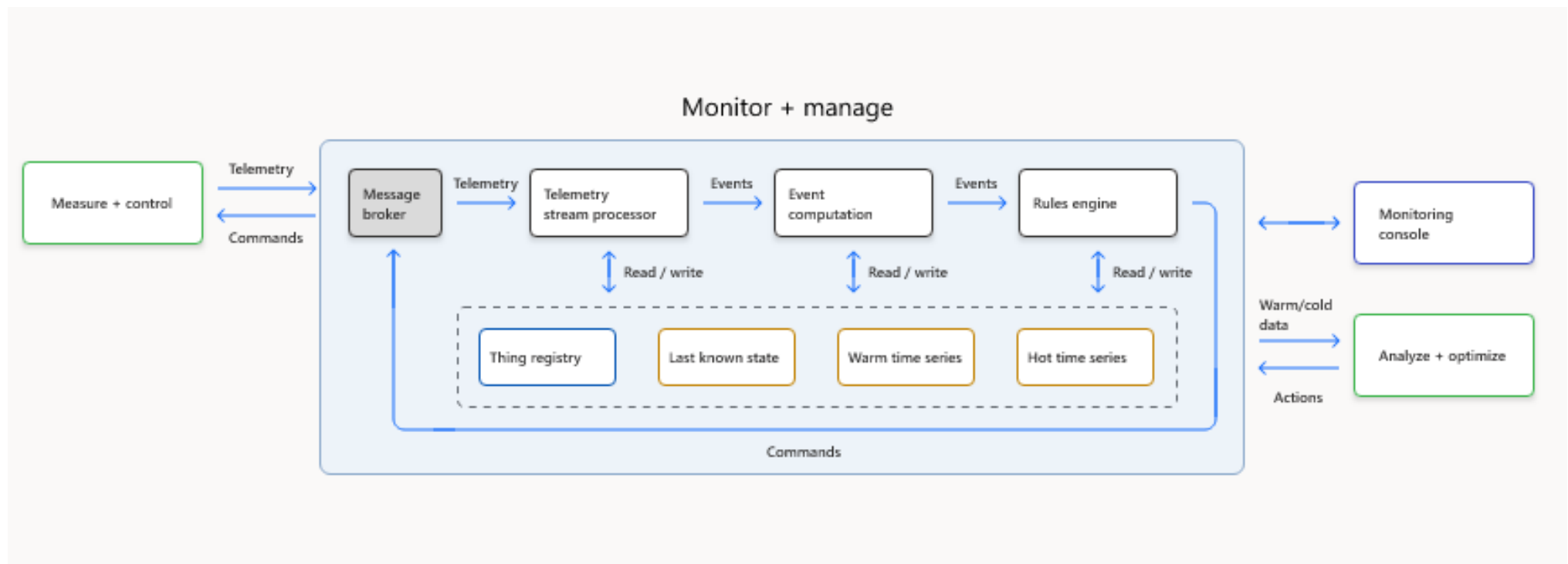
Зображені на рисунку (а) часові ряди для вхідного та вихідного трафіку у випадку складного, неперіодичного характеру функціонування номенклатури пристроїв сегменту С мережі (див. рисунок) відповідають хаотичному режиму роботи модельної системи генератора, створеного на база логістичного відображення на основі виразу (4). Якісно це можна оцінити із форми «дивного атрактора» на фазовій площині (рисунок,б-в).

Запропонований підхід може бути застосований для інтелектуального моніторингу параметрів локальних мереж Інтернету речей у вигляді самостійного моніторингового програмного засобу, який працює на локальному сервері чи хмарному середовищі, або у складі існуючої системи в якості інтегрованого модуля.

3 МОДЕЛЬ ЦИКЛУ МОНІТОРИНГУ ТА КЕРУВАННЯ ПАРАМЕТРІВ МЕРЕЖІ ПРИБОРІВ ІНТЕРНЕТУ РЕЧЕЙ

Моніторинг та керування Інтернетом речей

Далі приведені загальні відомості про архітектуру, характеристики та компоненти циклу моніторингу та управління процесами Інтернету речей (згідно Microsoft).



Цикл управління та моніторингу Інтернет речами – це механізм нагляду та контролю, який гарантує, що система функціонує в межах робочих порогових значень параметрів. Кілька пристроїв у системі повинні працювати спільно, щоб досягти необхідного стану в прийнятному діапазоні та підтримувати його.

Цикл управління і моніторингу відстежує і зіставляє тенденції сигналів активної телеметрії з декількох пристроїв, щоб визначити поточний стан. Логіка поєднує ці тенденції з "теплим" журналом часових рядів (нещодавно отриманих) і сигналами корпоративних систем для отримання нових цінних відомостей. Потім цикл моніторингу та управління надсилає аналітичні відомості через обробник правил, щоб створити виконавчі команди або генерувати попередження, якщо потрібно.

Цикли моніторингу та управління мають такі характеристики:

-пристрої моніторингу можуть розташовуватися віддалено або близько до фізичних пристроїв IoT. Об'єкти, віддалені через свою природу, наприклад нафто- і газопроводи, силові трансформатори, інтелектуальні дверні дзвінки, небезпечні середовища і засоби відстеження ресурсів, не можуть підтримувати цю інфраструктуру. У цих середовищах цикли моніторингу та управління працюють з віддаленими об'єктами, такими як загальнодоступні або приватні хмари. У переробних галузях, таких як нафтоперероблення і виробництво хімічних продуктів, цикли моніторингу та управління можуть бути розгорнуті ближче до пристроїв;

-обчислюються стани залежних пристроїв і формуються послідовності подій для зовнішніх систем обробки;

-можливість інтеграції із пристроями та корпоративними системами через мережеві протоколи HTTP, MQTT і AMQP.

- час циклу може становити кілька секунд залежно від сценарію Інтернету речей. Під час використання мережевих протоколів, що не враховують час, наприклад, MQTT, HTTP і AMQP, може мати місце відхилення або коливання затримки мережевих пакетів.

Цикли моніторингу та управління включають такі компоненти:

Брокери повідомлень обробляють дані телеметрії та надсилають команди пристроям.

Реєстр пристроїв, або система записів і єдине джерело істинних даних для всіх пристроїв IoT, зберігає метадані про пристрої та відносини між ними. Обробник потоку даних телеметрії використовує відомості з реєстру для розуміння структури повідомлень телеметрії, а також для аналізу і виконання логіки потокової обробки. Брокер повідомлень користується реєстром для перевірки запитів на під'єднання пристроїв і ухвалення рішень про маршрутизацію повідомлень.

Обробник потоку даних телеметрії (TSP) отримує дані телеметрії пристроїв, визначає на їхній основі стан окремих пристроїв і наборів пристроїв, а також виявляє помилки і відхилення від необхідного стану. TSP надсилає стани помилок і агреговані або необроблені точки даних відповідним обробникам подій, а також у сховище для подальшого опрацювання і зберігання записів.

Журнал часових рядів – це високошвидкісне сховище, що знаходиться в пам'яті або у віддаленому кеші. Воно надає останні відомі метрики пристроїв і набір точок даних для виявлення тенденцій в реальному часі.

«Теплий» журнал часових рядів зберігає точки даних за кілька тижнів, що допомагає зіставляти тенденції, визначені в реальному часі, з довгостроковими тенденціями і виявити можливі відхилення від бажаного стану. Цикли моніторингу та управління також можуть використовувати індексоване сховище для попереднього обчислення тенденцій.

Обчислення подій визначає події, що потребують реакції. Для цього об'єднуються відомості про події обробника потоків, останні відомі стани пристроїв, тенденції, отримані майже в реальному часі від «теплого» журналу часових рядів, і про сам журнал часових рядів, якщо це потрібно.

Обробник правил використовує і обробляє бізнес-події, налаштовуючи потрібні стани пристроїв за допомогою відповідних команд. Він також може публікувати події та сповіщення на консолі моніторингу.

Консоль моніторингу забезпечує візуальне відображення і втручання людини в разі потреби.

4 ВСТАНОВЛЕННЯ ТА НАЛАШТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ WEB-ОРІЄНТОВАНОГО ІНТЕРФЕЙСУ КОРИСТУВАЧА – АДМІНІСТРАТОРА ЛОКАЛЬНОЇ МЕРЕЖІ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ

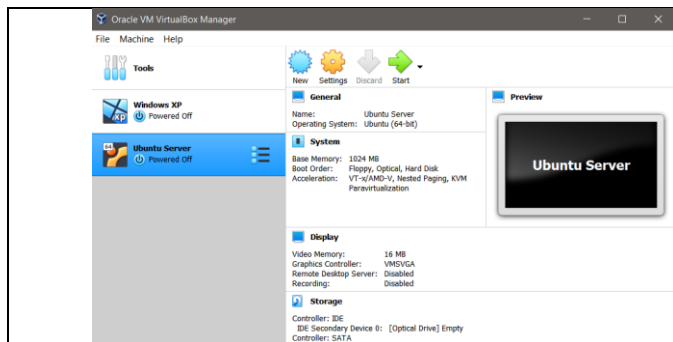
Встановлення та налаштування серверної операційної системи Ubuntu та системи управління Web-контентом Wordpress

Архітектура запропонованої системи моніторингу параметрів пристроїв Інтернету речей передбачає наявність централізованого сервера збору та візуалізації даних, який виконуватиме такі *функції*:

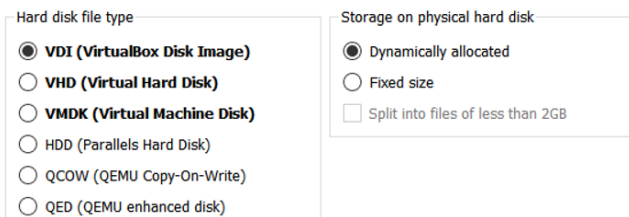
- збір та візуалізації даних, отриманих від пристроїв (речей), які оснащені сенсорами та виконують функції вимірювання та контролю фізичних величин;
- моніторинг параметрів локальної мережі пристроїв Інтернету речей з метою контролю мережевого трафіку та моніторингу підключення усіх пристроїв в реальному часі;
- забезпечення Web-орієнтованого інтерфейсу для користувачів та системних адміністраторів із реалізацією функції розмежування доступу;
- резервне копіювання бази даних та конфігураційних файлів системи.

Встановлення та налаштування серверної операційної системи Ubuntu server на віртуальну машину VirtualBox.

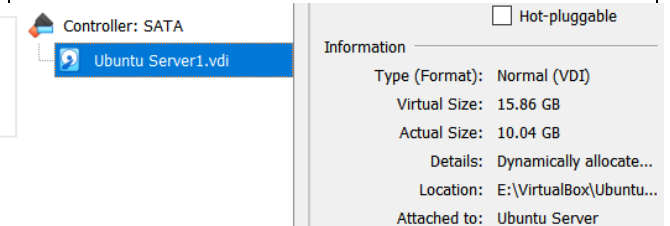
В якості серверної операційної системи було обрано Ubuntu Server версії 20.04. На слайді наведені деякі скріншоти процесу встановлення та налаштування цієї системи, а також HTTP-сервера Apache2. Робота проводилась в середовищі віртуалізації VirtualBox.



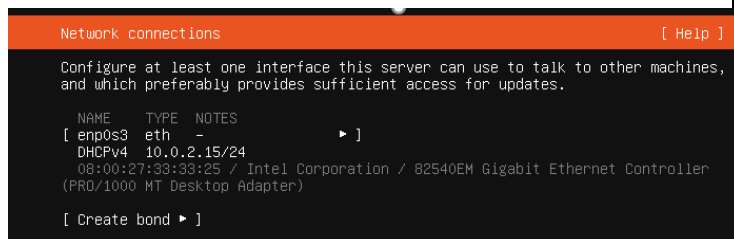
Інтерфейс Oracle VM Virtual Box



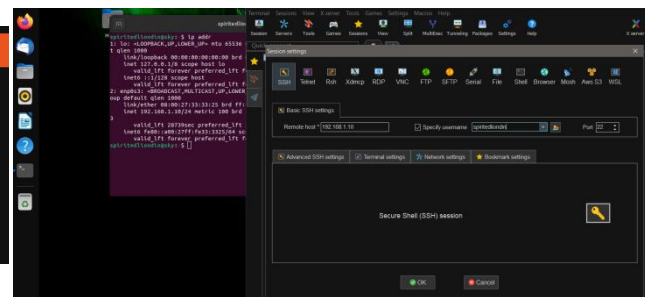
Налаштування диску для системи



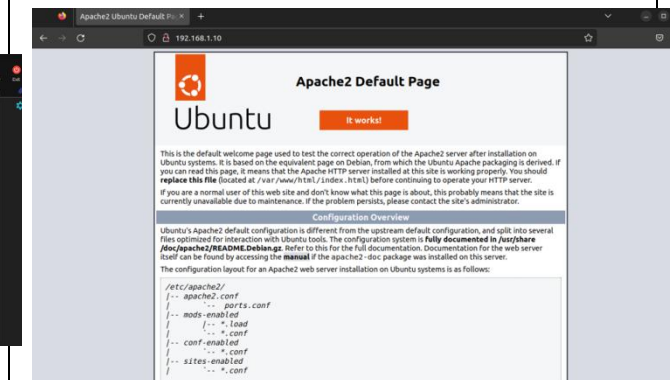
Віртуальна файлова система



Налаштування IP-адреси



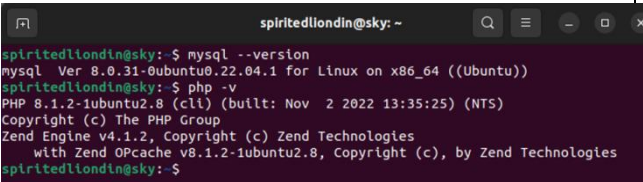
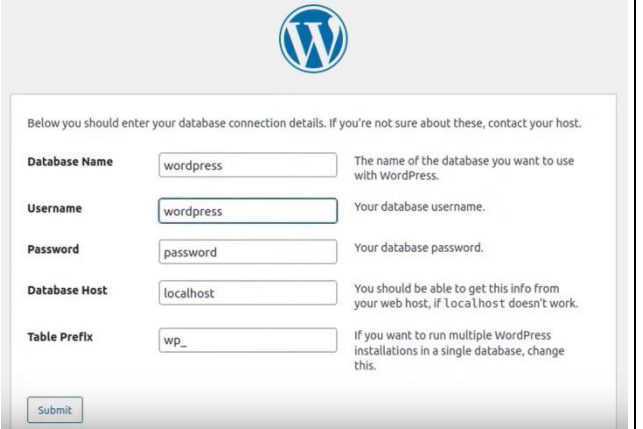
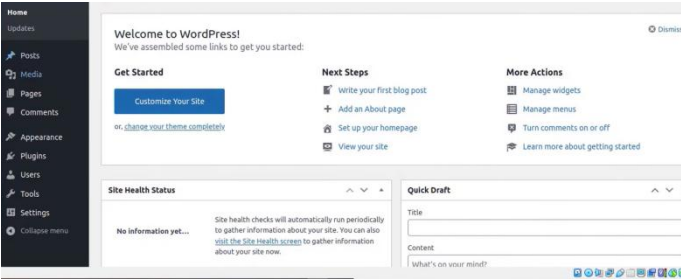
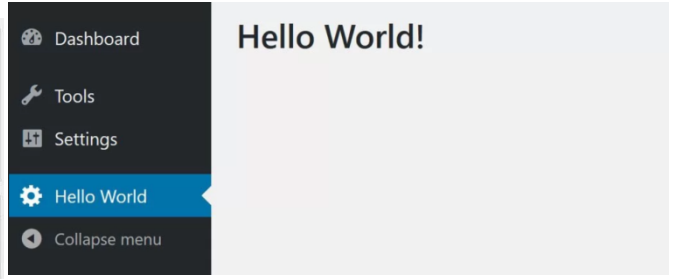
Віддалене підключення по SSH



Встановлення сервера Apache2

Встановлення системи управління Web-контентом WordPress

Система WordPress – це програмне забезпечення, яке автоматизує розробку Web-орієнтованих додатків та інтерфейсів користувача складних програмних систем для Linux-подібних операційних систем. Для роботи цієї програми необхідно додатково встановити сервер баз даних MySQL та інтерпретатор PHP. На слайді наведені деякі скріншоти процесу встановлення програми WordPress та її компонентів на Ubuntu Linux.

 <p>spiritedlondin@sky: ~ spiritedlondin@sky: \$ mysql --version mysql Ver 8.0.31-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu)) spiritedlondin@sky: \$ php -v PHP 8.1.2-1ubuntu2.8 (cli) (built: Nov 2 2022 13:35:25) (NTS) Copyright (c) The PHP Group Zend Engine v4.1.2, Copyright (c) Zend Technologies with Zend OPcache v8.1.2-1ubuntu2.8, Copyright (c), by Zend Technologies spiritedlondin@sky: \$</p> <p>Встановлення MySQL та PHP</p>	<pre> inflatng: wordpress/wp-admin/js/media.js inflatng: wordpress/wp-admin/js/editor-expand.min.js inflatng: wordpress/wp-admin/js/media-gallery.min.js inflatng: wordpress/wp-admin/js/common.min.js inflatng: wordpress/wp-admin/js/tags-box.min.js inflatng: wordpress/wp-admin/js/svg-painter.min.js inflatng: wordpress/wp-admin/js/custom-background.js inflatng: wordpress/wp-admin/js/color-picker.min.js inflatng: wordpress/wp-admin/js/auth-app.js inflatng: wordpress/wp-admin/js/code-editor.js inflatng: wordpress/wp-admin/js/common.js inflatng: wordpress/wp-admin/js/set-post-thumbnail.min.js inflatng: wordpress/wp-admin/js/postbox.min.js inflatng: wordpress/wp-admin/js/color-picker.js inflatng: wordpress/wp-admin/js/password-strength-meter.js inflatng: wordpress/wp-admin/js/customize-nav-menus.js inflatng: wordpress/wp-admin/js/editor-expand.js inflatng: wordpress/wp-admin/js/code-editor.min.js </pre> <p>Завантаження WordPress та розпаковка архіву</p>	<pre> MariaDB [(none)]> show databases; +-----+ Database +-----+ information_schema mysql performance_schema wordpress wordpress +-----+ 5 rows in set (0.007 sec) MariaDB [(none)]> create user "wordpress"@"%" identified by "password"; Query OK, 0 rows affected (0.003 sec) MariaDB [(none)]> grant all privileges on wordpress.* to "wordpress"@"%"; Query OK, 0 rows affected (0.004 sec) MariaDB [(none)]> exit Bye </pre> <p>Створення користувача та бази даних</p>
 <p>Below you should enter your database connection details. If you're not sure about these, contact your host.</p> <p>Database Name: <input type="text" value="wordpress"/> The name of the database you want to use with WordPress.</p> <p>Username: <input type="text" value="wordpress"/> Your database username.</p> <p>Password: <input type="password" value="password"/> Your database password.</p> <p>Database Host: <input type="text" value="localhost"/> You should be able to get this info from your web host, if localhost doesn't work.</p> <p>Table Prefix: <input type="text" value="wp_"/> If you want to run multiple WordPress installations in a single database, change this.</p> <p><input type="button" value="Submit"/></p> <p>Авторизація користувача</p>	 <p>Home Updates Posts Media Pages Comments Appearance Plugins Users Tools Settings Collapse menu</p> <p>Welcome to WordPress! We've assembled some links to get you started:</p> <p>Get Started: Customize Your Site</p> <p>Next Steps: Write your first blog post, Add an About page, Set up your homepage, View your site</p> <p>More Actions: Manage widgets, Manage menus, Turn comments on or off, Learn more about getting started</p> <p>Site Health Status: No information yet...</p> <p>Quick Draft: <input type="text" value="Hello World!"/></p> <p>Вхід в панель налаштувань WordPress</p>	 <p>Dashboard Hello World!</p> <p>Tools Settings Hello World Collapse menu</p> <p>Створення тестової Web-сторінки</p>

ВИСНОВКИ

Загальні висновки до роботи:

1. В роботі проведено аналіз сучасної технології Інтернету речей, її архітектури та структурних рівнів організації з позиції розробки функцій моніторингу мережі.
2. Показна необхідність та доцільність застосування методів інтелектуального аналізу параметрів мереж пристроїв Інтернету речей на етапі проектування, а також в рамках модернізації існуючих проектних рішень. Запропоновано метод підвищення ефективності моніторингу за допомогою використання графоаналітичного методу аналізу фазового портрету, отриманого в результаті реконструкції часового ряду зміни пропускної спроможності каналу передачі даних.
3. Отримав подальшого розвитку графоаналітичний метод моніторингу параметрів мережі пристроїв Інтернету речей на основі аналізу фазових портретів завантаженості каналу передачі даних, які були побудовані на основі часових рядів миттєвих значень швидкості прийому та передачі даних між сервером та кінцевими пристроями мережі.
4. Запропонований метод може бути застосований для інтелектуального моніторингу параметрів локальних мереж Інтернету речей у вигляді самостійного моніторингового програмного засобу, який працює на локальному сервері чи хмарному середовищі, або у складі існуючої системи в якості інтегрованого модуля.

Доповідь завершено

ДЯКУЮ ЗА УВАГУ!

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XIV Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2022»

18-19 листопада 2022

Хмельницький 2022

УДК 004:37:001:62

Збірник наукових праць за матеріалами XIV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2022». Хмельницький – 2022. – 331с.

У збірнику наукових праць подані перспективні практичні розробки аспірантів, студентів та здобувачів в області сучасних інформаційних технологій. Розглянуто актуальні проблеми комп'ютерних наук, комп'ютерної інженерії, прикладної математики й інженерії програмного забезпечення, приведено ряд робіт по впровадженню інформаційних технологій у виробництво та управління. Висвітлено перспективні розробки сучасних систем пошуку, обробки й захисту інформації, медійних та комунікаційних системи.

УДК 004:37:001:62

Матеріали конференції відтворені з авторських оригіналів. При макетуванні можливі незначні зміни компоновки контенту авторських оригіналів.

Участь у конференції та складові всіх її етапів (розгляд праць, макетування, публікація збірника наукових праць та видача сертифікатів) є безкоштовними для всіх учасників. Оргкомітет конференції висловлює подяку учасникам конференції та сподівається на подальшу співпрацю.

З питань проведення конференції та подальшого обміну інформацією звертатись на e-mail конференції: apkt.khnu@gmail.com

Денисенко В.О., Мельников О.Ю. Додаток для виявлення незаконної вирубки лісу.....	104
Дмітрієв Б.В., Яцків В.В. Метод та програмно-технічні засоби виявлення дипфейків	109
Долгополов С.Ю. Використання штучного інтелекту для багатозначної класифікації професійних напрямків діяльності при проведенні професійної орієнтації учнів загальної середньої освіти	112
Дрозд А.І. Розподілена система виявлення зловмисного програмного забезпечення на основі еволюційних алгоритмів.....	117
Дьоміна А.І. Використання методу пошуку новизни для автоматичної генерації тестових даних	119
Захарченко О.О., Бузнік О.О., Марченко А.В. Інформаційна система аналізу збитків від техногенних та природних катастроф ..	124
Іваненко В.В., Слободян М.О. Метод моніторингу параметрів мережі пристроїв інтернету речей на основі аналізу фазових портретів.....	126
Канішев В.О., Мельников О.Ю. Розробка програмного забезпечення для визначення кольорів	131
Клейн О.М. Метод та засоби виявлення аномалій в системах комп'ютерного зору.....	139
Кльоц Ю.П., Петляк Н.С., Блаута В.В. Виявлення аномального трафіку у загальнодоступних комп'ютерних мережах	142
Ковальчук О.В., Слободзян В.О., Мазурець О.В., Бармак О.В. Метод формування бінарного класифікатора україномовного інтернет-контенту	146
Ковтонюк М.О., Шпилюк О.В. Метод та алгоритм відтворення 3D-об'єктів за допомогою доповненої реальності	152
Кожушан М.Г. Автоматизація пошуку термінів у тлумачному словнику.....	158

УДК 004.7

Іваненко В.В., Слободян М.О.

Хмельницький національний університет

МЕТОД МОНІТОРИНГУ ПАРАМЕТРІВ МЕРЕЖІ ПРИСТРОЇВ ІНТЕРНЕТУ РЕЧЕЙ НА ОСНОВІ АНАЛІЗУ ФАЗОВИХ ПОРТРЕТІВ

Показна необхідність та доцільність застосування методів інтелектуального аналізу параметрів мереж пристроїв Інтернету речей на етапі проектування, а також в рамках модернізації існуючих проектних рішень. Запропоновано метод підвищення ефективності моніторингу за допомогою використання графоаналітичного методу аналізу фазового портрету, отриманого в результаті реконструкції часового ряду зміни пропускної спроможності каналу передачі даних.

The necessity and expediency of applying methods of intelligent analysis of the parameters of networks of Internet of Things devices at the design stage, as well as in the modernization of existing design solutions, are shown. A method for improving the efficiency of monitoring by using the graph-analytical method of analyzing the phase portrait obtained as a result of reconstructing the time series of changes in the data transmission channel capacity is proposed.

Для ефективного проектування, розробки та впровадження інтелектуальних пристроїв Інтернету речей (Internet of Things, IoT) необхідно вирішити ряд інженерно-технічних задач щодо оптимізації роботи та забезпечення відмовостійкості пристроїв, особливо в умовах нестабільного підключення, хакерських атак, зовнішнього джерела завад тощо. На етапі, який передусе безпосередній розробці та апаратно-програмній реалізації IoT-мереж – етапі моделювання та проектування, критично важливим є якісна та кількісна оцінка параметрів мережі, в першу чергу завантаження та пропускної здатності міжсегментних ліній передачі, що дозволить в майбутньому уникнути непередбачуваних відмов, зменшити витрати, пов'язані із експлуатацією і т.д.

З іншого боку, вже реалізована і функціонуюча система IoT пристроїв, яка на даний момент дозволяє доволі ефективно вирішувати поставлені на етапі проектування технологічні задачі, може бути схильна до непередбачуваних відмов та дестабілізації функціонування, отже, описані підходи також актуальні для модернізації існуючих рішень та систем, які вже введені в експлуатацію [1, 2].

Таким чином, метою роботи є підвищення ефективності моніторингу параметрів мережі пристроїв IoT за рахунок використання графоаналітичного

методу аналізу фазового портрету, отриманого в результаті реконструкції часового ряду зміни пропускної спроможності каналу передачі даних [3].

Для довільної динамічної системи (ДС) із двома змінними стану система рівнянь матиме вигляд:

$$\begin{cases} \frac{du}{dt} = P(u, v) \\ \frac{dv}{dt} = Q(u, v) \end{cases}, \quad (1)$$

де $u(t), v(t)$ – фазові змінні; P, Q – нелінійні функції двох змінних, отримані в результаті поліноміальної апроксимації вихідного часового ряду.

Система (1) описує динаміку нелінійної ДС на заданому проміжку часу (за умови існування та однозначності рішень функцій правих частин системи P, Q). Геометрично стан (поточну фазу) системи (1) можна зобразити на фазовій площині в декартових координатах, де миттєвому стану системи буде поставлена у відповідність точка $M(u_k, v_k)$. Таким чином, еволюція системи представлена деякою фазовою траєкторією, яку описує точка M , напрямок та швидкість руху якої визначається вектором фазових швидкостей.

Для системи (1) рівняння фазових траєкторій матиме вигляд:

$$v(u) = \int \frac{Q}{P} du + C, \quad (2)$$

де C – постійна інтегрування, що визначається на основі початкових значення координат v_0 та u_0 для моменту часу $t = 0$.

У разі неможливості аналітичного розв'язування системи (1) та виразу (2), фазовий портрет системи може бути побудовано за допомогою ітеративного відображення, отриманого для системи (1) шляхом заміни диференціалів функцій скінченними різницями, або за допомогою чисельного розв'язку системи (1), наприклад, методом Рунге-Кутти.

Проведемо дослідження на прикладі локальної мережі, схема якої зображена на рисунку 1. Дана комп'ютерна мережа складаються із трьох сегментів (А, В і С), маршрутизатора 1804 та трьох комутаторів 2960 [4, 5].

Розглянемо завантаження каналу передачі даних між сервером, який знаходиться в сегменті А, та усіма пристроями, розміщеними в сегментах В та С (рисунком 1).

Позначимо $x_{B,rx}$ та $x_{B,tx}$ – миттєві значення швидкості прийому та передачі даних (Mbit/s) між сервером та кінцевими пристроями сегменту В. Аналогічно $x_{C,rx}$ та $x_{C,tx}$ – відповідні значення для пристроїв сегменту С.

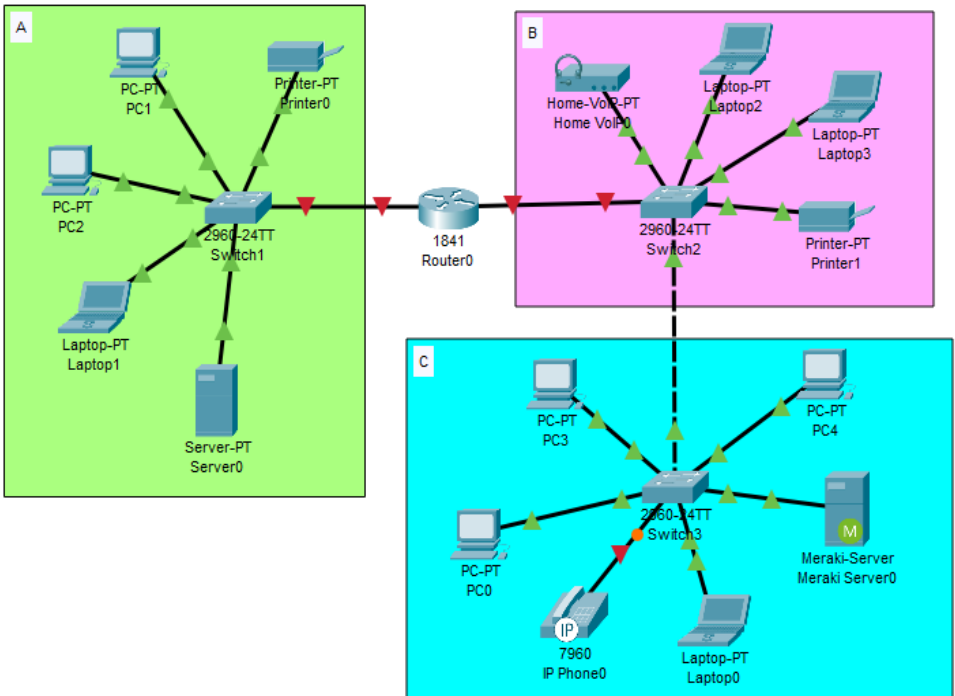


Рисунок 1 – Схема досліджуваної мережі

Нехай завантаження каналу пристроями сегменту мережі В носить детермінований і постійний характер, тому для деякого інтервалу часу Δt можна вважати:

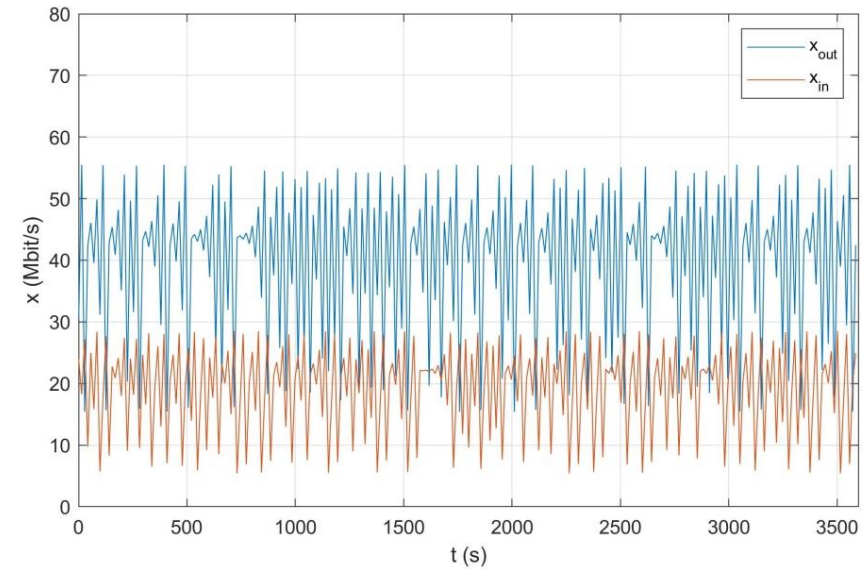
$$\begin{aligned} x_{B,rx} &= X_{rx} = const \\ x_{B,tx} &= X_{tx} = const \end{aligned} \quad (3)$$

В той же час, номенклатура та режими роботи кінцевих пристроїв сегменту мережі С обумовлюють складний неперіодичний характер завантаження мережі, що можна порівняти із режимами детермінованих хаотичних коливань.

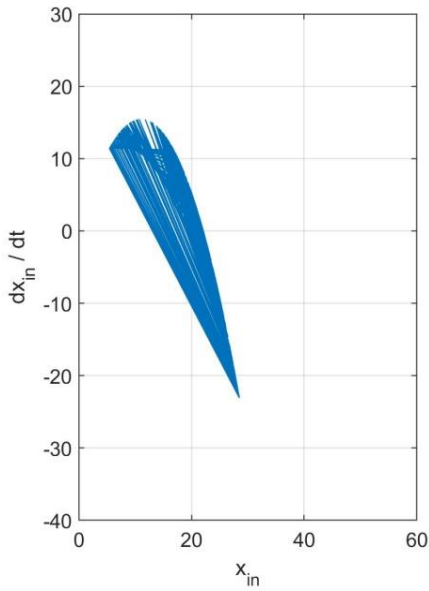
Розглянемо логістичне відображення в якості математичної моделі джерела таких коливань:

$$x_{n+1} = rx_n (1 - x_n), \quad (4)$$

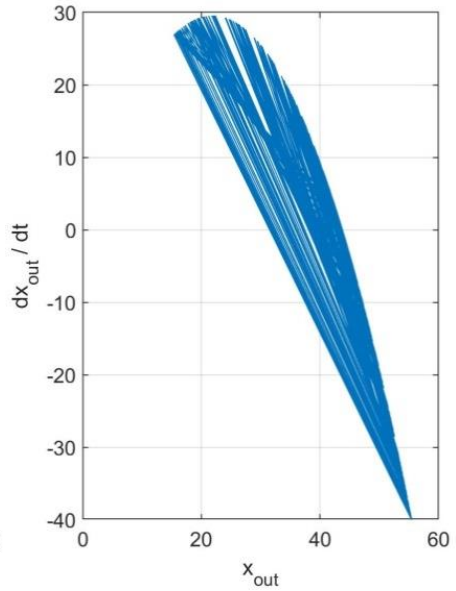
де x_n – фазова змінна, r – біфуркаційний параметр.



(a)



(б)



(в)

Рисунок 1 – Результат моделювання: часові ряди (а), фазові портрети для вхідного (б) та вихідного (в) трафіку для сегменту мережі С

Отже, зображені на рисунку 2,а часові ряди для вхідного та вихідного трафіку у випадку складного, неперіодичного характеру функціонування номенклатури пристроїв сегменту С мережі (див. рисунок 1) відповідають хаотичному режиму роботи модельної системи генератора, створеного на база логістичного відображення на основі виразу (4). Якісно це можна оцінити із форми «дивного атрактора» на фазовій площині (рисунок 2,б-с).

Запропонований підхід може бути застосований для інтелектуального моніторингу параметрів локальних мереж Інтернету речей у вигляді самостійного моніторингового програмного засобу, який працює на локальному сервері [6] чи хмарному середовищі, або у складі існуючої системи в якості інтегрованого модуля.

Перелік посилань

1. F. Zola, L. Seguro-Gil, J.L. Bruse, M. Galar, R. Orduna-Urrutia, Network traffic analysis through node behaviour classification: a graph-based approach with temporal dissection and data-level preprocessing, Computers & Security, Vol. 115, 2022,102632,doi: 10.1016/j.cose.2022.102632.
2. Азарова, А. О., Лисак Н. В. Комп'ютерні мережі та телекомунікації : навчальний посібник– Вінниця : ВНТУ, 2012. – 293 с.
3. Нікітчук Т.М. Використання методу фазової площини для дослідження пульсової хвилі / Т.М. Нікітчук, Ю.А. Поліщук // Вісник ЖДТУ. Тенічні науки. – 2011. – №2 (57). – С.80-87.
4. Комп'ютерні мережі - Частина 1 навчальний посібник: навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення» та 126 «Інформаційні системи та технології» / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 336 с.
5. M. L. Rajaram, E. Koungianos, S. P. Mohanty and U. Choppali, "Wireless sensor network simulation frameworks: A tutorial review: MATLAB/Simulink bests the rest," in IEEE Consumer Electronics Magazine, vol. 5, no. 2, pp. 63-69, April 2016, doi: 10.1109/MCE.2016.2519051.
6. Адміністрування серверних операційних систем: методичні вказівки до виконання лабораторних робіт для студентів напряму підготовки «Програма інженерія» / І.В. Гурман. – Хмельницький ХНУ, 2015. – 50 с.

Завідувачу кафедри
телекомунікацій, медійних та
інтелектуальних технологій (ТМІТ)
Підченку Сергію Костянтинівичу
здобувача вищої освіти, студента 2 курсу,
гр. ТРМ-21-1
Іваненка Валерія Вікторовича

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщена та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

12.12.22
дата


підпис

Іваненко В.В.

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 10%

ID: 109483 Название: Метод моніторингу параметрів IoT - мережі на основі аналізу фазових портретів Добавлено в БД: 2022-12-12 Авторы: Іваненко Валерій Вікторович Руководители: Підченко Сергій Костянтинівч Консультанты: Опоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	63660	925	1389 (2%)	22 (2%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

Ім'я користувача:
Kafedra TMIT KhNU

ID перевірки:
1013283436

Дата перевірки:
12.12.2022 22:15:18 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
12.12.2022 22:23:07 EET

ID користувача:
100005657

Назва документа: Іваненко_ТРМ-21

Кількість сторінок: 72 Кількість слів: 10220 Кількість символів: 81707 Розмір файлу: 1.49 MB ID файлу: 1013041982

254 слова позначені як "вилучені" та не враховуються у підрахунку слів

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

9.54% Схожість

Найбільша схожість: 2.08% з Інтернет-джерелом (http://www.dut.edu.ua/uploads/p_421_27805132.pdf)

7.65% Джерела з Інтернету 124 Сторінка 74

3.01% Джерела з Бібліотеки 49 Сторінка 75

1.12% Цитат

Цитати 8 Сторінка 76

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 5

Підозріле форматування 16 сторінок

РІШЕННЯ КАФЕДРИ

ТЕЛЕКОМУНІКАЦІЙ, МЕДІЙНИХ ТА ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: *Метод моніторингу параметрів IoT-мережі на основі аналізу фазових портретів*

Автор: **Іваненко Валерій Вікторович**

Спеціальність: **172 Телекомунікації та радіотехніка**

Освітня програма: **Телекомунікації та радіотехніка**

Науковий керівник: **д.т.н., доц. Підченко Сергій Костянтинович**

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	<u>Відповідає</u>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження: Запозичення у розмірі 9,54% відносяться до загальноновживаних фраз та не є плагіатом. Найбільша схожість з одним джерелом – 2,08%

Відповідальний за контроль

плагіату за системою Unicheck та антиплагіатизм (ХНУ):

12.12.22р.

Олег ПИВОВАР

Зав. каф. ТМІТ

Сергій ПІДЧЕНКО

12.12.22р.

ВІДГУК
на магістерську дипломну роботу
студента групи ТРм-21-І Валерія ІВАНЕНКО
«Метод моніторингу параметрів IoT-мережі на основі аналізу фазових портретів»

Метою роботи є підвищення ефективності моніторингу параметрів мережі пристроїв IoT за рахунок використання графоаналітичного методу аналізу фазових портретів, отриманих в результаті реконструкції часових рядів зміни пропускної спроможності каналу передачі даних.

В дипломній роботі розглянуто процес функціонування та міжмережевої комунікації пристроїв Інтернету речей у складі локальних комп'ютерних мереж. Представлений метод моніторингу параметрів IoT-мережі на основі аналізу фазових портретів. Розроблена математична модель методу моніторингу параметрів мережі пристроїв Інтернету речей на основі аналізу фазових портретів. Запропоновано модель циклу моніторингу та керування параметрів мережі пристроїв Інтернету речей. Наведено приклад встановлення та налаштування програмного забезпечення для Web-орієнтованого інтерфейсу користувача – адміністратора локальної мережі пристроїв Інтернету речей

У процесі виконання дипломної роботи Валерій ІВАНЕНКО проявив себе старанним, ініціативним фахівцем з достатнім рівнем знань та вмінням їх застосовувати для вирішення завдань в галузі електроніки та телекомунікацій.

В цілому магістерська дипломна робота виконана на високому науково-технічному рівні, а її автор Валерій ІВАНЕНКО заслуговує на оцінку «добре».

Керівник:
д.т.н., професор



Сергій ПІДЧЕНКО

РЕЦЕНЗІЯ
на магістерську дипломну роботу
студента групи ТРм-21-І Валерія ІВАНЕНКО
«Метод моніторингу параметрів IoT-мережі на основі аналізу фазових портретів»

Для ефективного проектування, розробки та впровадження інтелектуальних пристроїв Інтернету речей (Internet of Things, IoT) необхідно вирішити ряд інженерно-технічних задач щодо оптимізації роботи та забезпечення відмовостійкості пристроїв, особливо в умовах нестабільного підключення, хакерських атак, зовнішнього джерела завад тощо. На етапі, який передуює безпосередній розробці та апаратно-програмній реалізації IoT-мереж – етапі моделювання та проектування, критично важливим є якісна та кількісна оцінка параметрів мережі, в першу чергу завантаження та пропускну здатності міжсегментних ліній передачі, що дозволить в майбутньому уникнути непередбачуваних відмов та зменшити витрати, пов'язані із експлуатацією. Тому актуальність теми магістерської дипломної роботи Валерія ІВАНЕНКО не викликає сумнівів.

В роботі проведено аналіз сучасної технології Інтернету речей, її архітектури та структурних рівнів організації з позиції розробки функцій моніторингу мережі. Показна необхідність та доцільність застосування методів інтелектуального аналізу параметрів мереж пристроїв Інтернету речей на етапі проектування, а також в рамках модернізації існуючих проектних рішень. Запропоновано метод підвищення ефективності моніторингу за допомогою використання графоаналітичного методу аналізу фазового портрету, отриманого в результаті реконструкції часового ряду зміни пропускну спроможності каналу передачі даних.

Отримав подальшого розвитку графоаналітичний метод моніторингу параметрів мережі пристроїв Інтернету речей на основі аналізу фазових портретів завантаженості каналу передачі даних, які були побудовані на основі часових рядів миттєвих значень швидкості прийому та передачі даних між сервером та кінцевими пристроями мережі. Запропонований метод може бути застосований для інтелектуального моніторингу параметрів локальних мереж Інтернету речей у вигляді самостійного моніторингового програмного засобу, який працює на локальному сервері чи хмарному середовищі, або у складі існуючої системи в якості інтегрованого модуля

Апробацію результатів дослідження було здійснено на XIV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук», яка проходила в м. Хмельницькому 18-19 листопада 2022 року.

В цілому магістерська дипломна робота Валерія ІВАНЕНКО є актуальною в області сучасних технологій телекомунікацій та радіотехніки та заслуговує оцінки «добре».

Рецензент:

зав. кафедри АКІТ, д.т.н., професор



Валерій МАРТИНЮК