

# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

## ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ

Кафедра радіотехніки та радіоелектронних систем

«На правах рукопису»

Робота допущена до захисту в ЕК  
рішенням кафедри радіотехніки та радіоелектронних систем  
від \_\_\_\_\_ 2024 року, протокол № \_\_\_\_\_.  
Завідувач кафедри доктор фіз.-мат. наук, професор  
\_\_\_\_\_ Ігор АНІСІМОВ

### ДИПЛОМНА РОБОТА МАГІСТРА

на тему:

«ВПРОВАДЖЕННЯ МЕРЕЖІ 5G ДЛЯ ПОКРАЩЕННЯ ЯКОСТІ  
ОБСЛУГОВУВАННЯ (QoS) ІНТЕРНЕТУ РЕЧЕЙ (IoT)»

#### **Виконала:**

студентка 2-го курсу магістратури  
денної форми навчання  
спеціальності 172 - Телекомунікації та радіотехніка  
ОНП «Інформаційна безпека телекомунікаційних систем і мереж»  
Галецька Катерина Олександрівна \_\_\_\_\_

#### **Науковий керівник:**

к. т. н., асистент  
Аль Шурайфі Муштак Таліб \_\_\_\_\_

#### **Рецензент:**

к. т. н., доцент кафедри інформатики та програмної інженерії,  
Факультету інформатики та обчислювальної техніки, КПІ ім. Ігоря Сікорського  
Зенів Ірина Онуфріївна \_\_\_\_\_

Засвідчую, що у цій дипломній роботі  
немає запозичень з праць інших авторів без  
відповідних посилань  
Студентка \_\_\_\_\_ Катерина ГАЛЕЦЬКА

Київ – 2024

## РЕФЕРАТ

Дипломна робота: 41 с., 6 табл., 12 рис., 22 джерела.

5G МЕРЕЖА, ІоТ, ІНТЕРНЕТ РЕЧЕЙ, QoS, ЗОНА ПОКРИТТЯ, ШВИДКІСТЬ ПЕРЕДАЧІ ДАНИХ, СПЕКТРАЛЬНА ЕФЕКТИВНІСТЬ, ПРОПУСКНА ЗДАТНІСТЬ

Об'єкт дослідження – мережа 5G та її вплив на якість обслуговування Інтернету речей.

Мета роботи – моделювання впровадження мережі 5G з акцентом на покращення QoS в системах ІоТ. Робота включає розробку і валідацію моделей для ефективного розподілу ресурсів, управління трафіком і забезпечення високих стандартів QoS, а також оцінку продуктивності.

Робота базується на комп'ютерному моделюванні на мові C++, застосуванні програмних інструментів для симуляції мережевих процесів та аналіз результатів для визначення оптимальних стратегій управління трафіком та розподілу ресурсів.

Результати моделювання можуть бути використані для планування і впровадження мереж 5G операторами зв'язку, що дасть можливість значно підвищити якість обслуговування і задоволеність кінцевих користувачів.

Передбачається, що розроблені моделі та алгоритми нададуть можливість забезпечити високий рівень взаємодії між різними компонентами мережі, оптимізувати управління трафіком і підвищити загальну продуктивність систем ІоТ.

Ця магістерська робота вносить значний вклад у розуміння і практичне застосування мережі 5G для покращення сервісів в Інтернеті речей, підкреслюючи важливість інноваційних підходів в телекомунікаційній індустрії та значення 5G як ключової технології майбутнього.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ .....	4
ВСТУП .....	5
1. ТЕХНОЛОГІЯ 5G ТА ДІАПАЗОНИ ЧАСТОТ .....	6
1.1. Огляд технології 5G та її специфікації для підтримки IoT .....	6
1.2. Діапазони частот у 5G NR.....	7
2. ПАРАМЕТРИ QoS ДЛЯ 5G NR .....	11
3. АРХІТЕКТУРА МЕРЕЖІ.....	14
4. МОДЕЛЮВАННЯ .....	16
4.1. Обробка сигналу .....	16
4.2. Моделювання радіоканалу.....	18
4.3. HARQ .....	22
4.4 Керування QoS .....	24
4.5 Керування ресурсами.....	26
4.6 Моделювання базових станцій.....	27
4.7 Моделювання IoT пристроїв.....	29
5. ОЦІНКА ТА АНАЛІЗ ПРОДУКТИВНОСТІ .....	31
6. РЕЗУЛЬТАТИ .....	37
ВИСНОВКИ .....	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	39

## ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

QoS – Quality of Service (якість обслуговування)  
IoT – Internet of Things (інтернет речей)  
FR1 – Frequency Range 1 (діапазон частот 1)  
FR2 – Frequency Range 2 (діапазон частот 2)  
NR – New Radio  
WRC19 – World Radio Conference 2019  
MIMO – Multiple Input Multiple Output  
MMTC – Massive Machine Type Communications  
URLLC – Ultra Reliable Low Latency Communications  
eMBB – enhanced Mobile Broadband  
5GQI – QoS Identify  
ARP – Allocation and Retention Priority  
GFBR – Guaranteed Bit Rate  
MBR – Maximum Bit Rate  
PDB – Packet Delay Budget  
PER – Packet Error Rate  
TDD – Time Division Duplex  
NMS – Network Management System  
NFV – Network Functions Virtualization  
SDN – Software Defined Networking  
LDPC – Low-Density Parity-Check  
QAM – Quadrature Amplitude Modulation  
HARQ – Hybrid Automatic Repeat Request  
TCP – Transmission Control Protocol  
TBS – Transport Block Size  
UE – User Equipment

## ВСТУП

У наші дні технології швидко розвиваються, звертаючи особливу увагу на поліпшення якості та ефективності комунікацій. Цифрова трансформація впливає на всі сфери життя, і це значною мірою пов'язано з поширенням Інтернету речей (IoT). IoT об'єднує велику кількість пристроїв, які збирають та обмінюються даними. Це ставить перед комунікаційною інфраструктурою нові виклики, особливо щодо якості обслуговування (QoS). Важливий аспект - надання хорошої надійності та ефективності зв'язку. Запровадження мереж 5G розширює можливості для стабільного підключення із низькою затримкою й високою надійністю, що є ключовим для IoT. [1]

5G - це п'яте покоління мобільних мереж, яке принесло значне поліпшення порівняно з попередніми: збільшена пропускна здатність, зменшені затримки, покращена зона покриття та можливість обслуговувати значно більшу кількість пристроїв на одиницю площі.[2] Такі характеристики роблять 5G чудовою платформою для розробки IoT-рішень у різноманітних галузях, наприклад, промисловість, медицина, транспорт та смарт-міста.[3]

Дана магістерська робота спрямована на дослідження та аналіз можливостей, які 5G мережа відкриває для IoT, з особливим акцентом на якість обслуговування (QoS) клієнтів. Робота полягає у вивченні архітектурних рішень, стандартів і практичної реалізації моделювання мережі для оцінки впливу технологій 5G на функціонування систем IoT. Через глибокий аналіз передбачається розробка рекомендацій щодо оптимізації параметрів мереж, що дозволить ефективно та надійно використовувати ресурси для IoT.

## 1. ТЕХНОЛОГІЯ 5G ТА ДІАПАЗОНИ ЧАСТОТ

### 1.1. Огляд технології 5G та її специфікації для підтримки IoT

5G NR, або New Radio, — це інноваційна технологія мобільного зв'язку, яка забезпечує значно вищу пропускну здатність, малу затримку та більшу надійність порівняно з попередніми поколіннями. Це робить 5G NR ідеальною для підтримки різноманітних додатків, таких як Інтернет речей (IoT), автономне керування транспортом та віртуальна реальність.[4]

Основними характеристиками технології 5G є:

- Збільшена пропускну здатність. 5G надає можливості передачі даних до 20 Гбіт/с, що значно перевищує показники 4G (1 Гбіт/с). Це сприяє одночасному підключенню великої кількості пристроїв і обробці значних обсягів даних, наприклад, високоякісне відео.[5]

- Мала затримка. Значення затримки у мережі 5G становить лише кілька мілісекунд, що робить її корисною для застосовань у автономному керуванні автомобілями та віртуальній реальності.[6]

- Вища надійність. Мережа 5G гарантує кращу надійність порівняно із попередніми поколіннями мобільних мереж. Це дає можливість стабільного з'єднання з пристроями навіть у складних умовах, наприклад, щільна забудова міста чи віддалені локації.[6]

Сфера застосування технології 5G для підтримки Інтернету речей широка. Приклади застосування наведені нижче.

Автономна навігація транспортних засобів: завдяки низькій затримці 5G забезпечує ефективне автономне керування транспортними засобами.

Віртуальна і доповнена реальність: використовуючи свою пропускну здатність і низьку затримку, 5G дозволяє надавати послуги віртуальної і доповненої реальності з ефектом повного занурення.

Розумні будинки та міста: 5G дозволяє підключати цілий ряд пристроїв до мережі як в будинках, так і в "розумних" містах. Він відіграє важливу роль у

підтримці промислового Інтернету речей, пропонуючи широкосмуговий зв'язок з низькою затримкою і надійністю. [7]

NR (New Radio) - це новий інтерфейс, який покликаний підтримати широкий діапазон частот і багато одночасних з'єднань. NR є ключовим елементом стандарту 5G і був розроблений для швидкого сполучення та ефективного використання спектру на різних частотних діапазонах. Технологія орієнтована на скорочення затримки, збільшення пропускної здатності та поліпшення ефективності мережі через передові технології такі, як MIMO (Multiple Input Multiple Output) і beamforming.[8]

MMTC (Massive Machine Type Communications) – категорія 5G, що має на меті підтримку сценаріїв з великою кількістю машинних підключень. [9]

URLLC (Ultra Reliable Low Latency Communications): Стандарт для забезпечення підключення з ультранизькою затримкою і високою надійністю. [9]

eMBB (enhanced Mobile Broadband): Забезпечує значно вищі швидкості і пропускну здатність мережі.[9]

Таке різноманіття специфікацій та можливостей 5G створює ідеальну основу для розвитку IoT інфраструктур від промисловості та автоматизації до смарт-міста та телемедицини, відкриває нові перспективи для інновацій і технологій.

## 1.2. Діапазони частот у 5G NR

Діапазони частот — це частини спектру електромагнітних хвиль, які використовуються для передачі даних. Технологія 5G NR підтримує два діапазони частот, що позначаються як FR1 (діапазон частот 1) і FR2 (діапазон частот 2)(табл.1.1). [10] Потреби у спектрі для технології 5G показано на рис.1.1.

Таблиця 1.1 – Діапазони частот для технології 5G

Позначення діапазону	Діапазон частот (МГц)
FR1	410-7125
FR2	24250-52600

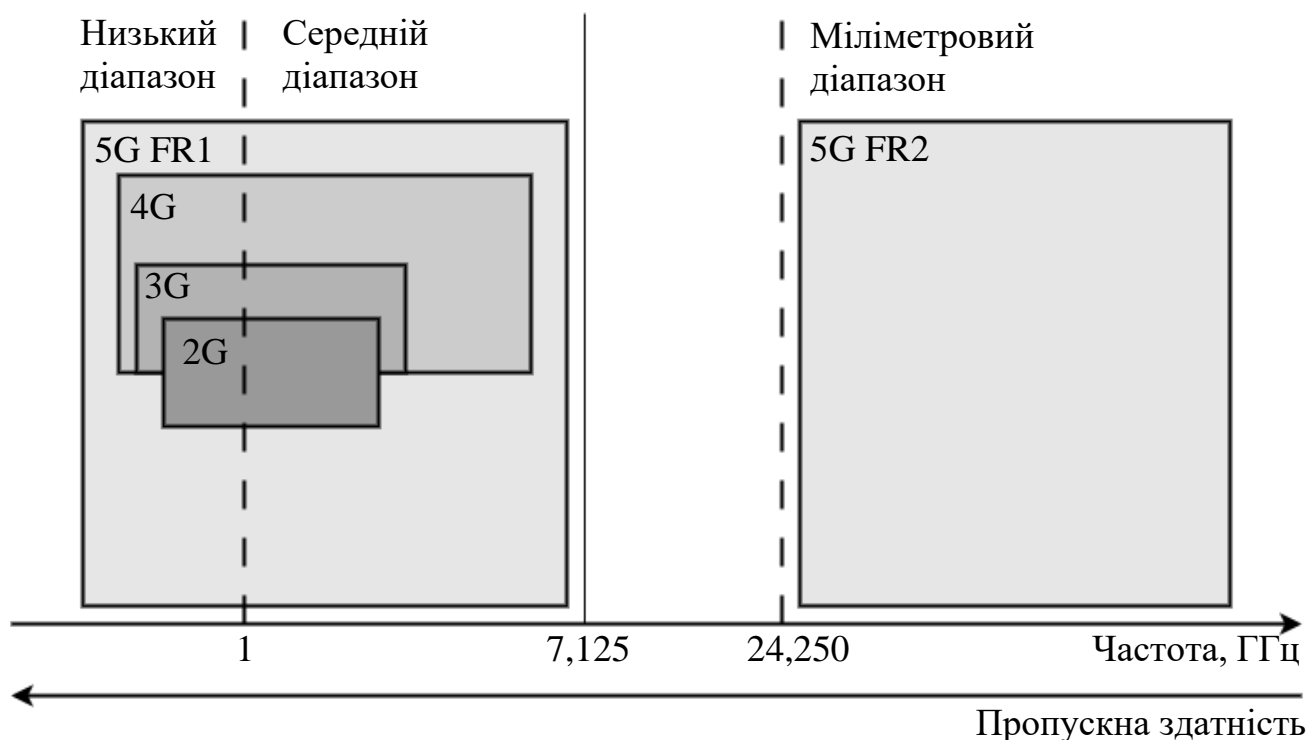


Рисунок.1.1 – Розподіл спектру частот [11]

FR1 (Frequency Range 1) - це діапазон частот нижче 7 ГГц з гарною пропускнуою здатністю, призначений для охоплення великих територій. Очікується, що смуги в діапазоні FR1 передаватимуть більшу кількість трафіку стандартного мобільного зв'язку через свою хорошу пропускну здатність. FR1 має хорошу проникаючу здатність, що означає, що сигнали можуть проходити через стіни та інші перешкоди. Цей діапазон також застосовується попередніми поколіннями мобільного зв'язку і оператори вже мають значний обсяг спектру в даному діапазоні. Це означає, що пристрої FR1, як правило, дешевші, ніж пристрої FR2.[12]

Спочатку діапазон FR1 був до 6 ГГц, але з очікуванням додатковим розподілом спектру FR1 було розширено до 7,125 ГГц після Всесвітньої радіоконференції WRC19 у 2019 році, де узгодили глобальний розподіл спектру. [13]

FR2 (Frequency Range 2) — це діапазон частот вище 6 ГГц, який забезпечує більш високу пропускну здатність порівняно з FR1, що робить його ідеально підходящим для додатків із великою кількістю одночасно підключених пристроїв, таких як “розумне” місто або додатків, що потребують передачі великих обсягів даних, таких як відео в реальному часі та віртуальна реальність. Але при цьому FR2 має меншу проникну здатність. Цей діапазон спрямований на надання дуже високої швидкості передачі даних для 5G. Для цих вищих частотних діапазонів потребуються додаткові смуги пропускання. Діапазон частот FR2 має меншу затримку порівняно з FR1, що робить його ідеальним для критичних додатків, наприклад, автономне керування транспортними засобами.[12]

У табл. 1.2 показано пріоритети використання смуг частот FR2.

Таблиця 1.2 – Використання нових смуг частот вище 6ГГц[5]

Смуга частот	Призначення	Розмір смуги частот на одного оператора
6 – 20 ГГц	Розширення смуг для традиційних Small Cells	100-500 МГц
20 – 40 ГГц	Високошвидкісний доступ, у тому числі для забезпечення покриття у приміщенні	500-1000 МГц
40 – 100 ГГц	Високошвидкісний доступ, зазвичай не у приміщенні і Self backhauling	Більше 1 ГГц

Вибір діапазону частот для 5G NR завжди повинен ґрунтуватися на конкретних потребах кожного застосунку. Для додатків, що потребують глибокого проникнення сигналу та доступності (наприклад, охоплення великих площ), найкращим може бути діапазон FR1. У разі якщо потрібна висока пропускну здатність, низька затримка або більша пропускну здатність на користувача, то варто розглядати можливість побудови мережі у діапазоні FR2.

Оператори мереж комунікацій зазвичай використовують комбінацію фрагментів спектрів FR1 і FR2, щоб задовольнити потреби різних застосунків. Наприклад, оператор може застосовувати фрагмент діапазона FR1 для охоплення

великих площ, а фрагмент спектра частот FR2 – для забезпечення високої пропускної здатності в районах з високою населеністю.

Найбільш вдалим вибором буде FR 1 для подальшої роботи з пристроями Інтернету речей.

## 2. ПАРАМЕТРИ QoS ДЛЯ 5G NR

Якість обслуговування (QoS) - це рівень сервісу, який забезпечує мережа мобільного зв'язку. Параметри QoS визначають, наскільки ефективно мережа може виконати вимоги конкретного застосунку. Технологія 5G NR підтримує широкий спектр параметрів QoS для задоволення потреб різних застосунків. Стандарт 3GPP 23.501 описує такі параметри як частину імплементації QoS для 5G (див. рисунок 2.1). [14]

5G NR (New Radio) призначений для підтримки різноманітних послуг та додатків, кожен із них має свої унікальні вимоги до якості обслуговування (QoS). Врахування цих вимог є ключем до оптимальної роботи мережі та гарантує належний рівень сервісу для всіх видів трафіку. Нижче розглянемо деякі з цих параметрів:

- 5G QoS Identifier (5QI)
- Allocation and Retention Priority (ARP)
- Guaranteed Bit Rate (GFBR)
- Maximum Bit Rate (MBR)
- Packet Delay Budget (PDB)
- Packet Error Rate (PER) [15]

На рис.2.1 показано усі загальні параметри QoS для 5G NR.

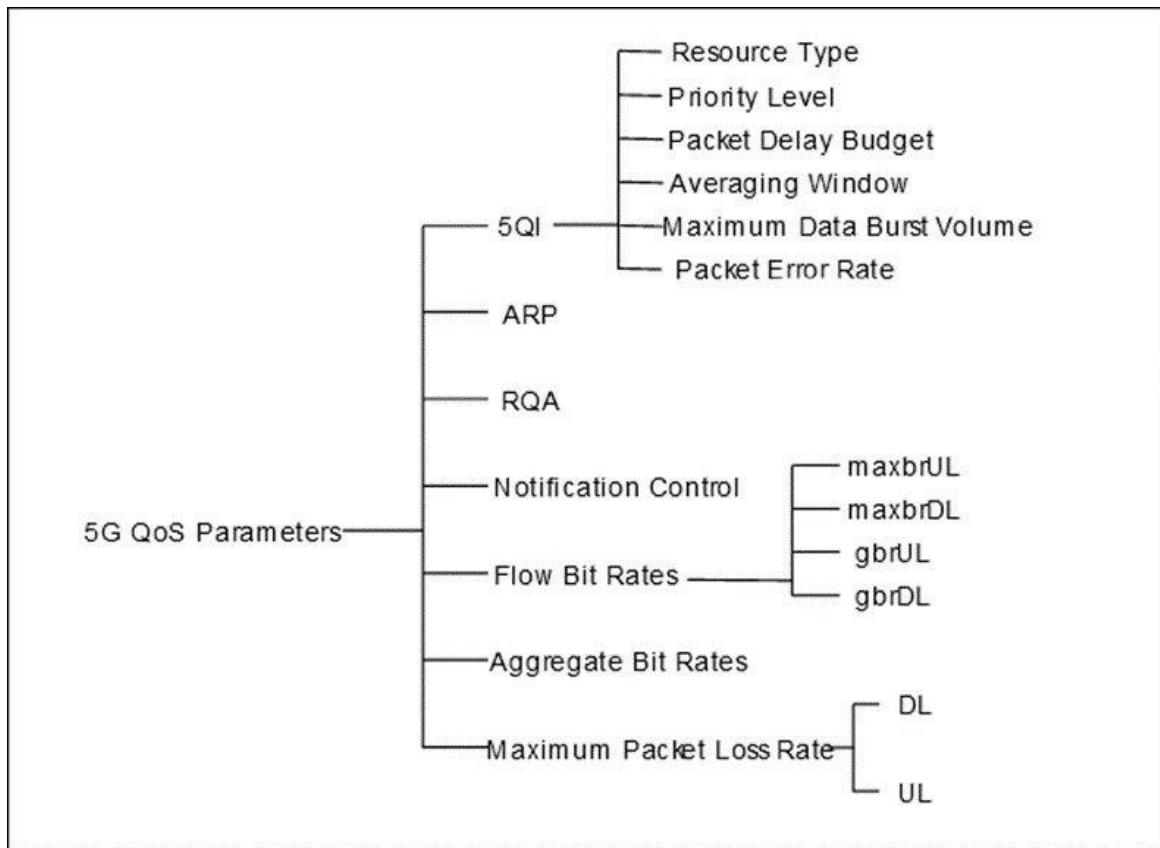


Рис. 2.1. Параметри QoS для 5G NR[15]

## 1. QoS ідентифікатори (5G QoS Identifiers, 5QI)

Ідентифікатори якості обслуговування дозволяють мережам розпізнати потреби кожного потоку даних і застосовувати відповідні налаштування для забезпечення необхідного рівня обслуговування. В стандарті 5G ідентифікатори якості обслуговування поділяються на два основних типи:

Стандартизовані ідентифікатори якості обслуговування (5QI): Заздалегідь визначені параметри, що описують характеристики трафіку, такі як пріоритет, GBR, MBR тощо.

Нестандартизовані ідентифікатори якості обслуговування: Кастомізовані параметри для конкретних потреб операторів або корпоративних мереж.[16]

## 2. Гарантова бітова швидкість (Guaranteed Bit Rate, GBR)

GBR визначає мінімальну пропускну здатність, яка гарантується потоку даних. Цей параметр критично важливий для застосунків, що вимагають стабільної

пропускної здатності, наприклад, для голосових викликів або відеоконференцій. [16]

### 3. Максимальна бітова швидкість (Maximum Bit Rate, MBR)

MBR встановлює ліміт максимальної швидкості, яку може досягнути потік даних. Цей параметр допомагає запобігти перевантаженню мережі від окремих користувачів або застосунків, забезпечуючи справедливий розподіл ресурсів між усіма користувачами. [16]

### 4. Затримка пакета (Packet Delay Budget, PDB)

PDB встановлює максимально допустимий час, за який пакет повинен бути доставлений від джерела до приймача. Затримка є критичним параметром для застосунків, чутливих до затримок, таких як управління транспортними засобами або оперативні системи відгуку. [17]

### 5. Помилка пакета (Packet Error Rate, PER)

PER визначає припустимий відсоток пакетів, які можуть бути втрачені або прийняті з помилками. Низький PER є ключовим для стабільних комунікацій, особливо у критичних сферах застосування, таких як телемедицина або автономне водіння. [18]

### 6. Пріоритизація трафіку

Процес пріоритизації трафіку дозволяє мережам 5G NR встановлювати пріоритети для різних потоків даних, забезпечуючи перевагу критичних застосунків, наприклад екстреним службам, над менш критичними. [19]

Ці параметри і механізми QoS гарантують ефективне задоволення різноманітних потреб застосунків та послуг мереж 5G - від простого перегляду сторінок у мережі до життєво важливих застосунків у режимі реального часу. Оператори зв'язку можуть користуватися параметрами QoS для управління мережею та забезпечення її ефективності для різноманітних додатків.

### 3. АРХІТЕКТУРА МЕРЕЖІ

Для впровадження технології 5G у Інтернеті речей потрібно розробити відповідну архітектуру системи, яка забезпечить ефективне керування ресурсами мережі і надасть високу якість обслуговування для додатків IoT. Архітектура складається з таких компонент:

1. Базові станції (gNodeB)
2. Пристрої IoT
3. Мережева інфраструктура
4. Технології та протоколи

Базові станції є ключовими елементами мережі, що забезпечують бездротовий зв'язок із пристроями IoT. Кожна базова станція підтримує TDD (дуплекс із тимчасовим поділом) для ефективного використання спектру. Також передбачено агрегацію каналів із 16 компонентами несучих для покращення загальної пропускної здатності. Реалізовано підтримку чотирьох MIMO шарів для поліпшення пропускної здатності та надійності з'єднання, модуляцію 64QAM для високої ефективності передачі даних.

Робоча частота складає 3500 МГц, що належить до FR1 діапазону, ідеально підходить для збалансованого покриття та пропускної здатності.

Програмне керування та налаштування параметрів виконується через централізовану систему управління мережею.

Пристрої IoT інтегровані в мережу з періодичною генерацією випадкових даних з використанням асинхронних таймерів для імітації реальних умов.

Забезпечено динамічне підключення до gNodeB, з можливістю вибору станції на основі сили сигналу та навантаження.

Є функціонал шифрування даних для забезпечення безпеки передаваної інформації.

Інфраструктура мережі включає ядро мережі, яке відповідає за обробку даних, маршрутизацію, аутентифікацію користувачів та інші важливі функції.

Також у склад мережевої інфраструктури входять мережеві шлюзи для з'єднання з Інтернетом та іншими зовнішніми мережами, а також системи управління мережею (NMS), які дозволяють динамічно керувати ресурсами та конфігурацією мережі.

У рамках цієї архітектури передбачено використання протоколів забезпечення якості обслуговування (QoS), що дозволяють пріоритизувати трафік; програмно визначені мережі (SDN) для ефективного управління; підтримка розподілу спектру і ресурсів для оптимізації використання частотного діапазону.

Такий підхід не лише задовольняє поточні вимоги стандарту до 5G мереж, але й дає можливість масштабувати та адаптувати до подальших інновацій і змін технології.

## 4. МОДЕЛЮВАННЯ

У розділі описується процес моделювання 5G NR мережі та його особливості з 1 базовою станцією та 10 IoT пристроями, що підключаються до неї. Мова написання коду – C++.

### 4.1. Обробка сигналу

SignalProcessor клас відповідає за три ключові аспекти обробки сигналу у модельованій 5G мережі: кодування, чергування та модуляцію даних. Розглянемо кожен метод детальніше.

Метод encodeData (рис.4.1) призначений для кодування даних за допомогою низькощільного методу перевірки на парність (LDPC), який є стандартним підходом у сучасних телекомунікаційних системах для виправлення помилок.

```

class SignalProcessor {
private:
    std::random_device rd;
    std::mt19937 gen;
    std::uniform_real_distribution<> dis;

public:
    SignalProcessor() : gen(rd()), dis(0, 1) {}

    void encodeData(std::vector<char>& data) {
        // Простий приклад LDPC кодування
        // Тут використано заглушку, в реальних системах це буде складний процес
        std::vector<char> encodedData;
        for (auto& byte : data) {
            // Кодування кожного байту
            encodedData.push_back(byte); // Символ даних
            encodedData.push_back(byte ^ 0xFF); // Приклад контрольного біту
        }
        data.swap(encodedData);
    }
}

```

Рисунок 4.1 – Код методу encodeData у класі SignalProcessor

У нашому випадку кодування виконується шляхом додавання контрольного біта, який представляє собою XOR оригінального байта з 0xFF. Це не є реальною реалізацією LDPC, а лише ілюструє процес.

Метод `interleaveData` (рис. 4.2) використовується для зміни порядку байтів у даних з метою покращення стійкості передачі інформації та виправлення помилок. У коді черговий чергування виконується шляхом перестановки байтів у блоках фіксованого розміру (`blockSize`). Це надає змогу більш ефективно коригувати помилки, які часто з'являються послідовно, стають розподіленими по всьому потоку даних. У нашому випадку для DL `blockSize = 6794` символи, для UL `blockSize = 7278` символів (кількість ефективних бітів поділена на порядок модуляції).

```
void interleaveData(std::vector<char>& data) {
    // Блочне чергування для підвищення стійкості до помилок
    int blockSize = 163062; // Величина блоку для чергування
    if (data.size() > blockSize) {
        std::vector<char> interleavedData(data.size());
        int numBlocks = data.size() / blockSize;
        for (int i = 0; i < blockSize; ++i) {
            for (int j = 0; j < numBlocks; ++j) {
                interleavedData[j * blockSize + i] = data[i * numBlocks + j];
            }
        }
        data.swap(interleavedData);
    }
}
```

Рисунок 4.2 – Код методу `interleaveData` у класі `SignalProcessor`

Метод `modulateData` (рис.4.3).

Модуляція — це процес перетворення інформації з цифрового формату у формат, придатний для передачі по радіоканалу. У коді використовується проста модель QAM (квадратурна амплітудна модуляція), де параметр `modulationOrder` задає кількість різних станів сигналу (наприклад, 64-QAM має 64 стани).

```

void modulateData(std::vector<char>& data, int modulationOrder) {
    // Приклад QAM модуляції
    // Замість реальної QAM модуляції, використовуємо заглушку
    int bitsPerSymbol = static_cast<int>(std::log2(modulationOrder));
    std::vector<char> modulatedData;
    modulatedData.reserve(data.size() * bitsPerSymbol);
    for (auto& byte : data) {
        for (int i = 0; i < bitsPerSymbol; ++i) {
            // Генеруємо псевдо-символи
            modulatedData.push_back(dis(gen) < 0.5 ? '0' : '1');
        }
    }
    data.swap(modulatedData);
}

```

Рисунок 4.3 – Код методу modulateData у класі SignalProcessor

У цьому прикладі кожен символ модулюється, як послідовність випадково обраних бітів, що не є типовим для реальної QAM модуляції, але демонструє процес.

SignalProcessor клас можна інтегрувати у систему gNodeB або кінцевого пристрою для проведення попереднього оброблення даних, включаючи кодування для захисту від помилок, чергування для збільшення стабільності до конкретних помилок у каналі та модуляція для ефективного бездротового зв'язку. Такий підхід дозволяє забезпечити надшвидке та ефективне використання радіоресурсів у складних умовах передачі, характерних для сучасних мобільних мереж.

## 4.2. Моделювання радіоканалу

RadioInterface клас (рис. 4.4) виконує дві основні функції у моделюванні 5G мережі: керує передачею даних через радіоканал і координує управління якістю обслуговування (QoS).

```

class RadioInterface {
public:
    RadioChannel channel;
    QoSManager qosManager;

    RadioInterface(double pathLossExponent, double shadowingVariance, double frequency)
        : channel(pathLossExponent, shadowingVariance, frequency) {}

    void transmitData(BaseStation& baseStation, const std::string& data, const std::string& deviceId) {
        double distance = 100.0; // Припустима відстань у метрах
        double pathLoss = channel.calculatePathLoss(distance);
        double fading = channel.simulateFading();
        double shadowing = channel.simulateShadowing();

        double receivedPower = baseStation.getTransmitPower() - (pathLoss + fading + shadowing);

        if (receivedPower < baseStation.getThreshold()) {
            std::cout << "Data lost in transmission due to high path loss and fading." << std::endl;
        } else {
            baseStation.receiveData(data, deviceId);
        }
    }

    void manageQoS(BaseStation& baseStation) {
        // Управління QoS для приєднаних пристроїв
        std::vector<std::string> devices = baseStation.getConnectedDevices();
        qosManager.manageQoS(devices);
    }
};

```

Рисунок 4.4 – код для RadioInterface класу

RadioChannel channel є екземпляром RadioChannel класу, який визначає затухання сигналу на основі відстані та моделює ефекти затінення і згасання. Параметри ініціалізації включають коефіцієнт затухання (pathLossExponent), варіативність затінення (shadowingVariance) та частоту (frequency), що дозволяє точно моделювати радіоканал у реалістичних умовах зв'язку.

QoSManager qosManager відповідає за керування QoS у мережі. Це включає моніторинг і корегування якості обслуговування для кожного підключеного пристрою на основі потреб QoS, які можуть бути різними для різних типів трафіку (наприклад, дані з медіа потребують вищої пріоритизації порівняно з звичайними даними).

Метод transmitData метод забезпечує передачу даних від gNodeB до пристрою. Враховується затухання сигналу, згасання та затінення для визначення

якості сигналу на приймачі. Якщо отримана потужність сигналу менша за поріг прийому (`baseStation.getThreshold()`), то дані будуть вважатися втраченими, що імітує реальні умови передачі, де високе згасання або згасання може призвести до втрати пакетів.

Метод `manageQoS` метод керує параметрами якості обслуговування для всіх підключених пристроїв. Він використовує інформацію про підключені пристрої через `gNodeB` (`baseStation.getConnectedDevices()`) та розподіляє мережеві ресурси для оптимізації загальної якості обслуговування. Це має важливе значення для 5G мереж, де різним застосункам необхідно враховувати затримку, швидкість і надійність.

`RadioInterface` грає ключову роль у моделюванні реалістичного радіоканалу та керування QoS, як фундаментальний аспект для підтримки IoT застосунків у 5G мережах. Він забезпечує інтеграцію між фізичним радіо середовищем та вищими шарами мережевої архітектури, щоб всі пристрої були забезпечені на необхідному рівні сервісу.

Клас `RadioChannel` (рис. 4.5) охоплює різні аспекти радіопередач, такі як згасання сигналу, затемнення та вигорання, які є ключовими для реалістичного моделювання радіосигналу в різних умовах.

```

class RadioChannel {
public:
    double pathLossExponent;
    double shadowingVariance;
    double carrierFrequency;

    RadioChannel(double exponent, double variance, double frequency)
        : pathLossExponent(exponent), shadowingVariance(variance), carrierFrequency(frequency) {}

    double calculatePathLoss(double distance) {
        // Розширене обчислення затухання сигналу
        return 20.0 * log10(distance) + 20.0 * log10(carrierFrequency) + (44.9 - 6.55 * log10(30)) * log10
    }

    double simulateShadowing() {
        static std::normal_distribution<double> dist(0.0, shadowingVariance);
        static std::mt19937 generator(std::random_device{}());
        return dist(generator);
    }

    double simulateFading() {
        static std::exponential_distribution<double> dist(1.0);
        static std::mt19937 generator(std::random_device{}());
        return -dist(generator); // Значення атенюації в dB
    }
};

```

Рисунок 4.5 – Код для RadioChannel класу

`pathLossExponent` використовується для обчислення експоненційного затухання, яке залежить від відстані. Використовується при розрахунку, як швидко сигнал слабшає зі збільшенням відстані між передавачем і приймачем.

`shadowingVariance` – це параметр, що характеризує варіативність затінення, яке показує випадкові коливання амплітуди сигналу через перешкоди на шляху поширення (прикладом можуть бути будинки).

`carrierFrequency` – частота несучої хвилі на якій працює радіосистема, впливає на послаблення сигналу оскільки вищі частоти піддаються більшому затухання при проходженні через середовище.

Метод `calculatePathLoss(double distance)` розраховує загальне затухання сигналу на заданій відстані. Формула включає логарифмічну залежність від відстані та частоти, що є типовою для моделей затухання у вільному просторі, а також додаткові коригування для врахування урбанізованого середовища.

Метод `simulateShadowing()`: Випадкова генерація значень затінення за нормальним законом розподілу. Це дозволяє імітувати ефекти, коли реальні об'єкти блокують або змінюють шлях поширення сигналу, що призводить до флуктуацій.

Метод `simulateFading()` для імітації згасання за допомогою експоненціального закону розподілу. Згасання часто моделюється як Релеївське згасання для безлінійних середовищ, де сигнал розсіюється різними об'єктами.

`RadioChannel` моделює фізичні процеси радіопередачі, що дозволяє іншим компонентам мережі (наприклад, `RadioInterface`) нормально реагувати на зміни в середовищі передачі. Це забезпечує динамічне налаштування параметрів підключення в залежності від поточних умов радіопередачі.

### **4.3. HARQ**

Клас `HARQProcess` (рис. 4.6) реалізує логіку гібридного автоматичного повторення запиту (HARQ), слугує для забезпечення надійності передачі даних у сучасних мережах зв'язку, особливо в мережах 5G.

```

class HARQProcess {
public:
    std::queue<std::vector<char>> retryQueue;
    int maxRetries;

    HARQProcess(int retries = 3) : maxRetries(retries) {}

    bool transmit(const std::vector<char>& data) {
        // Simulate transmission success or failure
        static std::random_device rd;
        static std::mt19937 gen(rd());
        static std::uniform_int_distribution<> dis(0, 1); // 50% chance of needing retry

        bool success = dis(gen) > 0; // Simulate transmission success (50% chance)
        if (!success) {
            if (retryQueue.size() < maxRetries) {
                retryQueue.push(data);
                std::cout << "Transmission failed, adding to retry queue." << std::endl;
            } else {
                std::cout << "Transmission failed after max retries." << std::endl;
            }
        } else {
            std::cout << "Transmission successful." << std::endl;
        }
        return success;
    }
}

```

Рисунок 4.6 – Ініціалізація класу HARQProcess

retryQueue зберігає пакети даних, які потребують повторної передачі. Це забезпечує механізм буферизації для пакетів, що не були успішно передані.

maxRetries визначає максимальну кількість спроб повторної передачі для кожного пакету. Це обмеження допомагає уникнути нескінченних циклів повторення при постійних помилках передачі.

Метод transmit (рис. 4.6) виконує спроби передати пакет даних. Використовуючи генератор випадкових чисел, він імітує успішне або несправне пересилання. У разі невдачі пакет додається до черги повторних спроб. Успішна передача виводить повідомлення про успіх.

Метод processQueue (рис. 4.7) обробляє всі пакети в черзі повторних спроб. Кожен пакет знову передається через метод transmit. Цей процес триває до тих пір,

поки черга не стане порожньою, гарантуючі, що всі дані спробують бути передані знову згідно з політикою повторних спроб.

```
void processQueue() {  
    while (!retryQueue.empty()) {  
        auto data = retryQueue.front();  
        retryQueue.pop();  
        transmit(data);  
    }  
}
```

Рисунок 4.7 – Метод processQueue класу HARQProcess

HARQ допомагає знизити втрати пакетів та покращити загальну продуктивність системи.

Цей механізм дозволяє системам ефективно розв'язувати проблеми передачі в складних радіочастотних умовах, що є важливим для різноманітних застосувань у Інтернеті речей (IoT).

#### 4.4 Керування QoS

Клас QoSManager відповідає за управління якістю обслуговування (QoS) в мережі та гарантує, що вимоги до затримки та пропускної здатності для кожного пристрою дотримуються.

deviceQoSRequirements – це словник, який ставить у відповідність ідентифікатори пристроїв їхнім вимогам до QoS, таким як затримка або пропускна здатність.

deviceCurrentPerformance – це словник для відстеження поточних показників затримки та пропускної здатності кожного пристрою.

Метод setQoSRequirement установлює або оновлює вимоги QoS для конкретного пристрою (затримка, пропускна здатність чи інший метричний параметр).

Метод `getQoSRequirement` повертає поточну вимогу QoS для заданого пристрою. Якщо вимоги не встановлені, повертає значення помилки або дефолтне значення.

Метод `updateDevicePerformance` оновлює поточні показники продуктивності для пристрою.

Метод `manageQoS` (рис.4.8) виконує основні функція управління QoS, порівнюючи поточну продуктивність зі встановленими вимогами і регулюючи ресурси для підтримки цих вимог.

```
void manageQoS(std::vector<std::string>& activeDevices) {
    // Adjust resources based on current performance compared to requirements
    for (const auto& deviceId : activeDevices) {
        auto it = deviceQoSRequirements.find(deviceId);
        if (it != deviceQoSRequirements.end()) {
            double requiredQoS = it->second;
            auto performance = deviceCurrentPerformance.find(deviceId);

            if (performance != deviceCurrentPerformance.end()) {
                double currentLatency = performance->second.first;
                double currentThroughput = performance->second.second;

                // Check if current performance meets the QoS requirements
                if (currentLatency > requiredQoS || currentThroughput < requiredQoS) {
                    std::cout << "Adjusting resources for Device " << deviceId << std::endl;
                    adjustResources(deviceId, requiredQoS, currentLatency, currentThroughput);
                }
            }
        }
    }
}
```

Рисунок 4.8 – Метод `manageQoS` класу `QoSManager`

Додатковий метод `adjustResources` (рис. 4.9) відповідає за адаптивне керування ресурсами відповідно до поточної продуктивності та встановлених вимог QoS. Його завданням є отримання поточних показників затримки та пропускнуої спроможності, а також оптимізація ресурсів (збільшення пріоритету планування або виділення додаткових ресурсних блоків (PRB)).

```

class QoSManager {
    void adjustResources(BaseStation& baseStation) {
        // Отримання поточних параметрів затримки та пропускної спроможності для кожного пристрою
        std::map<std::string, double> currentLatencies;
        std::map<std::string, double> currentThroughputs;
        // Симуляція отримання даних
        for (const auto& deviceId : baseStation.getConnectedDevices()) {
            // Симуляція вимірювання затримки
            currentLatencies[deviceId] = getRandomLatency();
            // Симуляція вимірювання пропускної спроможності
            currentThroughputs[deviceId] = getRandomThroughput();
        }
        // Перевірка і корегування ресурсів на основі вимог QoS
        for (const auto& deviceId : baseStation.getConnectedDevices()) {
            if (deviceQoSRequirements.find(deviceId) != deviceQoSRequirements.end()) {
                double requiredQoS = deviceQoSRequirements[deviceId];
                double currentLatency = currentLatencies[deviceId];
                double currentThroughput = currentThroughputs[deviceId];
                // Необхідне зменшення затримки
                double latencyImprovementNeeded = currentLatency - requiredQoS;
                // Необхідне збільшення пропускної спроможності
                double throughputImprovementNeeded = requiredQoS - currentThroughput;
                if (latencyImprovementNeeded > 0) {
                    // Збільшення пріоритету планування для зменшення затримок
                    baseStation.increaseSchedulingPriority(deviceId);
                }
                if (throughputImprovementNeeded > 0) {
                    // Збільшення кількості виділених ресурсних блоків (PRBs)
                    baseStation.allocateAdditionalPRBs(deviceId);
                }
            }
        }
    }
}

```

Рисунок 4.9 – Метод adjustResources класу QoSManager

QoSManager забезпечує оптимальну продуктивність, адаптацію до змінних умов та потреб користувачів, а також підтримку стабільності зв'язку в мережі. Його можливості адаптувати ресурси втілюють основний принцип мереж 5G, який ґрунтується на гнучкості та врахуванні різноманітних вимог різних типів трафіку та пристроїв у мережах.

#### 4.5 Керування ресурсами

Клас DynamicResourceAllocator є важливим компонентом у керуванні ресурсами у мережі, забезпечуючи справедливе та ефективне розподілення доступної пропускної здатності між підключеними пристроями на основі їхніх пріоритетів. Детальніше про компоненти та методи класу:

Клас `DynamicResourceAllocator` забезпечує справедливий та ефективний розподіл доступної пропускної здатності між підключеними пристроями на основі їх пріоритетів.

`devicePriorities` – це словник, що зберігає пріоритет кожного пристрою для визначення частки пропускної здатності, яка буде виділена кожному пристрою.

`totalBandwidth` – загальна доступна пропускна здатність мережі, яка має бути розподілена між пристроями.

Метод `DynamicResourceAllocator` (конструктор) ініціалізує об'єкт з вказаною загальною пропускною здатністю.

Метод `setDevicePriority` встановлює або оновлює пріоритет для конкретного пристрою.

Метод `allocateResources` розподіляє пропускну здатність між підключеними пристроями на основі їх пріоритетів. Метод обчислює відсоток загальної пропускної здатності, який повинен бути виділений кожному пристрою, і виконує реальне розподілення ресурсів.

#### **4.6 Моделювання базових станцій**

Клас `BaseStation` є центральним компонентом у модельованій мережі 5G, який виконує кілька функцій від керування з'єднаннями та пропускною здатністю до симуляції циклів часового поділу дуплексу (TDD).

`stationID`: Унікально ідентифікує `gNodeB`.

`stationType`: Описує тип `gNodeB`.

`connectedDevices`: Зберігає ідентифікатори приєднаних пристроїв.

`devicePriorities`: Відображає ідентифікатори пристроїв на їхні пріоритети, впливаючи на те, як розподіляються ресурси.

`lastActivity`: Відстежує останній час активності для кожного з'єданого пристрою для ефективного управління з'єднаннями.

`availableBandwidth`: Представляє загальну пропускну здатність, яку `gNodeB` може розподілити між з'єднаними пристроями.

`mimoLayers`, `modulationOrder`, `codeRate`: Визначають характеристики фізичного шару, які використовуються під час передачі даних.

`carrierFrequency`, `numPRB`: Вказують робочу частоту та кількість фізичних ресурсних блоків.

`overhead`, `symbolDuration`: Представляють додаткові деталі передачі.

`tcpAcceptor`, `udpSocket`: Мережеві сокети для обробки TCP та UDP з'єднань.

`prbUsage`: Вектор, що вказує статус використання кожного PRB.

`tddConfig`, `dlUlPattern`: Управління конфігурацією TDD і шаблоном спуску/підйому.

`harq`: Екземпляр `HARQProcess` для управління повторними передачами.

`radioInterface`: Управляє радіоасpekтами, такими як втрати сигналу та QoS.

Метод `updatePRBUsage` динамічно розподіляє PRB на основі вимог та пріоритетів пристроїв.

Метод `initializeTddPattern` налаштовує шаблон TDD згідно з передвизначеною конфігурацією.

Метод `manageTddCycles` симулює цикли TDD, перемикаючись між спуском та підйомом згідно з шаблоном TDD.

Метод `modulateData` заглушка для модуляції даних, що вказує, де зазвичай відбувається модуляція.

Метод `allocateSpectrum` розподіляє спектр між з'єднаними пристроями на основі їхніх потреб та загальнодоступних PRB.

Метод `startAccept` ініціює прийом нових з'єднань на TCP сокеті.

Метод `handleAccept` управляє новими з'єднаннями після їх встановлення.

Метод `connectDevice` підключає пристрій до `gNodeB` і ініціалізує його пріоритет та відстеження активності.

Метод `sendData` управляє відправкою даних до пристроїв, включаючи спроби повторної передачі через HARQ, коли це необхідно.

Метод `receiveData` управляє вхідними даними від пристроїв, читаючи з сокета.

Метод `processDataQueue` обробляє чергу даних, які потрібно відправити, дотримуючись обмежень пропускної здатності.

Метод `calculateDataBandwidth` оцінює вимоги до пропускної здатності для блоку даних.

`updateLastActivity()`: Оновлює час останньої активності для підключеного пристрою.

`disconnectInactiveDevices()`: Відключає пристрої, які не виявили активність протягом визначеного часу.

Клас `BaseStation` демонструє, як `gNodeB` можуть управляти мережевим трафіком, надавати пріоритет комунікаціям пристроїв та керувати динамічним розподілом ресурсів у реальному середовищі. Крім того, вона поєднує ряд аспектів управління мережею, таких як QoS, HARQ для надійності передачі та TDD для ефективного використання частотного спектру.

#### **4.7 Моделювання IoT пристроїв**

Клас `IoTDevice` представляє пристрій Інтернету речей (IoT) у симуляції мережі 5G, який може генерувати та відправляти дані, а також здійснювати зв'язок з базовою станцією.

`deviceID`: Унікальний ідентифікатор пристрою.

`deviceType`: Тип пристрою, який може описувати його функціональність або роль у мережі.

`isConnected`: Логічний прапорець, який вказує, чи з'єднаний пристрій з `gNodeB`.

`connectedBaseStation`: Вказівник на об'єкт `gNodeB`, до якої підключено пристрій.

`timer`: Таймер для планування періодичної генерації та відправки даних.

`encryptData`: Функція для шифрування даних перед їх відправкою.

Метод `generateAndSendDataPeriodically` генерує дані випадковим чином та відправляє їх `gNodeB`, використовуючи задану функцію шифрування. Відправка даних планується на повторення кожні 5 секунд.

Метод `connectToBaseStation` з'єднує пристрій з `gNodeB`, встановлюючи статус з'єднання на `true` та реєструючи пристрій у `gNodeB`.

Метод `formatDataFor5G` форматує дані відповідно до специфікацій 5G, що може включати трансформацію або підготовку даних перед їхньою передачею.

Метод `sendDataToBaseStation` відправляє дані `gNodeB`. Якщо пристрій підключено, дані відправляються після форматування згідно зі специфікаціями 5G. Якщо з'єднання відсутнє, виводиться повідомлення про відсутність з'єднання.

Метод `disconnect` відключає пристрій від `gNodeB`, встановлюючи статус з'єднання на `false` і відключаючи пристрій у `gNodeB`.

IoT-пристрій має вирішальне значення для моделювання взаємодії між кінцевими пристроями та `gNodeB` у контексті 5G, що підтримує функціональну реалізацію IoT, таких як збирання даних, їх шифрування та надсилання в мережу з урахуванням вимог до якості обслуговування та надійності з'єднання.

У табл. 4.1 наведено швидкість передачі даних та час передачі 1Мбіт даних у змодельованій мережі.

Таблиця 4.1 – Отримані результати моделювання

		1 gNodeB + 1 пристрій	1 gNodeB + 10 пристроїв
Час передачі 1Мбіт	DL	12,64 мс	14,23 мс
	UL	13,11 мс	17,647 мс
Швидкість передачі даних	DL	87,4 Мбіт/с	71,4 Мбіт/с
	UL	81,5 Мбіт/с	66,7 Мбіт/с

Моделювання проводилося для 2 сценаріїв: 1 `gNodeB` + 1 пристрій та 1 `gNodeB` + 10 пристроїв.

## 5. ОЦІНКА ТА АНАЛІЗ ПРОДУКТИВНОСТІ

У розділі розглядаються основні характеристики змодельованої 5G NR мережі та проводиться аналіз продуктивності роботи з IoT пристроями.

У таблиці 5.1 наведено основні параметри змодельованої мережі 5G NR.

Таблиця 5.1 – Характеристики змодельованої 5G мережі

Параметри моделювання	5G NR
Режим	TDD
Кількість агрегованих каналів J	16
Максимальна кількість шарів MIMO v(j)	4
Порядок модуляції Q(j)m	6 біт/символ
Цільова швидкість коду R x [1024]	948
Порядок модуляції Q(j)m	64QAM
Максимальна цільова швидкість коду Rmax	0.92578125
Коефіцієнт масштабування f(j) (3GPP 38.306)[20]	1
Пропускна здатність BW(j), MHz (3GPP 38.104)[21]	BW:20MHz FR1
Накладні ОН(j) витрати для режиму контролю каналів[20]	0,14 FR1 DL 0,08 FR1 UL
Частотний діапазон FR(j) 3GPP 38.104[21]	FR1 (450 MHz – 6000 MHz)
Значення конфігурації оператора μ(j) (3GPP 38.211)[22]	1
Ширина спектру піднесучої μ(j)	30kHz (14 символів/с)
Максимальна кількість PRB для обраних параметрів NbwPRB(j)	51
Середня тривалість символу OFDM у підкадрі для значення μ(i) для нормального циклічного префікса $T_S^\mu = \frac{10^{-3}}{14 \times 2^\mu}$	0.00003571428571428572

Розрахуємо максимально можливу пропускну здатність у режимах DL (downlink) та UL (uplink) для цих мереж.

Розрахунок максимальної швидкості передачі даних 5G NR можна зробити за формулою [4]:

$$\begin{aligned} \text{Швидкість, } \frac{\text{Мбіт}}{\text{с}} &= \\ &= 10^{-6} \cdot \\ &\cdot \sum_{j=1}^J (v_{\text{Layers}}^{(j)} \cdot Q_m^{(j)} \cdot f^{(j)} \cdot R_{\text{max}} \cdot \frac{N_{\text{PRB}}^{\text{BW}(j),\mu} \cdot 12}{T_S^\mu} \cdot (1 - OH^{(j)})) \end{aligned} \quad (5.1)$$

де

$J$  – кількість агрегованих каналів

$v_{\text{Layers}}^{(j)}$  – кількість шарів MIMO

$Q_m^{(j)}$  – порядок модуляції

$f^{(j)}$  – коефіцієнт масштабування

$R_{\text{max}}$  – цільова швидкість коду

$N_{\text{PRB}}^{\text{BW}(j),\mu}$  – максимальна кількість PRB для обраних параметрів у смузі

пропускання

$T_S^\mu = \frac{10^{-3}}{14 \times 2^\mu}$  – середня тривалість символу OFDM у підкадрі для значення  $\mu(i)$

$\mu$  – ширина спектру піднесучої

$OH^{(j)}$  – накладні витрати

Розрахунок базується на стандарті 3GPP TS 38.306: можливості радіодоступу обладнання користувача (UE) і використовує формулу для отримання швидкості передачі даних 5G NR Throughput у DL (низхідній лінії зв'язку) та UL (висхідній лінії зв'язку).

У таблиці 5.2 наведено рораховану пропускну здатність змодельованої мережі 5G NR для низхідної та висхідної ліній зв'язку.

Таблиця 5.2 – Швидкість передачі даних мережі

DL (низхідна лінія)	2795,108 Мбіт/с
UL (висхідна лінія)	2608,989 Мбіт/с

Частота для змодельованої gNodeB становить 3500 МГц, що зазвичай відноситься до діапазону C-band. Він широко використовується для 5G через здатність балансувати пропускну здатність та покриття.

Розрахунок спектральної ефективності змодельованої 5G мережі базується на формулі:

$$\text{Спектральна ефективність 5G} = \frac{\text{5G NR Пропускна здатність,біт/с}}{\text{Смуга пропускання каналу,Гц}} \quad (5.2)$$

Отже, за цією формулою для висхідної лінії спектральна ефективність складає 139,76 біт/с/Гц, а для низхідної – 130,5 біт/с/Гц.

Розмір транспортного блоку знаходиться за наступною формулою стандарту в'язку 3GPP 38.214:

$$N_{info} = N_{RE} R Q_m \nu, \quad (5.3)$$

$$N_{RE} = N_{sc}^{RB} \cdot N_{symb}^{sh} - N_{DMRS}^{PRB} - N_{oh}^{PRB}, \quad N_{RE} = \min(156, N_{RE}) \cdot n_{PRB}$$

$$1) \text{ Якщо } N_{info} \leq 3824, \quad N_{info} = \max\left(24, 2^n \times \left\lceil \frac{N_{info}}{2^n} \right\rceil\right), \text{ де } n = \max(3, \lceil \log_2(N_{info}) \rceil - 6)$$

$$2) \text{ Якщо } N_{info} > 3824, \quad N_{info} = \max\left(3840, 2^n \times \text{round}\left(\frac{N_{info}-24}{2^n}\right)\right),$$

$$\text{де } n = \lceil \log_2(N_{info} - 24) \rceil - 5$$

$$3) \text{ Якщо } R \leq \frac{1}{4}, TBS = 8 \cdot C \cdot \left\lceil \frac{N_{info}+24}{8 \cdot C} \right\rceil - 24, \text{ де } C = \left\lceil \frac{N_{info}+24}{3816} \right\rceil$$

$$4) \text{ Інакше якщо } N_{info} > 8424, TBS = 8 \cdot C \cdot \left\lceil \frac{N_{info}+24}{8 \cdot C} \right\rceil - 24, \text{ де } C = \left\lceil \frac{N_{info}+24}{8424} \right\rceil \text{ інакше } TBS = 8 \cdot \left\lceil \frac{N_{info}+24}{8} \right\rceil - 24$$

Для нашої мережі розмір транспортного блоку (5G NR TBS) становить 48168, а розрахована максимальна пропускна здатність: 96,336 Мбіт/с.

Для розрахунку часу на відправку пакетів в 5G мережі з нашими параметрами можна використати наступні формули:

- 1) Обчислення кількості бітів на символ OFDM. Кожен PRB складається з 12 піднесучих

$$\text{Кількість бітів на PRB на OFDM символ} = 12 \times 6 = 72 \text{ біт} \quad (5.4)$$

- 2) Загальна кількість бітів на слот:

$$\text{Кількість бітів на PRB на слот} = 72 \times 14 \times 51 = 51408 \text{ біт} \quad (5.5)$$

- 3) Розрахунок ефективних бітів на слот

$$\begin{aligned} \text{Ефективних бітів на слот (DL)} &= 51408 \times \frac{948}{1024} \times (1 - 0,14) \\ &= 40\,765,45 \text{ біт} \end{aligned} \quad (5.6)$$

$$\begin{aligned} \text{Ефективних бітів на слот (UL)} &= 51408 \times \frac{948}{1024} \times (1 - 0,08) \\ &= 43\,673,57 \text{ біт} \end{aligned} \quad (5.7)$$

- 4) Обчислення кількості слотів, необхідних для передачі 1 Мбіт (1 000 000 біт):

$$\text{Слоти (DL)} = \frac{1\,000\,000}{40\,765,45} = 24,53 \text{ біт} \quad (5.8)$$

$$\text{Слоти (UL)} = \frac{1\,000\,000}{43\,673,57} = 22,89 \text{ біт} \quad (5.9)$$

- 5) Розрахунок часу на відправку 1Мбіт(1,000,000 бітів) даних:

Кожен слот триває 0,5мс

$$\text{Час для DL} = 24,53 \times 0,5 = 12,265 \text{ мс} \quad (5.10)$$

$$\text{Час для UL} = 22,89 \times 0,5 = 11,445 \text{ мс} \quad (5.11)$$

Таким чином час на відправку 1 Мбіт даних складає приблизно 11,5 – 12,5мс.

Зона покриття 5G мережі значною мірою залежить від використовуваної частоти, потужності передавача, типу антен та топографії місцевості. Чим вища частота, тим менший радіус покриття, але вища пропускна спроможність. На частоті 3500 МГц, якщо мова йде про звичайні урбанізовані умови:

Міські умови: Покриття може становити від 200 до 500 метрів від однієї gNodeB, враховуючи високу щільність забудови та інші перешкоди.

Сільські умови: Покриття може бути значно ширшим, досягаючи кількох кілометрів, залежно від відсутності перешкод та висоти антен.

Розрахунок можна зробити за допомогою формули Фріса для вільного простору, щоб оцінити максимально можливе покриття:

$$PL = 20\log_{10}(d) + 20\log_{10}(f) + 20\log_{10}\left(\frac{4\pi}{c}\right) + L_{extra} \quad (5.10)$$

де

- $PL$  – втрати потужності сигналу (дБ)
- $d$  – відстань між передавачем і приймачем (м)
- $f$  - частота сигналу (Гц)
- $c$  - швидкість світла в вакуумі (м/с)
- $L_{extra}$  – додаткові втрати (дБ), викликані перешкодами, погодними умовами тощо

Параметри для розрахунку:

$$f = 3500 \times 10^6 \text{ Гц}$$

$$c = 3 \times 10^8 \text{ м/с}$$

$L_{extra}$  залежить від умов:

- Міські умови: може становити до 20 дБ в хорошу погоду та до 30 дБ в несприятливу погоду
- Сільські умови: може становити до 10 дБ в хорошу погоду та до 20 дБ в несприятливу погоду.

$PL_{max}$  (максимально припустимі втрати) для 5G зазвичай становить близько 100дБ.

У табл. 5.3 наведена оцінка мінімальної дальності зони покриття для міста та сільської місцевості при різних погодних умовах.

Таблиця 5.3 – Оцінка зони покриття для міських та сільських умов

	Гарна погода	Погана погода
Місто	68м	22м
Село	216м	68м

На рис. 5.1 показано, як впливають перешкоди на зону покриття.



Рисунок 5.1 – Графік залежності відстані зони покриття від перешкод

Видно, що на перешкодах, що мають значення 57дБ, зона покриття становить вже менше 1 метру, а саме 0,96м. Максимально можлива відстань зони покриття з нульовими перешкодами становить 686,62м.

## 6. РЕЗУЛЬТАТИ

У розділі представлені результати розрахунків показників продуктивності змодельованої 5G мережі.

Отримані розрахункові характеристики 5G мережі:

- Максимальна швидкість передачі даних мережі:
  - Низхідна лінія: 2795,108 Мбіт/с
  - Висхідна лінія: 2608,989 Мбіт/с
- Спектральна ефективність:
  - Низхідна лінія: 139,76 біт/с/Гц
  - Висхідна лінія: 130,5 біт/с/Гц
- Максимальна пропускна здатність: 2,8 Гбіт/с
- Загальна пропускна здатність:
  - Низхідна лінія: 2,61 Гбіт/с
  - Висхідна лінія: 2,8 Гбіт/с
- Час на передачу 1 Мбіт даних:
  - Низхідна лінія: 11,445 мс
  - Висхідна лінія: 12,265 мс
- Реальний час передачі 1Мбіт даних
  - 1 gNodeB + 1 пристрій: низхідна лінія 12,64 мс, висхідна лінія 13,11 мс
  - 1 gNodeB + 10 пристроїв: низхідна лінія 14,23 мс, висхідна лінія 17,647мс
- Максимальна дальність зони покриття: 686,62 м

У місті при хорошій погоді зона покриття складає близько 68 м, а при поганій – 22 м. У сільській місцевості ця відстань буде значно більшою: при гарній – 216м, а поганій – 68 м.

## ВИСНОВКИ

Показано, що запровадження технології 5G значно збільшує пропускну здатність мережі, дозволяючи передавати дані зі швидкістю до 2,8 Гбіт/с. Це робить можливим ефективне обслуговування великої кількості IoT пристроїв

Технологія 5G забезпечує малу затримку передачі даних, що є критично важливим для багатьох застосувань IoT, таких як автономні транспортні засоби та телемедицина. Розрахунки показують, що час на передачу 1 Мбіт даних становить від 11,5-12,5 мс. На практиці цей час може змінюватися залежно від перешкод, розподілу пропускну здатності між більшою кількістю пристроїв, навантаження на мережу у пікові години. Ці показники є значно кращими за мережі на основі попередніх поколінь.

Використання 5G дозволяє забезпечити високу надійність з'єднань навіть у складних умовах. Це досягається завдяки застосуванню сучасних методів кодування, модуляції та обробки сигналів, а також механізмів автоматичного повторного запиту (HARQ). Реалістичне моделювання радіоканалу з урахуванням затухання, затінення та згасання сигналів дозволяє ефективно адаптувати параметри мережі до змінних умов середовища, забезпечуючи стабільність зв'язку.

Використання частоти 3500 МГц забезпечує значну зону покриття. В умовах міської забудови покриття може сягати 68 м при гарній погоді та 22 м при поганій. У сільській місцевості ці показники зростають до 216 м та 68 м відповідно.

Впровадження мережі 5G відкриває нові можливості для Інтернету речей завдяки скороченню часу затримки, покращенню швидкості передачі даних та збільшенню кількості можливих одночасних підключень. Це дозволяє пристроям IoT більш ефективно взаємодіяти у реальному часі.

Подальше дослідження у напрямку оптимізації алгоритмів керування ресурсами, забезпечення безпеки та конфіденційності, а також розширення мережевої інфраструктури може сприяти покращенню впровадження та ефективності технології 5G у контексті Інтернету речей.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. G. Singh, J. Singh, D. Mitra and C. Prabha, "A Roadmap Toward Prospects for IoT Enabled 5G Networks," 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2023, pp. 1405-1410, doi: 10.1109/ICCMC56507.2023.10084183.
2. M. Agiwal, H. Kwon, S. Park and H. Jin, "A Survey on 4G-5G Dual Connectivity: Road to 5G Implementation," in IEEE Access, vol. 9, pp. 16193-16210, 2021, doi: 10.1109/ACCESS.2021.3052462.
3. L. Chettri and R. Bera, "A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems," in IEEE Internet of Things Journal, vol. 7, no. 1, pp. 16-32, Jan. 2020, doi: 10.1109/JIOT.2019.2948888.
4. How 5G is Transforming the Internet of Things (IoT). URL: <https://medium.com/@abidaparveen0124/how-5g-is-transforming-the-internet-of-things-iot-f0cf6925ec62> (дата звернення: 22.10.2023).
5. Dahlman, E., Parkvall, S., and Skold, J. (2020). 5G NR (2nd ed.). Elsevier Science. Retrieved from <https://www.perlego.com/book/1896253/5g-nr-the-next-generation-wireless-access-technology-pdf> (Original work published 2020).
6. Hiroyuki Atarashi; Mikio Iwamura; Satoshi Nagata; Takehiro Nakamura; Antti Toskala, "5G Targets and Standardization," in 5G Technology: 3GPP Evolution to 5G-Advanced , Wiley, 2024, pp.13-26, doi: 10.1002/9781119816058.ch2.
7. 5G unlocking the true potential of IoT. <https://www.iotinsider.com/smart-world/5g-unlocking-the-true-potential-of-iot> (дата звернення: 13.03.2024).
8. S. Mishra, A. Chaudhary and A. Kumar, "Recent advances in 5G and beyond," 2022 7th International Conference on Computing, Communication and Security (ICCCS), Seoul, Korea, Republic of, 2022, pp. 1-4, doi: 10.1109/ICCCS55188.2022.10079254.
9. 5G Malaysia Progressing Humanity, National 5G Task Force Report, 5G Key Challenges and 5G Nationwide Implementation Plan, December 2019. URL:

<https://www.mcmc.gov.my/skmmgovmy/media/General/pdf/The-National-5G-Task-Force-Report.pdf>.

10.X. Wan, H. Ghazzai and Y. Massoud, "Leveraging Personal Navigation Assistant Systems Using Automated Social Media Traffic Reporting," 2020 IEEE Technology & Engineering Management Conference (TEMSCON), Novi, MI, USA, 2020, pp. 1-6, doi: 10.1109/TEMSCON47658.2020.9140144.E.

11.RF and 5G new radio: top 5 questions answered. URL: <https://www.exfo.com/es/recursos/blog/rf-5g-new-radio-top-5-questions/> (дата звернення: 23.10.2023).

12.R. Dilli, "Analysis of 5G Wireless Systems in FR1 and FR2 Frequency Bands," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 2020, pp. 767-772, doi: 10.1109/ICIMIA48430.2020.9074973.

13.5G Radio Spectrum and WRC-19. URL: <https://www.mpirical.com/blog/5g-radio-spectrum-and-wrc-19> (дата звернення: 23.10.2023).

14.3GPP TS 23.501 version 17.5.0 Release 17: 5G; System architecture for the 5G System (5GS).

15.5G NR QoS Parameters and 5G QoS characteristics. URL: <https://www.prodevelopertutorial.com/5g-nr-qos-parameters-and-5g-qos-characteristics/> (дата звернення: 18.10.2023).

16.Exploring 5G Quality of Service (QoS): Unlocking the Potential of Next-Generation Networks. URL: <https://hemantra.medium.com/exploring-5g-quality-of-service-qos-unlocking-the-potential-of-next-generation-networks-2166ca2f7302> (дата звернення: 10.09.2023).

17.Mamane, Asmae & Fattah, Mohammed & El Ghazi, Mohammed & Bekkali, Moulhime. (2022). Packet Delay Budget-Based Scheduling Approach for 5G Time Division Duplex. 10.1007/978-3-031-02447-4\_33.

18.V. Inzillo, F. D. Rango and A. A. Quintana, "Packet Error Rate and Channel Performance Evaluation in 5G Wireless Networks with Massive MIMO Module

Extending Omnet++," 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2019, pp. 1-6, doi: 10.1109/CCNC.2019.8651743.

19.Kamal, Muhammad Ayoub et al. "Resource Allocation Schemes for 5G Network: A Systematic Review." Sensors (Basel, Switzerland) vol. 21,19 6588. 2 Oct. 2021, doi:10.3390/s21196588.

20. (3GPP TS 38.306 version 17.0.0 Release 17: User Equipment (UE) radio access capabilities.

21.3GPP TS 38.104 version 16.6.0 Release 16: 5G; NR; Base Station (BS) radio transmission and reception.

22.3GPP TS 38.211 version 16.2.0 Release 16: 5G; NR; Physical channels and modulation.