

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий інститут телекомунікаційних систем

Кафедра телекомунікацій

До захисту допущено:

Завідувач кафедри

_____ Сергій КРАВЧУК

«__» _____ 2025 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія та програмування
інфокомунікацій»**

спеціальності 172 «Телекомунікації та радіотехніка»

**на тему: «Побудова телекомунікаційних мереж на базі цифрових
двійників»**

Виконав:

студент ІV курсу, групи ТЗ- 12
Смельянов Михайло Васильович

Керівник:

Доцент кафедри ТК НН ІТС, к.т.н. доцент,
Живков Олександр Петрович

Консультант:

Професор кафедри ТК НН ІТС, д.т.н, професор,
Романов Олександр Іванович

Рецензент:

Доцент кафедри ЕКІР НН ІТС, к.т.н. доцент,
Бердников Олег Михайлович

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій КРАВЧУК

«__» _____ 2025 р.

ЗАВДАННЯ

на дипломну роботу студенту

Ємельянову Михайлу Васильовичу

1. Тема роботи «Побудова телекомунікаційних мереж на базі цифрових двійників», керівник роботи Живков, Олександр ,Петрович, к.т.н., доцент, затверджені наказом по університету від «26» травня 2025 р. № 1755-с.

2. Термін подання студентом роботи 9 червня 2025 р.

3. Вихідні дані до роботи:

1. Поняття цифрових двійників та їх роль в технологіях

2. Використання цифрових двійників для управління мережами 5G

4. Зміст роботи:

1. Проаналізувати поняття, структуру та принципи функціонування цифрових двійників

2. Реалізація цифрових двійників у телекомунікаційних мережах стандартів 5G/6G

3. Дослідити методи оптимізації та захисту інформації в цифрових двійниках телекомунікаційних мереж

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

Плакат №1 (слайд) Тема роботи;

Плакат №2 (слайд) Мета роботи, об'єкт, предмет та актуальність дослідження;

Плакат №3 (слайд) Основні поняття та принципи цифрових двійників (DT);

Плакат №4 (слайд) Архітектура цифрових двійників у телекомунікаціях;

Плакат №5 (слайд) Інтеграція цифрових двійників у мережі 5G/6G;

Плакат №6 (слайд) Застосування глибокого навчання з підкріпленням (DRL) для оптимізації мереж;

Плакат №7 (слайд) Впровадження цифрових двійників у телекомунікаціях для управління ресурсами мережі;

Плакат №8 (слайд) Використання цифрових двійників для зниження затримок та покращення пропускної здатності;

Плакат №9 (слайд) Висновки та перспективи використання цифрових двійників у майбутніх мережах;

Плакат №10 (слайд) Висновки;

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Поняття цифрових двійників та їх роль в технологіях	Професор кафедри ТК НН ІТС., д.т.н, професор, Романов Олександр Іванович		

7. Дата видачі завдання 06.11.2024 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	06.11.2024	виконано
2	Огляд літератури	03.02.2025 - 03.03.2025	виконано
3	Розгляд основних понять пов'язаних з технологією цифрового двійника	04.03.2025 - 16.03.2025	виконано
4	Огляд використання цифрових двійників для управління мережами	17.03.2025 - 22.03.2025	виконано
5	Дослідження впровадження цифрових двійників для управління мережами	23.03.2025 - 14.04.2025	виконано
6	Огляд структури мереж для цифрового двійника	15.04.2025 - 24.04.2025	виконано

7	Огляд оптимізація нарізки мереж 5G/6G за допомогою NTN	25.04.2025 - 31.05.2025	виконано
8	Оформлення дипломної роботи	01.06.2025 -07.06.2025	виконано
9	Отримання допуску до захисту	09.06.2025	виконано

Студент

Михайло ЄМЕЛЬЯНОВ

Керівник

Олександр ЖИВКОВ

РЕФЕРАТ

Текстова частина дипломної роботи: 120 стр, 60 рис, 40 джерел.

Метою цієї дипломної роботи є розробка та обґрунтування методів побудови телекомунікаційних мереж на базі цифрових двійників для забезпечення ефективного моніторингу, прогнозування та оптимізації їх роботи.

У першому розділі було розглянуто основні принципи, структуру та особливості функціонування цифрових двійників, їх роль у моделюванні та управлінні складними мережевими системами. Описано переваги та перспективи впровадження DT у сфері телекомунікацій. Другий розділ присвячений аналізу архітектури цифрових двійників та їх інтеграції у мережі 5G/6G. Було розглянуто моделі синхронізації цифрових двійників з фізичними мережами та застосування технологій штучного інтелекту, зокрема глибокого навчання з підкріпленням (Deep Reinforcement Learning, DRL), для оптимізації ресурсів мережі та управління трафіком. У третьому розділі запропоновано фреймворк DT-DRL, який використовує алгоритм глибокого детерміністичного градієнта політики (DDPG) для динамічного управління ресурсами мережі, що дозволяє суттєво зменшити затримки та покращити ефективність використання пропускної здатності.

КЛЮЧОВІ СЛОВА: ЦИФРОВИЙ ДВІЙНИК, DIGITAL TWIN, 5G, 6G, ТЕЛЕКОМУНІКАЦІЙНА МЕРЕЖА, ГЛИБОКЕ НАВЧАННЯ З ПІДКРІПЛЕННЯМ, DDPG, DRL, ШТУЧНИЙ ІНТЕЛЕКТ, ОПТИМІЗАЦІЯ, СИНХРОНІЗАЦІЯ.

ABSTRACT

The text part of the thesis: 120 pages, 60 figures, 40 references.

The purpose of this thesis is the development and justification of methods for constructing telecommunications networks based on digital twins to ensure effective monitoring, forecasting, and optimization of their operation.

The first chapter explores the fundamental principles, structure, and operational features of digital twins, emphasizing their role in modeling and managing complex network systems. The advantages and future perspectives of implementing digital twins in telecommunications are outlined. The second chapter analyzes the architecture of digital twins and their integration into 5G/6G networks. Models for synchronizing digital twins with physical networks and the application of artificial intelligence technologies, particularly Deep Reinforcement Learning (DRL), for network resource optimization and traffic management, are discussed. In the third chapter, a DT-DRL framework is proposed, utilizing the Deep Deterministic Policy Gradient (DDPG) algorithm for dynamic network resource management, significantly reducing latency and enhancing bandwidth utilization efficiency.

KEYWORDS: DIGITAL TWIN, 5G, 6G, TELECOMMUNICATION NETWORK, DEEP REINFORCEMENT LEARNING, DDPG, DRL, ARTIFICIAL INTELLIGENCE, OPTIMIZATION, SYNCHRONIZATION.

ЗМІСТ

ВСТУП	11
РОЗДІЛ 1 ПОНЯТТЯ ЦИФРОВИХ ДВІЙНИКІВ ТА ЇХ РОЛЬ В ТЕХНОЛОГІЯХ.....	13
1.1. Історія розвитку та основні етапи впровадження цифрових двійників	13
1.2 Загальні принципи побудови та функціонування цифрових двійників	16
Висновки	18
РОЗДІЛ 2 ОСНОВНІ КОМПОНЕНТИ ТА АРХІТЕКТУРА ЦИФРОВИХ ДВІЙНИКІВ ДЛЯ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖ.....	19
2.1. Моделювання фізичних компонентів мережі	19
2.2. Архітектури цифрових двійників	26
Висновки	29
РОЗДІЛ 3 ВИКОРИСТАННЯ ЦИФРОВИХ ДВІЙНИКІВ ДЛЯ УПРАВЛІННЯ МЕРЕЖАМИ 5G	30
3.1. Особливості управління мережами 5G за допомогою цифрових двійників	30
3.2. Реальні приклади впровадження цифрових двійників у мережах 5G	36
3.3. Проектування цифрового двійника мережі для мережі радіодоступу в 5G	41
Висновки	58
РОЗДІЛ 4 ЦИФРОВИЙ ДВІЙНИК ДЛЯ МЕРЕЖ ТА ПЕРСПЕКТИВА МОДЕЛЮВАННЯ ПРОДУКТИВНОСТІ НА ОСНОВІ ДАНИХ	59
4.1. Вимоги до моделей продуктивності	59
4.2. Вибрані моделі продуктивності.....	64
4.3. Виклики та можливості	72
Висновки	75
РОЗДІЛ 5 ЦИФРОВИЙ ДВІЙНИК O-RAN.....	76
5.1. Емулятор бездротової мережі colosseum.....	76
5.2. Наскрізне відкрите твінінгове дослідження RAN у колізеї.....	78
5.3. Радіочастотний твінінг	81
5.4. Автоматизація для цифрового двійника Colosseum.....	84
5.5. Поєднання реального світу та цифрового двійника	87
5.6 Приклади використання Colosseum як Open RAN Digital Twin.....	90
5.7 Colosseum для тестування O-Ran та інтеграція	99

Висновки	100
РОЗДІЛ 6 ОПТИМІЗАЦІЯ НАРІЗКИ МЕРЕЖ 5G/6G ЗА ДОПОМОГОЮ NTN	102
6.1. Модель системи.....	102
6.2. Запропонована структура ШІ-керованої DT	106
6.3. Результати моделювання.....	109
Висновки	112
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ	114
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	116

ПЕРЕЛІК СКОРОЧЕНЬ

DT	Digital Twin – Цифровий двійник
NDT	Network Digital Twin – Цифровий двійник мережі
DRL	Deep Reinforcement Learning – Глибоке навчання з підкріпленням
DDPG	Deep Deterministic Policy Gradient – Глибокий детерміністичний градієнт політики
RL	Reinforcement Learning – Навчання з підкріпленням
AI	Artificial Intelligence – Штучний інтелект
XAI	Explainable Artificial Intelligence – Пояснюваний штучний інтелект
ML	Machine Learning – Машинне навчання
UE	User Equipment – Абонентське обладнання
RAN	Radio Access Network – Радіомережа доступу
RIC	RAN Intelligent Controller – Інтелектуальний контролер RAN
NTN	Non-Terrestrial Network – Позаземна мережа
RT	Real-Time – У реальному часі
FBS	Flying Base Station – Літальна базова станція
LXC	Linux Containers – Лінукс-контейнери
KPI	Key Performance Indicator – Ключовий показник ефективності
IAB	Integrated Access and Backhaul – Інтегрований доступ і зворотна передача
IEEE	Institute of Electrical and Electronics Engineers – Інститут інженерів електротехніки та електроніки
SLA	Service Level Agreement – Угода про рівень обслуговування

SDR	Software Defined Radio – Радіо, визначене програмно
SINR	Signal to Interference plus Noise Ratio – Співвідношення сигнал/шум+перешкоди
URLLC	Ultra-Reliable Low-Latency Communications – Надійне з'єднання з малою затримкою
MIMO	Multiple Input Multiple Output – Багатоантенна технологія
MAC	Medium Access Control – Контроль доступу до середовища
TCP	Transmission Control Protocol – Протокол керування передачею
VPN	Virtual Private Network – Віртуальна приватна мережа
API	Application Programming Interface – Програмний інтерфейс додатка
MQTT	Message Queuing Telemetry Transport – Протокол телеметрії через чергу повідомлень
ШІ	Штучний інтелект
БС	Базова станція
ТБ	Терабайти
IAB	Integrated Access and Backhaul – Інтегрований доступ і зворотна передача
OAI	OpenAirInterface.
OFDM	Orthogonal Frequency-Division Multiplexing - метод множинного доступу та мультиплексії
CSI	Channel State Information
gBS	gNodeB глобальна базова станція в мережах 5G
xApps	Extended Applications
CU	Centralized Unit - Централізовані блоки
DU	Distributed Unit - Розподілені блоки
UE	User Equipment- Обладнання користувача
SMO	Service Management and Orchestration

ВСТУП

З впровадженням новітніх технологій, таких як 5G та Інтернет речей (IoT), телекомунікаційні мережі стають значно складнішими, що створює нові виклики для операторів. Виникає потреба в точних і гнучких моделях, здатних оперативно реагувати на зміни в мережі, виявляти можливі збої та оптимізувати роботу в реальному часі. Цифрові двійники дозволяють створити такі моделі, що забезпечують ефективний моніторинг та управління мережами, сприяють підвищенню якості обслуговування (QoS) та зниженню часу на відновлення після технічних несправностей. Це дозволяє значно зменшити вплив аварійних ситуацій на якість обслуговування кінцевих користувачів і забезпечити високу доступність сервісів. Цифрові двійники, інтегровані у систему управління мережею, сприяють підвищенню адаптивності та стабільності телекомунікаційної інфраструктури, забезпечуючи її стійкість до збоїв і здатність до швидкого самовідновлення. Завдяки цьому оператори зв'язку можуть досягти вищого рівня SLA (Service Level Agreement) та зменшити витрати на усунення інцидентів. Ураховуючи ці переваги, використання цифрових двійників у телекомунікаційних мережах набуває все більшої актуальності і є перспективним напрямом для покращення операційних процесів у цій галузі. У майбутньому цифрові двійники здатні стати основою для побудови самовідновлюваних, адаптивних та інтелектуальних мереж, які зможуть самостійно приймати рішення на основі аналізу великих обсягів даних, автоматично усувати аномалії та адаптуватися до змінних умов експлуатації. Це особливо важливо в умовах швидко спрощенню процесів технічного супроводу. Крім цього, цифрові двійники забезпечують підвищену прозорість усіх операційних процесів і дозволяють реалізовувати механізми предиктивного обслуговування, коли потенційні проблеми виявляються ще до їх фактичного виникнення. Таким чином, цифрові двійники відіграють роль цифрового мосту між фізичною мережею та аналітичною надбудовою, що підвищує якість експлуатації та підтримки в умовах масштабних і динамічних мережевих

середовищ. Дана робота належить до сфери телекомунікаційних технологій, зокрема до підходів у проектуванні та управлінні телекомунікаційними мережами за допомогою цифрових двійників. Це включає в себе аналіз, розробку та впровадження моделей цифрових двійників для різних типів мереж, включаючи 5G, IoT та інші сучасні технології, що зростають у популярності. Метою цієї дипломної роботи є розробка та обґрунтування методів побудови телекомунікаційних мереж на базі цифрових двійників для забезпечення ефективного моніторингу, прогнозування та оптимізації їх роботи.

РОЗДІЛ 1

ПОНЯТТЯ ЦИФРОВИХ ДВІЙНИКІВ ТА ЇХ РОЛЬ В ТЕХНОЛОГІЯХ

1.1. Історія розвитку та основні етапи впровадження цифрових двійників

На мій погляд, digital twin - це набір функціональних блоків, які реалізовані програмно і вирішують ті ж самі завдання, що і реальний фізичний об'єкт. Це віртуальне представлення фізичних об'єктів і процесів, складається з трьох основних компонентів: фізичного об'єкта в реальному просторі, цифрового близнюка у формі програмного забезпечення та даних, які пов'язують ці два елементи. Цифрові близнюки вже використовуються як для зниження витрат, так і для підвищення ефективності в багатьох галузях, і їхній потенціал для ще більшого вдосконалення можливостей стає дедалі очевиднішим.

Розглянемо інші визначення, які надають всесвітньо відомі компанії: Наприклад, IBM. Цифровий двійник - це віртуальне представлення об'єкта або системи, розроблене для точного відображення фізичного об'єкта. Він охоплює життєвий цикл об'єкта, оновлюється на основі даних у режимі реального часу та використовує моделювання, машинне навчання та логічні методи для прийняття рішень [1].

Amazon Web Services. Цифровий двійник - це віртуальна модель фізичного об'єкта. Вона охоплює життєвий цикл об'єкта та використовує дані в режимі реального часу, що надсилаються з датчиків на об'єкти, для моделювання його поведінки та моніторингу операцій. Цифрові двійники можуть відтворювати багато реальних об'єктів, від окремих одиниць обладнання на заводі до повноцінних установок, таких як вітрові турбіни та цілі міста. Технологія цифрових двійників дозволяє контролювати роботу активу, виявляти потенційні несправності та приймати більш обґрунтовані рішення щодо технічного обслуговування та життєвого циклу [2].

Autodesk. Цифровий двійник - це віртуальне представлення фізичного

об'єкта, системи або середовища. Завдяки актуальним даним, зібраним в одному місці, цифровий двійник розвивається разом із потоком інформації від датчиків, інформаційних моделей будівель, об'єктів Інтернету речей тощо [3].

Університет Тампере, лабораторія прогнозного суспільства та аналізу даних. Цифровий двійник - це інтегроване багатофізичне, багатомасштабне, ймовірнісне моделювання побудованого транспортного засобу або системи, яке використовує найкращі доступні фізичні моделі, оновлення датчиків, історію парку тощо, щоб відобразити термін служби відповідного літаючого двійника [4].

В охороні здоров'я цифровий двійник позначає бачення комплексного віртуального інструменту, який послідовно та динамічно інтегрує клінічні дані, отримані з часом для окремої людини, використовуючи механістичні та статистичні моделі. Це запозичує, але розширює концепцію цифрового двійника, що використовується в машинобудівних галузях, де представлення фізичної системи, такої як двигун або вітрова електростанція, використовуються для оптимізації процесів проектування або управління, з підключенням у режимі реального часу між фізичною системою та моделлю [5].

Перший поштовх до розробки цифрових двійників виник у 1970 році, коли NASA використало дзеркальні системи для порятунку постраждалої космічної місії «Аполлон-13». Наступний великий прорив відбувся 33 роки тому, коли американський професор доктор Майкл Гривз обговорював ідею на університетському курсі. Нижче перелічені основні віхи в розвитку теми цифрових двійників, визначені GlobalData [6]:

- 1970 - NASA використало дзеркальні системи - попередники цифрових двійників - для порятунку космічної місії «Аполлон-13».
- 1999 - Кевін Ештон з Procter & Gamble ввів термін Інтернет речей (IoT).
- 2003 - Доктор Майкл Гривз обговорював ідею цифрових двійників на університетському курсі з управління життєвим циклом продукту.
- 2010 - Ціни на датчики почали падати.
- 2010 - Концепцію цифрового двійника назвав Джон Вікерс з NASA у

звіті про дорожню карту за 2010 рік.

- 2013 - Хмарні обчислення набули достатньої популярності, щоб підтримати впровадження цифрових двійників.
- 2014 - Стенфордський університет та Hewlett Packard Enterprise (HPE) співпрацювали над інтегративним симулятором функції серця людини.
- 2014 - Сінгапур запустив план вартістю 54 мільйони доларів зі створення Віртуального Сінгапуру, динамічної тривимірної цифрової моделі міста.
- 2016 -Компанія Arctic Wind, що займається вітровими електростанціями, використовувала цифрових двійників для відстеження потреб у технічному обслуговуванні вітрових турбін.
- 2017 - Для підтримки використання технологій у будівництві Великої Британії було створено Центр цифрової забудови Британії (CDBB).
- 2017 - Цифрова платформа Huawei для розумних міст включала ідею створення цифрового двійника міста.
- 2018 -CDBB запустив принципи Gemini для досягнення консенсусу щодо розробки національного цифрового двійника Великої Британії.
- 2018 -Сінгапур оголосив про завершення проекту віртуального цифрового двійника Сінгапуру.
- 2018 - Lloyd's Register підкреслив роль цифрових двійників у своїй системі гарантування для морської та шельфової галузі.
- 2018 - Футуролог Томас Фрей передбачив, що до 2022 року 85% платформ Інтернету речей включатимуть моніторинг цифрових двійників.
- 2018 - Інженерні та технологічні організації Великої Британії проводять Національний день цифрових двійників для стимулювання співпраці в галузі цифрових двійників.
- 2019 - ВР заявила на конференції з офшорних технологій, що цифрові двійники допомогли їй заощадити мільярди доларів.

- 2019 -Object Management Group, ANSYS, Dell, Lendlease та Microsoft запустили Консорціум цифрових двійників.
- 2020 - Sidewalk Labs представила цифрового двійника 35-поверхової дерев'яної будівлі для свого запланованого «розумного міста» в Торонто.
- 2020 - В Австралії уряд Нового Південного Уельсу запустив просторового цифрового двійника Західного Сіднея.
- 2020 - Національний інститут стандартів і технологій США опублікував структуру для цифрового двійника у виробництві.
- 2020 - Наукова рада при президенті США наголосила на важливості цифрових двійників на фабриках майбутнього.
- 2020 - CDBB опублікував звіт, у якому досліджується роль цифрових двійників у створенні системи управління інформацією.
- 2020 - Sidewalk Labs скасувала запланований проект «розумного міста» в Торонто.
- 2021 рік - За оцінками, до 2021 року половина всіх великих промислових компаній використовуватимуть цифрових двійників.

1.2 Загальні принципи побудови та функціонування цифрових двійників

Цифрові двійники функціонують на основі постійної двосторонньої взаємодії між фізичним об'єктом і його цифровою копією. Якщо дані передаються між існуючим фізичним об'єктом та цифровим об'єктом, і вони повністю інтегровані в обох напрямках, це являє собою еталонний цифровий двійник. Зміна, внесена до фізичного об'єкта, автоматично призводить до зміни до цифрового об'єкта і навпаки. На відміну від цифрової моделі, у якій не повинно бути автоматичного обміну даними між фізичною та цифровою моделлю, та цифрової тіні, де зміна стану фізичного об'єкта призводить до зміни цифрового об'єкта, а не навпаки [7].

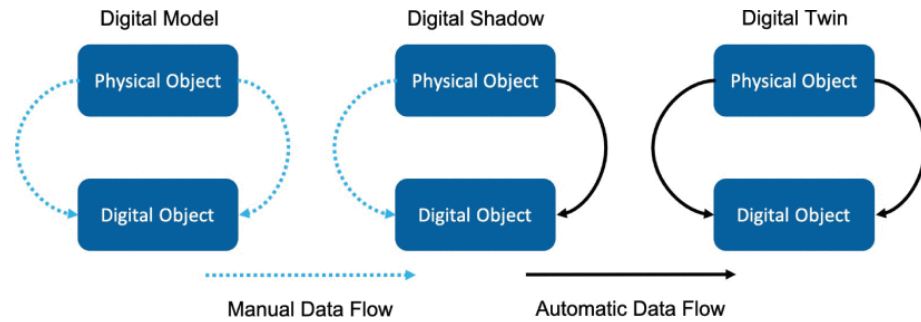


Рис.1.1. Візуалізація рівнів інтеграції для цифрової моделі, цифрової тіні та цифрового двійника [7]

Основними принципами побудови цифрових двійників є забезпечення постійного надходження та обробки даних у реальному часі від різноманітних сенсорів, IoT-пристроїв, SCADA-систем або інших джерел телеметричної інформації; забезпечення ідентичності структури, за якої цифрова модель максимально точно відображає топологію, функціональні зв'язки та параметри фізичного об'єкта; використання аналітичних моделей різного рівня складності - від простих алгоритмів до складних систем на базі штучного інтелекту - для інтерпретації та аналізу отримуваних даних; наявність зручного й інтерактивного інтерфейсу користувача, який дозволяє у реальному часі взаємодіяти з цифровим двійником через візуальні панелі або панелі керування; а також реалізація механізмів зворотного зв'язку, що дають змогу результатам аналізу чи симуляцій безпосередньо впливати на реальну фізичну систему, наприклад, шляхом автоматичного регулювання параметрів її роботи. Типовий цифровий двійник складається з трьох рівнів: фізичного рівня, що включає датчики та інші фізичні компоненти, комунікаційного рівня для передачі даних і цифрового рівня для аналітики, візуалізації та прийняття рішень. Завдяки цим принципам цифрові двійники дозволяють в режимі реального часу: ідентифікувати аномалії, прогнозувати навантаження, планувати технічне обслуговування, оптимізувати конфігурацію системи [7].

Висновки

Цифрові двійники є потужним інструментом для моделювання та управління фізичними системами в реальному часі. Вони складаються з трьох основних компонентів: фізичного об'єкта, цифрової моделі та даних, що забезпечують двосторонню інтеграцію між реальним і віртуальним світом. Завдяки цьому цифрові двійники дозволяють точно відображати стан об'єктів і процесів, забезпечуючи їх моніторинг, прогнозування та оптимізацію в реальному часі. Технологія цифрових двійників швидко розвивається, і сьогодні активно використовується у різних галузях, таких як телекомунікації, енергетика, охорона здоров'я та виробництво. Вона дозволяє знижувати витрати на технічне обслуговування, підвищувати ефективність роботи систем і знижувати ризики, оскільки дає можливість оперативно виявляти проблеми та оптимізувати роботу мереж. Цифрові двійники використовують дані з різних джерел, таких як сенсори та IoT-пристрої, що дозволяє здійснювати їх оновлення в реальному часі. Принципи побудови цифрових двійників забезпечують високу точність і ефективність у прийнятті рішень, що робить їх важливим інструментом для розвитку сучасних технологій. Вони сприяють оптимізації роботи складних мереж і систем, особливо в умовах швидкого розвитку технологій 5G та IoT, що робить їх перспективними для використання в майбутніх мережах.

РОЗДІЛ 2

ОСНОВНІ КОМПОНЕНТИ ТА АРХІТЕКТУРА ЦИФРОВИХ ДВІЙНИКІВ ДЛЯ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖ

2.1. Моделювання фізичних компонентів мережі

У контексті телекомунікацій цифровий двійник слугує інтегрованою віртуальною копією фізичної інфраструктури мережі, яка здатна відображати топологію, обладнання, канали зв'язку, параметри трафіку та події в режимі реального часу. Для ефективного функціонування цифрового двійника необхідно відтворити не лише структуру, а й динаміку роботи системи, що включає безперервний обмін даними, оновлення стану елементів і аналіз телеметрії. Комунікаційні мережі забезпечують міцну основу для впровадження різноманітних застосувань цифрових двійників. Водночас, з огляду на зростання типів бізнесу, масштабів та складності, сама мережа також повинна використовувати технологію цифрових двійників для пошуку вдосконалених та оптимізованих рішень порівняно з використанням виключно реальної мережі. Мотивацію для створення мережевого цифрового двійника можна певним чином простежити до деяких раних концепцій, таких як «shadow MIB», методи індуктивного моделювання, паралельні системи тощо. З 2017 року застосування технології цифрових двійників у сфері комунікаційних мереж поступово досліджується [8].

Повна модель цифрового двійника складається з трьох компонентів: даних, моделі та програмного забезпечення, як показано на Рис.2.1. [8].

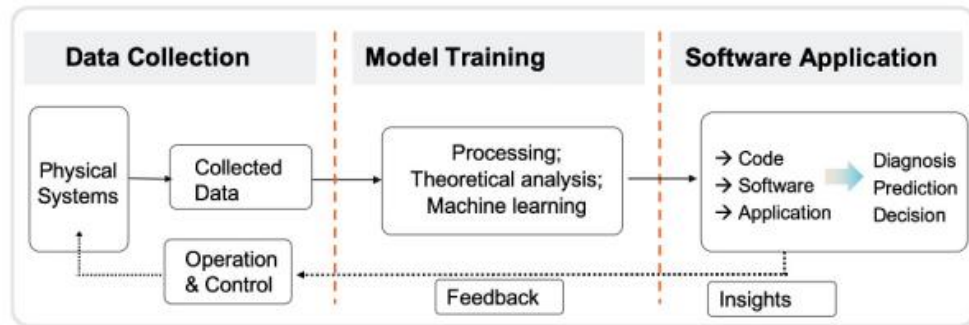


Рис. 2.1. Цифровий двійник: дані-модель-програмне забезпечення [8]

Дані, основа: Через те, що створення цифрових двійників значною мірою залежить від історичних і реальних даних, дані є основою всієї моделі цифрового двійника. Фізичні системи містять дані, принципи роботи сутностей та контролер, який може налаштовувати роботу фізичних систем [8].

Модель, основний компонент: Моделі (як теоретичні моделі, так і моделі на основі даних) є основним компонентом цифрових двійників. Теоретичні моделі будуються на основі принципів фізичних систем. Моделі на основі даних тренуються шляхом збирання великої кількості історичних і реальних даних з фізичних систем. Для навчання моделей на основі даних можуть використовуватися різноманітні технології, включаючи штучний інтелект (ШІ) та обробку даних. Навчання моделей на основі даних є ітеративним процесом, у якому моделі постійно тренуються та оновлюються, у процесі самоосвіти [8].

Програмне забезпечення, невід'ємний носій: Програмне забезпечення є носієм цифрових двійників і забезпечує інтерфейс між фізичними системами та цифровим простором. Моделі цифрових двійників, що складаються з алгоритмів, коду та програмного забезпечення, реалізуються шляхом розробки відповідного програмного забезпечення. Функції уявлення, діагностики, прогнозування та прийняття рішень розгортаються у програмно визначений спосіб, що надає фізичному контролеру оптимізовану команду для покращення роботи фізичних систем [8].

Цифровий двійник забезпечує точне відображення фізичного світу у цифровому просторі з підтримкою взаємодії в реальному часі. Він усуває розрив

між фізичним і цифровим простором шляхом безперервної синхронізації та оновлення. Основні характеристики цифрових двійників можна узагальнити таким чином [8]:

- Точне відображення (mapping): Цифровий двійник встановлює відповідність між фізичними об'єктами та цифровим простором. Історичні дані та поточний стан фізичних об'єктів синхронізуються для подальшої обробки й аналізу. Це дозволяє повністю відобразити стан об'єктів і досягти низької похибки у цифровому відтворенні.
- Робота в реальному часі: На відміну від традиційного моделювання, цифрові двійники постійно синхронізуються з фізичними об'єктами. Дані збираються та обробляються з використанням edge-комп'ютингу для побудови та оновлення цифрової моделі об'єкта.
- Розподіленість: У телекомунікаційних мережах фізичні об'єкти (сенсори, IoT-пристрої, системи) розташовані у просторі нерівномірно. Їх цифрові двійники також розміщуються на різних edge-серверах. У такому середовищі для побудови моделей застосовуються розподілені алгоритми штучного інтелекту.
- Інтелектуальність: Цифрові двійники не лише відображають поточний стан, а й інтегрують поведінкові моделі фізичних об'єктів. AI-алгоритми аналізують масиви даних і дозволяють системі прогнозувати, оптимізувати та ухвалювати рішення.
- Двостороння взаємодія: Цифрові двійники отримують дані від фізичних об'єктів, але й можуть впливати на них, передаючи оптимізаційні рішення. Це забезпечує зворотний зв'язок у реальному часі, що є ключовою особливістю цієї технології.

Цифрові близнюки можуть надати фізичним системам такі переваги завдяки процесам оптимізації, прогнозуванню та автоматизації [8]:

- Вища продуктивність: Завдяки оптимізації операцій та моделюванню робочих сценаріїв цифровий двійник дозволяє покращити ефективність фізичних систем, зменшити втрати й покращити процеси

проектування.

- Поглиблений моніторинг: Безперервне оновлення даних дозволяє цифровим двійникам проактивно стежити за станом об'єктів, поєднуючи історичні, поточні та прогнозовані дані.
- Зниження витрат на обслуговування: Цифровий двійник забезпечує предиктивне обслуговування, замість жорстко розписаного заздалегідь. Це значно знижує витрати та підвищує ефективність.
- Підвищення надійності: Тестування та симуляція сценаріїв в цифровому середовищі дозволяє виявляти слабкі місця без ризику пошкодження фізичних систем.
- Зменшення ризику збоїв: Віртуальні експерименти з високою точністю дозволяють уникнути критичних помилок, що можуть спричинити вихід системи з ладу в реальності.

Завдяки вищезазначеним перевагам, цифрові близнюки можуть бути застосовані в різноманітних сценаріях для підвищення продуктивності фізичної системи [8]:

- Розумне виробництво: Цифрові двійники інтегрують всі етапи життєвого циклу продукту, від проектування до обслуговування, з реальним часом. Це ключовий інструмент у рамках Індустрії 4.0.
- Авіація: Вперше впроваджено саме в авіаційній галузі для моніторингу технічного стану літаків. Дає змогу виконувати діагностику, планування маршрутів і прогнозне обслуговування. Але є виклики, пов'язані з точністю моделі та стабільністю зв'язку.
- Інтелектуальний транспорт: Допомогає в управлінні дорожнім рухом, запобіганні заторам та аваріям, оптимізації планування обслуговування інфраструктури.
- Охорона здоров'я: Цифрові двійники використовуються для моніторингу стану пацієнтів, прогнозування їхнього стану та навіть підтримки віддалених операцій. Проблема - захист конфіденційності

даних.

- **Мережі 6G:** Для досягнення ультранизьких затримок, інтелектуальної маршрутизації та адаптивного керування каналами в умовах надвеликих обсягів даних. ЦД може відображати стан базових станцій, керувати розподілом навантаження, забезпечувати стійкість до втрат зв'язку.

Далі розглянемо функції цифрових двійників в телекомунікаціях. Цифрові двійники дозволяють здійснювати моніторинг трафіку в реальному часі, аналізувати навантаження на різні частини мережі, оцінювати рівень пропускної здатності каналів зв'язку та ефективність роботи окремих компонентів (наприклад, базових станцій чи маршрутизаторів). За допомогою цих даних можна оптимізувати розподіл ресурсів, а також передбачати і усувати потенційні перевантаження мережі. Наприклад, виявивши, що певний маршрутизатор або точка доступу наближається до межі своєї пропускної здатності, система може автоматично налаштувати балансування навантаження або перенаправити трафік на інші канали. Цифрові двійники забезпечують можливість моделювання змін у конфігурації мережі, що дозволяє оперативно адаптувати мережу до нових умов, наприклад, при додаванні нових пристроїв або зміні топології. Це може бути корисним при розширенні мережі, налаштуванні нових базових станцій або при переході на нові протоколи зв'язку (наприклад, перехід на 5G). Моделювання змін дозволяє передбачити потенційні проблеми до того, як вони виникнуть в реальній мережі, таким чином знижуючи ризики помилок і збоїв. Прогнозування відмов є однією з основних функцій цифрових двійників у телекомунікаціях. Завдяки збору даних про стан мережі, таких як температура обладнання, рівень сигналу, завантаженість каналів та інші критичні параметри, цифровий двійник може передбачити відмови або збої в мережі до того, як вони стануть критичними. Це дозволяє операторам здійснювати профілактичне обслуговування, запобігаючи серйозним порушенням роботи мережі та мінімізуючи час простою. Цифрові двійники можуть автоматизувати процеси відновлення зв'язку в разі збоїв або аварійних ситуацій. Завдяки моніторингу в

реальному часі та аналітичним алгоритмам, цифровий двійник може автоматично визначити місце виникнення несправності та ініціювати необхідні дії, такі як перенаправлення трафіку, активація резервних каналів зв'язку або зміну параметрів конфігурації мережі. Це значно зменшує час відновлення зв'язку, дозволяючи операторам швидше реагувати на інциденти. Однією з ключових функцій цифрових двійників є оптимізація маршрутів даних в мережах. За допомогою цифрового двійника можна виявити найбільш ефективні маршрути для передачі даних у реальному часі, враховуючи поточне навантаження на мережу, швидкість з'єднання та інші параметри. Наприклад, якщо один із каналів зв'язку перевантажений, цифровий двійник може автоматично змінити маршрут трафіку, щоб уникнути затримок або втрат пакету. Це дозволяє забезпечити високу якість обслуговування (QoS) та мінімізувати затримки у мережах з високими вимогами до швидкості [8].

DT Networks (DTNs) визначається як мережа багато до багатьох, що складається з кількох одно-до-одного DT. Іншими словами, DTN використовує передові комунікаційні технології для реалізації інформаційних взаємодій в реальному часі між фізичним об'єктом і його віртуальним близнюком, віртуальним близнюком та іншими віртуальними близнюками, а також фізичним об'єктом і іншими фізичними об'єктами. DTN реалізує динамічні взаємодії та синхронізовану еволюцію кількох фізичних об'єктів і віртуальних близнюків, використовуючи точне моделювання DT, комунікації, обчислення та технології обробки фізичних даних. У DTN фізичні об'єкти та віртуальні близнюки можуть комунікувати, співпрацювати, обмінюватися інформацією, виконувати завдання один з одним і формувати мережу обміну інформацією, підключаючи кілька вузлів DT. Спочатку ми проаналізуємо різницю між DT і DTN, далі - комунікації в DTN. Крім того, ми зобразимо кілька типових сценаріїв застосування DTN, таких як виробництво, 6G мережі та інтелектуальні транспортні системи. Рис. 2.2 порівнює концепції DT та DTN з точки зору сценаріїв застосування, структури складу та відносин зіставлення [8].

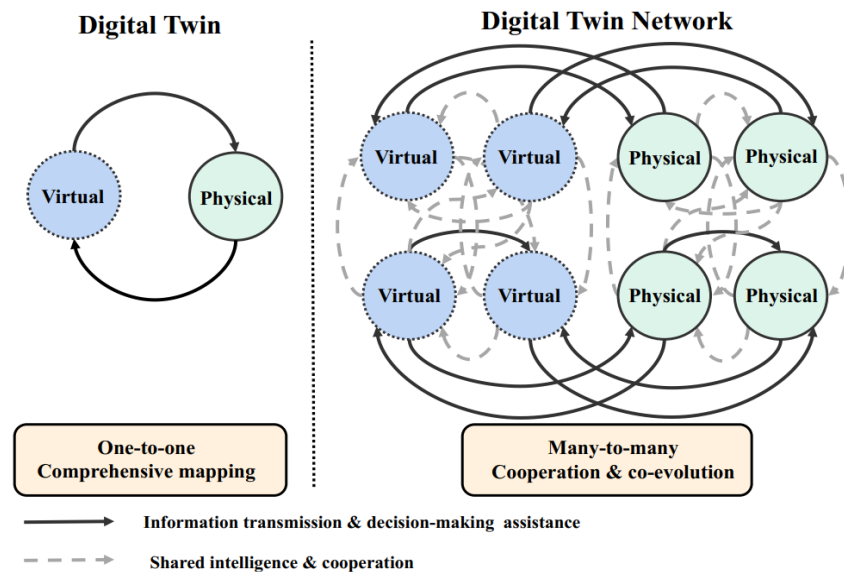


Рис.2.2 Порівняння між DT і DTN [8]

По-перше, з точки зору сценаріїв застосування, концепції DT і DTN відрізняються. DT підходить для відображення єдиного незалежного об'єкта, тоді як DTN моделює групу об'єктів зі складними внутрішніми взаємодіями. Наприклад, моделювання будівлі у віртуальному просторі за допомогою підходу DT допомагає оптимізувати весь життєвий цикл будівлі з точки зору проектування, обслуговування тощо. Модель будівлі залежить лише від аналізу та прийняття рішень на основі даних стану будівлі. На відміну від цього, при створенні віртуальної моделі виробничої лінії промислової автоматизації слід використовувати DTN для моделювання та відображення колаборативних відносин між кількома промисловими компонентами, залученими до виробничого процесу. Одним із критичних етапів створення цифрового двійника є точне моделювання фізичної інфраструктури. У сфері телекомунікацій це включає [8]:

- Активне мережеве обладнання: маршрутизатори, комутатори, базові станції, оптичні мультиплексори, сервери граничних обчислень (MEC).
- Кабельна інфраструктура: оптичні та мідні канали, муфти, кроси, розподільні шафи.
- Антенні системи: для стільникового зв'язку, мікрохвильових лінків,

супутникових терміналів.

- Джерела живлення та аварійні засоби підтримки: джерела безперебійного живлення (UPS), генератори, акумуляторні батареї.

Модель повинна враховувати географічне розміщення, фізичні параметри компонентів (потужність, чутливість, швидкість комутації) та їх стан у реальному часі. Для збору даних використовуються протоколи SNMP, NetFlow, телеметрія gRPC, а також інтеграція з системами OSS/BSS. Ці дані дозволяють цифровому двійнику підтримувати актуальний стан моделі й забезпечувати сценарне моделювання: наприклад, впливу збою живлення на сегмент мережі, або тестування маршрутизації при виході з ладу певного вузла. [8]

2.2. Архітектури цифрових двійників

Існує декілька підходів до архітектурного побудови цифрових двійників, що мають різний ступінь гнучкості, масштабованості та функціональності [9]:

- Монолітна архітектура - Усі компоненти цифрового двійника реалізуються в межах одного програмного модуля або системи. Цей підхід є простим у реалізації, але має обмежену масштабованість і погано підходить для великих телекомунікаційних систем, де відбуваються часті зміни структури або конфігурації.
- Модульна архітектура - Система розбивається на окремі модулі (наприклад: управління живленням, маршрутизація, моніторинг трафіку), які працюють автономно, але взаємодіють між собою через стандартизовані API. Такий підхід дозволяє оновлювати або замінювати компоненти без зупинки всієї системи та адаптувати цифровий двійник до складних мережевих сценаріїв.
- Інтегрована архітектура - Цей варіант поєднує цифрового двійника з існуючими інструментами OSS/BSS, системами керування ресурсами, візуалізацією, NMS/EMS-платформами. Наприклад, в одному

середовищі відображаються як фізичні зв'язки, так і параметри SLA, інформація про абонентів або KPI. Це забезпечує глибоку інтеграцію з реальними бізнес-процесами провайдера.

- Розподілена архітектура - Використовується для побудови цифрових двійників масштабних мереж, що охоплюють географічно віддалені об'єкти (наприклад, магістральні або міжнародні мережі). Дані обробляються частково на периферії (MEC/edge), частково в хмарі або на центральному сервері. Забезпечується висока швидкість реакції й відмовостійкість, зокрема в 5G-мережах.

Процес створення цифрового двійника можна розділити на окремі етапи, зокрема можна виділити етапи збору, зберігання та оброблення даних, аналізу даних та машинного навчання. Оскільки цифровий двійник передбачає двосторонню комунікацію фізичного об'єкта або процесу з його цифровою копією, то необхідно забезпечити надійні і безпечні канали зв'язку. Також цифровий двійник потребує накопичення великої кількості історичних даних для обробки і побудови моделі. Враховуючи ці вимоги, розгортання цифрового двійника для інтеграції з реальним об'єктом може бути доволі складною задачею, оскільки необхідно створити інфраструктуру з відповідним потужним серверним обладнанням. Більше того, заздалегідь складно визначити необхідні характеристики серверів, оскільки об'єми зберігання даних та необхідна обчислювальна потужність важко прогнозовані та залежать від типу моделі цифрового двійника. Тому для побудови цифрового двійника доцільно використовувати хмарні сервіси, які забезпечують масштабованість, гнучкість, високу надійність та безпеку. Це дозволить легко збільшувати або зменшувати ресурси відповідно до потреб, з можливістю оплати за фактичне використання, не купуючи фізичне обладнання і не витрачаючи кошти на його обслуговування та оновлення. Для побудови цифрового двійника на базі хмарних сервісів AWS застосовується шлюз обміну даними AWS IoT SiteWise Edge, який встановлюється на локальному сервері підприємства. Цей шлюз забезпечує захищену передачу даних від ПЛК котла по протоколу OPC UA до дата центру.

Отримані дані від AWS IoT SiteWise Edge обробляються і зберігаються в часовій базі даних AWS IoT SiteWise. Зі свого боку AWS IoT SiteWise передає дані для Amazon SageMaker для машинного навчання і побудови моделі. Отримана модель в Amazon SageMaker передається через AWS IoT SiteWise в AWS IoT TwinMaker для побудови цифрового двійника. Для візуалізації даних цифрового двійника використовується Amazon Grafana. Загальна архітектура показана на Рис 2.3. [9].

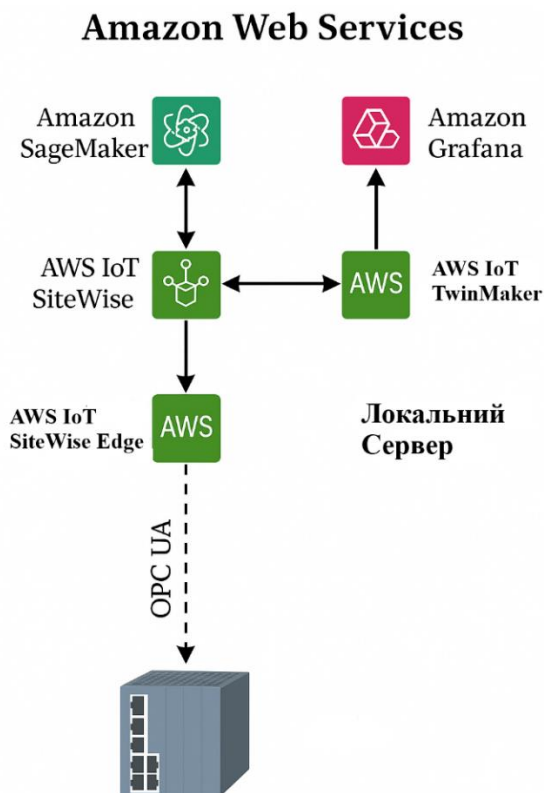


Рис. 2.3 Архітектура програмно-технічного комплексу для побудови цифрового двійника [9]

Висновки

У цьому розділі було розглянуто основні компоненти та архітектуру цифрових двійників для телекомунікаційних мереж. Цифрові двійники складаються з трьох основних компонентів: даних, моделі та програмного забезпечення, що забезпечує інтеграцію фізичної інфраструктури мережі з її цифровим відображенням. Розглянуто важливість збору та обробки реальних даних з фізичних систем для створення точних моделей, що забезпечують ефективне прогнозування та управління. Також було зазначено, що програмне забезпечення цифрових двійників має забезпечувати двосторонній зв'язок між фізичними та цифровими об'єктами для постійного оновлення даних і взаємодії з мережею в реальному часі. Особлива увага була приділена принципам побудови цифрових двійників, таким як точне відображення фізичних об'єктів, робота в реальному часі, розподіленість і інтеграція з технологіями штучного інтелекту. Це дозволяє здійснювати ефективне управління мережами, прогнозувати навантаження та оптимізувати конфігурацію системи. Крім того, застосування цифрових двійників дає змогу значно покращити продуктивність, знизити витрати на обслуговування та підвищити надійність телекомунікаційних мереж. Важливим аспектом є використання хмарних сервісів для побудови цифрових двійників, що дозволяє забезпечити масштабованість та гнучкість інфраструктури. Хмарні рішення, такі як AWS IoT SiteWise і Amazon SageMaker, забезпечують збереження та обробку даних, машинне навчання і побудову моделей цифрових двійників, що сприяє покращенню процесів управління та оптимізації мереж. Високий рівень інтеграції різних технологій забезпечує ефективне функціонування цифрових двійників у телекомунікаційних мережах і дає можливість адаптувати їх під різні сценарії використання.

РОЗДІЛ 3

ВИКОРИСТАННЯ ЦИФРОВИХ ДВІЙНИКІВ ДЛЯ УПРАВЛІННЯ МЕРЕЖАМИ 5G

3.1. Особливості управління мережами 5G за допомогою цифрових двійників

Цифрові двійники (DT)s, віртуальні копії фізичних систем, пропонують перспективний підхід до вирішення багатьох проблем. Оператори можуть моделювати, контролювати та оптимізувати продуктивність мережі в режимі реального часу, створюючи DT 5G RAN, що дозволяє здійснювати прогнозоване управління, ефективний розподіл ресурсів та швидке усунення несправностей. DTs використовуються в багатьох інших галузях, таких як виробництво, логістика, будівництво та розподіл енергії . В авангарді технології DT знаходяться DT Microsoft Azure . Це платформа як послуга (PaaS) для створення, підтримки та взаємодії з DT у хмарі. Хмара Azure, в якій розміщуються DT, є хмарною платформою виробничого рівня, що гарантує масштабованість, доступність та надійність. Крім того, до DT можна легко підключити інші хмарні сервіси зберігання, аналітики та візуалізації Azure. Таким чином, Azure DT представляє чудову можливість для об'єднання мереж 5G RAN. Стандарт O-RAN відіграватиме вирішальну роль у майбутньому зв'язку 5G, оскільки він забезпечує різноманітність постачальників, економічність та ефективність. Стандарт O-RAN вже забезпечує метод моніторингу продуктивності мережі, активності користувачів і загального стану системи 5G RAN . Ці повідомлення обробляються інтелектуальним контролером RAN в режимі реального часу (Near- Real-Time RAN Intelligent Controller, Near-RT RIC), який служить центральним вузлом для збору даних і управління мережею 5G RAN. Для ефективного використання цих даних існує спеціальний xApp, який взаємодіє з Near-RT RIC. Цей xApp витягує та фільтрує відповідну інформацію з індикаційних повідомлень O-RAN та оновлює екземпляр DT. Це гарантує, що

віртуальне представлення 5G RAN точно відображає поточний стан фізичної мережі. Далі розглянемо метод та архітектуру для створення та управління DT 5G RAN на хмарній платформі Azure. Підхід використовує стандартизоване моделювання та інтеграцію даних Azure DT, а також неінвазивні властивості O-RAN, які не залежать від постачальника. TwinRAN має три основні особливості, які підкреслюють його практичну корисність: Неінвазивний: він не вимагає змін у вихідному коді або апаратному забезпеченні 5G RAN. Підключивши xApp поверх архітектури 5G RAN, він будується безперешкодно над існуючими компонентами 5G. Незалежно від постачальника: може працювати з усіма постачальниками обладнання та програмного забезпечення, якщо вони сумісні зі стандартом O-RAN. Багатоцільовий: один і той самий DT може одночасно використовуватися для різних додатків, якщо всі параметри, необхідні для додатків, змодельовані в двійниках. Будь-яка програма може підключатися до DT для потокової передачі оновлень в реальному часі або завантаження архівних даних. Крім того, ця властивість розширює корисність TwinRAN на майбутнє: будь-який новий додаток може бути розгорнутий без модифікації коду TwinRAN. Далі проаналізуємо загальну інформацію про Azure DT та систему O-RAN. Стандарт O-RAN являє собою значну еволюцію в архітектурі радіорелейного зв'язку, спрямовану на забезпечення сумісності, гнучкості та інновацій в екосистемі 5G. Альянс O-RAN, глобальний консорціум мережевих операторів і постачальників, очолює цю ініціативу зі стандартизації відкритих та інтелектуальних інтерфейсів RAN, що забезпечує розгортання різних постачальників і розширені можливості управління мережею. Компоненти та структури повідомлень O-RAN [10]:

- 1) Вузол E2: Вузол E2 є посередником між gNB і Near-RT RIC. Він полегшує обмін телеметричними даними від gNB до Near-RT RIC і керуючими повідомленнями у зворотному напрямку. Цей обмін повідомленнями дозволяє Near-RT RIC керувати та оптимізувати функції RAN в режимі реального часу. TwinRAN використовує телеметричну інформацію, отриману від RIC Near-RT, для оновлення DT.

2) Near-RT RIC: Near-RT RIC призначений для виконання контролю та моніторингу елементів RAN в режимі, близькому до реального часу. Він обробляє дані, зібрані з вузлів E2, і доставляє їх до xApps. З іншого боку, він також дозволяє xApps контролювати елементи RAN через інтерфейс E2. Near-RT RIC зазвичай містить базу даних, яка збирає постійну інформацію від різних агентів E2. Послідовність обміну повідомленнями в TwinRAN зображено на Рис.3.1.: TwinRAN xApp використовує індикаційні повідомлення O-RAN від Near-RT RIC та оновлює DT, розміщений у хмарі Azure. Потрібні ключові показники ефективності (KPM) повинні бути підписані xApp через Near-RT RIC.

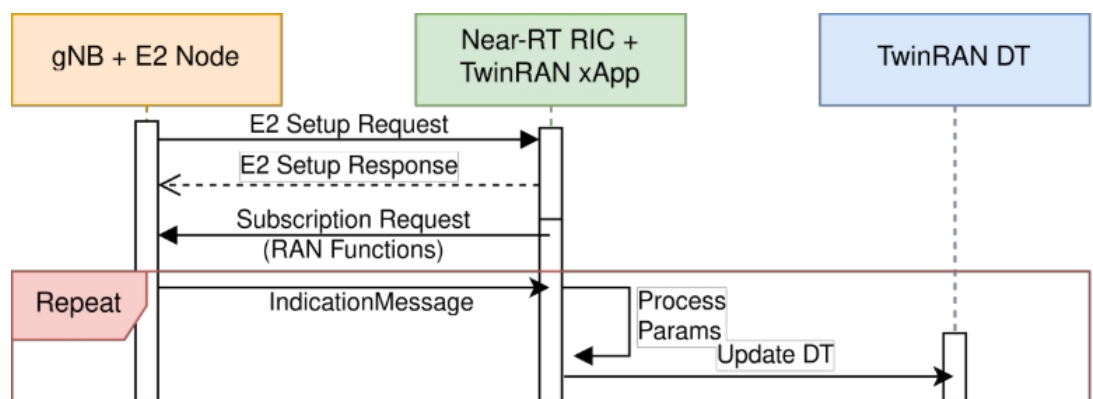


Рис. 3.1. Послідовність обміну повідомленнями в TwinRAN [10]

3) xApps: xApps - це модульні додатки, які працюють на Near-RT RIC, забезпечуючи можливості цілеспрямованого управління та оптимізації. Вони розроблені таким чином, що постачальники Near-RT RIC і xApps можуть бути відокремлені. Ці додатки використовують дані та аналітичні можливості Near-RT RIC в режимі реального часу для реалізації конкретних сценаріїв використання, підвищуючи ефективність системи. Деякі приклади використання включають керування трафіком, управління завадами і оптимізацію повторних джерел. DTs є новим варіантом використання, де потрібен новий xApp для оновлення DT. TwinRAN xApp фільтрує дані, доступні в ближньому RIC, перевіряє зміни в сутностях і оновлює двійників у хмарі.

4) Типи повідомлень: O-RAN визначає декілька типів повідомлень для

зв'язку між своїми компонентами. Індикація: Надає оновлення стану та телеметричні дані від елементів RAN до Near-RT RIC. Існує два типи індикаційних повідомлень: звіт і вставка. Перші надають періодичні оновлення за підпискою, а другі запускаються асинхронно, щоб повідомити xApp про певну подію в RAN. Політика: Передача політик від Near-RT RIC для керування поведінкою елементів RAN. Про передачу такої функції політики TwinRAN xApp повідомляє екземпляру TwinRAN. Control (Керування): Використовується для конфігурації та управління елементами RAN [10].

На Рис.3.2. показано, як TwinRAN знаходиться на вершині архітектури O-RAN. TwinRAN xApp використовує індикаційні повідомлення від вузла E2, а потім обробляє їх, щоб визначити, які повідомлення потрібно надіслати до екземпляра DT у хмарі Azure. Near-RT RIC підписується на відповідні функції RAN, доступні на вузлі E2. Важливою функцією RAN для TwinRAN є ключові показники продуктивності (KPM), визначені в стандарті O-RAN, що дає нам хорошу відправну точку для того, що може бути представлено за допомогою запропонованої нами архітектури.

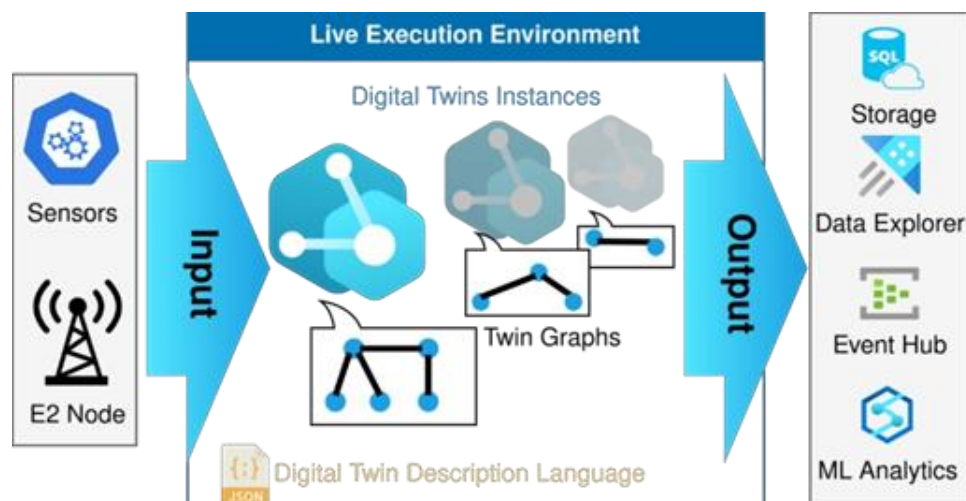


Рис.3.2. Цифрові двійники в хмарі Azure [10]

На Рис. 3.2 показано загальний огляд TwinRAN у хмарі Azure. Вузол E2 через xApp та інші пов'язані датчики надсилає дані до екземплярів DT. Моделі DT, визначені за допомогою DTDL, використовуються як каркас для кожного

екземпляра. Ці екземпляри виконуються в реальному середовищі, а граф двійників підтримує всі поточні стани та залежності двійників. Можна написати додаток для запиту стану екземплярів DT в режимі реального часу або налаштувати автоматичний потік змін DT до інших хмарних служб Azure, таких як Data Explorer. Обмеження, які накладає Azure DT, дають нам значний запас по розміру повідомлень. Нехай максимальна кількість двійників в екземплярі становить $Y \in \mathbb{N}^+$. Якщо кожен двійник в екземплярі оновлюється кожні $R \in \mathbb{R}^+$ секунди, то обмеження Azure DT становить $R \geq 0.1$, і ми отримуємо $Y = 1000 - R$. Системи 5G RAN є динамічними, і максимальна довжина кадру і слоту становить 10 і 1 мс відповідно. Azure DT спочатку був створений для представлення фізичного світу за допомогою IoT, де зміни в навколишньому середовищі відбуваються повільніше, ніж у 5G. Щоб підтримувати точність DT у хмарі в TwinRAN, швидкість оновлення повинна бути якомога вищою в межах обмежень, встановлених Azure DT. Оновлення всіх двійників з максимальною швидкістю $R = 0.1$ секунд, ми можемо мати максимум 100 близнюків в одному екземплярі [10].

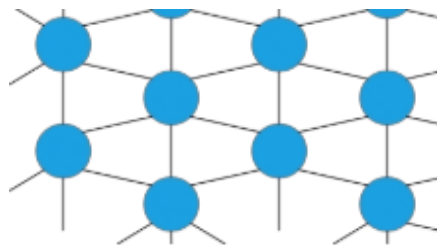


Рис. 3.3 TwinRAN – мульти [10]

Один екземпляр відстежує всю мережу. Одна модель DTDL визначає одну комірку. Сусідні комірки з'єднані відносинами, які в основному визначають кількість перешкод, спричинених сусідом [10].

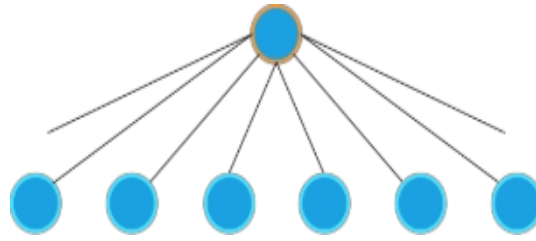


Рис. 3.4 TwinRAN-комірка [10]

Для кожної комірки створюється один екземпляр. GNB з'єднується з кожним UE через зв'язок. Між двома двійниками в одному екземплярі можна визначити будь-який зв'язок. Таким чином, виникає проблема в організації двійників для досягнення масштабованості при дотриманні обмеження в 100 двійників на екземпляр і збереженні всіх зв'язків між сутностями. У той же час, розмір моделей повинен бути невеликим, щоб зменшити затримку двійників. Моделі також повинні бути організовані так, щоб відповідати сутностям, які традиційно використовуються в мобільних мережах. Таким чином, пропонується розділити TwinRAN на два типи екземплярів. З точки зору оператора 5G RAN, завдання можна розділити на дві основні категорії: внутрішньостільникові і міжстільникові. Внутрішньостільникові завдання включають ефективне планування користувачів на основі необхідної якості обслуговування (QoS), мінімізацію операційних витрат, безпеку мережі тощо. Міжстільникові операції - це управління хендовер, зменшення завад тощо. Тому я пропоную два твін-графи, які одночасно підтримуються в хмарі: TwinRAN-cell та TwinRAN-multi. TwinRAN-мульти зберігає широкий огляд всієї мережі. Кожна комірка визначена в одній моделі DTDL, а сусідні комірки з'єднані відносинами. Кожен UE є однією властивістю gNB, значенням якої є SINR. TwinRAN-комірка породжує новий екземпляр для кожного gNB. Тут gNB і UE є двома різними моделями DTDL, між якими існує зв'язок, якщо UE підключено. Велика модель комірки з детальними моделями gNB і UE як компонентами дозволила б нам одночасно керувати 100 комітками, кожна з яких оновлюється 10 разів на секунду. Однак комітка, що містить 100 UE, зробить розміри оновлень великими і збільшить затримку двійників, час обслуговування і час запиту. Більше того, це вплине на

масштабованість системи як додатковий параметр в моделі UE і сильно вплине на загальну затримку. Ми можемо утримувати ці значення на низькому рівні, використовуючи дезагредований підхід, який одночасно підтримує TwinRAN-комірку та TwinRAN-мульти екземпляри [10].

3.2. Реальні приклади впровадження цифрових двійників у мережах 5G

Для кращого розуміння практичних аспектів впровадження цифрових двійників у телекомунікаційних мережах 5G розглянемо приклади їх застосування [10].

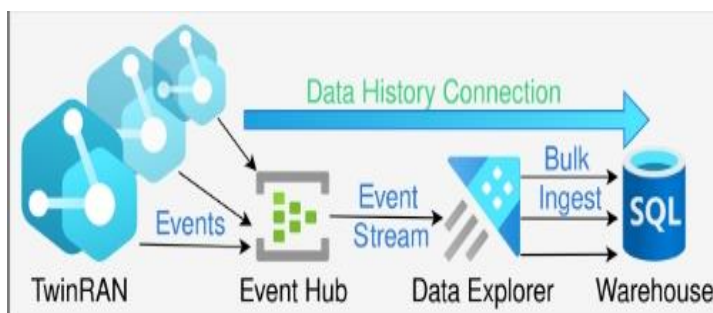


Рис. 3.5. Сховище даних для ML [10]

Один екземпляр відстежує всю мережу. Одна модель DTDL визначає одну комірку. Сусідні комірки з'єднані відносинами, які в основному визначають кількість перешкод, спричинених сусідом [10].

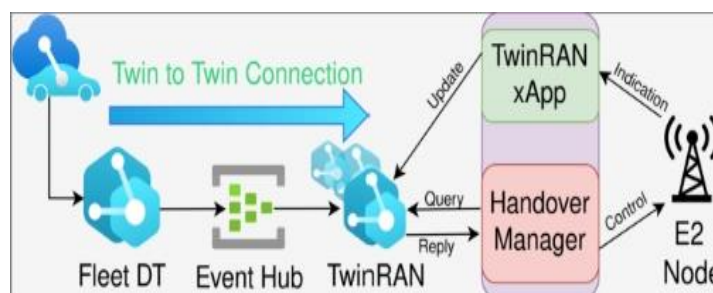


Рис. 3.6 TwinRAN-комірка [10]

Для кожної комірки створюється один екземпляр. GNB з'єднується з кожним UE через зв'язок [10].

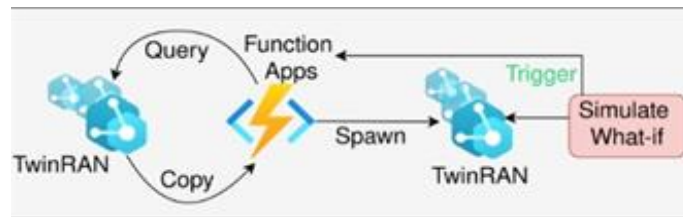


Рис. 3.7 Архітектура системи з використанням TwinRAN [10]

Поточний стан TwinRAN створюється шляхом одночасного запиту всього екземпляра і розгортання його на багатьох нових екземплярах [10].

Першим прикладом застосування є централізоване сховище даних, яке використовується для машинного навчання. Сховище даних - це централізоване сховище, яке зберігає великі обсяги структурованих даних з різних джерел. Доступ до таких великих обсягів структурованих даних дозволяє концептуалізувати навчання моделей ML. PEACH прогнозує якість каналу на основі зображень. Наприклад, якщо у вас є дані CQI для багатьох користувачів з TwinRAN і відзняті з позначкою часу кадри з камери спостереження, ви можете навчити модель PEACH для їх налаштування. Крім того, можна застосувати підхід безперервного навчання, періодично тренуючись на новому наборі даних. На рисунку 3.5 показано архітектуру в хмарі для розгортання цього прикладу використання. Екземпляри Azure DT створюють події щоразу, коли оновлюється або створюється двійник чи зв'язок. Ці події передаються до екземпляра Azure Event Hub, платформи для потокової передачі, обробки та поглинання даних. Оброблені події передаються в екземпляр Azure Data Explorer, систему зберігання та аналізу даних із вбудованою постійною базою даних. Запит мовою Kusto Query Language (KQL) з будь-якої платформи може завантажувати дані з бази даних до моделі ML. Другим прикладом застосування є система точного позиціонування для забезпечення ефективної передачі даних. Точне позиціонування може допомогти мережі приймати кращі рішення щодо

навігації. Однак точність триангуляції на основі лише сигналів NR набагато нижча порівняно з GPS або LiDAR. У деяких нещодавніх роботах запропоновано ефективне управління хендовером з використанням O-RAN. TwinRAN може забезпечити спрощений спосіб надання вхідних даних для цих алгоритмів. Оскільки розгортання DT стає все більш повсюдним, перехресне з'єднання між різними DT неминуче. У цьому прикладі використання ми уявляємо собі оператора автопарку або AGV розгортання власного DT. Цей комплексний DT є дуже великим і містить таку інформацію, як рівень заряду акумулятора, перевезений вантаж і пройдену відстань, а також поточне місцезнаходження транспортного засобу. Оператор мережі підписується на певний потік даних (координати місцезнаходження), необхідний для його роботи, і йому не потрібно підтримувати власний механізм збору даних про місцезнаходження. На Рис. 3.6 показано архітектуру для реалізації цього варіанту використання. Azure DT дозволяє створювати маршрути подій, які з'єднують їхній вхід і вихід (так звані кінцеві точки) з центром подій. Публікатор подій може публікувати свої дані в центрі подій з унікальним ідентифікатором, до якого інший маршрут подій може з'єднати свою кінцеву точку. Додаток xApp, який запускає передачу, запитує якість каналу, стан буфера, інформацію про перешкоди та місцезнаходження у TwinRAN DT, щоб передати їх своєму алгоритму, не звертаючи уваги на джерело даних про місцезнаходження. Третім прикладом застосування є реалізація сценаріїв «Що, якщо» для моделювання різних ситуацій та прийняття оптимальних рішень. DT у хмарі є найближчим представленням поточного цифрового стану системи. Надійність і безпека мережі можуть бути значно покращені шляхом проведення аналізу "що було б, якби" шляхом моделювання різних сценаріїв з використанням попереднього стану мережі. На рисунку 3.7 зображено архітектуру такої системи з використанням TwinRAN. Поточний стан TwinRAN створюється шляхом одночасного запиту всього екземпляра і розгортання його на багатьох нових екземплярах. Azure Functions - це безсерверний обчислювальний сервіс, який запускається подією або викликом API для створення таких копій "один-до-багатьох". Нові екземпляри TwinRAN

можна підключити до симуляції для проведення різних сценаріїв "що буде, якщо" і порівняння еволюції двійників. Можливість діагностики постачальника TwinRAN дозволяє здійснювати таке безперешкодне з'єднання. Спочатку ми заповнюємо екземпляр TwinRAN за допомогою FlexRIS одним реальним gNB OpenAirInterface та 7 E2 емуляторами агентів. Потім новостворені копії екземплярів TwinRAN були підключені до симуляції NS-3 для тестування сценаріїв «Що, якщо». Підсумовуючи результати проведеного дослідження, було змодельовано RAN-мережу, яка складається з восьми комірок ($M=8$), кожна з яких обслуговує 99 користувачів (UE). Для аналізу ефективності застосування цифрових двійників створено вісім екземплярів TwinRAN-мульти та по 100 двійників у кожній з комірок (TwinRAN-комірки), що дозволяє детально дослідити функціонування мережі та оцінити переваги інтеграції цифрових двійників у системи управління. Кожна комірка в екземплярі TwinRAN-multi має оновлюється п'ять разів на секунду, в той час як кожен Twin в екземплярі TwinRAN-комірки оновлюється 10 разів на секунду. Для кожного з трьох варіантів використання ми визначаємо відповідні KPI, які ми вимірюємо. Для першого варіанту використання вимірюється обсяг даних, які проходять через з'єднання з історією даних за годину і зберігаються в базі даних. Для другого варіанту використання вимірюється час затримки від sourceTime до updateTime TwinRAN. Нарешті, для останнього варіанту використання вимірюється час, необхідний для створення п'яти нових копій екземпляра TwinRAN. Для першого варіанту використання xApp генерує 240 кБ даних, а центр подій генерує в середньому 64 000 подій в секунду для першого варіанту використання. Це було виміряно за допомогою моніторингу вихідного порту xApp і журналів Azure Event Hub. Дослідник даних кешує кожну подію розміром 21 байт і пакетно завантажує її в базу даних кожні 5 хвилин. Таким чином, кожні 5 хвилин до бази даних потрапляє приблизно 4 МБ даних. Для таких невеликих обсягів даних достатньо одного екземпляра концентратора подій і 8vCPU в кластері data explorer, оскільки завантаження центрального процесора і кешу ніколи не перевищувало 20% протягом експерименту. Простір для зберігання даних

об'ємом 1,5 ТБ буде насичений через три з половиною роки експлуатації . Запізнення параметра місцезнаходження у TwinRAN було визначено для другого варіанту використання за допомогою методу. Оновлення від джерела спочатку діє на його двійника, потім проходить через центр подій і направляється до TwinRAN. На основі 10 000 вимірювань було визначено, що середня затримка для цієї схеми становить 140 мс, а максимальна - 562 мс. Якщо ми оновлюватимемо місцезнаходження UE 5 разів на секунду, ця затримка все одно дозволить отримати найсвіжішу інформацію про місцезнаходження в екземплярі TwinRAN. Однак об'єкт, що рухається зі швидкістю 200 км/год, переміститься приблизно на 10 метрів за 200 мс. Таким чином, висока затримка оновлення Twin-Twin обмежить другий варіант використання повільно рухомими об'єктами, такими як AGV. Ми створили п'ять копій TwinRAN за допомогою спеціальної функції Azure 100 разів. Ми засікли час роботи функції з моменту отримання нею тригера до моменту успішного оновлення останньої копії. В середньому цей час становив 22 секунди, з максимальним значенням 34 секунди. Було представлено концепцію та архітектуру для твінінгу 5G RAN у хмарі за допомогою Azure DT. Архітектура зосереджена на досягненні трьох основних властивостей TwinRAN: діагностика постачальника, багатоцільове використання та неінвазивність. TwinRAN включає і використовує концепцію O-RAN, поступово інтегруючи її в 5G RAN. Ретельні дослідження поведінки Azure DT допомагають нам приймати проектні рішення для TwinRAN, такі як одночасна підтримка двох різних екземплярів для міжклітинних і внутрішньоклітинних операцій. Дослідження також показують деякі обмеження - це не є підходящим рішенням для тісно пов'язаних замкнутих операцій, керованих даними, що вимагають затримок менше 100 мс, таких як ChARM для спільного використання спектру. Демонструється три випадки використання TwinRAN, які показують його багатоцільову природу. Проаналізовано витрати на експлуатацію TwinRAN за нинішніх загальнодоступних умов ціноутворення на хмарні послуги. Ці витрати наразі не дозволяють операторам постійно розгортати TwinRAN на всій своїй мережі. Тому, з поточною структурою цін,

TwinRAN може бути вигідним для розгортання в частині мережі оператора на обмежений проміжок часу для вирішення конкретних проблем, таких як інтенсивний трафік, дослідження "що було б, якби", збір даних для навчання або публікацій. Один і той самий екземпляр TwinRAN може виконувати всі три варіанти використання одночасно, і витрати не зростають із розгортанням більшої кількості додатків. Я вважаю, що якби оператор впровадив TwinRAN, він міг би домовитися з Azure про підписку за вигіднішою ціною і масштабувати її на більшу кількість екземплярів. Все вище згадане підкреслює доцільність і переваги технології DT в контексті мереж 5G. Це створює основу для майбутніх інновацій в управлінні та оптимізації мереж з використанням хмарних цифрових двійників. Ця архітектура має на меті сприяти створенню більш інтелектуальних, стійких та адаптивних мереж 5G шляхом подолання розриву між фізичним та цифровим мережевими середовищами [10].

3.3. Проектування цифрового двійника мережі для мережі радіодоступу в 5G

Мобільні мережі 5G були розроблені для забезпечення широкого спектру нових послуг та сценаріїв застосування (наприклад, розумні міста, віртуальна реальність, громадська безпека, промисловість) з численними та різномірними вимогами (наприклад, висока швидкість передачі даних, низька затримка, висока надійність) [11].

Для досягнення цієї мети 5G об'єднав кілька технологічних досягнень, що охоплюють технології радіодоступу (наприклад, використання міліметрових хвильових діапазонів, впровадження гнучких нумерологій, масивні множинні входи-виходи (MIMO)) та архітектурні (наприклад, сегментація мережі, функціональний поділ для дезагрегації базових станцій). Впровадження всіх цих досягнень сприяло безпрецедентному рівню складності, коли йдеться про управління мережею 5G. Крім того, бачення майбутніх мереж 6G як надгнучких передбачає, що складність майбутніх мереж буде ще більшою [12].

Проект RAN NDT повинен забезпечувати точну характеристику реальної

поведінки RAN та оцінку її продуктивності, а також бути гнучким для використання в різних застосуваннях та випадках. Враховуючи це, на Рис. 3.8 показано архітектуру RAN NDT, запропонованої далі. Для полегшення практичної реалізації запропонованої архітектури вона узгоджена з термінологією та концепціями, запропонованими в IETF у [13]. Архітектура RAN NDT включає три основні модулі: сховище даних, яке збирає та зберігає дані з реальної RAN, моделі відображення сервісів, які дозволяють представляти різні елементи та операції в реальній RAN, та управління цифровими двійниками, яке керує RAN NDT. Крім того, архітектура визначає різні потенційні застосування або програми, які можуть отримати вигоду від екземпляра RAN NDT запропонованої архітектури, налаштованого відповідно до її вимог. Далі наведено додаткові деталі щодо цих модулів [16].

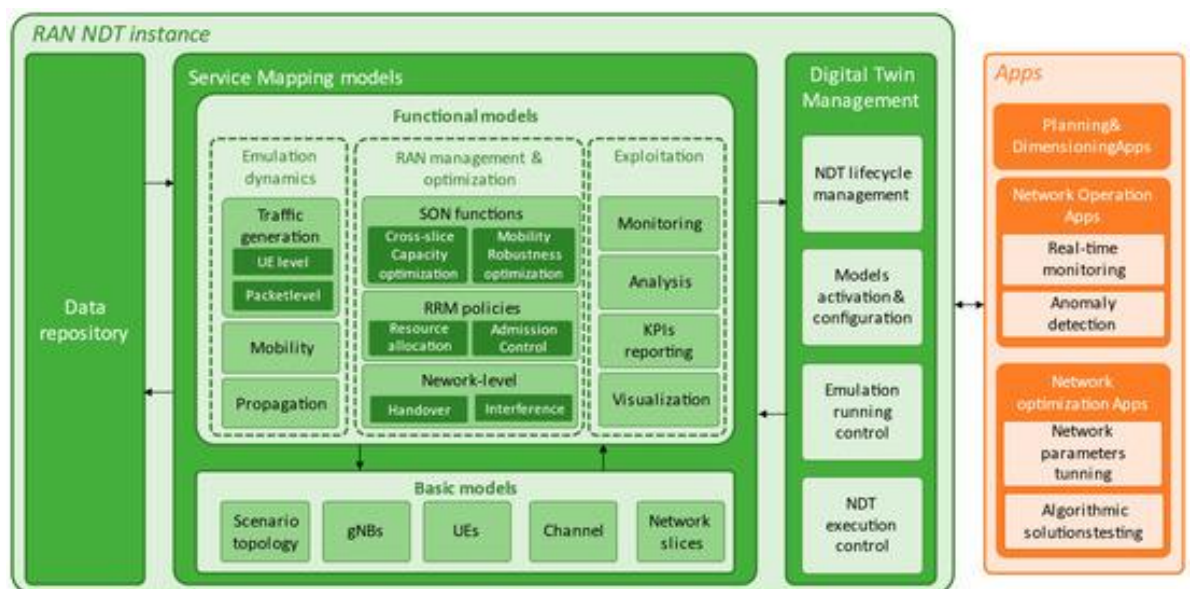


Рис. 3.8 Архітектура та додатки RAN NDT [16]

Архітектура RAN NDT включає сховище даних, моделі відображення сервісів для представлення елементів та операцій RAN, а також управління цифровими двійниками для керування екземпляром RAN NDT. Модуль сховища даних відповідає за збір даних з реального середовища RAN та їх зберігання для подання до RAN NDT. Це дозволяє NDT мати точне представлення реальної

RAN у режимі реального часу, а також надає історичні дані, які можуть використовуватися моделями в NDT. Типи даних, що містяться в репозиторії, можуть бути пов'язані з конфігураціями, робочими станами, топологією, трасами, ключовими показниками продуктивності тощо. Дані також можуть відрізнятися за рівнем деталізації (тобто, рівень пакетів, рівень часового інтервалу, рівень користувача), а також за часовим масштабом, в якому вони збираються (наприклад, мілісекунди, секунди, хвилини). Це залежатиме від мети екземпляра NDT (наприклад, навчання моделі RL на рівні MAC, що працює в мілісекундних часових масштабах, або політики RRM, що працює у порядку хвилин) та необхідних даних для різних моделей у NDT. Моделі відображення сервісів включають різні моделі для представлення різних елементів та функціональних можливостей RAN. Модульна конструкція розроблена для покращення програмованості мережевих сервісів та гнучкості експлуатації та розгортання. Моделі отримують дані зі сховища даних для оновлення своїх параметрів відповідно до реальної поведінки RAN. Різні моделі відображення сервісів взаємодіють між собою та поділяються на два основні типи: базові та функціональні моделі, які описані в наступних підрозділах із зазначенням деяких міркувань щодо їх проектування. Базові моделі стосуються елементів та сутностей мережі RAN, які дозволяють фіксувати конфігурацію та інформацію про середовище RAN. На Рис. 3.8. запропоновано деякі основні моделі, а саме: топологію сценаріїв, gNB, UE, канал та мережеві сегменти [16]:

- Модель топології сценарію: охоплює топографічну інформацію про район RAN, що розглядається для неруйнівного контролю, таку як детальні карти, включаючи будівлі, вулиці, парки [16].
- Модель gNB: емулює поведінку gNB (тобто вузлів 5G), включаючи такі аспекти, як інформація про їх розгортання (наприклад, положення gNB, висота gNB), конфігурація (наприклад, робоча частота, конфігурації пропускної здатності, нумерологія, модель формування променя) та експлуатація (тобто відповідність пакетів на різних рівнях) [16].
- Модель UE: емулює поведінку обладнання користувача (UE),

фіксуючи фізичні аспекти, такі як їхнє положення, висота, коефіцієнт підсилення антени або коефіцієнт шуму, а також інші, такі як тип UE (наприклад, пішохідний, транспортний, сенсорний) та/або вимоги UE до якості обслуговування (QoS) (наприклад, мінімальна швидкість передачі даних, затримка, надійність) [16].

- Модель каналу: характеризує зв'язки між UE та gNB, враховуючи вплив втрат поширення, перешкод або шуму відповідно до параметрів UE та gNB у NDT (наприклад, висота антени, коефіцієнт підсилення, коефіцієнт шуму) [16].
- Модель мережевих сегментів: для систем з кількома сегментами цей модуль охоплює всі аспекти, пов'язані з їхньою характеристикою. Для кожного сегмента це може включати UE, що належать до сегмента, Угоду про рівень обслуговування (SLA), яку необхідно гарантувати, та параметри для кожного сегмента, які необхідно враховувати у сценарії [16].

Функціональні моделі охоплюють мережевий аналіз, емуляцію, діагностику, прогнозування та гарантування. На Рис. 3.8 запропоновано три підгрупи функціональних моделей. Перша підгрупа - це динаміка емуляції, яка стосується тих моделей, що дозволяють керувати динамікою середовища RAN у неруйнівному контролі. Запропоновані моделі в цій підгрупі такі [16]:

- Модель генерації трафіку : дозволяє емулювати реалістичну поведінку попиту на трафік у реальній RAN на різних рівнях (наприклад, пакетний, користувацький, сервісний, мережевий сегмент або системний рівень), враховуючи часовий та просторовий розподіли в розглянутій області RAN. Генерація трафіку може базуватися або на доступних у літературі моделях (наприклад, генерація сеансів відповідно до розподілу Пуассона та тривалість сеансів, змодельована експоненціальним розподілом), або на геолокалізованих трасах та звітах про вимірювання, що генеруються UE під час їх підключення до мережі. Модель генерації трафіку взаємодіє з моделлю UE на Рис. 3.8. для генерації нових сеансів UE [16].
- Модель мобільності : емулює рух користувачів у зоні RAN, характеризуючи їхні траєкторії та швидкість. Для існуючих типів користувачів у

зоні RAN (наприклад, пішоходи, транспортні засоби, статичні користувачі, дрони тощо) можна визначити різні моделі. Ця модель взаємодіє з моделлю UE для зміни позицій UE та враховує модель топології сценаріїв для траєкторій користувачів (наприклад, пішоходи зазвичай ходять по тротуару, тоді як транспортні засоби їздять по дорозі) [16].

- Модель поширення : дозволяє генерувати різні умови поширення радіохвиль, враховуючи такі ефекти, як втрати на трасі, згасання через дифракцію та дощ, затінення або швидке згасання. Результати моделі поширення наведено в моделі каналу на Рис. 3.8 [16].

Друга підгрупа - це управління та оптимізація RAN. Моделі, що належать до цієї підгрупи, - це моделі управління та оптимізації, які застосовуються в реальній RAN і, таким чином, впливають на поведінку мережі. Це включає політики RRM (наприклад, контроль доступу, розподіл ресурсів, планування пакетів тощо), функції оптимізації на рівні мережі (наприклад, координація перешкод, передача обслуговування) та функції SON (наприклад, оптимізація ємності та покриття, балансування навантаження мобільності, оптимізація ємності між сегментами). Ці функції налаштовують та контролюють різні базові моделі. Остання група функціональних моделей - це експлуатаційні, що охоплює моделі для використання результатів решти моделей відображення послуг. Запропоновані моделі в цій групі - це модель моніторингу , яка дозволяє отримувати детальну інформацію про функціональні та базові моделі в RAN NDT, модель звітності за ключовими показниками ефективності (KPI) , яка генерує звіти з показниками KPI, що представляють інтерес (наприклад, використання ресурсів, пропускна здатність, задоволення QoS), модель аналізу для обробки отриманих KPI для оцінки продуктивності різних моделей NDT, та модель візуалізації , яка може надавати різні види графіків, такі як еволюція різних KPI, карти покриття різних gNB, включених до RAN NDT. Моделі відображення сервісів можна розглядати як аналітичні моделі, симулятори або моделі на основі машинного навчання (ML). Вибір того чи іншого підходу може залежати від різних аспектів, таких як доступність даних (тобто, якщо немає

доступних даних, модель ML не може бути навчена), рівень складності обраного підходу або необхідна точність у неруйнівному типі (NDT). Наприклад, модель генерації трафіку в NDT RAN можна розглядати як модель ML, навчену на основі реальних даних. Однак, якщо для навчання підходу ML недостатньо даних, модель також може бути розроблена як симулятор, параметри якого можна точно налаштувати для адаптації до реальних даних, зібраних з RAN. Модуль управління NDT на Рис. 3.8 виконує функцію управління NDT. Це включає управління життєвим циклом NDT, тобто розгортання, експлуатацію, оптимізацію, обслуговування та завершення роботи різних моделей у NDT. Це також охоплює передачу даних зі сховища даних до різних моделей та перевірку продуктивності моделей, а також взаємодії між ними. У випадку моделей на основі машинного навчання (ML), модуль управління NDT може забезпечувати підтримку протягом життєвого циклу AI/ML моделей на основі ML у NDT (тобто навчання на основі нових даних, тестування та експлуатація). Крім того, модуль управління NDT відповідає за активацію моделей, зазначених під час створення конкретного екземпляра NDT RAN. Іншою функціональністю є керування процесами емуляції на NDT, коли він використовується для оцінки та дослідження продуктивності мережі в різних ситуаціях (тобто, ситуації "що, якщо", такі як різні рівні попиту на трафік, різноманітні конфігурації вузлів тощо). Крім того, модуль управління неруйнівним контролем (NDT) відповідає за моніторинг виконання модулів NDT, що охоплює споживання ресурсів, виявлення проблем, зв'язок між NDT RAN та додатками, а також трафік, що виник. Запропонована архітектура враховує, що різні додатки можуть використовувати RAN NDT. Кожен додаток надається з екземпляром RAN NDT, налаштованим відповідно до конкретних вимог додатка. Це налаштування може включати активацію вибраних моделей відображення послуг, специфікацію даних, які потрібно збирати в сховищі даних, визначення необхідних ключових показників ефективності (KPI), специфікацію можливостей аналізу та візуалізації для додатка, специфікацію емуляцій, якщо потрібно. На Рис. 3.8 визначено різні відповідні програми, пов'язані з різними етапами життєвого

циклу RAN, які визначені як основні варіанти використання для неруйнівного контролю RAN [16]:

- Програми для планування та визначення розмірів: під час розгортання нових gNB у RAN або налаштування існуючих, емуляції можна виконувати на екземплярі NDT RAN, щоб визначити найбільш адекватні місця розгортання gNB та конфігурації для покращення покриття та пропускну здатності мережі. Для такого типу програми екземпляр NDT включатиме вже розгорнуті gNB та їхні конфігурації. Потім на екземплярі NDT RAN можна виконувати різноманітні емуляції, пов'язані з різними конфігураціями нового gNB [16].

- Програми для роботи мережі: Під час роботи мережі екземпляр RAN NDT може сприяти моніторингу мережі, розширюючи показники продуктивності, отримані з реальної мережі, додатковими показниками, отриманими з емуляцій на RAN NDT. На основі цього можна передбачити можливі збої або аномалії та вжити заходів для їх уникнення. Тому екземпляр RAN NDT для цієї програми повинен включати всі елементи (топологію сценарію, gNB, характеристику UE тощо), як і в реальній RAN, яка буде отримувати дані в режимі реального часу від RAN [16].

- Програми для оптимізації мережі: Оптимізація RAN може виграти від екземпляра RAN NDT для емуляції різних конфігурацій RAN, а також для тестування алгоритмічних рішень перед їх застосуванням у реальній RAN, що дозволяє налаштовувати та перевіряти їхні параметри. Тестування конфігурацій можна проводити шляхом їх застосування до екземпляра RAN NDT та виконання емуляцій. У випадку алгоритмічних рішень, незалежно від того, чи базуються вони на евристиці, аналітиці чи машинному навчанні, їх слід інтегрувати в екземпляр RAN NDT для оцінки їхньої продуктивності за допомогою емуляцій. Наприклад, якщо тестується політика RRM для контролю доступу, її слід завантажити як функціональну модель для управління та оптимізації RAN, а відповідні рішення щодо доступу слід приймати на моделях екземплярів RAN NDT для оцінки їхньої продуктивності. Крім того, зауважте, що у випадку рішень

на основі машинного навчання (ML), а особливо рішень на основі RL, їх навчання можна виконувати на екземплярі NDT [16].

NDT дозволяє виконувати навчання рішень RL для RAN. Процес навчання моделі RL полягає у навчанні оптимальної політики, яка вибирає найкраще рішення (тобто дію) для кожної можливої ситуації (тобто стану), яка отримує максимальну винагороду. Для досягнення цього, під час процесу навчання агент RL (тобто той, хто навчається) ітеративно взаємодіє з середовищем, де на кожній ітерації агент RL отримує стан, вибирає дію, а потім, в результаті останньої дії, отримує винагороду, яка оцінює, наскільки хорошою чи поганою була остання дія для останнього стану, разом з новим станом [14]. Дії вибираються відповідно до експлуатації, де дії вибираються відповідно до політики агента RL, та дослідження, де дії вибираються випадковим чином. З цієї взаємодії агент RL оновлює політику. Це повторюється до досягнення умови конвергенції, коли процес навчання завершується [16].

Рішення RL можуть бути застосовані до проблем RAN, що стосуються певного виду прийняття рішень, головним чином пов'язаних з функціями RRM, RNM або SON. Рішення RL для функцій RRM динамічно керують виділеними ресурсами, що працюють у різних часових масштабах від мілісекунд до кількох секунд, та охоплюють рішення для різних рівнів стеку протоколів. Прикладами застосовності RL для функцій RRM є кодування каналів, керування потужністю та динамічний доступ до спектру на фізичному рівні, планування на рівні MAC, керування допуском для керування радіоресурсами (RRC) та подвійне з'єднання на рівні протоколу конвергенції пакетних даних (PDCP). Крім того, RL може бути застосований до функцій RNM та SON для підтримки мережевого планування, розгортання та експлуатації RAN з точки зору конфігурації, оптимізації та управління несправностями, що працюють у довгострокових часових масштабах. Наприклад, приклади застосовності RL, пов'язані з функціями SON, включають оптимізацію ємності та покриття, оптимізацію стійкості мобільності, балансування навантаження мобільності або оптимізацію ємності між сегментами, також відому як розподіл ємності, для призначення

доступних ресурсів різним сегментам. Незважаючи на широке застосування RL для RAN, навчання цих рішень є актуальною проблемою з точки зору впровадження. Це пояснюється тим, що метод спроб і помилок, що виникає під час навчання рішень RL, може призвести до неприйняттого зниження продуктивності мережі, якщо навчання виконується безпосередньо на реальній RAN. Щоб вирішити цю проблему, навчання RL можна безпечно виконувати на RAN NDT. Це можна здійснити, розглядаючи рішення RL як додаток на Рис. 3.8, який взаємодіє з екземпляром RAN NDT під час навчання. На Рис. 3.9 зображено запропоновану функціональну модель для навчання моделей RL для RAN за допомогою RAN NDT. Для полегшення її практичної реалізації модель узгоджена з основними компонентами та функціональними можливостями, що беруть участь у навчанні моделей машинного навчання (ML) у робочому процесі ML O-RAN Alliance. Ці функціональні можливості позначені синім кольором на Рис. 3.9. Як запропоновані компоненти, функціональна модель включає екземпляр RAN NDT, позначений зеленим кольором, та різні компоненти для підтримки навчання моделей на основі RL на ньому, позначені помаранчевим кольором [16].

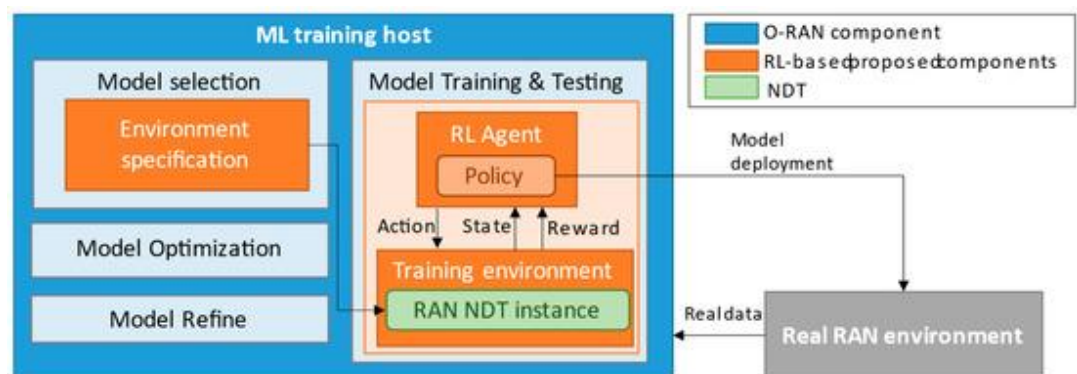


Рис. 3.9. Функціональна модель процесу навчання моделі RL на NDT [16]

Навчання виконується на хості навчання ML, який включає різні функціональні можливості, визначені в робочому процесі ML O-RAN. Навчання моделей RL виконується на хості навчання ML, що включає різні компоненти, а саме: навчання та тестування моделі, вибір моделі, оптимізацію моделі та

уточнення моделі, всі вони запропоновані в робочому процесі ML O-RAN Alliance. Серед цих компонентів хоста для навчання машинного навчання, навчання моделі RL виконується агентом RL в рамках навчання та тестування моделі, де процес ітерації політики RL відбувається через ітеративну взаємодію агента RL з екземпляром RAN NDT, який становить навчальне середовище, використовуючи дослідження та експлуатацію. Отже, під час процесу навчання агент RL ітеративно отримує стани та винагороди від екземпляра RAN NDT та надсилає йому дії. Потім, відповідно до станів, дій та винагород, політика оновлюється. Конкретний алгоритм оновлення політики залежить від розглянутого методу RL (наприклад, для методів Q-навчання та DQN відповідні алгоритми можна знайти в [14,15] відповідно). Під час навчання контролюються різні ключові показники ефективності (KPI) для оцінки збіжності моделі та визначення того, чи завершено процес навчання. Після цього відбувається тестування моделі, що полягає у перевірці поведінки навченої моделі. Це виконується шляхом вибору дій відповідно до навчених політик у режимі експлуатації та їх запуску на екземплярі RAN NDT. Під час цього процесу продуктивність навчених політик може бути оцінена за допомогою різних ключових показників ефективності (KPI), пов'язаних з RAN (наприклад, пропускна здатність, коефіцієнти помилок). Після завершення навчання та перевірки модель RL (тобто навчена політика) може бути упакована, наприклад, як контейнер, та розгорнута в реальному середовищі RAN, де дії моделі RL виконуються на етапі, відомому як висновок. У навчальному хості ML функція вибору моделі вибирає конфігурації для логічного висновку та навчання моделі RL на основі вимог рішення RL, наприклад, точності навчання, часу навчання, апаратних ресурсів для навчання та логічного висновку, швидкості логічного висновку тощо. Для керування та налаштування навчального середовища пропонується включити функцію специфікації середовища до функціональності вибору моделі. Зокрема, ця функція налаштовує екземпляр RAN NDT, вказуючи моделі екземпляра RAN NDT, необхідні для навчання моделі RL. З цією метою специфікація середовища взаємодіє з модулем керування NDT на Рис. 3.8 для

налаштування необхідних моделей у функціональності активації та налаштування моделі. Наприклад, це може включати активацію та налаштування моделей топології gNB та сценаріїв на Рис. 3.8 екземпляра RAN NDT відповідно до конфігурацій та топології gNB у реальній RAN відповідно. Крім того, для забезпечення високої продуктивності навчання моделі RL, агент RL повинен випробувати під час навчання ті відповідні ситуації (тобто стани), які будуть випробувані пізніше під час логічного висновку. Функція специфікації середовища також відповідає за конфігурацію параметрів екземпляра RAN NDT для емуляції цих ситуацій у функціональності керування емуляцією роботи управління NDT. Наприклад, для емуляції можна налаштувати різні рівні трафіку або розподіл навантаження між різними gNB. Оскільки екземпляр RAN NDT може отримувати дані з реального середовища RAN, які будуть використовуватися моделями відображення послуг, показаними на Рис. 3.8, ще одним обов'язком функції специфікації середовища є визначення даних, необхідних екземпляру RAN NDT. На основі цієї конфігурації дані збираються з реального середовища RAN та зберігаються у сховищі даних екземпляра RAN NDT [16].

Щодо компонента оптимізації моделі в навчальному хості ML, він дозволяє оптимізувати гіперпараметри моделі RL (наприклад, кількість шарів нейронної мережі у випадку глибокого RL, кількість нейронів на шар) на основі певних вимог до апаратного забезпечення або показників продуктивності, таких як точність моделі, розмір моделі, швидкість виведення, використана пам'ять тощо. Потім екземпляр RAN NDT може бути використаний для тестування продуктивності різних політик, пов'язаних з різними конфігураціями гіперпараметрів. Зрештою, компонент уточнення моделі дозволяє, за потреби, оновити модель шляхом перенавчання після того, як вона вже була розгорнута в реальній RAN. Це може знадобитися, коли продуктивність розгорнутих політик більше не є оптимальною через невідповідність між умовами, що враховувалися під час навчання, та умовами в реальному середовищі RAN. Наприклад, це може статися, коли до моделі gNB вносяться зміни (наприклад, розгортання нового

gNB) або конфігурації її елементів (наприклад, зміна конфігурації gNB), або також коли користувачі в реальній RAN починають поводитися по-іншому (наприклад, нові вимоги до UE, нові послуги пропонуються в реальній RAN). У цих ситуаціях компонент уточнення моделі отримує сповіщення і, відповідно, запускає процес перенавчання, що проводиться на екземплярі NDT RAN, оновленому відповідно до нових умов [16].

Рішення для розподілу потужності на основі RL виконує роль програми (App), як показано на Рис. 3.8. Зокрема, розглянуте рішення для розподілу потужності - це підхід до розподілу потужності на основі багатоагентного навчання з підкріпленням Deep Q-Network Multi-Agent-Reinforcement Learning (DQN-MARL), запропонований у роботі в [17]. Рішення дозволяє динамічно розподіляти доступну потужність в інфраструктурі RAN, що складається з N gNB, між K орендарями, кожен з яких має сегмент RAN. Кожен gNB n має загальну потужність s_n (біт/с). Рішення спрямоване на ефективне використання доступної потужності в gNB та, водночас, на задоволення SLA орендарів. SLA, встановлена для k -го орендаря, визначається з точки зору: (а) агрегованої гарантованої швидкості передачі даних за сценарієм, SAGBR k , яка є агрегованою потужністю, що надається всіма gNB орендарю k за запитом, та (b) максимальної швидкості передачі даних комірки, MCBR k,n , яка є максимальною швидкістю передачі даних, яка може бути надана орендарю k у комірці n . Рішення для спільного використання потужності DQN-MARL враховує, що кожен орендар пов'язаний з різним агентом RL. Кожен агент налаштовує квоту ресурсів (тобто частку потужності), призначену сегменту орендаря в різних gNB, з кроками часу тривалістю Δt , у часових масштабах порядку хвилин. Щоб дізнатися політику, яка налаштовує квоту ресурсів для кожного орендаря, на кожному кроці часу агент RL отримує стан орендаря в різних gNB середовища, який визначається як кортеж з різними метриками, включаючи використання ресурсів та квоту ресурсів орендаря, квоту ресурсів, не призначену жодному орендарю, ресурси, що не використовуються в gNB, та параметри SLA орендаря. Відповідно до отриманого стану агент RL вирішує, які

дії слід виконати в кожній gNB, які можуть полягати у збільшенні квоти ресурсів у Δ , зменшенні її в Δ або збереженні її незмінною. Щоб оцінити, наскільки застосована квота ресурсів у кожній gNB була придатною, кожному агенту RL на наступному кроці часу надається винагорода. Визначення винагороди сприяє задоволенню параметрів SLA та мінімізації ситуацій перевитрати. Під час навчання рішення агенти RL взаємодіють з екземпляром NDT RAN, з якого отримуються сигнали стану та винагороди, а дії, вибрані за допомогою ϵ -жадібної стратегії, застосовуються в різних gNB. На основі цієї взаємодії політика, яка налаштовує квоти ресурсів, оновлюється до досягнення конвергенції. Реалізований RAN NDT для навчання рішення для спільного використання потужності DQN-MARL відповідає конкретному екземпляру RAN NDT, показаному на Рис. 3.8. Відповідно до функціональної моделі на Рис. 3.9, модуль специфікації середовища в навчальному хості визначає моделі відображення послуг екземпляра RAN NDT та їх конфігурацію для емуляції різних реалістичних умов RAN (тобто станів агентів RL), застосування дій RL та отримання винагород. Зокрема, реалізований екземпляр RAN NDT дозволяє реалістично фіксувати просторовий розподіл трафіку в області з різними gNB та емулювати поведінку мережі за різних динамічних умов, коли UE рухаються за сценарієм та генерують трафік. Далі наведено опис базових та функціональних моделей у реалізованому NDT (тобто екземплярі RAN NDT). Впроваджений метод неруйнівного контролю (НК) підійшов до базових моделей наступним чином [16]:

- Модель топології сценарію: відповідає міському сценарію розміром 700×700 м у Барселоні, показаному на Рис. 3.10, який охоплює різні вулиці, проспекти, парк та різні семиповерхові будівлі висотою попереху 3,5 м [16].
- Модель gNB: включає gNB, розгорнуті в цій зоні. Кожна з них моделюється як зовнішня мікростільника та визначається такими параметрами: положення gNB, висота, робоча частота, загальна передана потужність, загальна пропускна здатність, розділення піднесучих, загальна кількість блоків фізичних ресурсів (PRB), дуплексний режим, коефіцієнт підсилення та коефіцієнт шуму

антени gNB. Враховується лише напрямок низхідного каналу (DL). Положення gNB в NDT показано на Рис. 3.10. Щоб уникнути ефектів меж, передбачається, що дослідження алгоритму розподілу ємності виконується лише в п'яти центральних gNB, які пронумеровані та виділені жовтим кольором на рисунку [16].

- Модель UE : визначається щільністю користувачів у зоні (користувачі/км²), типом UE, який може бути стаціонарним, пішохідним або транспортним, а також положенням UE, яке змінюється динамічно, висотою UE, коефіцієнтом підсилення антени, коефіцієнтом шуму та необхідною швидкістю передачі даних R_b [16].

- Модель каналу : забезпечує співвідношення сигнал/перешкода та шум (SINR) для кожного UE, враховуючи параметри UE та gNB та отримуючи втрати на трасі відповідно до положення UE та моделі поширення нерозривного детектора (NDT). Крім того, модель каналу забезпечує спектральну ефективність S_{eff} у біт/с/Гц для кожного UE відповідно до SINR та припускаючи, що максимально досяжне значення S_{eff} становить 7,4063 біт/с/Гц, і що UE перебуває в стані збою, якщо S_{eff} нижче 1 біт/с/Гц. Крім того, були враховані коефіцієнти неефективності, зумовлені циклічним префіксом передачі 5G та накладними витратами, які приймають значення 14/15 та 0,8 відповідно [16].

- Модель мережевого сегмента: визначається пов'язаною угодою про рівень обслуговування (SLA), яка гарантується для різних мережевих сегментів (тобто параметрами SAGBR_k та MCBR_k в алгоритмі розподілу потужності DQN-MARL), вимогами до швидкості передачі даних для UE, що належать до мережевого сегмента R_b , параметрами генерації трафіку для мережевого сегмента та відсотком користувачів різних типів у мережевому сегменті. Крім того, тут також налаштовується квота ресурсів мережевого сегмента в різних gNB протягом поточного часового кроку [16].



Рис. 3.10. Карта, включена до моделі топології сценаріїв реалізованого RAN [16]

На Рис. 3.11. піддіаграми з (a)-(e) зображують еволюцію пропонованого навантаження (тобто необхідної бітової швидкості) на одного орендаря та сукупне пропоноване навантаження обох орендарів у gNB від 1 до 5 відповідно, обидва нормалізовані до середньої ємності gNB [16].

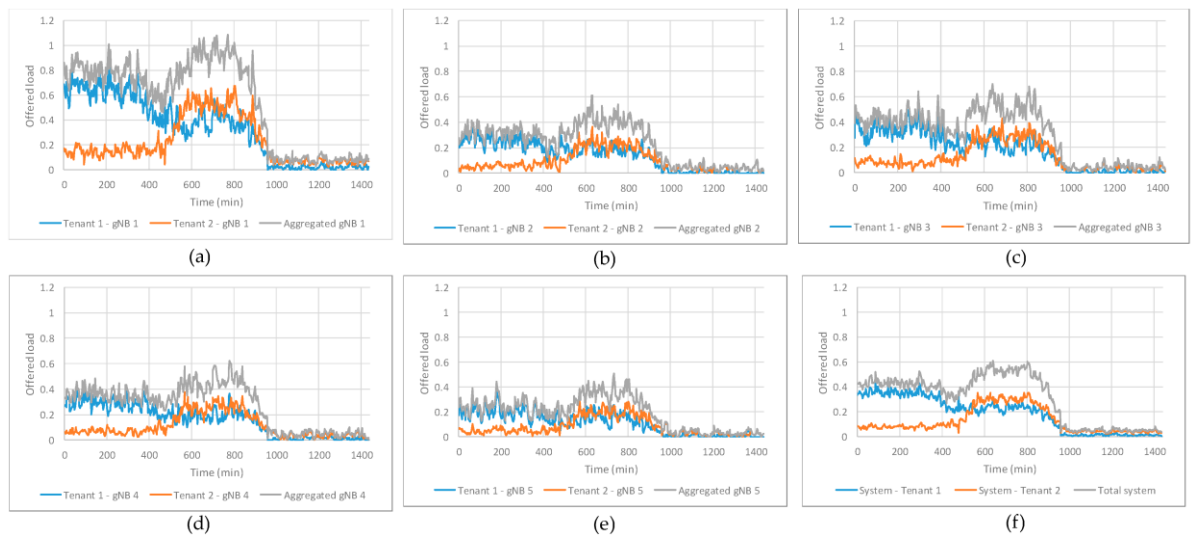


Рис. 3.11. Пропоновані навантаження на одного орендаря та агреговані для (a) gNB 1, (b) gNB 2, (c) gNB 3, (d) gNB 4, (e) gNB 5 та (f) на системному рівні [16]

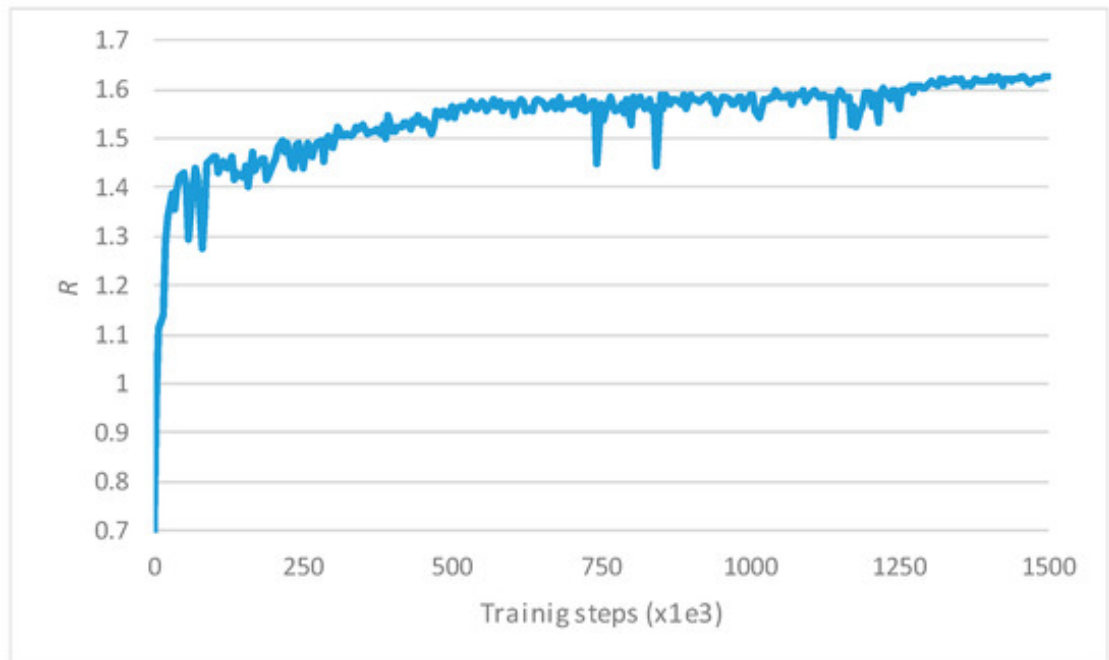


Рис. 3.12 Зміна сукупної винагороди орендарем 1 та орендарем 2 під час навчання [16]

Як результат моделі візуалізації в RAN NDT, на Рис. 3.12 зображено еволюцію агрегованої середньої винагороди обох орендарів, позначеної як R , в результаті оцінок, що проводилися періодично під час навчання. На рисунку видно, що протягом перших $250 \cdot 10^3$ кроків навчання R різко зростає, зазнаючи значних коливань. Після цього періоду R зростає повільніше з меншими коливаннями, доки не стабілізується на значенні близько 1,6 після приблизно $1500 \cdot 10^3$ кроків навчання. Ці результати підкреслюють корисність впровадженого RAN NDT для проведення навчання рішення для спільного використання потужностей DQN-MARL [16].

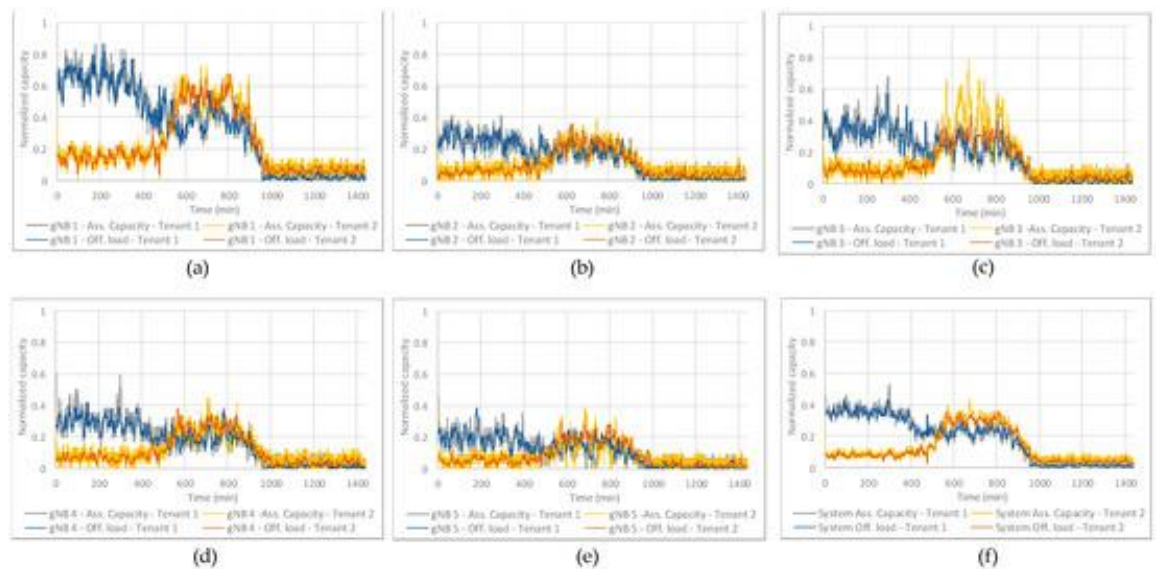


Рис. 3.13 Порівняння запропонованих навантажень та призначеної потужності на одного орендаря для (а) gNB 1, (б) gNB 2, (с) gNB 3, (д) gNB 4, (е) gNB 5 та (ф) на системному рівні [16]

Необхідні потужності загалом задовольняються, що свідчить про успішне навчання політик на RAN NDT. Модель візуалізації в RAN NDT також надає Рис. 3.13, який порівнює призначену потужність на одного орендаря за вивченою політикою після конвергенції до їхніх відповідних запропонованих навантажень у різних gNB (піддіаграми (а)-(е)) та на системному рівні (піддіаграма (ф)). На піддіаграмах (а)-(е) як запропоновані навантаження, так і призначені потужності нормалізовані до потужності gNB. Результати на цих піддіаграмах ілюструють, що призначена потужність на комірку адаптується до запропонованого навантаження у всіх gNB. Крім того, піддіаграма (ф) порівнює запропоновані навантаження та призначені потужності обом орендарям в агрегованому вираженні по всіх gNB, обидва нормалізовані до системної потужності. З цієї піддіаграми можна спостерігати той факт, що вивчені політики дозволяють задовольнити вимогу SAGBR k , оскільки агреговані запропоновані навантаження на одного орендаря, які нижчі за їхні відповідні SAGBR k , задовольняються. Отже, вивчені політики на реалізованому RAN NDT дозволяють адаптувати призначену потужність до запропонованих навантажень, одночасно задовольняючи SLA обох орендарів [16].

Висновки

У розділі розглянуто використання цифрових двійників для управління мережами 5G. Цифрові двійники дозволяють значно покращити ефективність управління мережею 5G, надаючи можливість моделювати, контролювати та оптимізувати продуктивність мережі в реальному часі. Використання xApps на Near-RT RIC дозволяє ефективно працювати з даними, що надходять з елементів мережі, і оновлювати віртуальну модель мережі (DT) для відображення поточного стану фізичної мережі. Платформа Microsoft Azure забезпечує надійний хмарний сервіс для підтримки цифрових двійників, що дозволяє легко інтегрувати та масштабувати рішення для різних сценаріїв використання. Архітектура O-RAN забезпечує сумісність між постачальниками обладнання і програмного забезпечення, створюючи гнучку платформу для управління 5G-мережами. Технологія TwinRAN демонструє високу практичну корисність, завдяки своїй неінвазивності, можливості інтеграції з різними постачальниками та багатоцільовому використанню. Важливими перевагами є забезпечення реального часу оновлення цифрових двійників, можливість аналізу «що, якщо», а також адаптація ресурсів для оптимального функціонування мережі. Важливим аспектом є використання xApps для підтримки мережевих функцій, таких як керування трафіком, управління завадами та оптимізація маршрутизації. Визначено, що для належного функціонування цифрових двійників важливо забезпечити інтеграцію з існуючими інфраструктурами через стандартизовані інтерфейси, що дозволяє досягти високої ефективності та гнучкості в управлінні мережами 5G. Отже, цифрові двійники стають критично важливим елементом для управління мережами 5G, оскільки вони дозволяють забезпечити точність, гнучкість і масштабованість при управлінні складними системами та передбачати проблеми до їх виникнення, що є особливо важливим для майбутніх технологій, таких як 6G.

РОЗДІЛ 4

ЦИФРОВИЙ ДВІЙНИК ДЛЯ МЕРЕЖ ТА ПЕРСПЕКТИВА МОДЕЛЮВАННЯ ПРОДУКТИВНОСТІ НА ОСНОВІ ДАНИХ

4.1. Вимоги до моделей продуктивності

Нескінченна гонитва за високою пропускнуою здатністю та низькою затримкою спонукає до розвитку нових технологій (наприклад, 5G, хмарних обчислень). Тим часом, високопродуктивні мережеві сервіси, в свою чергу, породили цілу низку нових додатків (наприклад, прямі трансляції, віртуальна реальність, хмарні ігри). Ці технології та додатки роблять мережу безпрецедентно складною і неоднорідною, що призводить до дорогих і ризикованих практик у фізичній мережі. У цьому контексті мережа цифрових двійників (DTN) [18] може значно полегшити навантаження на практиків. DTN - це віртуальне представлення фізичної мережі зв'язку, яке постійно оновлюється даними про продуктивність, технічне обслуговування та стан здоров'я останньої. На відміну від типових технологій цифрових двійників, які замінюють фізичні об'єкти, DTN впершу чергу займається абстрагуванням станів та поведінки мережі. DTN має на меті створення цифрових двійників для універсальних мереж зв'язку і не обмежується конкретними додатками чи контекстами. Мережеві оператори часто прагнуть розробити методи оптимізації для підвищення продуктивності мережі, які в основному включають в себе налаштування конфігурації і вивчення нових політик. Нові конфігурації і політики повинні бути повністю перевірені перед розгортанням у фізичних мережах. DTN можна використовувати як безпечне і економічно ефективне середовище для оцінки продуктивності. Оператори можуть досліджувати і перевіряти свої нові технології в DTN, уникаючи складних і ризикованих операцій на фізичних мережах. Оптимізація та інші сценарії вимагають оцінки продуктивності для сценаріїв "Що, якщо" [19], що означає, що DTN може визначити, якою буде продуктивність мережі при зміні факторів впливу (наприклад, обсягів трафіку,

конфігурацій пристроїв, схем маршрутизації, топологій). Оцінка продуктивності мережі привертала інтерес дослідників протягом десятиліть. Експерименти та вимірювання - це два методи оцінки продуктивності фізичних виробничих мереж, які характеризуються високим ризиком, високими накладними витратами та високою складністю для сценаріїв "Що, якщо". У віртуальних середовищах симуляція та моделювання є двома фундаментальними підходами для оцінки продуктивності за сценарієм "Що, якщо". Симулятори роботи мережі (наприклад, NS- 2, NS-3, OMNet++) обробляють віртуальні пакети за заздалегідь визначеними механізмами (наприклад, алгоритми контролю перевантажень, політики формування черг) і генерують метрики продуктивності (наприклад, пропускну здатність, затримку, рівень втрат), які дозволяють збирати довільну інформацію, не впливаючи на поведінку системи. Такі симулятори пакетного рівня мають делікатний дизайн і тісно пов'язані між собою, що призводить до неефективного виконання. У цьому контексті варто звернути увагу на 3- 4 порядки повільніше, ніж у реальному часі [20], визначають, що сучасні симулятори є неприйнятними для DTN. Моделювання значно відрізняється від імітації, яка безпосередньо встановлює взаємозв'язки між факторами впливу та показниками ефективності. Традиційні методи аналітичного моделювання (наприклад, розрахунок чистої роботи, теорія черг) використовують пуассонівський процес для імітації процесу надходження і відправлення пакетів, який не може описати інкаст [21] і призводить до неточної оцінки. Імітатори важкі, а аналітика формалістична, і обидва вони не гнучкі до швидкої і безперервної еволюції мережі. З відродженням машинного навчання в останні роки методи, що базуються на даних, особливо нейронні мережі (НМ), здаються перспективними для оцінки продуктивності DTN. Методи, засновані на даних, зазвичай попередньо визначають ряд можливих функцій відображення і використовують велику кількість даних для визначення набору параметрів (тобто навчання), які точно відображають фактори впливу на показники ефективності. Функція відображення здатна описувати різні взаємозв'язки, що забезпечує гнучкість для моделювання складних мережевих механізмів. Навчені

моделі, керовані даними, мають невелику вагу і можуть ефективно генерувати результати за допомогою лише одноразових прямих обчислень. Дослідники розробили спеціалізовані структури для конкретних завдань (наприклад, згорткові нейронні мережі для комп'ютерного зору, рекурентні нейронні мережі для обробки природної мови) та прикладні методи (наприклад, регуляризації, відсіювання) для вирішення проблем надмірної підгонки та узагальнення, що робить навчені моделі надійними. Завдяки цим перевагам, методи, керовані даними, можуть допомогти пом'якшити проблеми неефективності, негнучкості та неточності. У контексті застосування таких моделей, особливо важливо визначити чіткі вимоги до моделей продуктивності. Дослідники запропонували загальну архітектуру DTN [18], що включає три рівні, як показано на Рис. 4.1. Вона повинна періодично збирати статичні дані (наприклад, топологію, конфігурацію) і безперервно збирати дані під час виконання (наприклад, використання каналів, обсяг трафіку) і завантажувати їх в DTN. Потім DTN реконструює внутрішні взаємозв'язки зібраних даних для відображення фізичного стану мережі, що забезпечує можливість "що було б, якби". DTN принесе користь мережевій практиці, надаючи середовище оцінки продуктивності в режимі реального часу і без ризику. Дослідимо чотири типові мережеві сценарії планування, експлуатації, оптимізації та модернізації, щоб визначити конкретні вимоги до моделей продуктивності [33].

У сценарії планування проектувальники повинні переконатися, що загальна продуктивність мережі відповідає вимогам заданої топології, конфігурації і попиту на трафік. Модель повинна генерувати результати продуктивності при різних топологіях, конфігураціях і навантаженнях трафіку, що вимагає від моделі точності при різних комбінаціях вхідних даних. Під час експлуатації інженери сподіваються знати про зміни продуктивності в реальному часі і швидко реагувати на потенційні аномалії. Ці аномалії повинні бути швидко локалізовані або виявлені, як тільки вони з'являться. Моніторинг у реальному часі та виявлення аномалій вимагають, щоб модель ефективно відображала фізичну продуктивність мережі. Оптимізація мережі зазвичай передбачає

налаштування конфігурацій та вивчення нових політик. Конфігурації та політики повинні бути повністю перевірені перед розгортанням на практиці. Модель є ідеальним середовищем з нульовим рівнем ризику для дослідження схем та оцінки їхньої продуктивності за різних сценаріїв перед розгортанням. Модернізація мережі часто включає в себе зміну топології та розширення каналів зв'язку. Оператори можуть задатися питанням, як змінити топологію і де розширити пропускну здатність при обмеженому бюджеті ресурсів. Модель повинна оцінювати продуктивність при зміні топології та пропускну здатності каналів зв'язку, а також визначати вузькі місця, щоб максимізувати ефективність модернізації [33].



Рис. 4.1. Архітектура мережі цифрових двійників [33]

З вищезазначених аспектів впливають три вимоги (тобто, точність, ефективність і гнучкість), яких повинна прагнути досягти модель ефективності. Точність - це основна вимога, яка забезпечує точність для всіх сценаріїв. Ефективність має важливе значення для експлуатації та оптимізації через часте оцінювання продуктивності в режимі реального часу. Гнучкість в основному означає, що модель може оцінювати продуктивність при змінах в мережі, що полегшує планування, оптимізацію та модернізацію мережі. Далі ми зупинимося

на трьох вимогах, які наведені нижче [33]:

- Точність показує, наскільки точно метрики з моделей продуктивності відповідають фізичній мережі. Поділяється точність на три рівні (довгострокова, короткострокова, один до одного) з точки зору часової перспективи. Довгострокова представляє статистичні результати за певний період часу, короткострокова описує детальні зміни в часових інтервалах, а точність "один до одного" точно відображає кожен пакет фізичної мережі. Довгострокова стабільна оцінка вимагає від моделі розумного абстрагування складних механізмів, таких як RouteNet [22], що може представляти довільні схеми маршрутизації. Крім того, для представлення короткострокових процесів потрібно описати умови мережі та часові залежності, наприклад, xNet [23] вивчає функцію переходу стану між часовими кроками. Один до одного означає, що ми повинні точно змодельювати вплив різних механізмів на кожен пакет. MimicNet [20] зберігає функцію кінцевих хостів і моделює вплив мережевих кластерів на пакети, що є першим кроком до моделювання "один до одного" [33].

- Ефективність складається з двох частин. Перша полягає в тому, що моделі продуктивності можуть бути швидшими, ніж фізична мережа в реальному часі. З іншого боку, моделі повинні бути простими у розгортанні та споживати раціональні ресурси. Швидкість, вища за фізичну, дозволяє користувачам прогнозувати продуктивність і заздалегідь реагувати на потенційні аномалії. На відміну від симуляторів, вона повинна спрощувати складні механізми і відкинути непотрібні деталі (наприклад, корисне навантаження пакетів, процес комутації), щоб пришвидшити оцінку. Часто існує компроміс між деталізацією моделювання (наприклад, на рівні пакетів [20], потоків [23], [24], шляхів [22], [25]) і швидкістю, яка визначається цільовою постановкою задачі. Модель повинна враховувати компроміс і слугувати економічно ефективним бекендом для оцінки метрик продуктивності [33].

- Гнучкість: Мережа швидко розвивається. Коли змінюються топологія, конфігурації або механізми, модель також повинна точно генерувати метрики продуктивності. Модель може бути ітеративно модернізована до нових

механізмів, що означає, що модель повинна бути гнучкою. Зазвичай у мінливій системі залишаються незмінні частини, що надихає на використання багаторівневої або модульної філософії для вирішення проблеми гнучкості. Шари і модулі роз'єднані, але організовано функціонують в одній моделі, де ми можемо окремо конструювати кожен модуль. Нові механізми не впливатимуть на всю модель, а лише на пов'язані з нею модулі, які можна буде або модернізувати, або замінити. Гнучкість прискорить процес побудови точної моделі продуктивності [33].

4.2. Вибрані моделі продуктивності

За останні десять років дослідники мереж виявили великий інтерес до моделювання продуктивності на основі даних. Дані, модель і застосування - це три основні аспекти підходу, заснованого на даних. Дані необхідні для навчання моделі, що відрізняє його від традиційних аналітичних або імітаційних підходів. Певною мірою кількість і якість наявних даних обмежує точність моделі, побудованої на основі даних. Тим часом, моделі повинні бути розумно розроблені, і добре розроблена модель може ефективно витягувати зв'язки між змінними для побудови самої себе. Дослідники також розробили спеціальні методи щодо архітектури моделі та функції втрат для отримання більш точних результатів. Більшість моделей застосовуються не лише для цільових завдань, але й використовуються для інших сценаріїв, демонструючи широкі перспективи моделей ефективності [33].

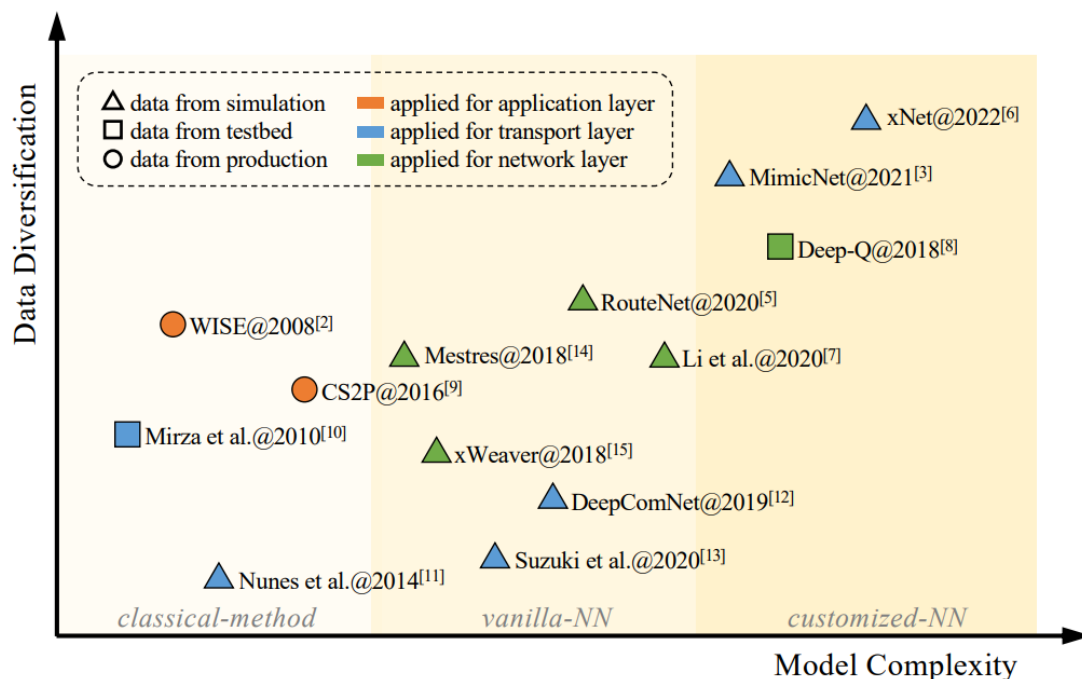


Рис. 4.2. Якісне порівняння диверсифікації даних, складності моделі та сценаріїв застосування обраних моделей ефективності на основі даних [33]

Ці моделі уможливили широкі сценарії застосування для різних рівнів, в основному, включаючи мережевий, транспортний та прикладний рівні. Чим нижчий рівень, тим детальніші абстракції потрібні для моделювання. З розвитком методів моделювання спостерігається тенденція переходу від сценаріїв вищих рівнів до сценаріїв нижчих рівнів. Дослідники також усвідомили великий потенціал моделей продуктивності і використовували їх для вирішення проблем, які в минулому здавалися складними, наприклад, моделювання схем маршрутизації [22]. Тут можна навести якісне порівняння диверсифікації даних, складності моделей та сценаріїв застосування вибраних моделей, як показано на Рис. 4.2. Джерелом даних в основному є моделювання, тестові стенди та виробництво, де складність вимірювання різко зростає. Однак багато робіт збирають дані з середовища моделювання. Вони часто використовують симулятори на рівні пакетів або потоків, які є недорогими і відносно простими у розгортанні. Хоча різні дані зручно збирати в такому обмеженому середовищі, точність даних може відрізнятись від реальних. Крім

того, великомасштабні симуляції на симуляторах пакетного рівня можуть зайняти багато часу. У роботах [25], [27] для отримання даних будуються тестові стенди. Незважаючи на високу вартість, тестові стенди ближчі до реального середовища і мають високу швидкість. Дослідники можуть збирати велику кількість чистих даних з контрольованих тестових стендів. Однак, як симулятори, так і тестові стенди мають проблеми з моделлю трафіку. Структуру і розподіл потоків важко описати через складну і різноманітну поведінку дорожнього руху. Без надійних моделей трафіку достовірність зібраних даних знижується. Безпосереднє вимірювання даних з виробничих мереж є виграшним, оскільки час моделювання та модель трафіку більше не є проблемою. WISE [19] використовував дані tcpdump з глобальної мережі доставки пошукового контенту Google (CDN), а Sun та ін. [26] використовували власний набір даних вимірювання пропускної здатності HTTP з операційної платформи iQIYI. Реальні дані можуть забезпечити більшу достовірність і дозволяють розгортати моделі продуктивності для практичного використання. Однак збір великої кількості послідовних і чистих даних з виробничого середовища все ще залишається дорогим. Для проведення вимірювань часто потрібне спеціалізоване обладнання та методики. Навіть попри це, деякі метрики (наприклад, довжина черги [21]) не можуть бути виміряні безпосередньо. Високоточна синхронізація часу на пристроях також перешкоджає точному вимірюванню метрик, пов'язаних з часом [33].

Вхідні дані є різними для різних задач. Деякі роботи використовують декілька дискретних або неперервних змінних як вхідні дані. WISE [19] проаналізував набір даних tcpdump і виявив, що ряд змінних (наприклад, мітка часу, кількість надісланих пакетів, регіон клієнта, час в обидва боки (RTT), час відгуку) відповідають за час відгуку сервісу. Мірза та ін. [27] припустили, що включення властивостей шляху (наприклад, затримки в черзі, втрати, доступна пропускна здатність) в якості вхідних даних може покращити прогнозування пропускної здатності TCP на основі історії. CS2P [26] вибрав заданий набір ознак з усіх можливих комбінацій ознак (наприклад, IP клієнта, провайдера, AS, міста,

сервера) для агрегування кластерів сеансів. RouteNet [22] використовував стабільний трафік за різних топологій, схем маршрутизації та пропускної здатності каналів, щоб визначити затримку та рівень втрат на шляху. Лі та ін. [24] взяли розмір потоку, час початку, тип обслуговування (ToS), протокол і пропускну здатність каналу для моделювання часу завершення потоку (ЧЗП). Хоча це створює більший тягар, диверсифіковані вхідні дані дадуть точніші результати та сильнішу здатність до узагальнення. Існують також роботи, які використовують меншу кількість ознак як вхідні дані. Такі роботи часто зосереджуються на заданих сценаріях і вирішують конкретні завдання. Nunes та ін. [28] і Suzuki та ін. [30] використовували лише вимірні метрики (RTT/затримки до одних вузлів) для оцінки однорідних метрик (наступні RTT/затримки до інших вузлів). xWeaver [32], Deep-Q [25] і Mestres та ін. [31] надавали перевагу використанню матриць трафіку для моделювання наскрізних матриць пропускної здатності, де топологія та інші конфігурації були визначені. DeepComNet [29] використовували характеристики нижчого рівня (графічне представлення топології та потоків) для оцінки середньої продуктивності топології мережі. Менша кількість вхідних даних може зменшити витрати на вимірювання та уможливити швидке виконання, підвищуючи ефективність за певних налаштувань. Вихідні дані - це показники продуктивності (наприклад, RTT, затримка, пропускну здатність, коефіцієнт втрат, FCT) з меншою кількістю вимірів, ніж вхідні дані. Більшість робіт мають на меті оцінити довгострокову середню продуктивність в умовах сталого стану. Mirza та ін. [27] і CS2P [26] прогнозували пропускну здатність за період. xWeaver [32] і DeepComNet [29] оцінювали стабільну продуктивність даної топології. Mestres та ін. [31] і RouteNet [22] оцінювали середню затримку шляху, яка відображає загальну продуктивність шляху при стабільному трафіку. Suzuki та ін. [30] зробили висновок про наскрізні затримки пар вузлів при постійних потоках TCP. Лі та ін. [24] змоделювали FCT в DCN. Наведена вище довгострокова оцінка продуктивності полегшила прийняття рішень для цільових завдань. На відміну від наведених вище середніх показників, генерування розподілів показників

дозволяє користувачам оцінювати продуктивність з імовірнісної точки зору. WISE [19] оцінив розподіл часу відгуку сервісу за сценаріями "Що, якщо" і може забезпечити відсоткову гарантію для вимог угоди про рівень обслуговування (SLA). Deer-Q [25] перевіряв, що деякі показники якості обслуговування (QoS) є не скалярними, а випадковими величинами. Він зробив висновок про розподіл затримок і втрат на шляху проходження через певні проміжки часу. Deer-Q також показав зміни продуктивності з часом, що дозволило динамічно оцінювати продуктивність у часовому вимірі. Nunes та ін. [28] оцінювали наступний RTT за минулим вимірним RTT. xNet [23] уможливив тимчасове прогнозування на основі висновків QoS і FCT. MimicNet [20] моделював затримку для кожного пакета. Тимчасові метрики відображають зміни продуктивності з часом, що дозволяє використовувати більш цінні додатки (наприклад, моніторинг QoS, виявлення аномалій, вивчення політик). З огляду на побудови моделей, ми розділили останні досягнення на три етапи: етап класичного методу, етап ванільного NN та етап кастомізованого NN. Етап класичних методів: На цьому етапі класичні методи (наприклад, причинно-наслідкова байєсівська мережа (CBN), опорно-векторна регресія (SVR)) застосовуються для моделювання продуктивності. Хоча такі методи мають фіксовану і просту форму, вони показали хороші результати для конкретних завдань. WISE [19] визначив релевантні ознаки з великих змінних і використав CBN для побудови причинно-наслідкової структури цих ознак. Потім було застосовано статистичні інтервенції до змінених ознак для оцінки часу реагування за сценаріями "Що, якщо". Мірза та ін. [27] використовували SVR з функцією ядра Radial Basis для прогнозування пропускну здатності TCP. Функція втрат є ϵ - нечутливою до втрат з регуляризацією L2. SVR має міцне теоретичне підґрунтя, і йому надають перевагу на практиці завдяки його чудовій емпіричній ефективності. Нунес та ін. [28] використовували легкий Experts Framework для онлайн-навчання, яке показало високу швидкість з прийнятною точністю. Вони розробили спеціальну функцію кускових втрат для опису похибки між прогнозом і вимірюванням. CS2P [26] використовували приховану марковську модель (HMM) для фіксації

поведінки переходу від стану до стану для подібних кластерів, щоб прогнозувати їх для адаптації до бітрейту відео. На цьому етапі класичні методи, засновані на даних, часто мають фіксовані форми і мало емпіричних гіперпараметрів, де функції втрат легко визначити. Виходячи з цього, не так багато простору для розробки конкретних архітектур для конкретних завдань. Дослідникам потрібно абстрагуватися від проблем і звернути увагу на функціональну інженерію. Ці моделі є дуже ефективними і мають високу точність при правильному застосуванні, хоча вони можуть не бути гнучкими до еволюції мережі. З фіксованими формами їм важко моделювати складні мережеві механізми, що призводить до дуже обмеженого застосування. Стадія ванільного NN: З розвитком обчислювальних технологій та великої кількості доступних даних, NN виводять методи, керовані даними, в нову еру. ШН також застосовуються для моделювання продуктивності мережі, що дозволяє перейти до етапу ванільних ШН. xWeaver і DeepComNet використовували методи ШН для оцінки продуктивності заданої топології. xWeaver може дослідити оптимальну топологію для заданого трафіку, де для оцінки використовувався модуль підрахунку балів. Модуль оцінки використовував дві згорткові нейронні мережі (CNN) для вилучення інформації з трафіку і конфігурації топології окремо, з повністю підключеною NN для виведення оцінки продуктивності. DeepComNet представляв потоки і черги у вигляді вузлів графа. Коли потік проходить через чергу, ребро з'єднує вузол потоку і вузол черги. Потім для оцінки топології використовувалася нейронна мережа з закритим графом (GGNN). Местрес та ін. цікавилися, чи може NN точно моделювати затримку як функцію вхідного трафіку. Вони оцінили NN з різними гіперпараметрами (тобто кількістю прихованих шарів, кількістю нейронів у шарі, функцією активації, швидкістю навчання та параметром регуляризації) і дали ствердну відповідь. Судзукі провели напівкероване навчання за допомогою графових згорткових мереж (GCN) для визначення наскрізної затримки. Вони змоделювали проблему як питання класифікації і використовували логарифмічний класифікатор softmax. RouteNet використовував нейронну мережу передачі повідомлень (MPNN) для

моделювання взаємозв'язків між зв'язками і шляхами, виводячи середню затримку, джиттер і втрати на шляху. Лі абстрагувалися від потоків як від вузлів графа, а від зв'язків як від ребер графа. Вони вводять характеристики потоків і зв'язків у модель, побудовану за допомогою трьох блоків графових мереж (GN), щоб вивести FCT. У вищезгаданих роботах розглядалися проблеми з відповідними методами ШН, де структури моделей і функції втрат залишаються традиційними. Нові методи ШН безпосередньо адаптуються до конкретних задач. Їхні вхідні дані, як правило, структуровані (наприклад, структура графа) і не можуть бути безпосередньо оброблені класичними методами. Ці вхідні дані мають кілька вимірів, що дозволяє більшу гнучкість для моделювання довгострокових характеристик [33].

Кастомізований етап NN: Дослідники зрозуміли, що детальне моделювання продуктивності (наприклад, короткочасне і один до одного) є більш цінним і практичним. Deep-Q вперше змоделювали розподіл затримок і втрат у тракті на часових інтервалах. Вони поєднали варіаційний автокодер (VAE) з довгою короткочасною пам'яттю (LSTM). LSTM використовувався для вилучення інформації з послідовностей матриці трафіку, а VAE міг генерувати метричні розподіли. Спеціально розроблений модуль Cinfer-loss міг вимірювати похибку прогнозованих розподілів QoS, що забезпечило ефективне і точне навчання. MimicNet забезпечив прискорення на два порядки порівняно зі звичайним моделюванням для DCN з тисячами серверів. Вона використовувала модель, керовану даними, для заміни складних і повільних кластерів у DCN, зберігаючи при цьому масштабованість, гнучкість і точність. Для точного моделювання впливу (падіння, затримки, ECN тощо) на пакети вони розробили функцію втрат зі зваженою двійковою перехресною ентропією (BCE) та втратами Губера. xNet забезпечує загальний підхід до моделювання мережевих характеристик, що викликають занепокоєння з графічним представленням та конфігурованими блоками GNN. Він вивчає перехід стану між часовими кроками і розгортає його, щоб отримати всю траєкторію прогнозування. xNet використовував три мережеві графові мережі (NGN) для побудови моделі

переходу стану і застосовував втрати для переходів стану. Ці роботи моделювали мережу в динаміці та забезпечували вищу точність, коли входи та виходи змінювалися з часом. Вони розробили делікатні NN-моделі та застосували знання про предметну область для точного моделювання [33].

На початку не було багато застосувань для моделей, керованих даними. Багато робіт були присвячені одновимірному моделюванню. Ці роботи були розроблені для конкретних завдань і використовувалися лише в цільових сценаріях. WISE оцінював час відгуку сервісу за сценаріями "Що, якщо". Nunes використовували онлайн-навчання для оцінки RTT TCP-з'єднання. Це може бути реалізовано в ядрі Linux для підвищення точності оцінки RTT. Mirza і CS2P прогнозували пропускну здатність за певного фону для передачі файлів і адаптації бітрейту відео окремо. Mestres підтвердили, що NN може точно моделювати наскрізну затримку як функцію матриці трафіку. Вони не досліджували випадки використання для додатків. xWeaver був присвячений оцінці продуктивності топології. Він може бути використаний для визначення правильної топології для заданого трафіку. Перш за все, ми бачили дуже обмежене застосування цих робіт. Це може бути пов'язано з багатьма причинами, пов'язаними з методами збору даних і моделюванням. Менша кількість вхідних даних, одиничні вихідні дані та спеціалізована модель, яка підходить лише для конкретних завдань. Дослідники не розглядали можливість застосування цих моделей для більшої кількості сценаріїв. З розвитком різних методів ШН моделі стають все потужнішими. Широке застосування було запропоновано в роботах [20], [22]-[25], [29], [30]. Deep-Q сподівався вивести метрики продуктивності безпосередньо зі статистики трафіку в реальному часі. Його також можна було б використовувати для оптимізації продуктивності. DeepComNet був розроблений для оцінки продуктивності топології. Інженери можуть використовувати DeepComNet для прогнозування пропускну здатності TCP або наскрізної затримки UDP для планування мережі. RouteNet є чудовим прикладом застосування моделі для різних варіантів використання. Вони змоделювали середню затримку, джиттер і втрати на шляху, які використовуються для

маршрутизації з урахуванням QoS і модернізації мережі з обмеженим бюджетом. Лі також показали безліч застосувань для оптимізації трафіку, маршрутизації та планування потоків, управління топологією. Mim-icNet вдосконалив симулятор пакетного рівня і забезпечив масштабовану, швидшу і настроювану оцінку продуктивності для DCN. xNet запропонував структуру моделювання і продемонстрував три випадки використання для онлайн-моніторингу QoS, моделювання сценаріїв "Що, якщо" і мережевого планування. Очевидно, що моделі на основі даних мають більш широке застосування, ніж будь-коли раніше. Узагальнені методи на основі даних зменшили складність традиційних проблем за допомогою точних, гнучких та ефективних методів моделювання [33].

4.3. Виклики та можливості

Оцінка продуктивності на основі даних стрімко розвивається, і від цього виграють багато додатків. Незважаючи на яскраві перспективи, є ще багато перешкод, які потрібно подолати, а можливості залишаються позаду. Підсумуємо виклики та можливості щодо збору даних, побудови моделей та перспектив застосування наступним чином. Доступну для збору дані для навчання сильно впливають на точність і узагальненість моделей навчання. Дані з виробничих мереж часто мають вищу цінність, але їх кількість і типи обмежені. Реальні дані для навчання дозволяють представити практичну модель для розгортання, таку як WISE. У той же час, різноманітні дані повинні бути зібрані в різних конфігураціях, що нереально у виробничій мережі. Імітаційні середовища можуть бути гарною альтернативою для вирішення проблеми точності та диверсифікації. Однак такі симуляції часто забирають багато часу. Крім того, часто існує розрив між змодельованими даними та реальними даними, що заважає застосовувати навчені моделі у виробництві. Методи, що базуються на даних, широко розвинуті в широких галузях, де загальний набір даних і еталонів відіграє важливу роль, тоді як загально визнаних наборів даних і еталонів для мережевого моделювання досі не існує. Експерименти,

вимірювання та методи, що базуються на даних, є перспективними для вирішення цих проблем. Інженерам може знадобитися використання спеціалізованих методів (наприклад, високоточна синхронізація часу, внутрішньомережева телеметрія) та обладнання (наприклад, пов'язане з ескізами) для точного вимірювання цінних даних у виробництві. Тим часом, різні методи, засновані на даних, також допомагають пом'якшити цю проблему. Знання мережевої предметної області можуть бути накладені на кілька вимірних даних для доповнення. Також пропонуються інші методи навчання з меншою кількістю даних, такі як навчання з невеликою кількістю пострілів і самоконтрольоване навчання. Трансферне навчання може допомогти перетворити модель, навчену на імітаційних даних, на практичну модель з невеликою кількістю реальних даних. Мережі, керовані даними, швидко розвиваються, і ми впевнені, що будуть сформульовані стандартні набори даних і критерії, які принесуть значний прогрес у цій захоплюючій галузі. Моделі, керовані даними, повинні відображати складні механізми мережі. Існують як глобальні, так і локальні, просторові та часові зв'язки мережевих об'єктів, і вони переплітаються між собою. Алгоритми управління перевантаженнями, політика черг, схеми маршрутизації тощо - все це має суттєвий вплив на продуктивність мережі. Алгоритми керування перевантаженням функціонують на рівні потоку, але глобально, де використовуються як кінцеві хости, так і внутрішньомережева інформація. Політики керування чергою планують передачу пакетів у локальному вузлі комутації, впливаючи на поведінку кінцевих вузлів. Схеми маршрутизації керують потоками на рівні шляху, що має значний вплив на розподіл трафіку. Крім того, ці механізми чутливі до часу (тобто зміни поточного стану впливають на майбутні стани). Як точно змодельовати глобальні та локальні, просторові та часові механізми, все ще залишається складним завданням. Доцільно використовувати модульну або багаторівневу концепцію, а також комбінованого аналітично-нейронного мислення, щоб полегшити складну проблему моделювання. Заплутані механізми можуть бути роз'єднані за допомогою модульних або багаторівневих частин, які можуть бути побудовані

та оновлені незалежно. Аналітичні методи також можуть бути введені для забезпечення більшої точності та ефективності, які часто впливають із знань про предметну область. Поточні моделі, керовані даними, розробляються окремо, тоді як фундаментальна модель може бути досить корисною. Попередньо навчені фундаментальні моделі можуть слугувати бекендом, і нам потрібно лише доопрацювати їх для конкретних завдань. Фундаментальна модель для оцінки ефективності подібна до трансформатора для обробки природної мови, який може забезпечити універсальні та базові можливості моделювання. Проблемами для побудови фундаментальних моделей є складність модельних конструкцій та універсальна абстракція мережевих механізмів. Потенційним методом є опис мережі як послідовності пакетів, де різні механізми можуть розглядатися як впливи на інтервал часу між пакетами. Методи, керовані даними, мають переваги над традиційними методами в ефективності, точності та гнучкості, що створює великий потенціал для оптимізації конфігурації мережі. Політики оптимізації на основі пошуку будуть просуватися зі швидшою швидкістю оцінки. Більш того, тимчасові NN-моделі можуть ефективно виконувати послідовні оцінки, що дозволяє оптимізувати управління на основі моделей. Завдяки моделі, яка точно оцінює продуктивність мереж, DTN може забезпечити високоточне динамічне середовище. Для прийняття мережевих рішень часто використовують методи навчання з підкріпленням (RL). Агент RL повинен сприймати стани з навколишнього середовища і виконувати певні дії, щоб максимізувати винагороду. Віртуальне динамічне середовище DTN є ідеальним майданчиком для алгоритмів RL для безпечного дослідження нових політик. Існують також нові технології (наприклад, штучний інтелект для IT-операцій (AIOps), самокеровані мережі), які звільняють інженерів від важкої ручної роботи, де DTN може слугувати середовищем для дослідження і верифікації [33].

Висновки

У розділі розглянуто основні вимоги до моделей продуктивності мереж, зокрема в контексті використання цифрових двійників для оцінки та оптимізації мереж 5G. Одним із головних аспектів є необхідність забезпечення точності, ефективності та гнучкості моделей, що використовуються для прогнозування продуктивності мережі. Цифрові двійники, як віртуальні копії фізичних мереж, дозволяють оператором тестувати нові конфігурації та політики в безпечному віртуальному середовищі, значно знижуючи ризики, що виникають при впровадженні змін на реальних мережах. Моделювання продуктивності з використанням цифрових двійників також дає змогу ефективно виконувати аналіз сценаріїв «що, якщо», що дозволяє операторам прогнозувати зміни в продуктивності мережі при зміні факторів впливу, таких як конфігурації пристроїв або топології. Оцінка продуктивності в таких віртуальних середовищах дозволяє швидко реагувати на аномалії та проводити оптимізацію без ризику для реальної мережі. Технології, засновані на даних, такі як нейронні мережі, набувають значення у створенні більш точних моделей продуктивності, здатних передбачати вплив різних параметрів на ефективність мережі. Важливими аспектами є не тільки точність і ефективність, але й здатність адаптувати моделі до швидко змінюваних умов мережі, що є необхідним для забезпечення гнучкості та масштабованості мереж 5G. Таким чином, цифрові двійники для мереж зв'язку, зокрема DTN, є важливим інструментом для оцінки, оптимізації та вдосконалення мережевих технологій, забезпечуючи надійність і ефективність при мінімізації ризиків, що виникають під час реальних змін у мережах. Враховуючи перспективи розвитку 6G та подальших технологій, цифрові двійники будуть відігравати ще більшу роль у моніторингу та управлінні телекомунікаційними системами.

РОЗДІЛ 5

ЦИФРОВИЙ ДВІЙНИК O-RAN

5.1. Емулятор бездротової мережі colosseum

Архітектура Colosseum показана в нижній частині Рис. 5.1. Її основними компонентами є [34]:

- 128 пар типових обчислювальних серверів і SDR, які називаються стандартними радіовузлами (SRN);
- система емуляції каналів;
- найсучасніша інфраструктура AI/ML.

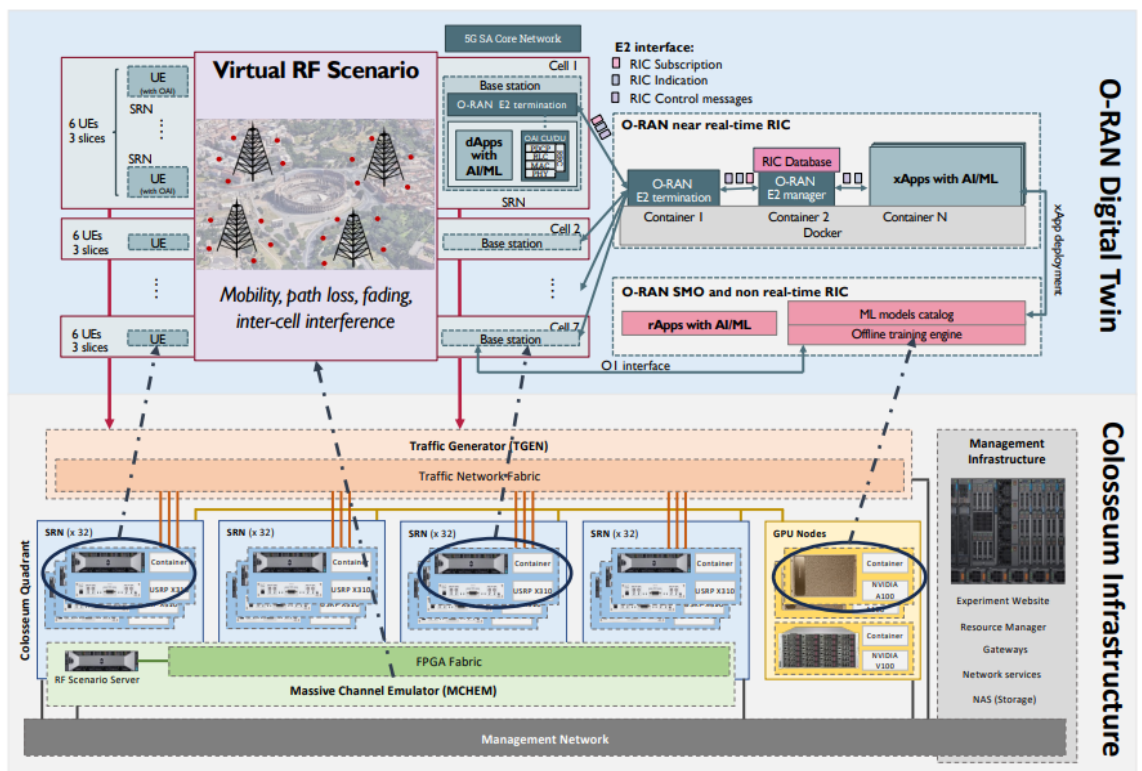


Рис. 5.1. Відкриті можливості твінінгу RAN у Colosseum [34]

Кожна SDR з'єднана з сервером Dell, підключеним до NI/Ettus USRP X310 SDR через виділений канал 10 Гбіт/с. Користувачі Colosseum можуть резервувати ці SRN і розгорнути власні або попередньо сконфігуровані стеки

протоколів через Linux Container (LXC). Образи контейнерів зберігаються у виділеній інфраструктурі зберігання, яка є частиною більшої інфраструктури управління, що включає такі сервіси, як веб- сайти, віртуальна приватна мережа (VPN) для зовнішнього підключення та мережеві функції. Емулятор каналу Colosseum, а саме Massive Channel Emulator (MCHEM), відповідає за відтворення цифрових зображень реальних сценаріїв розповсюдження радіосигналів, які відображають стан радіоканалу в задані моменти часу. Зокрема, кожна пара приймачів, тобто приймачі SDR, моделюється за допомогою лінії затримки на відгалуженнях (Tapped Delay Line, TDL) з чотирма ненульовими відгалуженнями, які представляють багатопроменеві компоненти імпульсної характеристики каналу (CIR) для кожної мілісекунди знятих сценаріїв. В основі MCHEM лежать чотири квадранти з 64 програмованими вентиляними матрицями (FPGA) і 128 SDR, які перетворюють сигнал між радіочастотним і базовим діапазонами. FPGA реалізують операції згортки цифрового сигналу базової смуги в реальному часі, моделюючи радіочастотні відгалуження бездротового каналу з урахуванням впливу реальних умов навколишнього середовища. Ці відгалуження, які мають максимальну затримку поширення 5,12 мкс, транслуються на FPGA зі спеціального сервера радіочастотних сценаріїв. Інфраструктура AI/ML Colosseum, показана жовтим кольором внизу на рисунку 6, складається з двох станцій NVIDIA DGX A100 з 8 графічними процесорами кожна, що забезпечує 10 петафлопс обчислювальної потужності, і одного суперсервера Supermicro Superserver 8049U- E1CR4T з 6 графічними процесорами NVIDIA V100 і 3 ТБ оперативної пам'яті. У порівнянні з системою віртуалізації SRN, GPU використовують контейнери Docker. Ресурси на цих вузлах управляються за допомогою фреймворків HashiCorp, таких як Nomad (для оркестрування робочих навантажень), Consul (для виявлення сервісів, конфігурації та підключення) і Traefik (для зворотного проксі та балансування навантаження), розгорнутих з резервуванням для відмовостійкості [34].

5.2. Наскрізне відкрите твінінгове дослідження RAN у колізеї

Можливості емуляції Colosseum можна використовувати для розгортання, вивчення та профілювання стеків бездротових протоколів у контрольованих і повторюваних середовищах. Як приклад, на Рис. 5.2. показано розгортання наскрізної, повністю програмованої стільникової системи Open RAN в Colosseum. У верхній лівій частині рисунка зображено RAN, яка складається з опорної мережі і базових станцій стільникового зв'язку - або монолітних, або розбитих на центральні блоки (CU) і розподілені блоки (DU) - розгорнутих на емуляції радіочастотного розгортання, що представляє інтерес, і обслуговує мобільне обладнання користувача (UE). Натомість у верхній правій частині рисунка показані системи управління та оркестрування послуг (SMO) і O-RAN RIC, які взаємодіють з базовими станціями RAN через відкриті інтерфейси і розширюють їхні можливості за допомогою розгорнутих в них додатків AI/ML, а саме xApps і rApps. У Colosseum ці O-RAN-орієнтовані рішення надаються користувачам через готові до використання образи LXC, які реалізують функціональні можливості, доступні через фреймворк OpenRAN Gum, який буде описано далі. Colosseum дозволяє експериментувати з різними стеками протоколів з відкритим вихідним кодом, такими як OAI для 5G RAN і srsRAN для 4G. Ці стеки протоколів, які надаються у вигляді готових до використання зображень, є тими самими, які можна використовувати для натурних експериментів з незначними варіаціями параметрів конфігурації, що обговорюються далі [34].

OpenAirInterface. OAI - це проект з відкритим вихідним кодом, який забезпечує еталонну реалізацію стеків протоколів стільникового зв'язку 3GPP, в тому числі 5G NR. Система NR на основі OAI складається з 5G RAN та опорної мережі, які працюють на обчислювальних платформах загального призначення на базі Linux і можуть керувати SDR, наприклад, NI/Ettus USRP, розгорнутими в Colosseum. Додатки OAI RAN підтримують інстанціювання баз вузлів наступного покоління (gNB) та UE. gNB може працювати в монолітному або

дезагрегованому режимі, який використовує опцію розділення CU/DU. Обидва режими доступні в Colosseum. У дезагрегованій базовій станції CU і DU обмінюються даними з керуючим і користувацьким рівнями через інтерфейси F1-C і F1-U, відповідно, за допомогою протоколу F1AP. CU може бути дезагрегований на площину управління центрального блоку (CU-CP) і площину користувача центрального блоку (CU-UP). Керуюча інформація між вузлами площини керування та площини користувача передається за протоколом E1AP. Користувачі Colosseum можуть розгорнути стільникові мережі 5G NR на Colosseum за допомогою готових зображень контейнерів. Однак, у порівнянні з повітряним середовищем, додаткові компоненти в Colosseum передають і приймають ланцюжки (наприклад, ті, що входять до складу MСHEM) вимагають коригування випередження часу під час встановлення початкового з'єднання між gNB та UE. Це можна зробити за допомогою вхідного аргументу та під час запуску програми базової станції [34].

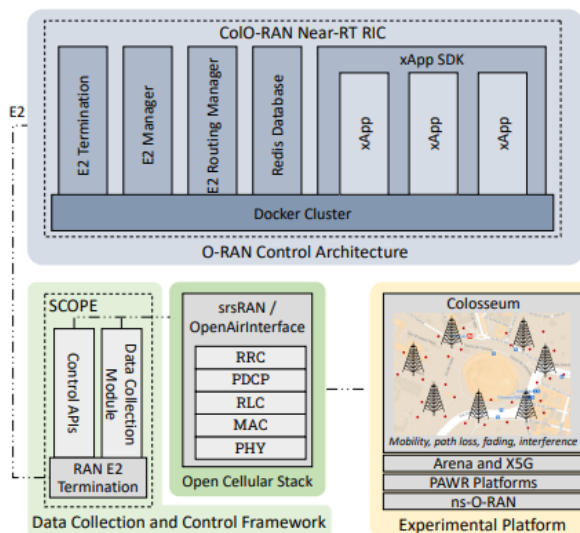


Рис. 5.2. Інтелектуальні контури управління з Колізеєм, на основі фреймворку OpenRAN GYM [34]

Для застарілих мереж 4G Colosseum надає зображення контейнерів для версії srsRAN для 4G. Подібно до OAI, це програмне забезпечення з відкритим вихідним кодом дозволяє користувачам проводити експерименти з програмно

керуваними розвиненими базами вузлів (eNB), абонентськими пристроями (UE) і опорною мережею. На відміну від мереж 5G, які використовують дуплексування з часовим розділенням каналів (TDD), системи 4G в основному базуються на дуплексуванні з частотним розділенням каналів (FDD), з низхідною і висхідною лініями зв'язку в двох окремих частотних діапазонах. У Colosseum їх потрібно розмістити в двох частинах корисної смуги частот 80 МГц за допомогою спеціальної конфігурації srsLTE eNB, яка вже попередньо налаштована в зображеннях LXC, доступних для користувачів Colosseum [34].

OpenRAN Gum - це фреймворк з відкритим вихідним кодом для експериментів з O-RAN, який дозволяє користувачам інстанціювати софтуалізовані RAN на основі стеків протоколів OAI та srsRAN і керувати ними через xApps, розгорнуті на RIC O-RAN Near-Real-time (RT) [35]. Її компоненти, показані на Рис. 5.2, включають [34]:

- SCOPE, обгортка для програмного забезпечення RAN, яка спрощує робочий процес експерименту для користувачів і дозволяє збирати дані в масштабі.
- ColO-RAN, спрощена версія Near-RT RIC спільноти O-RAN Software Community, яка надає комплект розробки програмного забезпечення (SDK) для розробки xApp, а також конвеєри для збору даних і навчання AI/ML моделей для RAN висновків і управління, які можна вбудовувати в xApps.

Загалом, OpenRAN Gum дозволяє користувачам створювати повністю сумісні Near- RT RIC за допомогою контейнерів Docker, а також xApps, які можуть взаємодіяти з RAN (наприклад, отримувати ключові показники ефективності (KPI) і застосовувати політики управління) через інтерфейс E2. Приклади розроблених xApps включають Агенти глибокого навчання з підкріпленням (Deep Reinforcement Learning, DRL) для оптимізації продуктивності RAN в реальному часі шляхом керування розкладом та конфігурацією нарізки. Підручники про те, як почати використовувати OpenRAN Gum на Colosseum, знаходяться у відкритому доступі на веб-сайті OpenRAN Gum. Нарешті, цей фреймворк також використовувався для

демонстрації перенесення експериментів з Colosseum на зовнішні платформи, наприклад, на тестовий стенд Arena і платформи з програми PAWR [34].

5.3. Радіочастотний твінінг

Емуляція каналу та трафіку, а також стеки протоколів стільникового зв'язку, які можна розгорнути на Colosseum, надають ключ компонентів для створення високоточних систем у реальному часі, цифрових реплік систем Open RAN з радіочастотним обладнанням у контурі. Далі описано набір інструментів, які розробили і створили для дублювання реальних радіочастотних сценаріїв у Колізеї. Через обчислювальну та часову складність запуску операцій трасування променів для отримання відгалуджень бездротових каналів для емуляції, радіочастотні сценарії наразі генеруються в автономному режимі, а потім відтворюються в реальному часі сервером сценаріїв. Для цього використовується процедура радіочастотного твінінгу фреймворку CaST. Цей інструментарій, який є загальнодоступним для дослідницької спільноти, дозволяє користувачам характеризувати реальні радіочастотні середовища і перетворювати їх на цифрові двійники для використання в емуляторах каналів, таких як Colosseum. Робочий процес і процедури CaST, зображені на Рис. 5.3, складаються з двох основних компонентів [34]:

- спрощена структура для створення реалістичних сценаріїв мобільного бездротового зв'язку (верхня частина рисунка);
- емулятора каналів на основі SDR для визначення характеристик емульованих радіочастотних каналів (нижня частина).

Поєднання цих двох компонентів дозволяє створювати цифрові подвійні радіочастотні сценарії та перевіряти їх, гарантуючи, що цифрові репліки бездротових середовищ відповідають їхнім реальним аналогам [34].

Основні етапи робочого процесу створення сценарію CaST є наступними [34]:

- визначити бездротове середовище, яке потрібно захопити, тобто

фізичне місце для двійників, яке може відрізнятися за розміром і типом, наприклад, внутрішнє, зовнішнє, міське, сільське;

- отримати 3D-модель середовища, використовуючи онлайн-бази даних, такі як Open Street Map (OSM), або програмне забезпечення для моделювання, таке як SketchUp;
- завантажити модель в програму для трасування променів, наприклад, MATLAB
- встановити розташування і траєкторії вузлів;
- зробити вибірку каналу за допомогою трасувальника променів, зафіксувавши такі ефекти, як рухливість вузлів.

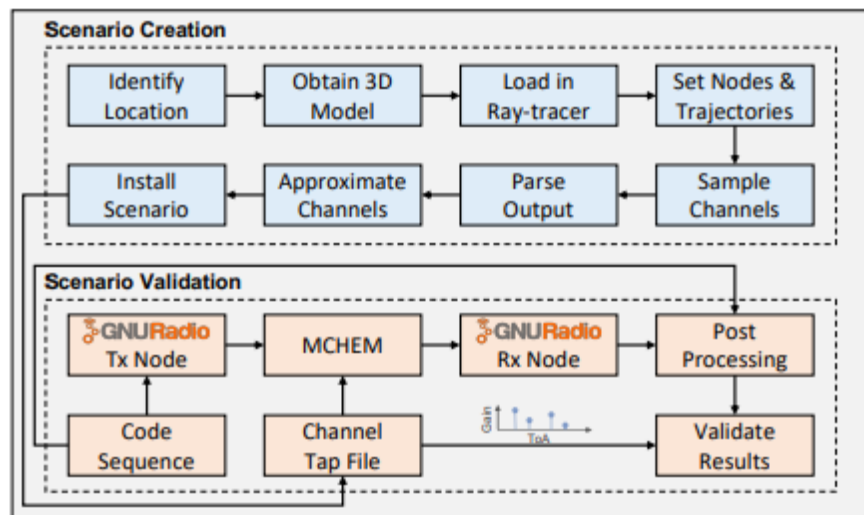


Рис. 5.3. Робочі процеси створення сценарію CaST (вгорі) та перевірки сценарію (внизу) [34]

Розглянемо робочі процеси перевірки сценарію [34]:

- проаналізувати вихідні дані трасувальника променів, щоб виділити канали для кожної пари вузлів для кожної мілісекунди емуляції сценарію;
- апроксимувати отриману інформацію про канал у формат, придатний для емулятора каналу Colosseum, тобто 512 відводів каналу з максимум 4 ненульовими значеннями, розташованими з інтервалом 10 нс і максимальним розкидом затримок 5,12 мкс;

- встановити сценарій у Colosseum, перетворивши його у формат, придатний для FPGA. Інструментарій створення сценаріїв є модульним і дозволяє користувачам вносити свої дані на будь-якому з вищезгаданих етапів.

Після того, як сценарій встановлено в Colosseum, він проходить валідацію за допомогою другої частини фреймворку CaST, щоб гарантувати, що він точно відповідає очікуваній поведінці в реальному світі. Це передбачає перевірку каналу, реалізовану за допомогою програмного забезпечення GNU Radio, і складається з наступних кроків [34]:

- передача відомої кодової послідовності, що використовується як опорний сигнал, через передавач GNU Radio;
- прийом через приймач GNU Radio сигналу, обробленого MСHEM, поверх якого були накладені характеристики каналу;
- постобробка отриманих даних за допомогою відомої кодової послідовності з додаванням каналних ефектів для вилучення CIR і шляхових втрат (PL) форми сигналу;
- перевірка результатів з оригінальними змодельованими відгалуженнями каналів. Ехолот CaST здатен досягти точності 20 нс для затримок CIR і 0,5 дБ для коефіцієнтів підсилення відгалуження.

Нарешті, оскільки SDR Colosseum оснащені двома ланцюгами прийому-передачі, Colosseum і MСHEM підтримують операції множинного вводу-виводу (MIMO). Таким чином, CaST може бути використаний для генерації спрощених каналів MIMO з використанням MATLAB або каналних модулів NR 3GPP від Sionna [34].

Дорожня карта на шляху до твінінгу в режимі реального часу Colosseum дозволяє користувачам відтворювати високоточні радіочастотні (RF) сценарії, які були попередньо змодельовані в автономному режимі і встановлені в системі. Хоча цей процес надзвичайно гнучкий і дозволяє моделювати радіочастотні сценарії за допомогою різних інструментів (наприклад, трасувальників променів, математичних моделей), він не підходить для тих випадків, коли канал між вузлами повинен динамічно змінюватися в реальному часі. Це, наприклад,

випадок, коли цифровий двійник повинен адаптуватися до стану фізичного середовища. Щоб подолати це обмеження, оновлюємо інфраструктуру Colosseum, щоб уможливити RF-двійник у реальному часі. Поетапний підхід поєднує оновлення сервера радіочастотних сценаріїв з більш гнучкою реалізацією MСHEM, що дозволяє транслювати відводи каналів, згенеровані в реальному часі, з рішеннями трасування променів у реальному часі. Далі розглянемо можливі підходи до цього, а також деякі виклики [34].

Використовуючи цей підхід, детальна модель RAN може бути відтворена в цифровому двійнику за допомогою методів моделювання сценаріїв і каналів, наприклад, 3D-моделей і трасування променів. Ця модель включає статичні та динамічні компоненти. Перша попередньо налаштована в системі і включає, наприклад, розташування базових станцій і будівель. Друга передається в реальному часі з реального радіочастотного діапазону, щоб відобразити зміни, такі як розташування абонентських пристроїв, потужність передачі. Нарешті, після реконструкції цифрової складової можна використовувати такі методи, як трасування променів або стохастичні моделі поширення, щоб охарактеризувати середовище поширення радіохвиль, а потім оновити MСHEM у реальному часі. Попередній підхід може бути розширений методами зондування каналів (наприклад, компонент зондування CaST), які безперервно оцінюють канали між пристроями або тестовими місцями в реальному середовищі. Ця оцінка каналу може безперервно надсилатися з RAN до цифрового двійника, а потім перетворюватися на вхідні дані, які можуть бути оброблені MСHEM. Крім того, можуть бути використані методи для отримання оцінки відгалужень каналу з еталонного сигналу інформації про стан каналу (CSI- RS) або еталонного сигналу зондування висхідної лінії зв'язку (SRS) фізичного рівня 5G NR [34].

5.4. Автоматизація для цифрового двійника Colosseum

Окрім радіочастотного та прикладного двійникування, другий компонент платформи цифрових двійників Colosseum дозволяє автоматизовано

відтворювати наскрізні повностекові відкриті системи RAN на реальних випробувальних стендах та в Colosseum. Colosseum реалізує конвеєри CI, CD і безперервного тестування (СТ) для автоматизації двійників і тестування стеків протоколів. Конвеєри CI і CD покладаються на фреймворк автоматизації Ansible від Red Hat, тоді як процес СТ покладається на пакетні завдання Colosseum. Схему цього робочого процесу показано на Рис. 5.4 [34].

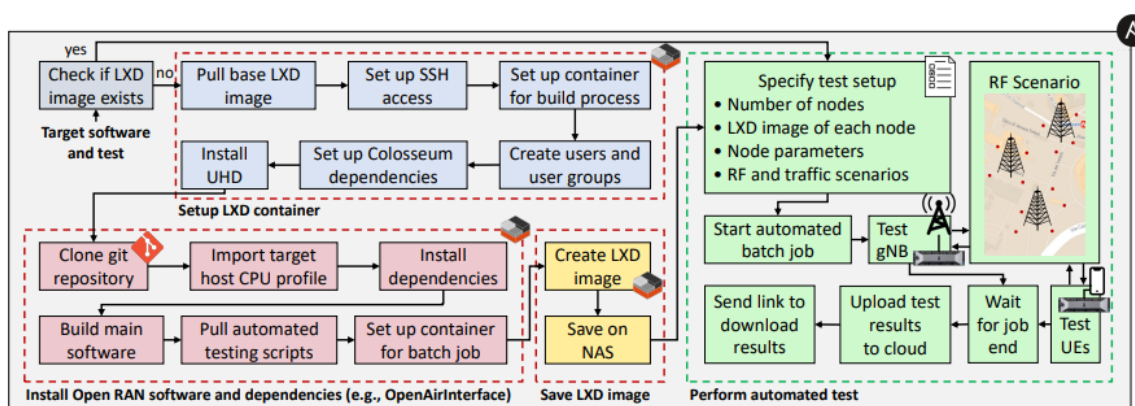


Рис. 5.4. Конвеєр Ansible для виконання автоматизованого тесту на Colosseum [34]

CI/CD складається з двох логічних кроків [34]:

- створення нового образу (позначено пунктирними червоними лініями)
- виконання автоматизованого тесту на створеному образі (пунктирна зелена лінія).

Після запуску конвеєра з тестом для виконання виконується перевірка на наявність LXC-образу програмного забезпечення, яке тестується. У разі позитивного результату цієї перевірки виконується автоматизоване тестування; в іншому випадку створюється новий LXC- образ. На високому рівні кроки зі створення нового LXC-образу включають три основні завдання [34]:

- налаштування нового LXC- контейнера,
- встановлення програмного забезпечення Open RAN та залежностей.
- збереження LXC-образу.

Налаштування нового контейнера LXC передбачає низку кроків, які починаються з витягування базового образу LXC, наприклад, базового образу Ubuntu 22.04, зі сховища із загальнодоступними образами LXC Colosseum. Процес налаштування продовжується налаштуванням доступу по SSH, щоб дозволити користувачам увійти до контейнера після його розгортання у Colosseum. Потім контейнер готується до процесу збирання шляхом встановлення необхідних компіляторів та інструментів, створення користувачів та груп, а також встановлення різних залежностей та інструментів, необхідних для Colosseum. Додатково встановлюється програмне забезпечення UHD, необхідне для зв'язку з SDR. Потім, задача встановлення програмного забезпечення Open RAN та залежностей клонує репозиторій програмного забезпечення для тестування (наприклад, програмного забезпечення OAI); імпортує профіль цільового вузла для створення бінарного файлу програмного забезпечення (наприклад, у випадку, якщо процес компіляції оптимізовано для певних наборів інструкцій процесора); встановлює необхідні залежності, наприклад, визначення ASN.1; створює основні бінарні файли програмного забезпечення; витягує деякі скрипти для автоматизації процесу тестування; і налаштовує контейнер для виконання пакетних завдань на Colosseum. Нарешті, завдання збереження результуючого LXC-образу експортує створений контейнер у вигляді LXC-образу і зберігає його у сховищі Colosseum. Після створення LXC-образу програмного забезпечення, яке тестується, виконується етап автоматизованого тестування за допомогою пакетних завдань Colosseum. Спочатку вказується налаштування тесту, в якому детально описуються параметри пакетного завдання. Ці параметри включають інформацію про кількість вузлів, що беруть участь у тестуванні, образ LXC для завантаження на кожен вузол (наприклад, різні вузли можуть діяти як gNB, UE або основна мережа, що вимагає різних образів), а також про конфігурацію RAN (наприклад, частоту і діапазон, що використовуються), а також сценарії радіочастот і трафіку Colosseum для запуску тесту. Потім запускається нове пакетне завдання в Colosseum. Це передбачає, наприклад, розгортання gNB, що надає послуги

стільниковим користувачам - обидва реалізовані через стек протоколів OAI - за допомогою гетерогенних радіочастотних сценаріїв Colosseum. Після завершення пакетного завдання результати завантажуються в хмару, а посилення для їх завантаження повертається конвеєром CI/CD/CT, щоб користувачі могли легко їх отримати [34].

5.5. Поєднання реального світу та цифрового двійника

Цифровий двійник Colosseum може бути інтегрований із зовнішніми платформами та інструментами або шляхом прямого підключення або шляхом перенесення експериментів між платформами. Крім того, можна розгортати і розширювати емульовані експерименти за допомогою мережевих симуляторів дискретних подій, таких як ns-3 і його розширення O-RAN [34].

Випробувальний стенд Arena - це повітряний критий програмований випробувальний стенд з 24 SDR, підключеними до сітки з 64 антени, що дозволяє експериментувати в частотному спектрі до 7,2 ГГц. Розміщений разом з X5G-5G та O-RAN-сумісним тестовим стендом на базі NVIDIA Aerial, який реалізує фізичний рівень NR на GPU, OAI та комерційних радіопристроях (RU) O-RAN. Використали спільний канал метро зі швидкістю 10 Гбіт/с, щоб створити локальну VPN між Colosseum та Arena і практично об'єднати ці дві мережі за допомогою спеціальних брандмауерів. Ця інтеграція дозволить користувачам Colosseum виконувати експерименти на тестовому стенді Arena через уніфікований веб-портал і LXC, що дозволить швидко переходити між реальними і емуляційними середовищами [34].

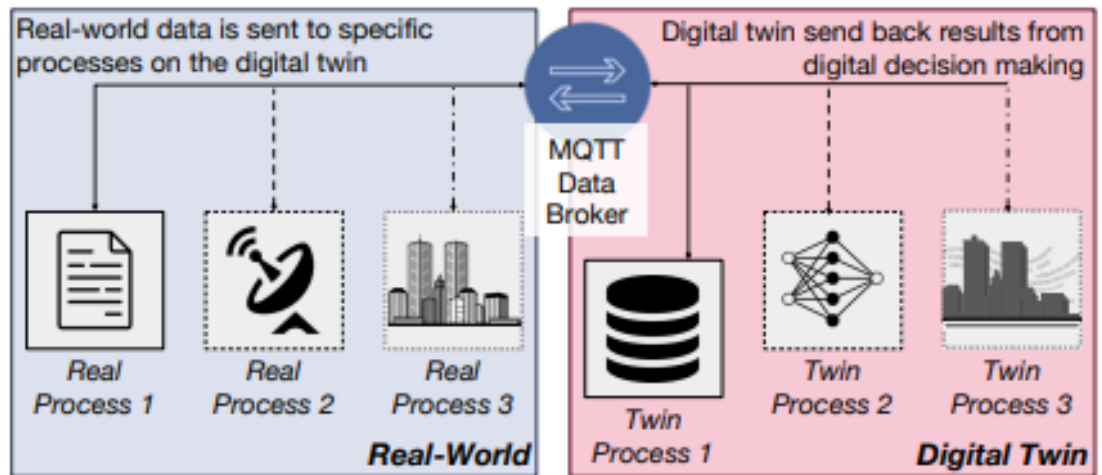


Рис. 5.5. Огляд того, як комунікаційний брокер MQTT полегшує двонаправлений розподіл даних для конкретних процесів, дозволяючи масштабовану реалізацію цифрових двійників [34]

Інтеграція Colosseum-Arena також була використана для створення прототипу і тестування програмного забезпечення для твінінгу в реальному часі реального сценарію на Arena та його цифрового аналога на Colosseum. Використали протокол MQTT для реалізації зв'язку в реальному часі між двома кінцевими точками. Завдяки механізму публікації/підписки цей протокол забезпечує безперебійний обмін даними між кількома пристроями, гарантуючи мінімальне споживання смуги пропускання, низьке енергоспоживання та стабільне з'єднання, що є ключовими для нашого твінінгу в реальному часі. Комунікаційний брокер полегшує зв'язок між пристроями та додатками, керуючи обміном повідомленнями за принципом "точка-точка". Це дозволяє одному суб'єкту контролювати потік даних в обох напрямках, що робить управління системою впорядкованим, централізованим і таким, що реагує на зміни вимог до даних. Ми розмістили брокера в середовищі цифрового двійника, як показано на Рис. 5.5, що гарантує відсутність ресурсних обмежень для реальної системи, яка виступає в ролі розподільника даних [34].

Таблиця 5.1.

Час надходження пакетів між цифровим двійником та реальним аналогом у мілісекундах (мс). (Усереднений час затримки для 100 вибірок) [34]

Розмір пакету	Затримка між реальним і твіном	Затримка між подвійною та реальною затримкою
1 Байт	15.318 мс	15.119 мс
100 Байт	15.547 мс	15.688 мс
1 кілобайт	15.742 мс	15.713 мс
10 кілобайт	22.261 мс	21.979 мс
100 кілобайт	31.101 мс	31.097 мс
1 мегабайт	47.967 мс	48.003 мс

У таблиці 1 показано затримку зв'язку між двома системами, яка показує, як розмір пакету впливає на затримку. Як і очікувалося, менші пакети дають меншу затримку зв'язку, наприклад, у випадку зчитування даних з датчиків або простого обміну командами. Пакети розміром до 100 байт показують схожі значення затримки. Це може бути, наприклад, у випадку передачі коротких повідомлень. Нарешті, більші розміри пакетів можуть підтримувати передачу більш значних даних, наприклад, невеликих наборів даних про спектр, зберігаючи при цьому прийнятні значення затримки. Можливості переміщення експериментів між Colosseum та зовнішніми тестовими стендами також були продемонстровані в роботі [35], де експериментальна кампанія спочатку була виконана на Colosseum, а потім перенесена на тестовий стенд Arena та на платформи програми PAWR. Зокрема, інструментарій OpenRAN Gum був використаний на Colosseum для проведення кампанії зі збору даних, навчання xApps ML і розгортання їх на Near-RT RIC для виконання управління на основі даних м'яко воєнізованою RAN, реалізованою за допомогою фреймворку SCOPE і інстанційованою на імітованому міському радіочастотному сценарії. Потім ті ж самі xApps і LXC-контейнери, що використовувалися на Colosseum, були перенесені на платформи PAWR і використані для управління на основі даних

РАН як у зовнішньому, так і у внутрішньому середовищі, що дозволило досягти порівнянних тенденцій в реальному і імітованому середовищах. Це продемонструвало можливість перенесення експериментів між Colosseum і зовнішнім випробувальним стендом [34].

5.6 Приклади використання Colosseum як Open RAN Digital Twin

Colosseum широко використовується для вивчення відкритих проблем алгоритмічного, архітектурного та системного дизайну систем Open RAN, використовуючи реалістичність емуляції сценаріїв радіочастотного діапазону та можливість запуску сучасних програмно-визначених стеків протоколів, що повторюють реальні сценарії. Розподіл радіочастотного діапазону дозволяє операторам динамічно розділяти доступну смугу частот для розміщення абонентських пристроїв і трафіку з різними вимогами до якості обслуговування (QoS), продуктивності та угод про рівень обслуговування (SLA). Поширеним прикладом UE і типу послуг в 5G є розширений мобільний широкосмуговий зв'язок (eMBB), зв'язок машинного типу (MTC) і наднадійний зв'язок з низькою затримкою (URLLC). Ще однією конфігурацією управління, яку ми розглянули, є вибір різних політик планування для кожного з вищезгаданих сегментів RAN. Для досягнення гнучкості в управлінні RAN агенти AI/ML можуть бути розроблені і вбудовані в xApps, а потім розгорнуті на Near-RT RIC і використовуватися для адаптації RAN до різних умов мережі і попиту. Зокрема, DRL можна використовувати при розробці таких рішень для управління відкритою RAN, оскільки вони можуть бути навчені заздалегідь і не потребують попередніх знань про динаміку мережі під час виконання. Оскільки ці рішення вимагають навчання на великих наборах даних перед розгортанням на виробничій RAN (щоб уникнути порушення її роботи), масштаб і контрольоване середовище цифрового двійника пропонують сприятливий ґрунт для їхньої розробки. Через фреймворк OpenRAN Gym, Colosseum надає інструменти та інфраструктуру для збору даних, проектування, навчання, тестування та оцінки

AI/ML xApps до їхнього розгортання в ефірі [34].

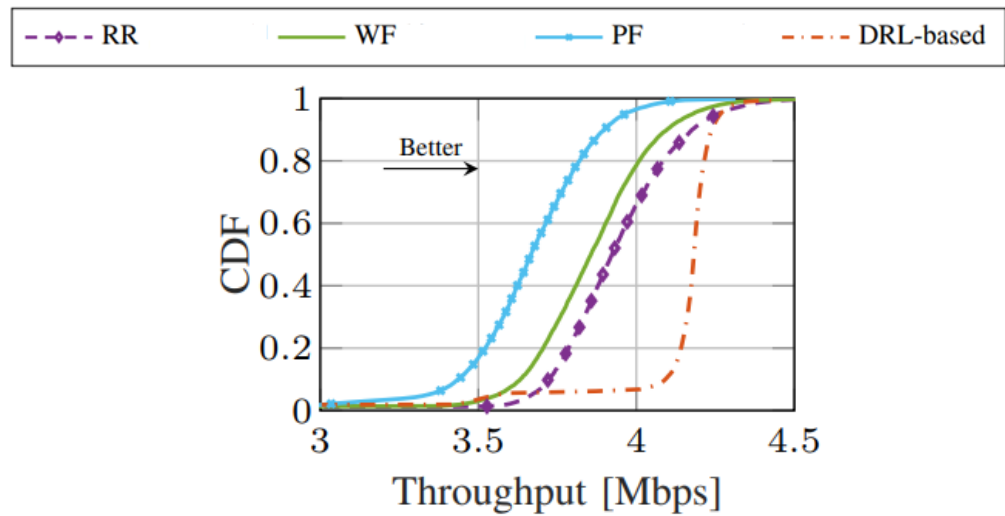


Рис. 5.6. Пропускна здатність eMBB [34]

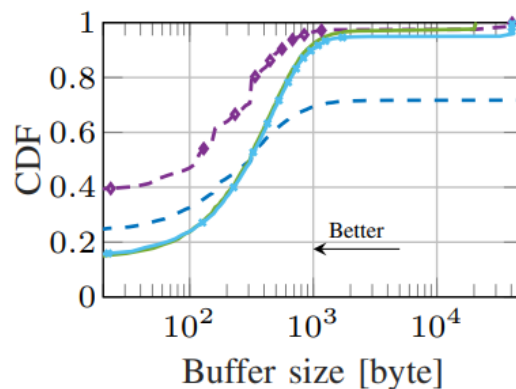


Рис. 5.7. Буфер URLLC [34]

На рисунка 5.6 та 5.7 показана продуктивність агента DRL, що керує політиками нарізки та планування. Ми порівнюємо xApp, керований даними, та базовий варіант без адаптації на основі DRL. В останньому випадку продуктивність базується на конфігурації нарізки, обраній з найбільшою ймовірністю найефективнішим агентом DRL, та трьох політиках планувальника. Як приклад, в [36] розробили, навчили та протестували xApps на основі DRL для керування розрізом та плануванням м'якої RAN на цифровому двійнику "Колізею". Ефективність одного з цих xApps в управлінні плануванням (на вибір між Round-robin (RR), Waterfilling (WF) та Proportionally Fair (PF)) та розбиттям (з точки зору виділених блоків фізичних ресурсів (PRB)) показано на рисунках

5.6 та 5.7. Зокрема, на Рисунку 5.6 показано пропускну здатність зрізу eMBB, ана Рисунку 6.6 - заповнення буфера зрізу URLLC, з нашим адаптивним xApp на основі DRL та статичними базовими лініями. Помітно, що адаптація конфігурації RAN в режимі, близькому до реального часу, дозволяє xApp на основі DRL перевершити базові показники для зрізу eMBB, в той же час відповідаючи найкращій статичній конфігурації для зрізу URLLC. Підготовлені таким чином xApps потім були перенесені на випробувальний стенд Arena, де вони адаптувалися до повітряних налаштувань за допомогою лише кількох раундів онлайн-навчання, а також на платформи PAWR, де вони продемонстрували тенденції, подібні до тих, що були досягнуті в Colosseum. Нарешті, розширивши цей аналіз, провівши бенчмаркінг 12 xApps, навчених з різними функціями винагороди, просторами дій та ієрархічним контролем. Ці xApps продемонстрували, як різні варіанти дизайну можуть забезпечити вищу продуктивність, тоді як інші можуть призвести до конкурентної поведінки між різними класами трафіку зі схожими цілями. Colosseum також використовувався для оцінки рішень щодо розподілу спектру для відкритого радіочастотного діапазону. Вони передбачають підвищення обізнаності про власників спектру за межами RAN, а також додаткові контрольні дії (наприклад, звільнення частини спектру або обнулення блоків ресурсів). Цифрова природа систем дозволяє безпечно оцінювати схеми спільного використання та співіснування без ризику завдати шкоди користувачам до того, як рішення будуть впроваджені в реальному світі [34].

На Рис. 5.8 показано приклад робочого процесу, який забезпечує Colosseum для досліджень з розподілу спектру. Він включає такі етапи, як твінінг хвильових форм і сценаріїв, тестування і валідація розроблених рішень для спільного використання [34].

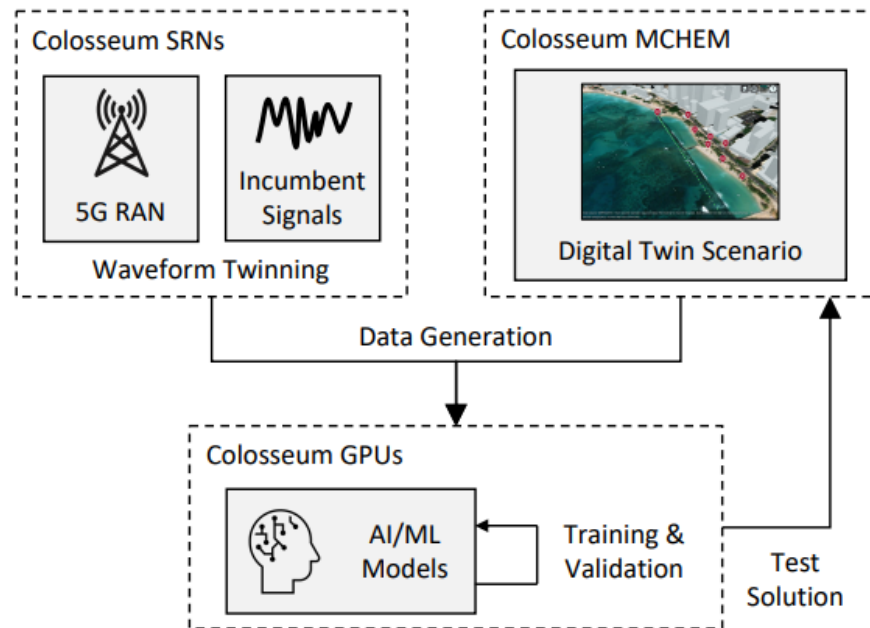


Рис. 5.8. Робочий процес Colosseum для розробки та тестування рішень з розподілу спектру на основі даних [34]

Цей робочий процес використовується для дослідження рішень на основі штучного інтелекту з метою покращення продуктивності RAN порівняно зі статичними схемами або підходами, заснованими на моделях. Зокрема, Colosseum використано для збору набору даних про користувачів Wi-Fi і стільникового зв'язку, які співіснують в одній смузі частот, і для навчання детектора для виявлення моделей використання спектру на основі рекурентних нейронних мереж. Аналогічно, продемонстровано використання Colosseum для проведення безпечних експериментів з радіолокатором, що працює в діапазоні CBRS, і для звільнення стільникового зв'язку від роботи в цьому діапазоні, якщо він там присутній. Хоча штучний інтелек (ШІ) буде наріжним каменем технології Відкритої RAN, одне питання, яке все ще залишається невирішеним, полягає в тому, як гарантувати, що рішення ШІ, вбудовані в xApps і rApps, є достатньо надійними, щоб приймати автономні рішення без необхідності участі людини. Дійсно, xApps та rApps можуть приймати рішення швидше та ефективніше, ніж людина, завдяки логіці, що ґрунтується на даних, яку вони вбудовують. Однак ці рішення можуть призвести до перебоїв у роботі та

неоптимальної продуктивності, якщо алгоритми штучного інтелекту не здатні адаптуватися до мінливих умов і обчислювати мережеві політики, стійкі до шуму та високостохастичних середовищ. У цьому контексті пояснювальний ШІ (Explainable AI, ХАІ) став технологією, яка пропонує платформу для розуміння процесу прийняття рішень на основі ШІ. Зокрема, ХАІ дає змогу дослідити взаємозв'язки між виходами алгоритмів ШІ (наприклад, керуючими діями, передбаченнями, прогнозами) та входами. Маючи можливість вивчати взаємодію між виходами та входами рішень ШІ, ХАІ надає необхідні механізми для аналізу алгоритмів ШІ, вбудованих в xApps та rApps, та перевірки правильності прийняття ними рішень. ХАІ може бути корисним багатьма різними способами, як показано нижче [34]:

- Паритет продуктивності: найважливішим аспектом для впровадження ШІ є паритет продуктивності зі старими системами RAN. Зокрема, необхідно продемонструвати, що xApps і rApps можуть не тільки досягти паритету продуктивності, але й вийти за його межі. ХАІ пропонує інструменти для визначення умови, в яких алгоритми ШІ не працюють належним чином, і порівнюють результати роботи ШІ з базовими показниками та експертними знаннями [34].

- Стійкість до шуму та атак: завдяки здатності будувати зрозумілі взаємозв'язки між входами та виходами, ХАІ дає змогу визначити стійкість рішень ШІ до шуму, стохастичних середовищ та атак. Наприклад, ХАІ можна використовувати для виявлення вразливих частин алгоритмів ШІ (наприклад, певного шару, функції активації), які піддаються коливанням вхідних даних, що можуть бути спричинені шумом у навколишньому середовищі або використані зловмисниками, які знають про цю вразливість [34].

- Покращення дизайну: після виявлення пасток і слабких місць алгоритмів ШІ, як зазначалося вище, ці знання можуть бути використані для виявлення частин архітектури ШІ, які не працюють належним чином (або які вразливі до шуму чи атак), і їх доопрацювання, щоб зробити їх більш надійними та надійними [34].

Colosseum можна використовувати для проектування, розробки та тестування методів ХАІ для додатків О-RAN. Зокрема, Colosseum унікальний тим, що може полегшити розробку ХАІ [34]:

- підтримує тестування xApps і rApps у різних топологіях, мобільності, каналах і умовах трафіку для аналізу їхньої поведінки в різних сценаріях;
- пропонує готове до використання середовище розробки для інтеграції алгоритмів ХАІ в xApps і rApps;
- забезпечує реалістичний цифровий двійник Open RAN, який дозволяє безпечний аналіз ХАІ xApps і rApps до їхнього фактичного розгортання у виробничій мережі, щоб уникнути потенційного зниження продуктивності та/або перебоїв у роботі.

Наприклад, фреймворк EXPLORA є інструментом ХАІ для xApps і rApps на основі DRL, розроблених на Colosseum через OpenRAN Gym. EXPLORA генерує високорівневі інтерпретовані пояснення, які описують, чому певні дії були обрані агентом DRL, а також операційні умови (наприклад, умови дорожнього руху), які призвели до певного вибору. Крім того, ХАІ-інсайти, згенеровані EXPLORA, можуть бути використані для виявлення неоптимальних пар "дія-стан" і запобігання застосуванню таких дій шляхом заміни їх іншими діями, які забезпечують вищу ефективність. ІАВ - це гнучка технологія бездротового транзитного зв'язку для стільникових систем, яка довела свою ефективність у зниженні витрат на розгортання надщільних RAN. Вона була представлена в специфікаціях стандарту 5G NR в рамках версії 16 3GPP і поставляється з новим типом об'єкта RAN, який називається ІАВ-вузол. Вузол ІАВ включає в себе мобільне закінчення (МТ) для бездротового транзитного зв'язку з висхідними вузлами, а також нижню частину стека протоколів стільникового зв'язку (тобто, рівні DU і RU). Вузли ІАВ не з'єднані дротовими каналами з опорною мережею, натомість вони з'єднуються один з одним, створюючи багатохоповий граф, який використовує ту ж саму форму сигналу, стек протоколів і, можливо, той же самий спектр. Цей бездротовий граф закінчується на базових станціях з дротовим підключенням до опорної мережі,

які називаються донорами IAB. Colosseum використовувався як цифровий майданчик для прототипування і тестування внутрішньосмугових мереж IAB на основі OAI. Хоча рішення не повністю відповідає стандартам, воно дозволяє розгорнути вузли та донори IAB з мінімальними змінами в OAI. Це дає змогу вивчати відповідні проблеми та випадки використання в домені IAB, такі як розподіл трафіку доступу та транзиту, затримки в багатохопових IAB, а також оптимальна маршрутизація та планування. Оскільки бездротова природа транзитного каналу в IAB збільшує простір і параметри для управління мережею, архітектура Open RAN і RIC можуть бути використані для організації та оптимізації розгортання IAB на основі підходу, заснованого на даних. Безпека є ще однією ключовою проблемою, а також сферою досліджень і розробок в рамках відкритих систем RAN. Віртуалізація, програмізація і відкриті інтерфейси розширюють можливості атак на стільникові системи, вимагаючи додаткового тестування і вивчення процедур управління мережею, протоколів і реалізацій. У той же час, відкриті інтерфейси дозволяють підвищити видимість динаміки і телеметрії цих систем, відкриваючи двері для надійних схем виявлення аномалій і вторгнень на основі даних. Нарешті, з поширенням технології стільникового зв'язку 5G зростає ризик більш складних атак типу "глушіння", що загрожує доступності, надійності і безпеці мережі. Це є підставою для продовження досліджень у сфері глушіння, щоб захистити майбутні комунікації від зловмисних перешкод. Як цифровий двійник, заснований на реалістичній радіочастотній емуляції в режимі реального часу, Colosseum можна використовувати як пісочницю для оцінки загроз, тестування контрзаходів і перевірки захисту від перешкод без ризику завдати шкоди реальним повітряним системам. Colosseum для оцінки безпеки відкритих RAN. Можливості цифрового двійника Colosseum були використані для оцінки безпеки інтерфейсів та реалізацій O-RAN. Ця оцінка була виконана для наскрізних стеків протоколів 5G і Open RAN, розгорнутих на інфраструктурі емуляції Colosseum, з акцентом на інтерфейсі E2 і на чисельній та експериментальній оцінці вартості, пов'язаної з підтримкою сучасних методів

шифрування для потоку E2 SCTP [34].

Окрім захисту інтерфейсів, існує великий інтерес до оцінки того, як відкритість архітектури O-RAN дозволяє розробляти та впроваджувати більш досконалі механізми виявлення аномалій та вторгнень. У роботах [37, 38] автори досліджують механізми класифікації трафіку для систем Open RAN, реалізованих на Near-RT RIC, для аналізу конкретних KPI, пов'язаних з наскрізними патернами додатків. У роботі [39] зроблено крок далі і поєднано ці дослідження з глибоким навчанням на зразках RAN I/Q (з а допомогою dApp) для виявлення аномалій, пов'язаних, наприклад, з клонованими модулями безпечної ідентифікації користувачів. Colosseum для оцінки атак на перешкоди. Розуміння еволюції методів глушіння має вирішальне значення для розробки ефективних контрзаходів і забезпечення безперебійного зв'язку для критично важливих сервісів, що покладаються на мережі 5G. Однак, часто закони та нормативні акти забороняють створювати навмисні перешкоди це може суттєво перешкоджати масштабним дослідженням завад, які проводяться в авторизованих системах радіозв'язку. Це відкриває двері для цифрових двійників, які пропонують покращені дослідження завад завдяки точній, реальній емуляції. Масштабні експерименти із завадами дають унікальний погляд на те, як атаки завад впливають на спектр. Ці сигнали мають конкретні цілі атаки, але існують також випадкові перешкоди, які можуть виникати в бездротовому зв'язку. Емулюючи реальне середовище за допомогою цифрових двійників, можна показати як навмисні, так і випадкові наслідки атак на завади і отримати важливе уявлення про них. Наприклад, Colosseum було використано для проведення великомасштабних експериментів з контрольованим глушінням з апаратними засобами в контурі та комерційною системою глушіння. Результати експериментів показують, що цифровий двійник досягає точності до 98% у відтворенні пропускної здатності, відношення сигнал/завада плюс шум (SINR) і структури стану лінії зв'язку порівняно з реальними експериментами із завадостійкості. Крім того, на рисунку 6.8 та 6.9 порівнюються результати для двох типів завад, вузькосмугових і широкосмугових, проти системи OFDM.

Вузькосмуговий сигнал займає ~ 156 кГц з 20 МГц спектру, що використовується для зв'язку, широкосмуговий сигнал займає 10 МГц. Дивлячись на рисунок, ми бачимо, що в реальних експериментах вузькосмугові завади призводять до зниження пропускної здатності на 37-43%, тоді як широкосмугові завади - на 94-96%. Подібні тенденції спостерігаються і в цифрових двійникових експериментах [34].

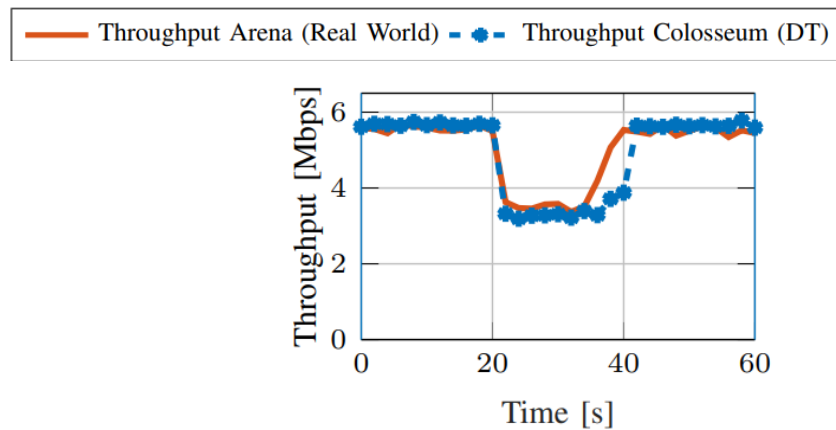


Рис. 5.9. Вузькосмугова перешкода. [34]

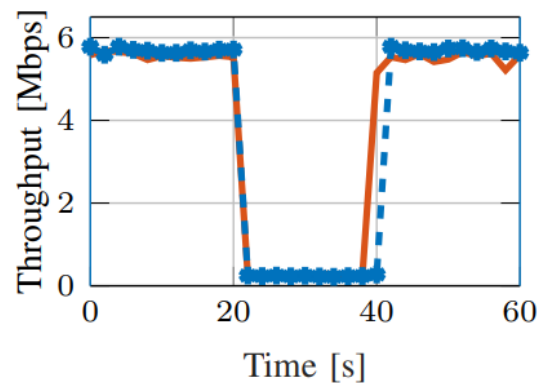


Рис. 5.10. Широкасмугова перешкода [34]

Рис. 5.9 та 5.10: Результати пропускної здатності в експериментах із завадами на Arena та Тестові стенди Colosseum, що демонструють вузькосмугові та широкосмугові корпуси [34].

5.7 Colosseum для тестування O-Ran та інтеграція

Цифровий двійник Colosseum Open RAN також знаходиться в центрі нещодавно створеного Центру відкритого тестування та інтеграції (OTIC) в Центрі Open6G Північно-Східного університету. OTIC - це центр, офіційно визнаний O-RAN ALLIANCE для тестування та інтеграції компонентів Open RAN від різних постачальників. В даний час основна діяльність з тестування та інтеграції, яку проводять OTIC по всьому світу, пов'язана з досягненням інтероперабельності в дезагрегованій архітектурі Open RAN. Наприклад, OTIC можуть оцінити відповідність інтерфейсу O-RAN або обладнання специфікаціям O-RAN, або сертифікувати сумісність пар компонентів і наскрізних систем. Хоча центр обробки даних "Colosseum" наразі обладнаний рішеннями для проведення такого тестування, ми також розглядаємо його подвійні можливості - разом з контрольованим і повторюваним Експериментальне середовище та його інфраструктура ШІ/ML - як ключовий компонент для тестування і, зрештою, сертифікації рішень зі штучного інтелекту для O-RAN. На сьогоднішній день дійсно ведуться дискусії щодо сертифікації інтероперабельності та відповідності інтерфейсів, пов'язаних з RIC, але тестування та оцінка ШІ є відкритою проблемою, яка потребує подальшого аналізу та розвитку. Рішення зі штучним інтелектом дійсно можуть бути недетермінованими і демонструвати різну поведінку при розробці різних сценаріїв, а також в динамічних умовах і при різному навантаженні на мережу. Це призводить до того, що визначення вимог до сертифікації ШІ є нетривіальним завданням. Можливою стратегією є використання еталонних сценаріїв, як це зазвичай робиться з наборами даних у спільноті комп'ютерного зору. У цьому сенсі Colosseum з його фреймворком OpenRAN Gum може реалізувати і надати такі еталонні сценарії спільноті, використовуючи різноманітні сценарії бездротового розгортання, стеки протоколів з відкритим вихідним кодом, а також платформи RIC з відкритим вихідним кодом або комерційні платформи. Тестування може передбачати досягнення мінімальних показників точності або ефективності в еталонних

сценаріях з використанням xApps або rApps, прийняття зрозумілих фреймворків ШІ і забезпечення рівнів довіри до агентів ШІ, які поведуться неправильно в невідомих умовах. Нарешті, Colosseum розвивається, щоб пристосувати користувацькі і COTS-радіостанції, підключені до цифрової інфраструктури подвійної радіоємуляції. Це робить можливим встановлення на борту радіоелектронного обладнання, яке дозволяє проводити оцінки, що виходять за рамки можливостей SDR USRP, наприклад, O-RAN-сумісних радіостанцій або комерційних радарів [34].

Висновки

У розділі розглянуто використання цифрового двійника Colosseum для тестування та розвитку мереж Open RAN. Це потужна платформа для емуляції бездротових мереж, яка дозволяє проводити дослідження та експерименти з реалістичними сценаріями радіочастотного покриття. Використання Colosseum для цифрового двійника дозволяє зібрати, перевірити та оптимізувати дані з мережевих компонентів, таких як базові станції та користувацькі пристрої, в режимі реального часу. Колізей, завдяки своїй гнучкості та інфраструктурі AI/ML, може бути використаний для проектування, тестування та сертифікації рішень Open RAN від різних постачальників, а також для створення надійних сценаріїв для випробувань в реальних умовах. Розгортання та тестування стеків протоколів за допомогою фреймворку OpenRAN Gum дозволяє проводити експерименти з різними типами налаштувань, такими як управління трафіком і оптимізація продуктивності. Платформа Colosseum має також великий потенціал для вдосконалення алгоритмів на основі даних та штучного інтелекту, що забезпечує більш ефективне управління мережею у реальному часі. Розширення можливостей Colosseum для тестування O-RAN-сумісних радіостанцій і оцінки систем у різноманітних умовах значно підвищує ефективність тестування і сертифікації цифрових двійників у реальних умовах. Інтеграція Colosseum з іншими платформами, такими як Arena та X5G, дозволяє безперешкодно переходити між реальними та емуляційними середовищами, що надає ще більше

можливостей для досліджень у сфері Open RAN. Завдяки використанню таких інструментів, як MQTT для обміну даними та фреймворку XAI для аналізу рішень штучного інтелекту, Colosseum забезпечує ефективне і безпечне середовище для тестування та верифікації інноваційних підходів у керуванні та оптимізації Open RAN. Цей підхід надає нові можливості для впровадження та тестування рішень на основі ШІ без ризику для реальних мереж, що є важливим кроком у розвитку стійких і надійних телекомунікаційних інфраструктур.

РОЗДІЛ 6

ОПТИМІЗАЦІЯ НАРІЗКИ МЕРЕЖ 5G/6G ЗА ДОПОМОГОЮ NTN

6.1. Модель системи

Розглядаємо мережу 5G/6G, що складається з gBS і FBS на базі БПЛА, яка обслуговує набір M UE в межах eMBB slice [40].

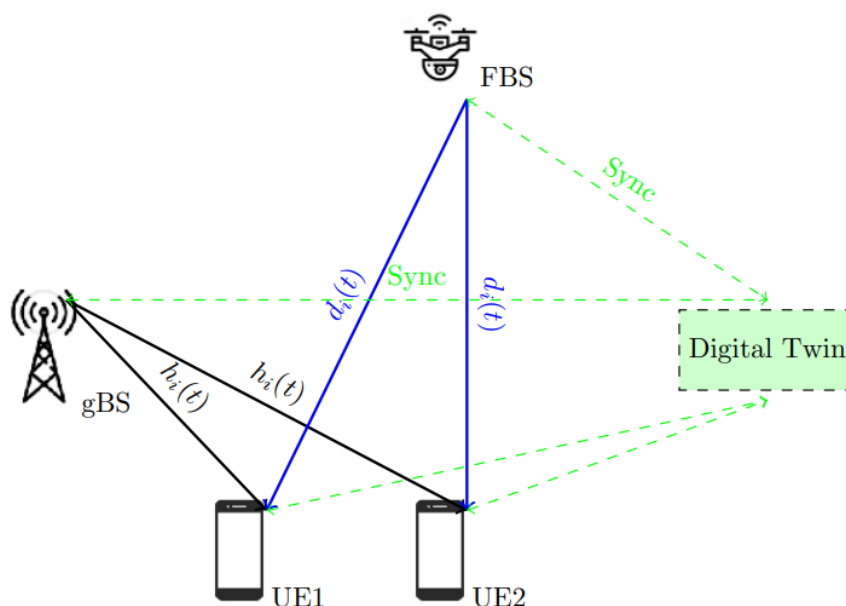


Рис. 6.1. Модель системи: Архітектура мережі з gBS, FBS, UEs та взаємодією DT [40]

Мережа використовує ортогональне мультиплексування з частотним розділенням (OFDM) з загальною смугою пропускання B , поділеною на N RBs. Рис. 6.1 ілюструє модель системи з архітектурою мережі. DT підтримує віртуальне представлення фізичної мережі, фіксуючи позиції UE, CSI, мобільність FBS та вимоги до трафіку [40].

Відстань, $d_i(t)$, між UE і та FBS враховує його мобільність і може бути

записана як [40]:

$$d_i(t) = \sqrt{(x_i - x_{\text{FBS}}(t))^2 + (y_i - y_{\text{FBS}}(t))^2 + h_{\text{FBS}}^2} \quad (6.2)$$

де (x_i, y_i) та $(x_{\text{FBS}}(t), y_{\text{FBS}}(t))$ - координати i -ї НР та FBS відповідно, а h_{FBS} - висота FBS [40].

Коефіцієнт підсилення каналу $h_{(i)}(t)$ для i -го ПП моделюється у вигляді [40]:

$$h_i(t) = \sqrt{\frac{\kappa}{d_i(t)^\beta}} g_i(t) \quad (6.3)$$

де κ - константа втрат на проходження, β - експонента втрат на проходження, а $g_i(t) \sim \text{CN}(0,1)$ - коефіцієнт дрібномасштабних завмирань за релеєвським розподілом. Завада I_i , що зазнала впливу де UE і включає внески від інших UE та вузлів NTN [40]:

$$I_i = \sum_{j \neq i}^M P_j |h_{j,i}|^2 + I_{\text{NTN}} \quad (6.4)$$

де $h_{(j,i)}$ -коефіцієнт підсилення каналу від UE j до базової щабілугове UE i , та $I_{\text{NTN}} \sim N(0, \sigma_{\text{NTN}}^2)$ моделює завади від інших вузлів NTN, наприклад, супутників [40].

Положення БС, pos_i , змінюється з часом відповідно до просторового розташування абонентів та їх відповідного трафіку, що впливає на відстань $D_{(i)}(t)$ між БС та i -м абонентом, як показано в (6.2). Відстань впливає на коефіцієнт підсилення каналу $h_{(i)}(t)$ у (6.3), що є ключовим фактором у визначенні відношення сигнал до завад і шуму (SINR) i , як наслідок, досяжної швидкості. ДТ динамічно відстежує рухливість FBS, точне налаштування стану s_t відповідно до умов мережі в реальному часі, щоб агент DRL міг зробити дані інформованими оптимально приймати рішення щодо розподілу ресурсів. Вважаємо, що траєкторія руху БС3 визначається швидкістю $\mathbf{v}_{\text{FBS}}(t) =$

$(v_x(t), v_y(t))$, яка оновлюється наступним чином [40]:

$$x_{\text{FBS}}(t + \Delta t) = x_{\text{FBS}}(t) + v_x(t)\Delta t \quad (6.5)$$

$$y_{\text{FBS}}(t + \Delta t) = y_{\text{FBS}}(t) + v_y(t)\Delta t \quad (6.5)$$

де $v_x(t)$ та $v_y(t)$ обмежені $\sqrt{v_x(t)^2 + v_y(t)^2} \leq v_{\text{max}}$. Для оптимізації покриття ми мінімізуємо середню відстань до UE наступним чином [40]:

$$J_{\text{FBS}}(t) = \frac{1}{M} \sum_{i=1}^M d_i(t) \quad (6.6)$$

при умові [40]:

$$x_{\text{FBS}}^{\min} \leq x_{\text{FBS}}(t) \leq x_{\text{FBS}}^{\max}, y_{\text{FBS}}^{\min} \leq y_{\text{FBS}}(t) \leq y_{\text{FBS}}^{\max} \quad (6.7)$$

Отже, для оновлення швидкості використовуються градієнти, як показано нижче [40]:

$$v_x(t) = -\eta \frac{\partial J_{\text{FBS}}}{\partial x_{\text{FBS}}} = -\eta \frac{1}{M} \sum_{i=1}^M \frac{x_{\text{FBS}}(t) - x_i}{d_i(t)} \quad (6.8)$$

$$v_y(t) = -\eta \frac{\partial J_{\text{FBS}}}{\partial y_{\text{FBS}}} = -\eta \frac{1}{M} \sum_{i=1}^M \frac{y_{\text{FBS}}(t) - y_i}{d_i(t)} \quad (6.9)$$

де η - розмір кроку.

Досяжна швидкість і затримка визначають ключові показники продуктивності для розбиття мережі на сегменти в 5G/6G. Досяжна швидкість $R_{(i)}(t)$ для i -го UE в момент часу t виводиться з формули пропускної здатності Шеннона, яка визначає максимальну швидкість передачі даних в смузі пропускання B при заданому SINR [40]:

$$C = B \log_2(1 + \text{SINR}), \quad (6.10)$$

де C - пропускна здатність. Для i -го UE замінимо C на $R_{(i)}(t)$ і B з виділеною смугою пропускання B_i . На SINR впливає коефіцієнт підсилення каналу $h_{(i)}t_j$ і завади, і може бути виражений як [40]:

$$\text{SINR}_i = \frac{P_i h_i(t)}{N_0 + I_i}, \quad (6.11)$$

де P_i -передана потужність, а $N_{(0)}$ - потужність шуму. Підставляючи $\text{SINR}_{(i)}$ в (9), досяжна швидкість для i -го UE стає [40]:

$$R_i(t) = B_i \log_2 \left(1 + \frac{P_i h_i(t)}{N_0 + I_i} \right) \quad (6.12)$$

На затримку впливають помилки синхронізації $\epsilon_i(t)$, спричинені розбіжностями між фізичною мережею та DT, а також обмеженнями ресурсів, що обмежують пропускну здатність та затримку обчислень. Вони моделюються у функцію винагороди r_t , яка балансує між максимізацією швидкості та мінімізацією затримки, вказуючи агенту DRL оптимізувати розподіл ресурсів для кожного зрізу. Затримка UE i у зрізі eMBB дорівнює [40]:

$$L_i(t) = \frac{D_i(t)}{R_i(t)} + T_{\text{proc}} + \frac{d_i(t)}{c} \quad (6.13)$$

де $D_i(t)$ - розмір даних, $T_{(\text{proc})}$ - затримка обробки, а $c = 3 \times 10^8$ м/с швидкість світла.

Синхронізація DT гарантує, що віртуальна модель відповідає реальній фізичній мережі 5G/6G, особливо в динамічному середовищі, спричиненому мобільністю FBS. DT постійно оновлюється за допомогою $h_i(t), D_i(t)$ і $\text{pos}_{\text{FBS}}(t)$, щоб забезпечити узгодженість DT з фізичною мережею, незважаючи на затримку, або неточності даних. DT оцінює попит на трафік $D_{(i)}(t)$ за допомогою моделі експоненціального ковзного середнього [40]:

$$\hat{D}_i(t+1) = \alpha D_i(t) + (1 - \alpha) \hat{D}_i(t) \quad (6.14)$$

де $\alpha \in [0,1]$ - коефіцієнт згладжування. Для синхронізації між DT та фізичною мережею DT коригує свій стан кожні Δt секунд відповідно до спостережень $h(t), d(t)$ та $D(t)$ у реальному часі. Похибка синхронізації представляє собою похибку ϵ_i надіслано як [40]:

$$\epsilon_i(t) = |D_i(t) - \hat{D}_i(t)|, \quad (6.15)$$

З метою мінімізації $\mathbb{E}[\epsilon_i(t)]$ через часті оновлення. Метою оптимізації є мінімізація середньої затримки для всіх UE [40]:

$$L_{\text{avg}} = \frac{1}{M} \sum_{i=1}^M L_i \quad (6.16)$$

при умові ресурсні обмеження наступним чином:

Модель системи враховує динаміку NTN шляхом оновлення (1) на основі швидкості $v_{(\text{FBS})}$ та траєкторії FBS, забезпечуючи точне представлення варіацій каналу в DT [40].

6.2. Запропонована структура ШІ-керованої DT

Тут розглянемо структуру, яка інтегрує DT та агента DRL для оптимізації мережі. Запропонований фреймворк використовує алгоритм DDPG для максимізації нарізки мережі для 5G/6G NTN-інтегрованих мереж. Рис. 6.17. та Рис. 6.18. ілюструють запропоновану архітектуру. На Рис. 6.17. фізична мережа надає дані в реальному часі, такі як $h_{(i)}(t)$ і $D_{(i)}(t)$, до DT через синхронізацію. Далі, DT перетворює ці дані в стан $s_{(t)}$, яку агент DRL використовує для генерації дій a_t через свою DNN та політику $\pi_{\theta}(s, a)$, передаючи їх назад до DT для симуляції. Потім DT повертає винагороду r_t до DRL агентом, що дозволяє постійно навчатися та оптимізувати мережеві активи. На Рис. 6.18. показано, як DT надає стан $s_{(t)}$ кожному вхідному вузлу повністю пов'язаної нейронної мережі агента DRL. Також НМ забезпечує винагороду r_t для всієї нейронної мережі, в той час як вихідні дії a_t , що представляють стохастичні політики для m зрізів мережі, циклічно повертаються назад до DT для моделювання. Цей ітераційний метод дозволяє агенту DRL навчитися ефективно управління ресурсами мережевого зрізу. На відміну від традиційних евристичних або статичних методів розподілу ресурсів, наша система використовує прогностичні можливості DT та адаптивна реакція DRL на специфічні для NTN виклики, такі як мобільність FBS та сплески трафіку.

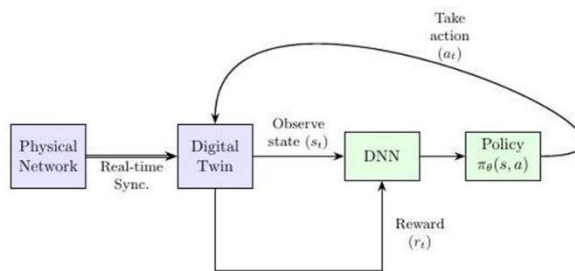


Рис. 6.17. Синхронізація між фізичною мережею та цифровим двійником [40]

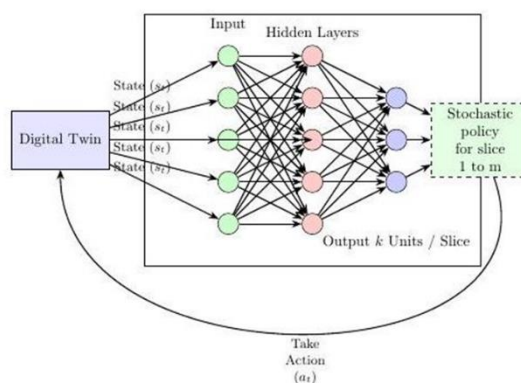


Рис. 6.18. Повністю пов'язана нейронна мережа агента DRL [40]

Запропонована архітектура ШІ-керованого ДІ. Новизна полягає в трьох аспектах. По-перше, запропонований нами метод включає синхронізацію в реальному часі. Фреймворк ДТ використовує (6.14) і (6.15) для підтримки точної віртуальної моделі, що дозволяє проактивно розподіляти ресурси. По-друге, наше рішення включає адаптацію на основі DRL, де алгоритм DDPG регулює розподіл смуги пропускання на основі динаміки NTN, перевершуючи статичні методи в середовищах, чутливих до затримок. Нарешті, наш фреймворк має перевагу оптимізації з урахуванням NTN. Наша система моделює мобільність FBS явно через (6.2), з гарантованою надійною продуктивністю в умовах аварійного відновлення або міського блокування. Задача DRL формулюється як марковський процес прийняття рішень (MDP), який складається зі стану s_t , дій a_t та винагороди $r(t)$ [9]. Стан у будь-який момент часу t можна записати у вигляді [40]:

$$s_t = \{h_i(t), D_i(t), \text{pos}_i, \text{pos}_{\text{FBS}}(t), \epsilon_i(t)\}_{i=1}^M, \quad (6.19)$$

де $\text{pos}_i = (x_i, y_i)$ та $\text{pos}_{\text{FBS}}(t) = (x_{\text{FBS}}(t), y_{\text{FBS}}(t))$ отриману з (6.2), а $\epsilon^{(i)}(t)$ - з (6.15). Тоді, дія, a_t для розподілу пропускної здатності записується так [40]:

$$a_t = \{B_i\}_{i=1}^M \quad (6.20)$$

де розподіл пропускної здатності задовольняє (6.19). Нарешті, ми можемо записати нашу винагороду DRL так [40]:

$$r_t = - \sum_{i=1}^M L_i - \lambda \sum_{i=1}^M |B_i - B_i^{\text{prev}}| - \gamma \sum_{i=1}^M \epsilon_i(t) \quad (6.21)$$

де λ карає за різкі зміни розподілу, а γ мінімізує помилки синхронізації. Агент DDPG використовує акторно-критичну архітектуру. Мережа акторів, параметризована θ^μ , зіставляє стани з діями за допомогою політики $\mu(s_t; \theta^\mu)$. Критична мережа, параметризована θ^Q , оцінює функцію "діяцінність" $Q(s_t, a_t; \theta^Q)$. Актор навчений максимізувати [40]:

$$J(\theta^\mu) = \mathbb{E}[Q(s_t, a_t; \theta^Q) \mid a_t = \mu(s_t; \theta^\mu)] \quad (6.22)$$

тоді як критик мінімізує похибку часової різниці (TD), використовуючи цільову мережу, яка оновлюється за допомогою м'яких оновлень наступним чином [40]:

$$\mathcal{L}(\theta^Q) = \mathbb{E}[(r_t + \gamma Q(s_{t+1}, \mu(s_{t+1}; \theta^{\mu'}), \theta^{Q'}) - Q(s_t, a_t; \theta^Q))^2] \quad (6.23)$$

$$-Q(s_t, a_t; \theta^Q))^2] \quad (6.23)$$

де $\theta^{Q'}$ та $\theta^{\mu'}$ параметри цільової мережі, що оновлюються за допомогою [40]:

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}, \theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'} \quad (6.24)$$

з $\tau = 0.001$. Актор оновлюється за допомогою градієнта політики [40]:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}[\nabla_a Q(s_t, a; \theta^Q) \big|_{a=\mu(s_t)} \nabla_{\theta^\mu} \mu(s_t; \theta^\mu)] \quad (6.25)$$

Похибка в ТД може бути записана як [40]:

$$\delta_t = r_t + \gamma Q(s_{t+1}, \mu(s_{t+1}; \theta^{\mu'}), \theta^{Q'}) - Q(s_t, a_t; \theta^Q) \quad (6.26)$$

з обмеженим очікуваним квадратом [40]:

$$\mathbb{E}[\delta_t^2] \leq 2R_{\max}^2 + 2\gamma^2 L_Q^2 \mathbb{E}[\|s_{t+1} - s_t\|^2] \quad (6.27)$$

де $|r(t)| \leq R_{\max}$, а L_Q -стала Ліпшиця Q -функції.

DT - це уявна пісочниця для агента DRL, яка оновлює стан s_t кожний Δt на основі вимірювань $h_i(t)$ ((6.3)), $D_i(t)$ ((6.14)) та $\text{pos}_{(\text{FBS})}(t)$ ((6.2)) у реальному часі. DT моделює потенційні дії, щоб спостерігати їх вплив на $L_{(i)}$ ((6.13)), не впливаючи на фізичну мережу, що зменшує ризики розвідки. DT також використовує нейронну мережу для уточнення (6.14) відповідно до минулих моделей трафіку, зменшуючи $\epsilon_i(t)$ ((6.15)). Це спільне прогнозування подвійного рольового стану та моделювання дій відрізняє наш фреймворк від інших підходів на основі DT без оптимізації, оптимізованої ШІ [40].

Інноваційність запропонованого фреймворку полягає в комбінації DT і DRL для розбиття мережі на частини з урахуванням NTN. Цей підхід прямо звертається до мобільності FBS і блокувань, які є ключовими для середовищ 6G NTN. Використання DPG сприяє створенню просторів безперервної дії, пропонуючи адаптацію пропускну здатності з високою деталізацією порівняно з рішеннями DRL дискретної дії. Функція винагороди в (6.21) балансує затримку, стабільність і синхронізацію для підтримки стійкості в невизначених середовищах. Запропонована структура найкраще підходить для аварійного відновлення, де мережі NTN забезпечують тимчасове з'єднання, а також для міських умов з щільним сигналом блокування [40].

6.3. Результати моделювання

Ми проаналізували запропоновану структуру, використовуючи симулятор мережі 5G з $M = 50\text{UE}$, $B = 20\text{ МГц}$ та FBS, що рухається зі швидкістю 10 м/с. DT оновлюється кожні 10 мс, а агент DDPG навчається протягом 50000 епізодів. Порівнюючи наш підхід DT-DRL зі статичним розподілом та евристичним методом, заснованим на пропорційній справедливості. На рисунках 6.28, 6.29 та 6.30 ілюструється загальне порівняння продуктивності між запропонованою архітектурою та базовими методами. На рис. 6.28 показує залежність середньої

затримки від часу. Рішення DT-DRL досягає затримки 33 мс при 50 секундах, перевершуючи статичний розподіл (54 мс) на 25% та евристичний метод (48 мс) на 14%. Покращення пояснюється точним прогнозуванням попиту дикцію DT ((6.14)) та адаптивність розподілу DRL, яка є стійкою до мобільності FBS ((1)) та мінливості каналів ((6.3)). Рис. 6.29 ілюструє використання ресурсів у часі для запропонованого методу DT-DRL та двох інших базових ліній. Метод DT-DRL досягає 92% при $t = 50$ с, порівняно з 78% для статичного розподілу та 85% для евристичного методу, що підкреслює його перевагу в роботі при динамічних налаштуваннях NTN. Причиною покращеної продуктивності DT-методу є те, що DRL - це адаптивний розподіл смуги пропускання на основі обмеження ресурсу в (6.18) [40].

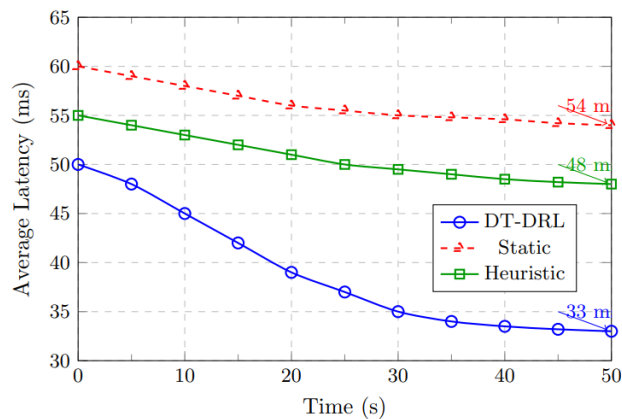


Рис. 6.28. Середня затримка в залежності від часу [40]

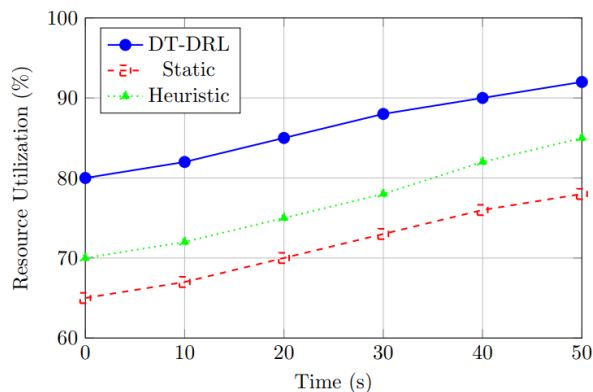


Рис. 6.29. Використання ресурсів проти часу [40]

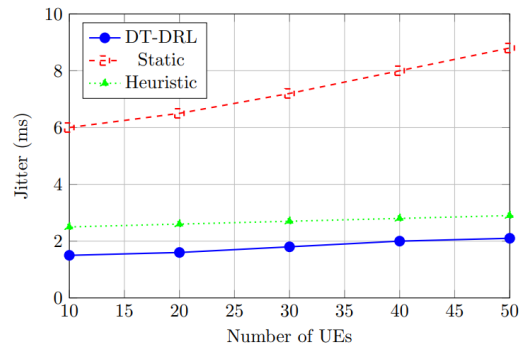


Рис. 6.30. Залежність джиттера від часу [40]

Розглянемо порівняння продуктивності запропонованого методу з базовими методами для зрізу eMBB. На відміну від статичного розподілу, який створює фіксовані розподіли пропускної здатності, DT-DRL використовує прогнозовані вимоги трафіку, виражені в (6.14), для перерозподілу розподілів пропускної здатності B_i відповідно до змін трафіку в реальному часі, на відміну від фіксованої пропускної здатності. Це ефективно в сценаріях NTN з мобільністю FBS зі змінними умовами каналу, як це змодельовано в (6.2), для ефективного використання ресурсів в таких додатках, як операції з ліквідації наслідків стихійних лих з високим попитом на трафік. Агент DRL оптимально розподіляє відповідно до функції винагороди в (6.21), і це зменшує затримку, стабільність розподілу та похибку синхронізації DT. Неперервний простір дії алгоритму DDPG дозволяє тонко контролювати B_i , що призводить до плавного зростання використання з 80% при $t = 0$ с до 92% при $t = 50$ с. Умова стабільності в (6.21) запобігає різким перемиканням розподілів, на відміну від евристичних рішень, заснованих на пропорційній справедливості, які досягають лише 85% використання. На рис. 6.30 показано часовий графік джиттера. Схема DT-DRL має низький джиттер 2,1 мс при $t = 50$ с, значно перевершуючи статичний розподіл (8,8 мс) і евристичну схему (2,9 мс), що свідчить про її ефективність для чутливих до затримок послуг eMBB в сценаріях NTN. Адаптивно налаштовуючи B_i , щоб динамічно слідувати за $D_{(i)}(t)$, підхід мінімізує варіації затримки в (6.13), таким чином мінімізуючи безпосередньо джиттер, особливо за наявності динамічних умов каналу через рухливість FBS

(6.2). Наприклад, у сценаріях з міськими заторами або ліквідацією наслідків стихійних лих, синхронні оновлення DT у реальному часі дозволяють ефективно оновлювати стан, стабілізуючи розподіл ресурсів. Агент DRL оптимізує затримку L_i і карає за різкі зміни розподілу, надаючи перевагу плавним змінам B_i . Це призводить до того, що джиттер DT-DRL дещо зростає з 1,5 мс при $t = 0$ с до 2,1 мс при $t = 50$ с, що охоплює термін стабільності в (6.21). З іншого боку, статичний розподіл страждає від великого джиттера (від 6,0 мс до 8,8 мс), оскільки константа B_i не реагує на сплески трафіку. Евристичний метод досягає прийняттого джиттера (від 2,5 мс до 2,9 мс), але є менш точним порівняно з DT-DRL, оскільки він має пропорційну залежність від справедливості. Низький джиттер, що підтримується системою DT-DRL з плином часу, підкреслює синергію прогнозного моделювання (6.14), оптимізації DRL (6.21) та адаптації з урахуванням мобільності (6.2). Це робить її дуже придатною для послуг eMBB, таких як відео високої чіткості або доповнена реальність в мережах 5G/6G, де низький джиттер повинен бути забезпечений в будь-який час [40].

Висновки

У цьому розділі досліджено використання цифрових двійників для оптимізації мереж 5G/6G з інтеграцією технології NTN (Non-Terrestrial Networks). Запропонована модель системи з використанням gBS і FBS на базі БПЛА, що обслуговує UE в межах eMBB slice, забезпечує ефективний моніторинг і управління ресурсами мережі. Впровадження цифрових двійників дозволяє динамічно відстежувати стан фізичної мережі, зокрема мобільність FBS та вимоги до трафіку, а також прогнозувати і адаптувати мережеві ресурси для покращення продуктивності. Порівняння запропонованого підходу DT-DRL з базовими методами показало значне покращення продуктивності в різних аспектах. Система DT-DRL зменшує затримку, покращує використання ресурсів і стабільність мережі, а також знижує джиттер, що критично важливо для послуг з високими вимогами до затримки, таких як відео високої чіткості або доповнена

реальність. Динамічний розподіл ресурсів через агент DRL забезпечує ефективну адаптацію до змінних умов мережі, таких як мобільність FBS і зміни в умовах каналу, що дозволяє оптимізувати продуктивність у реальному часі. Модель також демонструє високий потенціал для розв'язання задач, пов'язаних із динамічною адаптацією мережевих ресурсів і аварійним відновленням, що робить її ідеальною для умов 5G/6G NTN. Оптимізація за допомогою DRL, в поєднанні з цифровими двійниками, дає змогу значно покращити продуктивність і забезпечити стабільну роботу мережі в реальних умовах.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

Дослідження, проведене в межах цієї дипломної роботи, висвітлює застосування цифрових двійників (DT) у мережах 5G та 6G, зокрема в контексті інтеграції Non-Terrestrial Networks (NTN). Основний внесок роботи полягає в розробці моделі оптимізації мережі, яка використовує цифрові двійники для забезпечення ефективного моніторингу та управління ресурсами в реальному часі.

У роботі було показано, що цифрові двійники здатні надавати точне віртуальне відображення фізичних мереж, що дозволяє прогнозувати і адаптувати мережеві ресурси відповідно до змінних умов. Зокрема, інтеграція Deep Reinforcement Learning (DRL) із цифровими двійниками дозволяє оптимізувати пропускну здатність і мінімізувати затримки, що особливо важливо для послуг з високими вимогами до часу відгуку, таких як відео високої чіткості або доповнена реальність. Порівняння різних методів управління ресурсами (статичного, евристичного та DRL) показало перевагу підходу, заснованого на DRL, зокрема в контексті зниження джиттера та покращення використання ресурсів мережі.

Також було показано, що цифрові двійники є ефективним інструментом для моделювання та оптимізації в умовах динамічної мобільності та змінних умов каналу, що робить їх незамінними для мереж з високими вимогами до адаптивності, таких як мережі 5G/6G. Зокрема, модель на основі DRL, підтримувана цифровими двійниками, демонструє можливості для точного налаштування розподілу смуги пропускання та забезпечення ефективної роботи мережі в умовах реального часу.

Цифрові двійники також мають значний потенціал у підтримці аварійного відновлення, управлінні мобільністю та оптимізації навантаження в мережах з високим рівнем складності, забезпечуючи їх стійкість до втрат зв'язку та інших несправностей. Використання таких технологій у майбутніх мережах 5G та 6G може суттєво покращити якість обслуговування, зменшити витрати на підтримку

та обслуговування мереж, а також дозволити ефективно вирішувати завдання у реальному часі.

Враховуючи всі зазначені аспекти, можна стверджувати, що інтеграція цифрових двійників в управління мережами 5G/6G та NTN відкриває нові можливості для більш ефективного, гнучкого та адаптивного управління мережами з високими вимогами до якості обслуговування, що робить технологію перспективною для майбутнього розвитку телекомунікаційних інфраструктур.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is a digital twin? [Електронний ресурс] - <https://www.ibm.com/think/topics/what-is-a-digital-twin>
2. Що таке технологія цифрового двійника? [Електронний ресурс] - <https://aws.amazon.com/what-is/digital-twin/>
3. Технологія цифрових двійників [Електронний ресурс] - <https://www.autodesk.com/design-make/emerging-tech/digital-twin>
4. Emmert-Streib, F. (2023). Defining a Digital Twin: A Data Science-Based Unification. *Machine Learning and Knowledge Extraction*, 5(3), 1036-1054. [Електронний ресурс] - <https://doi.org/10.3390/make5030054>
5. Коррал-Асеро, Дж.; Маргара, Ф.; Марциняк, М.; Родеро, К.; Лончарич, Ф.; Фенг, Ю.; Гілберт, А.; Фернандес, Дж. Ф.; Бухарі, НА; Вайдан, А.; та ін. «Цифровий близнюк», щоб забезпечити бачення точної кардіології. *євро Heart J.* 2020,41, 4556-4564. [Електронний ресурс] - <https://doi.org/10.1093/eurheartj/ehaa159>
6. Digital Twins – Timeline [Електронний ресурс] - <https://www.verdict.co.uk/digital-twins-timeline/>
7. Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access*, 8, 108952-108971. [Електронний ресурс] - <https://doi.org/10.1109/ACCESS.2020.2998358>
8. «Цифровий двійник» Ян Чжан; ISBN: 978-3-031-51818-8 [Електронний ресурс] - <https://doi.org/10.1007/978-3-031-51819-5>
9. ЗАСТОСУВАННЯ ЦИФРОВОГО ДВІЙНИКА ТА МАШИННОГО НАВЧАННЯ ДЛЯ ОПТИМІЗАЦІЇ СПАЛЮВАННЯ ДЕРЕВНОЇ ТРІСКИ У КОТЛАХ І. А. Поліщук; Д. В. Столбов; [Електронний ресурс] - <https://doi.org/10.31649/1997-9266-2024-177-6-7-12>
10. TwinRAN: Twinning the 5G RAN in Azure Cloud Yash Deshpande, Eni Sulkaj, Wolfgang Kellerer [Електронний ресурс] - <https://arxiv.org/pdf/2407.13340>
11. Рост, П.; Банчс, А.; Берберана, І.; Брайтбах, М.; Долл, М.; Дросте, Х.;

- Маннвайлер, К.; Пуенте, М.А.; Самданіс, К.; Саяді, Б. Еволюція архітектури мобільної мережі в напрямку 5G. *IEEE Commun. Mag.* 2016 , 54 , 84-91. [Електронний ресурс] - DOI:10.1109/MCOM.2016.7470940
12. Арслан, Х.; Туша, С.Д.; Язар, А. Бачення 6G: надгнучка перспектива. *ITU J. Futur. Evol. Technol.* 2020 , 1 , 121-140.
13. Zhou, C.; Yang, H.; Duan, X.; Lopez, D.; Pastor, A.; Wu, Q.; Boucadair, M.; Jacquenet, C. Мережа цифрових двійників: концепції та еталонна архітектура. *Internet Eng. Task Force* 2022. [Електронний ресурс] - <https://datatracker.ietf.org/doc/draft-zhou-nmrg-digitaltwin-network-concepts/>
14. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2018. [Електронний ресурс] - <https://doi.org/10.1017/S0263574799271172>
15. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* 2015, 518, 529-533
16. Vilà, I., Sallent, O., & Pérez-Romero, J. (2023). On the Design of a Network Digital Twin for the Radio Access Network in 5G and Beyond. *Sensors*, 23(3), 1197. [Електронний ресурс] - <https://doi.org/10.3390/s23031197>
17. Vilà, I.; Pérez-Romero, J.; Sallent, O.; Umbert, A. A Multi-Agent Reinforcement Learning Approach for Capacity Sharing in Multi-tenant Scenarios. *IEEE Trans. Veh. Technol.* 2021, 70, 9450-9465 [Електронний ресурс] - DOI: 10.1109/MCOM.2016.7470940
18. C. Zhou, H. Yang, X. Duan, D. Lopez, A. Pastor, Q. Wu, M. Boucadair, and C. Jacquenet, "Digital Twin Network: Concepts and Reference Architecture," *Internet Engineering Task Force, Internet-Draft* [Електронний ресурс] - <https://datatracker.ietf.org/doc/html/draft-zhou-nmrg-digitaltwin-network-concepts-07>
19. M. Tariq, A. Zeitoun, V. Valancius, N. Feamster, and M. Ammar, "Answering what-if deployment and configuration questions with wise," in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, 2008, pp. 99-110.

20. Q. Zhang, K. K. Ng, C. Kazer, S. Yan, J. Sedoc, and V. Liu, "Mimicnet: fast performance estimates for data center networks with machine learning," in Proceedings of the 2021 ACM SIGCOMM 2021 Conference, 2021, pp. 287-304.
21. Y. Geng, S. Liu, Z. Yin, A. Naik, B. Prabhakar, M. Rosenblum, and A. Vahdat, "A simple and scalable method for sensing, inference and measurement in data center networks," in 16th Symposium on Networked Systems Design and Implementation (NSDI), 2019, pp. 549-564.
22. K. Rusek, J. Suarez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "Routenet: Leveraging graph neural networks for network modeling and optimization in sdn," IEEE Journal on Selected Areas in Communications, vol. 38, no. 10, pp. 2260-2270, 2020.
23. M. Wang, L. Hui, Y. Cui, R. Liang, and Z. Liu, "xnet: Improving expressiveness and granularity for network modeling with graph neural networks," in IEEE INFOCOM 2022 - IEEE Conference on Computer Communications (INFOCOM 2022), London, United Kingdom (Great Britain), May 2022.
24. J. Li, P. Sun, and Y. Hu, "Traffic modeling and optimization in datacenters with graph neural network," Computer Networks, vol. 181, p. 107528, 2020.
25. S. Xiao, D. He, and Z. Gong, "Deep-q: Traffic-driven qos inference using deep generative network," in Proceedings of the 2018 Workshop on Network Meets AI & ML, 2018, pp. 67-73.
26. Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "Cs2p: Improving video bitrate selection and adaptation with data-driven throughput prediction," in Proceedings of the 2016 ACM SIGCOMM Conference, 2016, pp. 272-285.
27. M. Mirza, J. Sommers, P. Barford, and X. Zhu, "A machine learning approach to tcp throughput prediction," IEEE/ACM Transactions on Networking, vol. 18, no. 4, pp. 1026-1039, 2010.
28. B. A. A. Nunes, K. Veenstra, W. Ballenthin, S. Lukin, and K. Obraczka, "A machine learning framework for tcp round-trip time estimation," EURASIP Journal on Wireless Communications and Networking, vol. 2014, no. 1, pp. 1-22, 2014.

29. F. Geyer, "Deepcomnet: Performance evaluation of network topologies using graph-based deep learning," *Performance Evaluation*, vol. 130, pp. 1-16, 2019.
30. T. Suzuki, Y. Yasuda, R. Nakamura, and H. Ohsaki, "On estimating communication delays using graph convolutional networks with semisupervised learning," in *2020 International Conference on Information Networking (ICOIN)*. IEEE, 2020, pp. 481-486.
31. A. Mestres, E. Alarcon, Y. Ji, and A. Cabellos-Aparicio, "Understanding ' the modeling of computer network delays using neural networks," in *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, 2018, pp. 46-52.
32. M. Wang, Y. Cui, S. Xiao, X. Wang, D. Yang, K. Chen, and J. Zhu, "Neural network meets dcn: Traffic-driven topology adaptation with deep learning," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 2, pp. 1-25, 2018.
33. Digital Twin for Networking: A Data-driven Performance Modeling Perspective Linbo Hui, Mowei Wang, Liang Zhang, Lu Lu, and Yong Cui [Электронный ресурс] - <https://arxiv.org/pdf/2206.00310>
34. Colosseum: The Open RAN Digital Twin Michele Polese, Leonardo Bonati, Salvatore D'Oro, Pedram Johari, Davide Villa, Sakthivel Velumani, Rajeev Gangula, Maria Tsampazi, Clifton Paul Robinson, Gabriele Gemmi, Andrea Lacava, Stefano Maxenti, Hai Cheng, Tommaso Melodia [Электронный ресурс] - <https://arxiv.org/pdf/2404.17317>
35. L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "OpenRAN Gym: AI/ML Development, Data Collection, and Testing for O RAN on PAWR Platforms," *Computer Networks*, pp. 1-12, November 2022.
36. M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "ColO- RAN: Developing Machine Learning-Based xApps for Open RAN Closed-Loop Control on Programmable Experimental Platforms", *IEEE Transactions on Mobile Computing*, vol. 22, no. 10, pp. 5787-5800, Oct 2023.
37. J. Groen, M. Belgiovine, U. Demir, B. Kim, and K. Chowdhury, "TRACTOR:

Traffic Analysis and Classification Tool for Open RAN,” arXiv preprint arXiv:2312.07896, 2023.

38. M. Belgiovine, J. Gu, J. Groen, M. Sirera, U. Demir, and K. Chowdhury, “MEGATRON: Machine Learning in 5G with Analysis of Traffic in Open Radio Access Networks,” in International Conference on Computing, Networking and Communications (ICNC), 2024.

39. A. Scalingi, S. D’Oro, F. Restuccia, T. Melodia, and D. Giustiniano, “Det-RAN: Data-Driven Cross-Layer Real-Time Attack Detection in 5G Open RANs,” in IEEE International Conference on Computer Communications, 2024, pp. 1-10.

40. AI-Driven Digital Twins: Optimizing 5G/6G Network Slicing with NTNs Afan Ali;Huseyin Arslan [Электронный ресурс] - <https://arxiv.org/pdf/2505.08328>