Bachelor of Science in Electrical and Electronic Engineering

EEE 400: Thesis

Proactive Handover in Multi-Cell Networks: A Digital Twin Approach for Low-Complexity and Scalable Performance

Submitted by

Anindha Dhar

201906194

Tapu Datta

201906049

Supervised by

Dr. Lutfa Akter

Professor

Department of Electrical and Electronic Engineering

Bangladesh University of Engineering and Technology

Dhaka, Bangladesh

March 2025

CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis, titled, "**Proactive Handover in Multi-Cell Networks: A Digital Twin Approach for Low-Complexity and Scalable Performance**", is the outcome of the investigation and research carried out by us under the supervision of **Dr. Lutfa Akter**.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Anindha Dhar
201906194

Tapu Datta
201906049

CERTIFICATION

This thesis titled, **”Proactive Handover in Multi-Cell Networks: A Digital Twin Approach for Low-Complexity and Scalable Performance”**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for EEE 400: Project/Thesis course, and as the requirements for the degree B.Sc. in Electrical and Electronic Engineering in March 2025.

Group Members:

Anindha Dhar

Tapu Datta

Supervisor:

Dr. Lutfa Akter
Professor
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology

ACKNOWLEDGEMENT

We begin by expressing our deepest gratitude to the Almighty, the most merciful and benevolent, for His endless blessings and the strength He has granted us to complete this research within the stipulated time. His guidance has been our constant companion throughout this journey. We are deeply indebted to our parents for their unconditional love, endless support, and prayers, which have been our anchor during challenging times. Their belief in us has given us the courage to pursue this work with determination.

We would like to extend our heartfelt appreciation to our supervisor, **Dr. Lutfa Akter**, Professor in the Department of Electrical & Electronic Engineering, Bangladesh University of Engineering and Technology (BUET), for her unwavering support, insightful advice, and patient mentorship. Her expertise and encouragement have been instrumental in shaping this work, and we are truly fortunate to have had the opportunity to learn under her guidance. We extend our appreciation to, **Dr. Imtiaz Ahmed**, Assistant Professor in the Department of Electrical Engineering and Computer Science, Howard University, Washington, DC, USA and **Dr. Zoheb Hassan**, Professeur Adjoint in the Département de génie électrique et de génie informatique, Université Laval, Québec City, QC, Canada. They generously shared their time and insights, making this research possible. Their cooperation and willingness to participate have enriched the quality of this study.

Our sincere thanks go to all the faculty members of the Department of Electrical & Electronic Engineering at BUET for their invaluable teachings and inspiration throughout our academic journey. This research is a testament to the knowledge and skills they have imparted to us.

Thank you all from the bottom of our heart.

Dhaka
March 2025

Anindha Dhar

Tapu Datta

Contents

List of Figures

List of Tables

ABSTRACT

This research project investigates the development and evaluation of a Digital Twin (DT)-based proactive handover framework for wireless networks, addressing the limitations of traditional distance-based handover methods in dense, multi-cell environments. With the rapid evolution of 5G and beyond, ensuring seamless connectivity and high-quality service (QoS) amidst complex RF environments, including path loss, shadowing, and interference, has become a critical challenge.

The study employs advanced simulation tools, including the Sionna library with ray tracing, to construct a virtual replica of a multi-cell network, integrating realistic 3D environments, user mobility modeling via the Gauss-Markov model, and RF propagation analysis. Coverage maps are generated to assess SINR performance, enabling a proactive handover strategy that optimizes transmitter selection based on environmental awareness rather than proximity alone.

Key findings indicate that the DT-based handover outperforms distance-based methods, achieving superior SINR values in obstructed scenarios compared to suboptimal or -inf dB results with distance-based approaches. The comparative analysis across simulated cases highlights reduced BER and enhanced network efficiency, particularly in urban settings with significant blockages.

The research underscores the transformative potential of DT technology in improving wireless network management, offering a scalable solution for next-generation networks. Limitations include the reliance on simulated data and the need for real-world validation. Future work will focus on comparing results with commercial telecom handover protocols and integrating 6G communication and non-RF sensors like LiDAR for enhanced accuracy.

These findings contribute significantly to the field, advancing network optimization and user experience in dynamic wireless environments.

Keywords: Digital Twin, proactive handover, wireless networks, RF environment awareness, Sionna ray tracing, SINR optimization, network efficiency, 6G communication.

Chapter 1

Introduction

1.1 Background and Context

Wireless communication systems have undergone a remarkable evolution over recent decades, fueled by an ever-growing demand for high-speed, reliable, and ubiquitous connectivity. The emergence of 5G and beyond has ushered in dense, multi-cell network architectures designed to meet these demands, characterized by a proliferation of base stations (BSs) and increasingly complex radio frequency (RF) environments. In such networks, ensuring seamless mobility and maintaining a high quality of service (QoS) are paramount challenges. Handover—the process of transferring an active connection from one BS to another as a user moves through the network—plays a critical role in achieving this goal. However, traditional handover strategies often rely on simplistic metrics such as received signal strength or physical distance, which fail to fully capture the intricate dynamics of modern RF environments, including path loss, shadowing, and interference. This can result in suboptimal performance, manifesting as high bit error rates (BER), low signal-to-interference-plus-noise ratio (SINR), and excessive handover attempts, ultimately compromising network efficiency and user satisfaction.

1.2 Problem Statement

The limitations of conventional handover approaches become particularly evident in dense urban settings, where environmental factors such as buildings introduce significant link blockages and multipath effects. For example, a user may be physically closer to a BS but experience poor signal quality due to obstructions, while a more distant BS with a clear line-of-sight (LoS) path could provide superior performance. Distance-based handover, a widely adopted low-complexity method, lacks the environmental awareness needed to make informed decisions in such scenarios. This often leads to suboptimal transmitter selection, degraded QoS, and ineffi-

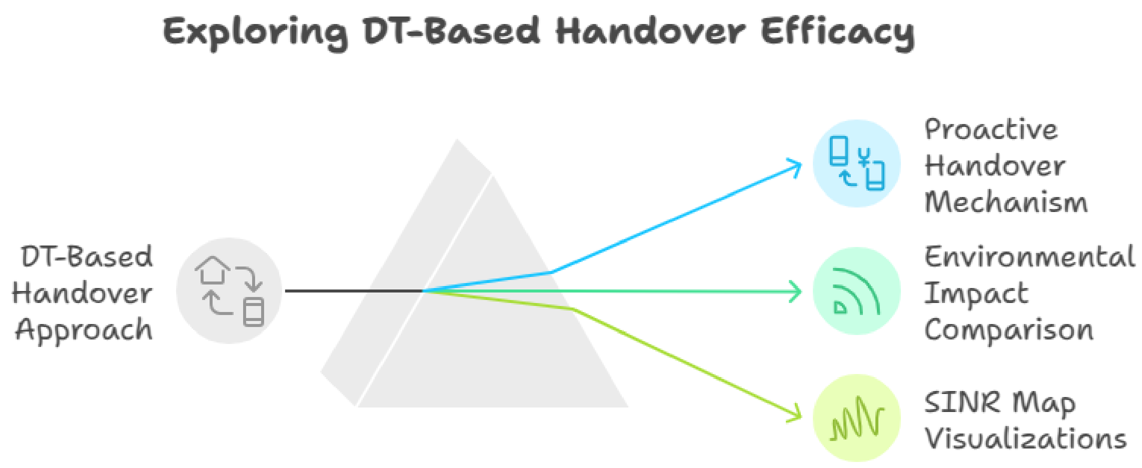

Figure 1.1: DT-Based Handover

cient resource utilization. As network densification continues to increase, these shortcomings highlight the urgent need for a more intelligent, proactive, and scalable handover framework capable of leveraging detailed RF environmental knowledge to optimize connectivity in multi-cell networks.

1.3 Proposed Solution: Digital Twin Approach

In this thesis, we propose a novel approach to proactive handover in multi-cell networks using a Digital Twin (DT) framework, enabled by advanced ray-tracing and coverage mapping techniques. A Digital Twin is a virtual replica of a physical system that mirrors its real-time behavior and characteristics, offering a powerful tool for simulation, analysis, and decision-making. By integrating the Sionna [1] library—a state-of-the-art open-source tool for wireless system simulation—we construct a DT of a multi-cell network based on the ”etoile” scene, a pre-built urban environment. This DT incorporates realistic RF propagation models, including LoS, reflection, and diffraction effects, to generate high-fidelity coverage maps. These maps assign metrics such as SINR to every point in the scene, providing a comprehensive view of the RF landscape across multiple transmitters. This approach is significant for enhancing network efficiency and user experience, especially in urban settings, and aligns with broader trends in network digital twins [2]. Our approach aims to overcome the limitations of traditional methods by pre-computing coverage maps and using them to efficiently determine optimal handover decisions, reducing computational complexity while enhancing performance.

Figure 1.2: Digital-Twin Based vs Distance Based Handover

1.4 Research Methodology and Simulations

Our work focuses on two key simulations to demonstrate the efficacy of the DT-based handover approach. In the first simulation, we develop a proactive handover mechanism that identifies the best-serving transmitter for a given user position by querying SINR values from pre-computed coverage maps. This method avoids the need for repeated path computations, offering a low-complexity alternative to conventional strategies. We evaluate this approach across three example cases, showcasing how the DT selects transmitters with the highest SINR, even when they are not the nearest, thus outperforming distance-based methods. The second simulation compares DT-based handover with distance-based handover in scenarios where environmental factors significantly impact performance. By analyzing a grid of user positions, we identify regions where the nearest transmitter fails to provide the best SINR due to blockages, while a more distant transmitter offers a clearer path. Through visualizations of SINR maps and cell-to-transmitter associations, we illustrate the DT's superior RF environment awareness and its ability to achieve better SINR performance.

1.5 Motivation and Significance

The motivation for this research stems from the pressing need for low-complexity, scalable solutions in next-generation wireless networks. As network densification intensifies, traditional handover algorithms become computationally prohibitive and less effective in dynamic, obstacle-rich environments. By leveraging pre-computed coverage maps and the predictive capabilities of a DT, our approach minimizes real-time computation overhead while maximizing handover accuracy. Furthermore, the use of Sionna's latest features, such as coverage map generation introduced in version 0.19.0, ensures that our framework is both cutting-edge and adaptable to future advancements in wireless simulation. This work has the potential to enhance network efficiency, improve user experience, and pave the way for smarter resource management in multi-cell networks.

1.6 Comparative Analysis Table

To further illustrate the comparison between DT-based and distance-based handover, the following table summarizes key aspects based on the simulations:

Table 1.1: Comparison of DT-Based and Distance-Based Handover

Aspect	**DT-Based Handover**	**Distance-Based Handover**
RF Environment Awareness	High, accounts for LoS, reflection, diffraction	Low, relies on proximity, ignores obstacles
Computational Complexity	Low, uses pre-computed coverage maps	High, requires repeated path computations
SINR Performance	Superior, especially in blocked areas	Suboptimal, may select blocked transmitters
Scalability	High, efficient for dense networks	Limited, scales poorly with network density
Example Outcome	Selects best SINR transmitter, e.g., 61.65 dB at certain positions	May choose nearest, e.g., -inf SINR if blocked

This table highlights the advantages of the DT approach, particularly in complex urban environments, as evidenced by the results.

1.7 Summary

In summary, this thesis introduces a proactive handover strategy that harnesses the power of Digital Twins to deliver low-complexity, scalable, and RF-aware performance in multi-cell networks. By bridging the gap between physical network dynamics and virtual simulation, we address the shortcomings of traditional handover methods and offer a forward-looking solution for next-generation wireless systems. Our work demonstrates how DT-enabled handover can optimize connectivity, reduce operational overhead, and enhance scalability, ultimately contributing to the development of smarter, more efficient networks capable of meeting the demands of tomorrow's connected world.

Chapter 2

Literature Review

This comprehensive literature review explores the evolution of handover techniques in wireless networks, with a particular focus on proactive handover approaches and the emerging role of digital twins. The review identifies significant advances in handover mechanisms from traditional signal-based methods to sophisticated proactive schemes leveraging machine learning and dual connectivity. Concurrently, digital twin technology has emerged as a powerful paradigm for modeling, simulating, and optimizing wireless networks. The integration of these technologies presents promising opportunities for developing low-complexity, scalable handover solutions that can meet the demanding requirements of future wireless networks.

2.1 Introduction to Handover and Proactive Handover

Handover, or handoff, is a critical process in wireless communication systems that ensures the seamless transfer of a mobile user's connection from one base station to another as they move within a network. This process is essential for maintaining service continuity, especially in heterogeneous networks where multiple technologies and access points coexist. Traditional handover techniques often rely on reactive methods, which may lead to service disruptions due to link degradation. Proactive handover, on the other hand, anticipates and initiates handover before any noticeable degradation in service quality, making it a vital feature for next-generation wireless networks [3] [4].

Proactive handover techniques leverage advanced technologies such as machine learning (ML) and deep learning to predict future link blockages and optimize handover decisions. For instance, deep-learning-based approaches have been proposed to predict line-of-sight (LoS) link blockages in dynamic urban environments, achieving prediction accuracies of up to 90% using wireless signatures alone [3]. These techniques are particularly relevant in 5G/6G networks, where millimetre wave (mmWave) frequencies are highly sensitive to blockages, necessitating

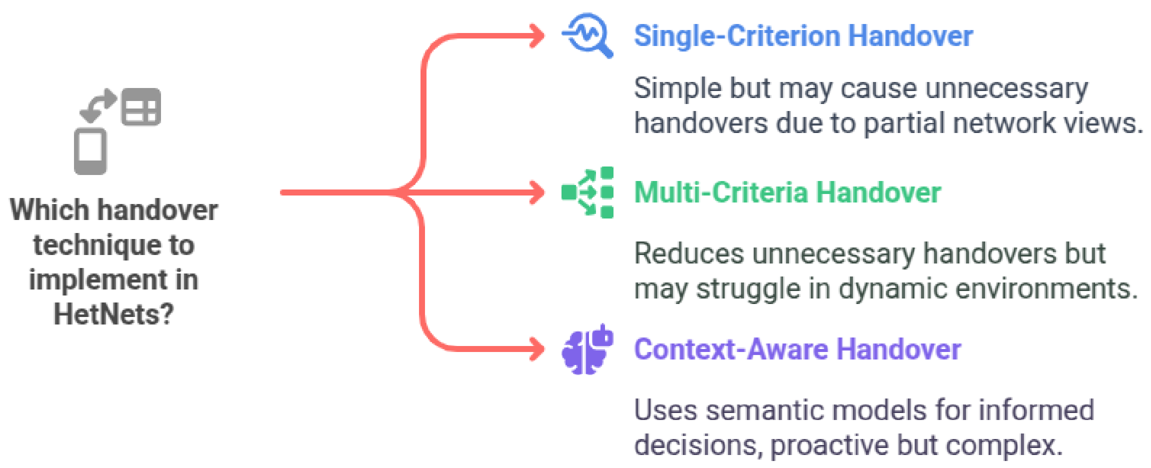

Figure 2.1: Techniques for Effective Handover Management in HetNets

robust, proactive handover solutions [3].

2.2 Handover Techniques in Heterogeneous Networks

Heterogeneous networks (HetNets) are characterized by the coexistence of multiple radio access technologies (RATs) and cell layers, such as macrocells, small cells, and relay nodes. Handover management in HetNets is more complex due to the diversity of network conditions and user requirements. Various handover techniques have been developed to address these challenges, including:

- **Single-Criterion Handover:** Based on a single parameter such as received signal strength (RSS) or signal-to-interference-plus-noise ratio (SINR). While simple, these methods may lead to unnecessary handovers due to partial network views [4] [5].
- **Multi-Criteria Handover:** Considers multiple factors such as link quality, user velocity, and network load. These methods reduce unnecessary handovers and improve network selection but may fail in dynamic environments [4] [5].
- **Context-Aware Handover:** Incorporates semantic information models and knowledge bases to make informed decisions. For example, the SIM-Know approach uses a semantic information model (SIM) and a distributed knowledge base profile (KBP) to enable context-aware and proactive handover decisions [4].

2.3 Proactive Handover Techniques

Proactive handover techniques aim to anticipate and mitigate potential service disruptions before they occur. These techniques are particularly important in scenarios where user mobility and dynamic network conditions can lead to sudden link failures. Key approaches include:

Table 2.1: Proactive Handover Techniques

Technique	**Description**
Predictive Analytics	Utilizes historical and real-time data to predict future link blockages. Deep-learning-based models have shown significant promise in this area, with accuracies exceeding 90% in certain scenarios [3].
Multi-Base Station Connectivity	Maintains simultaneous connections with multiple base stations to ensure seamless handover when the primary link degrades [3].
Non-RF Sensors and LiDAR	Leverages external sensors such as LiDAR and visual information to predict blockages and optimize handover execution [3].

2.4 Digital Twins: Concepts and Applications

Digital twins are virtual representations of physical systems that enable real-time monitoring, simulation, and optimization. They have gained significant attention in various domains, including manufacturing, healthcare, and telecommunications. In the context of wireless networks, digital twins offer several advantages, such as streamlined network development, enhanced productivity, and cost reduction [6] [7].

2.4.1 Key Features of Digital Twins

Digital twins are distinguished by their capacity to mirror the structure, behavior, and dynamics of physical systems across various domains. They leverage an array of advanced technologies, including the Internet of Things (IoT), artificial intelligence (AI), machine learning, and data analytics, to construct a virtual representation that can adapt to evolving operational conditions and environmental changes in real-time [8] [9]. These virtual models integrate diverse data sources—such as sensors, communication networks, and intelligent algorithms—to create comprehensive and adaptable representations of physical entities, spanning industries like manufacturing, healthcare, and infrastructure. The primary features of digital twins include:

Table 2.2: Key Features of Digital Twins

Feature	Description
Real-Time Monitoring	Continuous data collection from physical systems to update the virtual model [9].
Simulation and Optimization	Ability to simulate different scenarios and optimize system performance [10] [11].
Bi-Directional Feedback	Enables interaction between the physical and virtual systems, allowing for informed decision-making [9].

2.4.2 Applications of Digital Twins

Digital twins have a wide range of applications across various industries. In the context of wireless networks, they are particularly useful for:

- **Network Optimization:** Digital twins can be used to optimize network performance by simulating different configurations and predicting outcomes [6] [7].
- **Resource Allocation:** They enable efficient resource allocation by analyzing network traffic and user demand in real-time [7] [12].
- **Edge Caching:** Digital twins can be applied to edge caching scenarios to improve content delivery and reduce latency [6].
- **Proactive Handover:** Predicts future link blockages and optimizes handover decisions [3] [4].
- **Wireless Traffic Forecasting:** Used to predict wireless traffic patterns and optimize network performance [6].

2.5 Implementation of Digital Twins

The implementation of digital twins involves several steps, including data collection, model creation, and system integration. The process typically starts with the deployment of IoT sensors to collect real-time data from the physical system. This data is then used to create a virtual model that can simulate the behavior of the physical system [9].

2.5.1 Key Technologies for Digital Twin Implementation

Several technologies are essential for the successful implementation of digital twins, including:

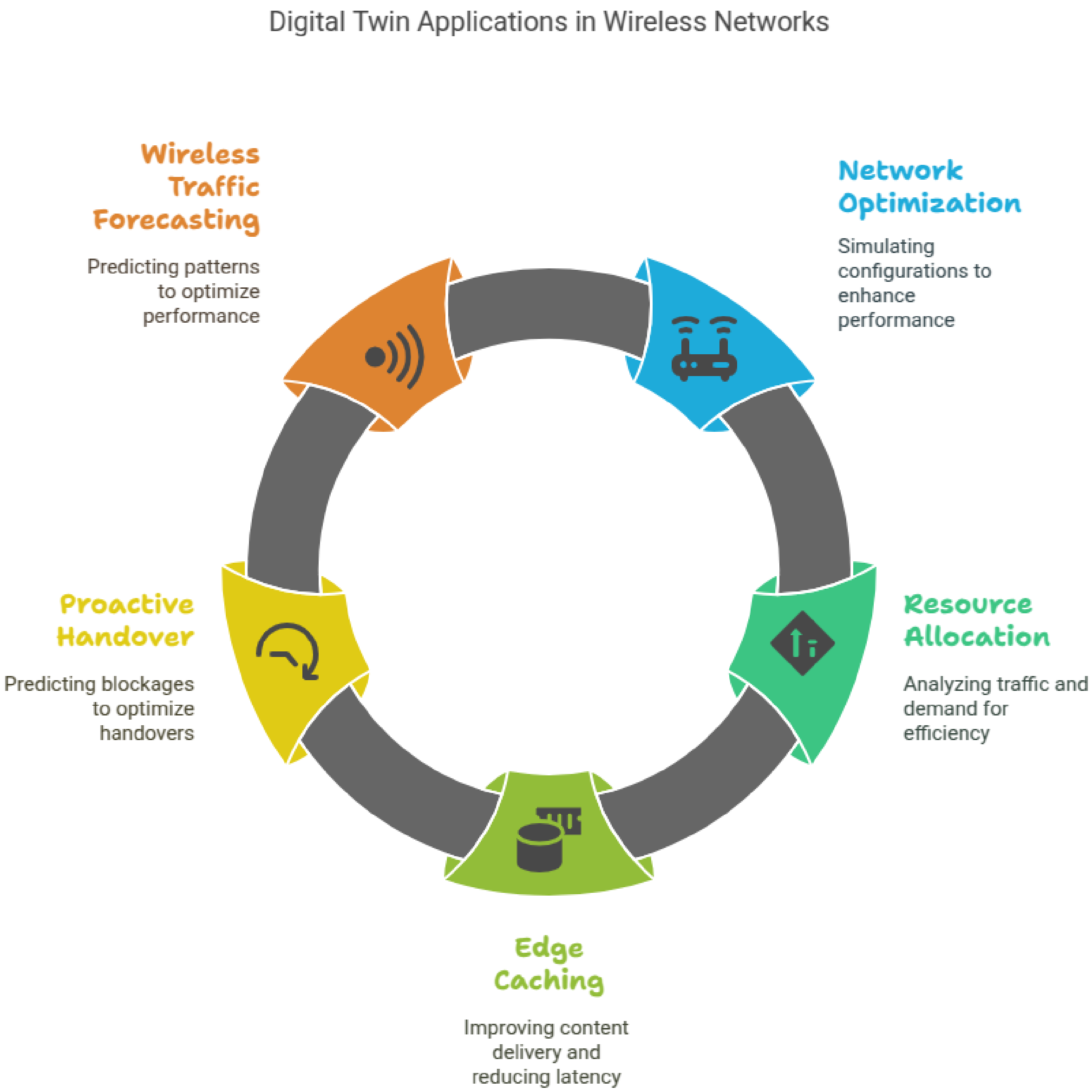

Figure 2.2: Unveiling the Power of Digital Twin in Wireless Networks

Table 2.3: Key Technologies for Digital Twin Implementation

Technology	**Description**
IoT Sensors	Collect real-time data from the physical system [8] [9].
Machine Learning	Used to analyze data and predict future trends [3] [8].
Cloud Computing	Provides the necessary infrastructure for data storage and processing [9] [13].
6G Communication	Enables fast and reliable data transfer between the physical and virtual systems [13].

2.5.2 Challenges in Digital Twin Implementation

Despite their potential, digital twins face several challenges during implementation. These include:

- **Data Integration:** Combining data from multiple sources can be complex and may require advanced data fusion techniques [8] [14].
- **Security and Privacy:** Protecting sensitive data from unauthorized access is a major concern [14].
- **Lack of Standardization:** The absence of standardized frameworks can hinder the widespread adoption of digital twins [14].

All of which require targeted solutions to ensure seamless adoption. Data integration can be particularly complex due to the need to combine information from multiple, often heterogeneous sources such as IoT sensors, legacy systems, and cloud platforms. To address this, organizations can adopt advanced data fusion algorithms that allow for more efficient merging of diverse datasets. Additionally, using middleware platforms or data lakes can help centralize and preprocess data before it is utilized by the digital twin system. Open APIs and interoperable communication protocols like MQTT or OPC UA can further enhance connectivity between different data sources, ensuring smoother integration.

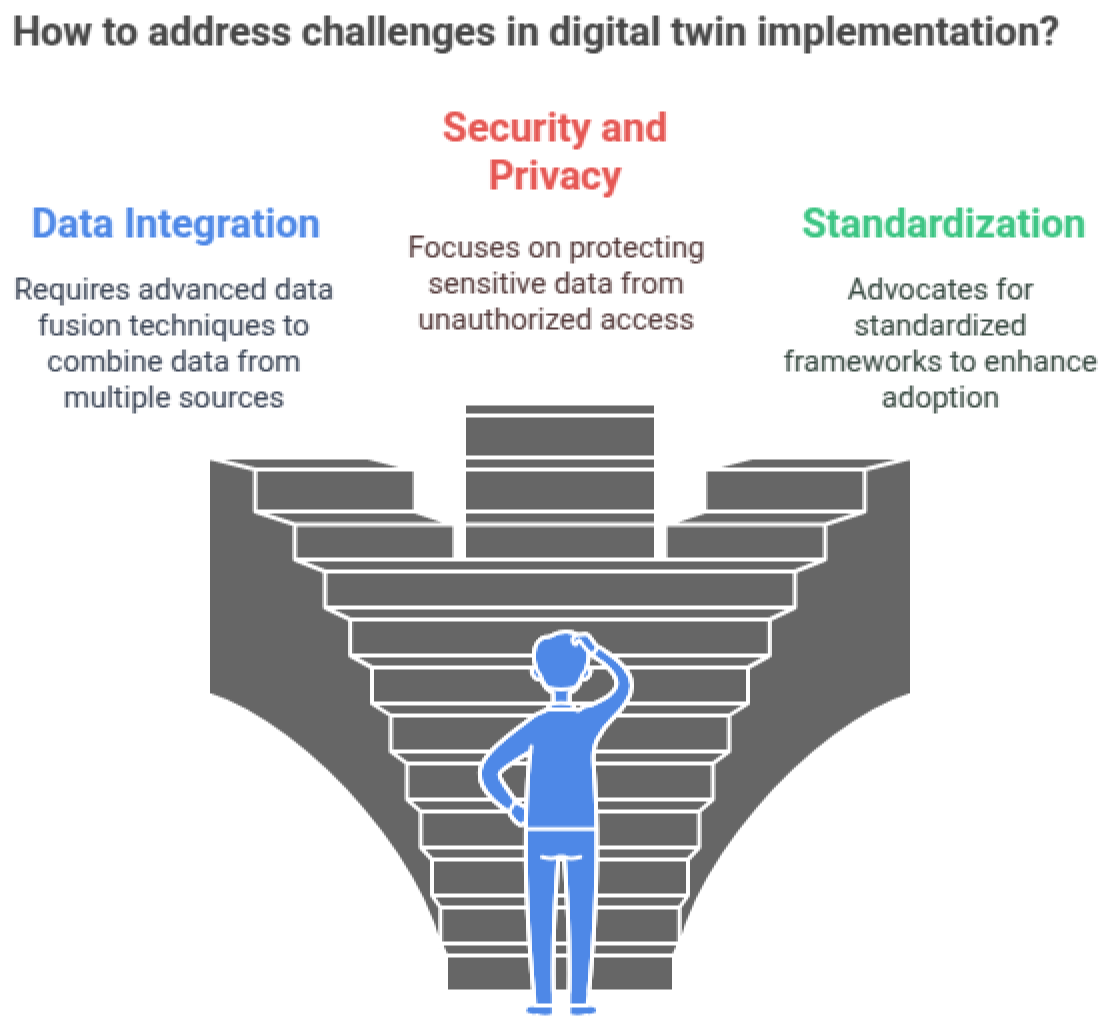

Figure 2.3: Challenges in DT

Security and privacy remain major concerns, especially when dealing with sensitive data in digital twin applications. Unauthorized access or data breaches could lead to significant risks, making it essential to implement end-to-end encryption for both data storage and transmission. Access control mechanisms, such as role-based access control (RBAC) or attribute-based access control (ABAC), can restrict unauthorized users from accessing critical systems or data. Regular security audits and vulnerability assessments should also be conducted to proactively identify and mitigate potential threats. Furthermore, adopting privacy-preserving technologies like differential privacy or federated learning can protect user data while still enabling valuable insights through analytics.

Finally, the lack of standardization across digital twin frameworks poses a barrier to widespread

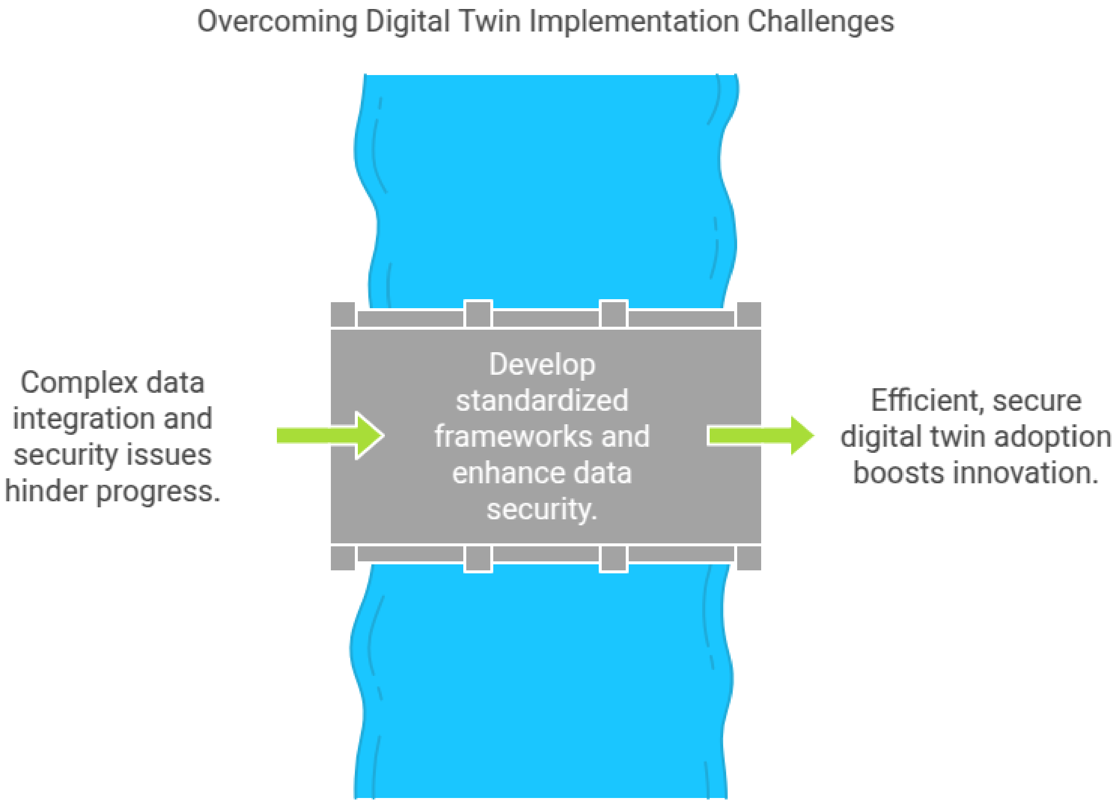

Figure 2.4: Overcoming the Challenges of DT

adoption, as inconsistent approaches can hinder interoperability and scalability. One solution is to engage with industry consortia and standards organizations, such as ISO or IEEE, to develop unified frameworks that facilitate compatibility across different digital twin implementations. Encouraging the use of open-source platforms can also promote collaboration and innovation, allowing developers to build on shared tools and resources. Establishing best practices and guidelines within industries will further ensure consistency in how digital twins are designed, deployed, and maintained. By addressing these challenges through advanced data integration techniques, robust security measures, and the promotion of standardized frameworks, organizations can overcome key barriers and fully leverage the transformative benefits of digital twin technology across various sectors.

2.6 Digital Twins for Proactive Handover

Digital twins are increasingly being applied to wireless networks to address the challenges of network optimization, resource allocation, and proactive handover. By creating a virtual model of the network, digital twins can simulate different scenarios and predict outcomes, enabling better decision-making [6] [7]. Digital twins can play a crucial role in proactive handover by predicting future link blockages and optimizing handover decisions. By simulating different scenarios, digital twins can identify potential issues before they occur, enabling proactive measures to be taken [3] [4].

2.7 Research Gaps, Future Directions & Our Research

The integration of digital twins and proactive handover techniques offers significant potential for improving the performance and reliability of wireless networks. Digital twins provide a powerful tool for network optimization and resource allocation, while proactive handover techniques ensure seamless service continuity in dynamic environments. However, several challenges need to be addressed to fully realize the benefits of these technologies, including data integration, security, and the lack of standardized frameworks. So, we can see there are several research gaps.

Future research should focus on developing unified frameworks for digital twin development and exploring new approaches for maximizing the benefits of digital twin technology in wireless networks. Additionally, the integration of emerging technologies such as 6G communication and edge computing will be crucial for advancing the field [6] [7] [13]. In our research, we aim to develop a comprehensive framework for Digital Twins specifically tailored to enhance proactive handover mechanisms. This framework seeks to harness the full potential of Digital Twins by addressing one of the most critical challenges in proactive handover: ensuring seamless and efficient connectivity transitions.

Chapter 3

Methodology

The methodology for this thesis is designed to develop and evaluate a Digital Twin (DT) framework for wireless network management, with a focus on enhancing network performance, optimizing handover processes, and improving user experience. This approach integrates advanced simulation tools, real-world data, and techniques to create a dynamic virtual replica of a wireless network. The methodology is structured into several key phases, each building upon the previous to ensure a robust and adaptable system.

3.1 Network Topology Modeling

The foundation of the DT framework involves constructing a realistic representation of the wireless network topology.

3.1.1 Creating Custom Scene

Our objective was to create a highly realistic Digital Twin (DT) of wireless networks by leveraging advanced tools such as SIONNA [1] and Ray Tracing to model network topology and environmental interactions with precision. To achieve this goal, we integrated multiple data sources and modelling techniques, utilizing OpenStreetMap [15] for geographical data and OpenCellID [16] for accurate base station locations. The process began by importing a specific geographic area—chosen for its complexity and relevance—into Blender [17], a powerful open-source 3D modeling tool. For this experiment, we selected the area around the iconic **Arc de Triomphe in Paris**, known for its dense urban environment and intricate layout.

Using Blender, we meticulously constructed a detailed and accurate 3D representation of the physical environment, ensuring that all structural elements, such as buildings, roads, and other urban features, were faithfully replicated. This level of detail is critical for capturing the real-

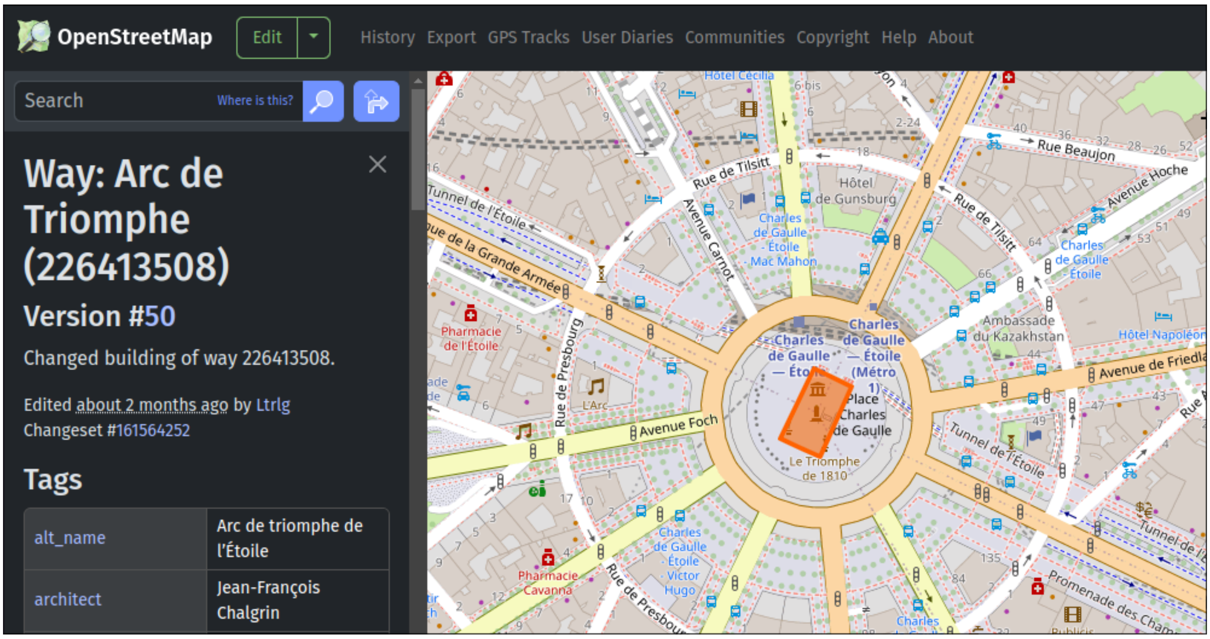

Figure 3.1: Importing a certain area from OpenStreetMap

world dynamics of wireless signal propagation. Next, we assigned appropriate radio materials to various elements within the scene, carefully reflecting their actual electromagnetic propagation characteristics, such as reflection, absorption, and scattering properties. These material assignments are essential for simulating how wireless signals interact with the environment in a realistic manner.

Figure 3.2: Real World

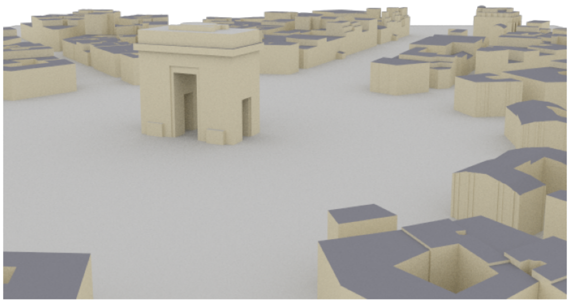

Figure 3.3: Digital Twin

Once the custom scene was fully prepared, it was exported in the Mitsuba File format, a format compatible with Sionna Ray Tracing (RT). The scene was then loaded into Sionna RT for further processing, enabling us to compute detailed signal propagation paths, analyze coverage maps, and evaluate network performance metrics. This comprehensive approach [18] allowed us to create any robust and realistic Digital Twin of any location, providing a solid foundation for conducting experiments and analyzing the impact of environmental factors on wireless

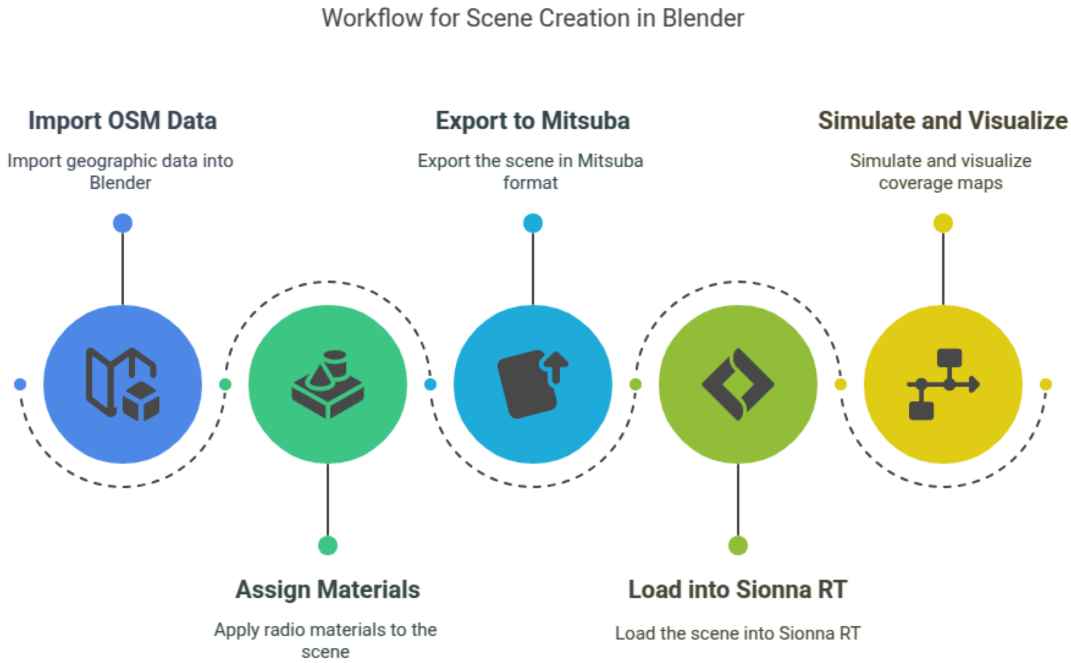

Figure 3.4: Creating any custom scene for the Digital Twin

communication systems. By combining these cutting-edge tools and methodologies, we established a framework capable of supporting advanced simulations, ultimately contributing to more efficient and reliable wireless network management.

3.1.2 3D Environment

To integrate real-world network elements into our model, we leveraged OpenCellID [16], a widely used open database for cell tower geolocation, to obtain precise locations of base stations. Using these accurate coordinates, we incorporated multiple base stations into our simulation scene, ensuring their positions matched real-world deployments. In our case we used three transmitters. This step was critical for achieving a high level of realism in the digital twin (DT) environment. Additionally, we populated the scene with multiple user equipment (UE) units to simulate user mobility and their interactions with the network, capturing dynamic scenarios such as movement between different network zones.

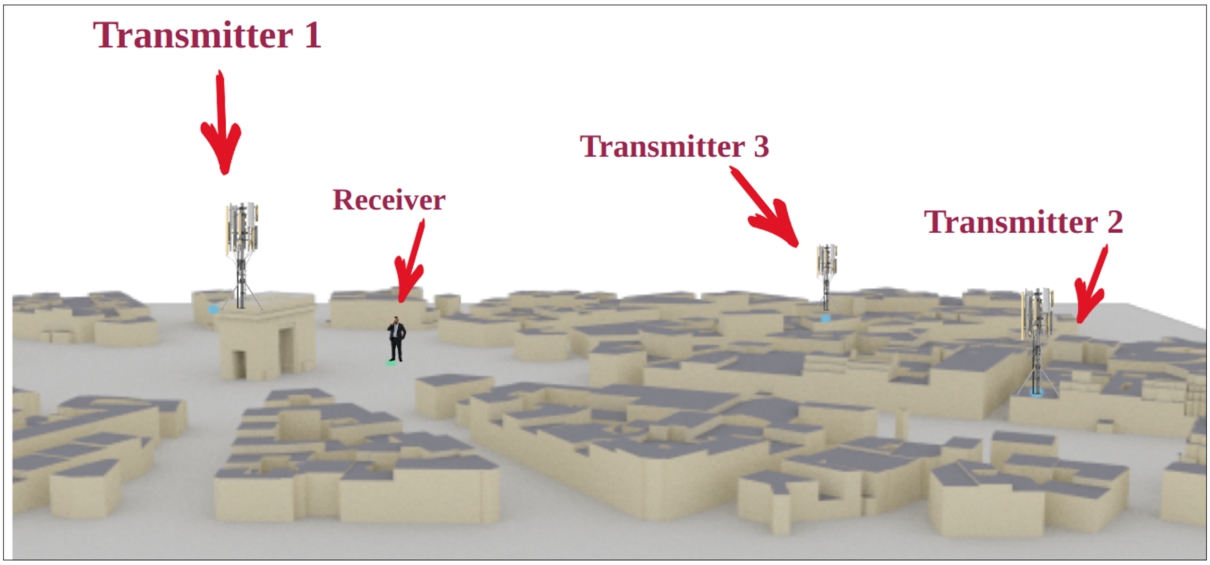

Figure 3.5: 3D Environment with Transmitters & a sample Receiver

3.1.3 RF propagation

Once the base stations and UEs were integrated into the scene, we utilized Sionna Ray Tracing to compute the signal propagation paths from the base stations to the users. This allowed us to analyze how signals behave within the modeled environment, taking into account factors such as reflection, diffraction, scattering, and attenuation caused by physical obstacles like buildings and other structures.

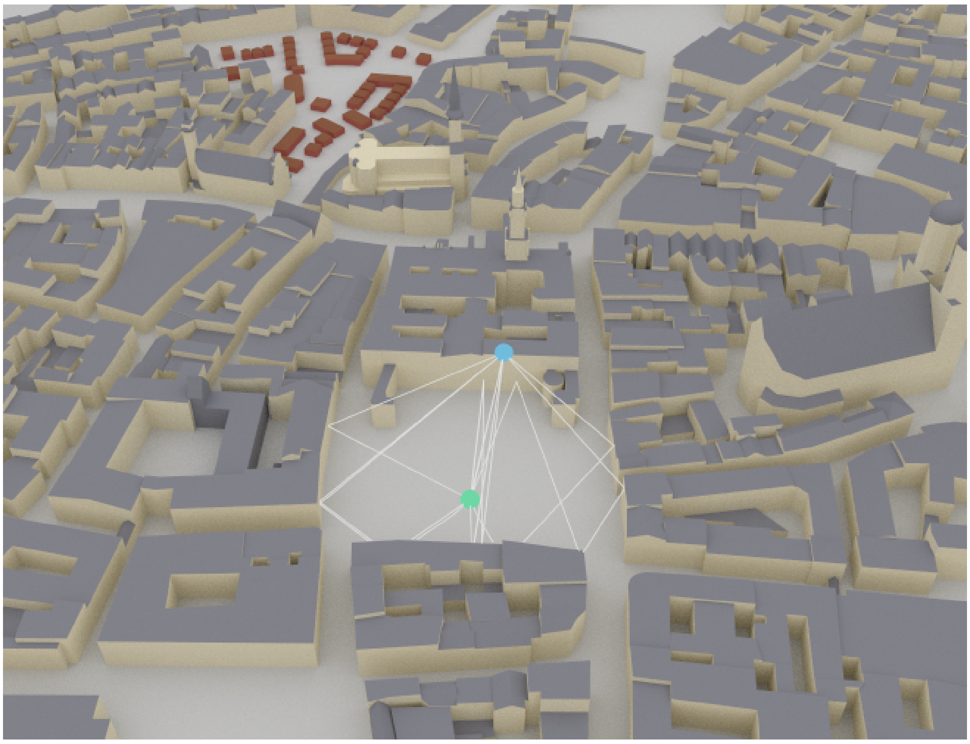

Figure 3.6: Sample Paths Visualization between a Transmitter & a Receiver

3.1.4 Coverage Map

A coverage map [19] in the Sionna Python library is a tool used to visualize and analyze wireless communication performance metrics across a given area. It assigns specific metrics, such as path gain, received signal strength (RSS), or signal-to-interference-plus-noise ratio (SINR), to points on a plane within a simulated environment. These maps provide detailed insights into how wireless signals propagate and interact with the environment, including effects like reflection, diffraction, and attenuation.

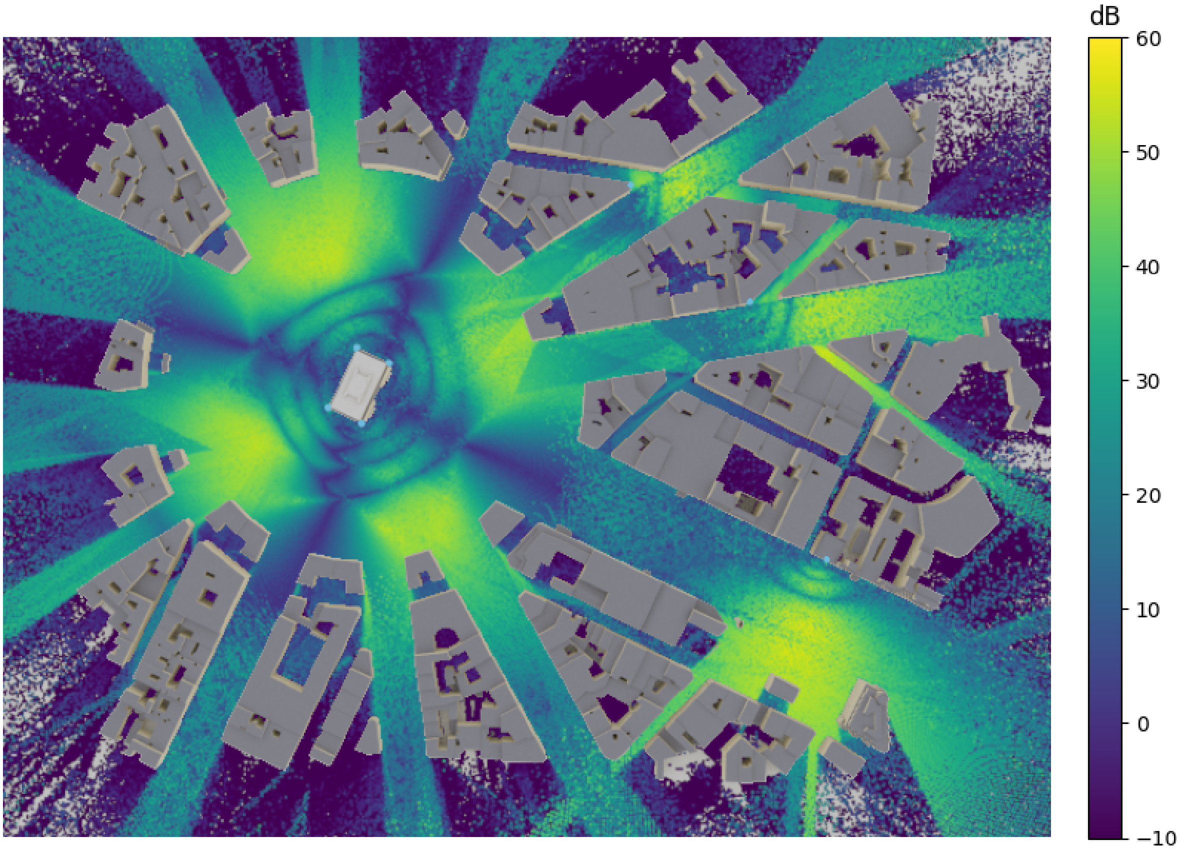

Figure 3.7: Coverage Map ; Metric: SINR

In Sionna, coverage maps are generated using the ray tracing module, which simulates the propagation paths of signals from transmitters to receivers in a 3D environment. The library computes these metrics for every point in the scene, enabling users to evaluate network performance, optimize base station placement, and analyze signal behaviour under various conditions. It helps researchers and engineers understand the quality of wireless communication in different regions of the modelled environment. This process integrates environmental data, such as radio materials and physical obstacles, to produce accurate and realistic simulations.

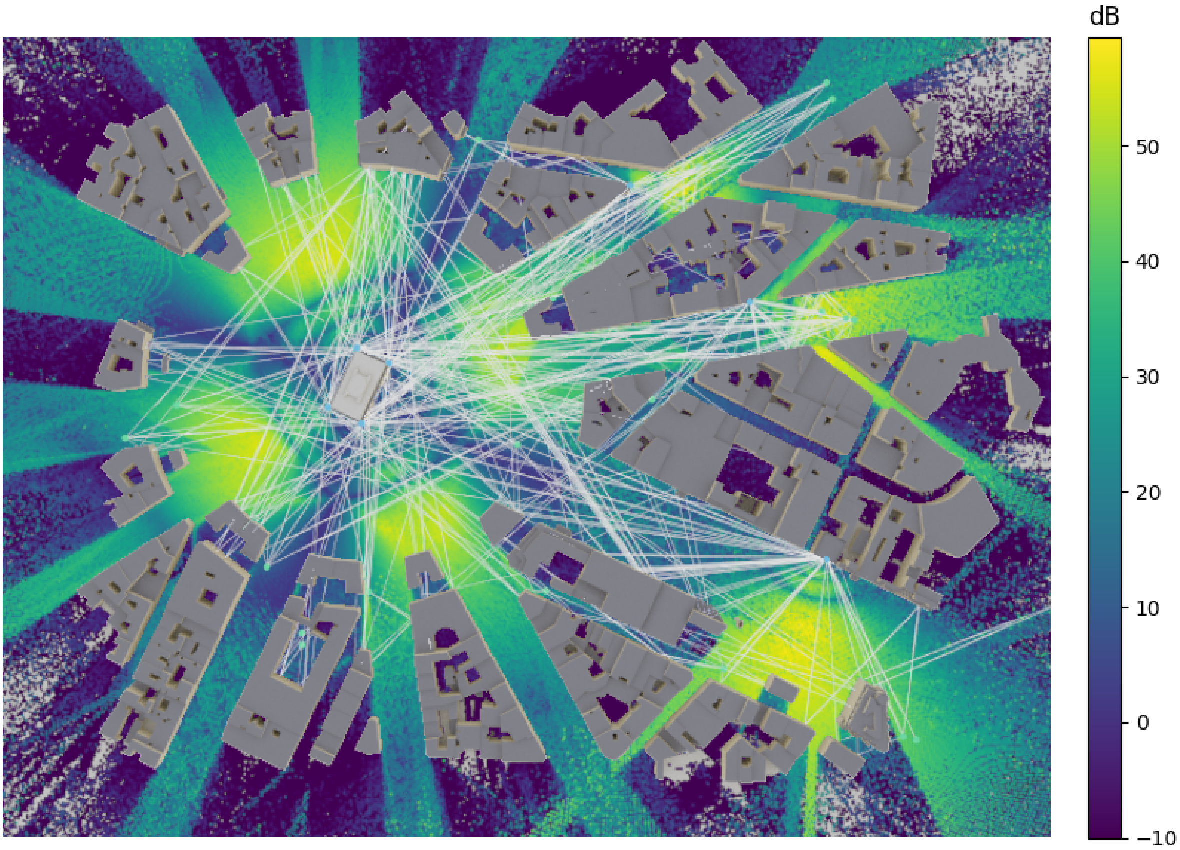

Figure 3.8: Coverage Map with Paths Visualization

This comprehensive and systematic approach—combining precise base station placement, realistic user mobility modelling, and advanced ray tracing techniques—enabled us to create a highly realistic Digital Twin (DT) of the wireless network. By accurately replicating both the physical and operational aspects of the network, this DT serves as a robust foundation for conducting further experiments and analyses. It not only enhances our ability to evaluate current network performance but also opens up opportunities for exploring advanced optimization strategies, such as proactive handover mechanisms and beamforming techniques, ultimately contributing to the development of more efficient and reliable wireless communication systems.

3.2 User Mobility

To accurately simulate real-world conditions, we integrated *user mobility* into our model, ensuring that the simulation reflects the dynamic behaviour of mobile users in wireless networks. To achieve this, both the transmitters and receivers were equipped with multiple antennas, enabling advanced signal processing techniques such as beamforming and spatial multiplexing. For modelling the movement patterns of users, we employed the **Gauss-Markov User Mo-**

bility Model [20], which is widely recognized for its ability to generate realistic and smooth trajectories by balancing randomness and predictability in user motion.

The value of speed and direction at the nth instance is calculated using the following formulas. The new speed and new direction can be easily calculated using the formulae:

$$\text{New Speed:} \quad s_n = \alpha s_{n-1} + (1-\alpha)\overline{s} + \sqrt{(1-\alpha^2)}s_{x_{n-1}}$$

$$\text{New Direction:} \quad d_n = \alpha d_{n-1} + (1-\alpha)\overline{d} + \sqrt{(1-\alpha^2)}d_{x_{n-1}}$$

Where:

- α: Index of Randomness
- $\overline{s}$: Average Speed
- $\overline{d}$: Average Direction
- $s_{x_{n-1}}$, $d_{x_{n-1}}$: Random Variables of Gaussian Distribution

To implement user mobility, we utilized **Sim2Net** [21], an open-source discrete-event simulator specifically designed for mobile ad hoc networks (MANETs) and written in Python. Sim2Net provides a flexible framework for simulating networks with a specified number of nodes, where each node moves according to a chosen mobility model, such as the Gauss-Markov model. This allows us to replicate realistic user movement patterns, including transitions between different network zones, which are critical for evaluating handover mechanisms and network performance under varying conditions.

In addition to modeling mobility, Sim2Net supports the execution of custom applications and facilitates communication exclusively through wireless links by sending application messages. This feature enables us to simulate real-world scenarios where users interact with the network dynamically, exchanging data while moving across the coverage area. By incorporating these tools and techniques, we ensured that our model captures the complexities of user mobility, providing valuable insights into how mobility impacts network performance, signal quality, and the overall user experience. This approach lays the groundwork for analyzing and optimizing critical network processes, such as handovers, in highly dynamic environments.

3.3 Path loss and BER calculation

In this Ray tracing scenario Sionna set up an ideal gain controller to fully compensate for the path loss. This is not practical as the path loss in real scenarios increases with distance. So, we introduced a power control scheme in Sionna to replicate the practical scenario.Different from Sionna's ideal gain control mechanism, we considered that a Base station(BS) can increase or decrease power up to a certain limit, and if the limit is not maintained, BS simply omits data transmission. Later happens when a user moves far from BS, and thus, it needs to be associated with another nearby BS to achieve good BER. Before We found the propagation paths among transmitters and receivers through ray tracing. Then, we calculated CIR from propagation paths.The channel path loss (in linear scale), including the effect of antenna gains, is calculated by [22]:

$$PL_{\text{MMW}} = \left(\frac{\lambda^2\beta}{8\pi\eta_0}\right)\left|\sum_{i=1}^{N_p} E_{\theta,i}g_\theta(\theta_i,\phi_i) + E_{\phi,i}g_\phi(\theta_i,\phi_i)\right|^2 \quad (1)$$

where $E_{\theta,i}$ and $E_{\phi,i}$ are the so-called theta and phi components of the electric field of the i^{th} path at the receiver point, while θ_i and ϕ_i are the parameters related to the direction of the arrival ray. $g_\theta(\theta_i,\phi_i)$ indicates the direction of arrival angles including the elevation and azimuth angles. Here, λ is the wavelength, η_0 is the impedance of RF space, and β is the overlap of the frequency spectrum of the transmitted waveform and the frequency sensitivity of the receiver. Here the ideal value of β is used, which is 1. BS implements the following channel inversion policy:

$$P = \begin{cases} \text{constant}/|h|^2 & \text{; if } d \leq d_0 \text{ (a threshold distance) or } |h|^2 \geq h_{\text{th}} \\ 0 & \text{; otherwise} \end{cases}$$

Using this P, we have to calculate the received SNR of the user. The SNR can be calculated using the formula:

$$SNR = \frac{P|h|^2}{(\text{Noise}\cdot BW)}$$

where $|h|^2$ is the amplitude of the ray tracing channel. We can consider the **equation (1)** for calculating $|h|^2$.

Then, we have to simulate BER for this SNR while using Sionna's standard BER calculation process.

3.4 Handover, Observed Issues & Our Proposed Solution

3.4.1 Initial Handover Procedure

Initially, the distance from each transmitter to the receiver is measured, and the transmitter with the shortest distance is selected for transmission. Transmission begins with this nearest transmitter. However, as the receiver moves according to the Gauss-Markov mobility model, the Bit Error Rate (BER) is continuously monitored. If the BER exceeds the predefined threshold value of **0.10**, the current transmission is halted, and a handover process is initiated.

During the handover, distances from the receiver to the remaining transmitters are reassessed. The receiver is then connected to the nearest transmitter, and transmission resumes with the BER being monitored anew. If the BER for the new transmitter also surpasses the threshold, a further handover occurs. This process continues iteratively, with the receiver switching to the next nearest transmitter each time the BER exceeds the threshold.

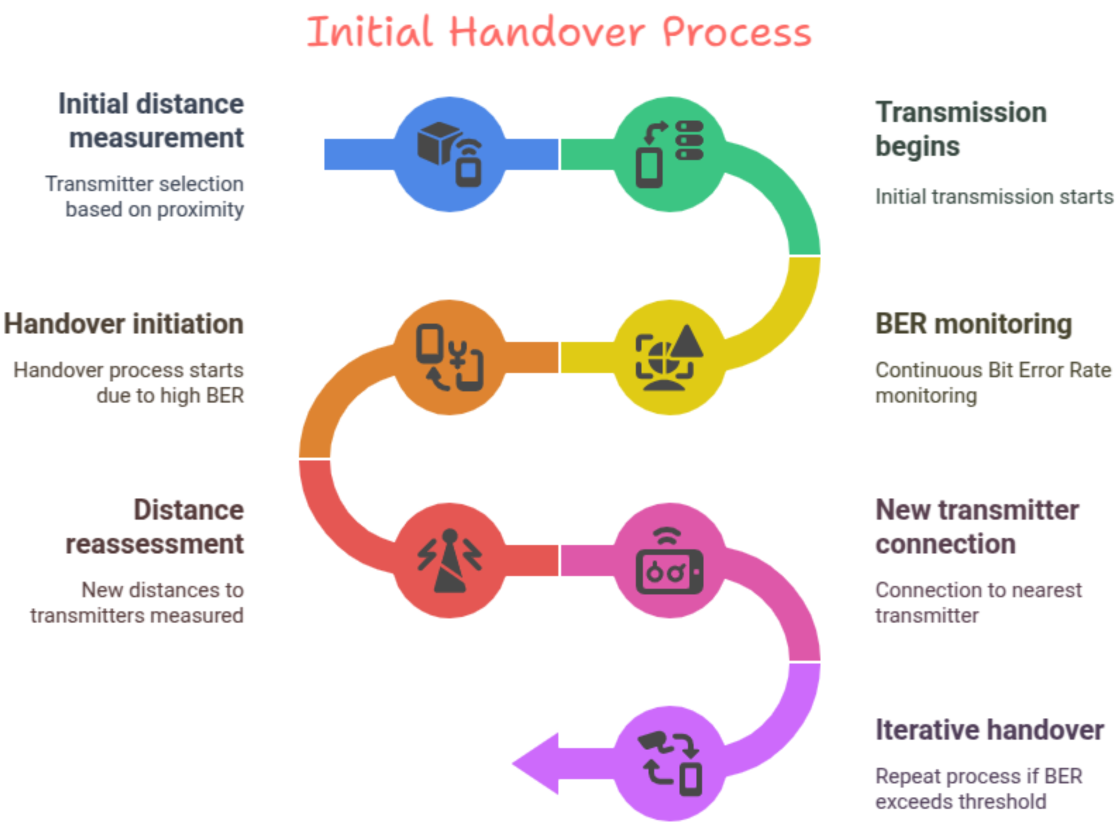

Figure 3.9: Initial Distance-Based Handover Procedure

3.4.2 Observed Issues

The simulation is conducted over a period of **10 seconds**. An issue observed during the simulation is that, despite transitioning through multiple transmitters, the BER may remain above the threshold value. Specifically, the transmission begins with *Transmitter A*, the closest to the receiver. If the BER with *Transmitter A* exceeds the threshold, a handover to *Transmitter B*, the next closest transmitter, occurs. Despite this handover, if the BER with *Transmitter B* also remains above the threshold, a further handover to *Transmitter C* is performed. If, after exhausting all available transmitters, the BER still exceeds the threshold, the value at this point is recorded, indicating that the BER consistently surpasses the desired threshold even after all possible handovers.

3.4.3 Proposed Solution: DT Based Proactive Handover

To address these limitations, an RF environment-aware DT-enabled handover strategy is developed. This method leverages **real-time SINR data from coverage maps** to select the BS offering the best signal quality, considering environmental factors like path loss and shadowing. A custom function *(get_sinr_at_position)* is implemented to compute SINR values at user positions, enabling dynamic handover decisions. Comparative analysis between DT-enabled and distance-based handovers is conducted, with results indicating improved SINR and reduced BER using the DT approach.

3.5 Design Considerations

The design of the Digital Twin (DT) framework for wireless networks must address public health, safety, cultural, societal, and environmental factors to ensure ethical and sustainable implementation.

3.5.1 Public Health and Safety Considerations

The DT framework must comply with EMF exposure guidelines (e.g. International Commission on Non-Ionizing Radiation Protection (ICNIRP)) by optimizing BS power and placement via RF propagation analysis. Data privacy is ensured through encryption and anonymization, protecting user data collected during real-time monitoring. Reliable handovers, enabled by proactive strategies, prevent service disruptions, supporting critical applications like emergency communications.

3.5.2 Cultural, Societal, and Environmental considerations

The framework should adapt to diverse mobility patterns across cultures, ensuring equitable access to connectivity, especially in underserved areas. Energy efficiency is prioritized through adaptive power control, minimizing the ecological footprint of cloud computing and BS operations. The DT can also support disaster resilience by reconfiguring networks during emergencies, enhancing societal benefits.

Chapter 4

Results and Discussion

We will explore **three distinct cases** to highlight the differences in handover approaches and their outcomes.

In the *first case*, we will examine the traditional **distance-based handover** mechanism, analyzing its limitations and the challenges it presents. Specifically, we will investigate issues such as suboptimal performance caused by environmental obstructions, high path loss, and shadowing effects that can lead to elevated Bit Error Rates (BER), even after multiple handovers.

In the *second case*, we will harness the capabilities of Digital Twin (DT) technology to implement a **proactive DT-based handover** strategy. This approach leverages real-time environmental awareness and predictive modeling to optimize handover decisions, ensuring improved signal quality and reduced BER by considering factors such as obstacles, RF propagation characteristics, and user mobility patterns.

Finally, in the *third case*, we will conduct a comparative analysis between DT-based handover and the conventional distance-based handover. Through this comparison, we aim to demonstrate that the DT-enabled approach significantly outperforms the traditional method, particularly in challenging environments where distance alone is insufficient for determining the optimal base station. By integrating environmental awareness and proactive strategies, the DT-based handover achieves superior connectivity and overall network performance.

This structured evaluation underscores the transformative potential of Digital Twin technology in revolutionizing wireless network management.

4.1 Case 1: Typical Distance-Based Handover

Initially, the distance from each transmitter to the receiver is measured, and the transmitter with the shortest distance is selected for transmission.

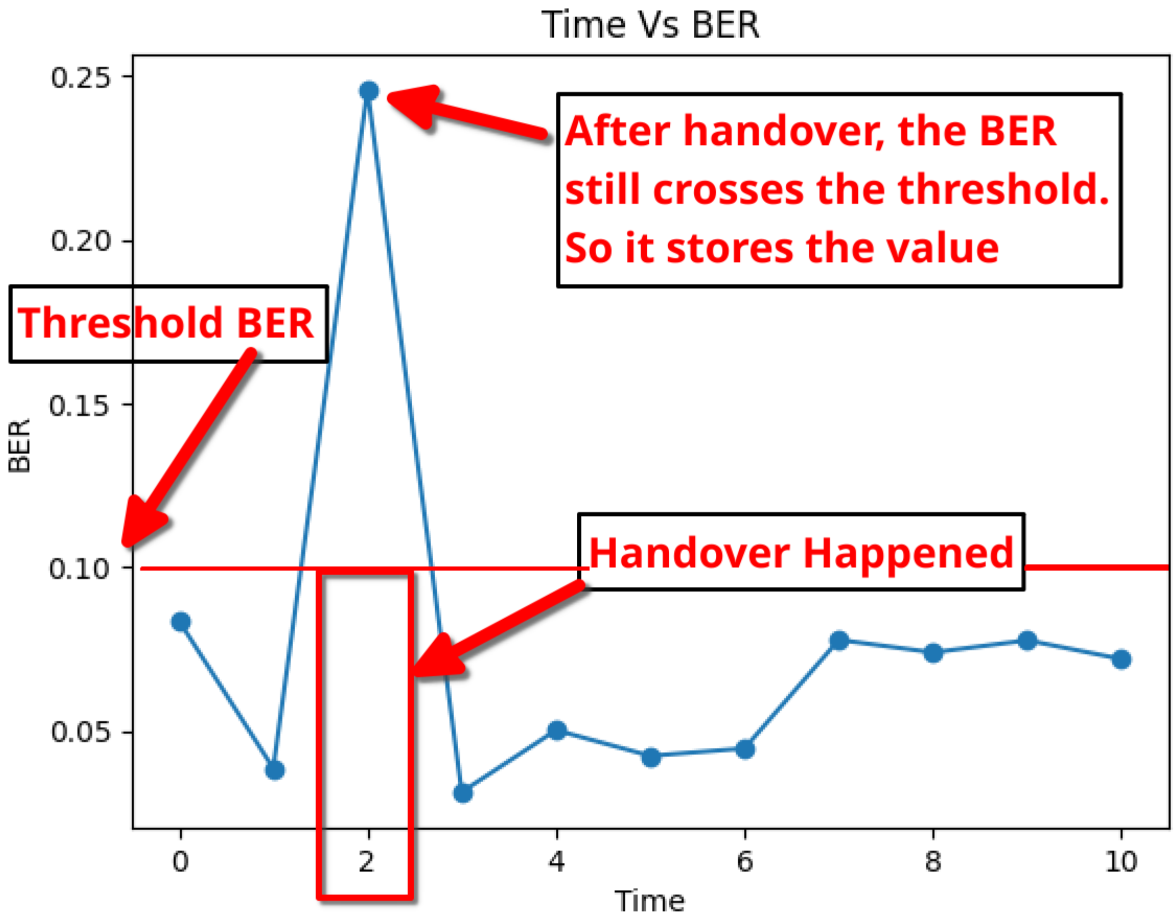

Figure 4.1: Time Vs BER Plot in Distance-Based Handover

Transmission begins with this nearest transmitter. However, as the receiver moves according to the *Gauss-Markov mobility model*, the Bit Error Rate (BER) is continuously monitored. If the BER exceeds the **predefined threshold value of 0.10**, the current transmission is halted, and a handover process is initiated.

During the handover, distances from the receiver to the remaining transmitters are reassessed. The receiver is then connected to the next nearest transmitter, and transmission resumes with the BER being monitored anew. If the BER for the new transmitter also surpasses the threshold, a further handover occurs. This process continues iteratively, with the receiver switching to the next nearest transmitter each time the BER exceeds the threshold.

4.1.1 Challenges

The simulation runs for **10 seconds**. During this time, a problem was noticed: even after switching between multiple transmitters, the Bit Error Rate (BER) —which measures how many errors occur in data transmission—stayed too high. Here's what happens:

- The process starts with **Transmitter A** , which is the closest to the receiver.
- If the BER with Transmitter A is too high (above the acceptable limit), the system switches to **Transmitter B**, the next nearest transmitter.
- If the BER is still too high with Transmitter B, the system tries again by switching to **Transmitter C**.
- This switching continues until all available transmitters have been tried.

However, even after trying all the transmitters, if the BER remains above the acceptable limit, the system records this high value. This shows that the BER stays too high no matter how many times the system switches transmitters.

This highlights a key issue:

Simply switching to the nearest transmitter doesn't always solve the problem, especially if there are obstacles like buildings causing signal loss or interference. Typically, distance-based handover is used because it's straightforward. However, this method lacks awareness of the radio frequency (RF) environment. As a result, it might connect a user to a nearby base station, which may have high path loss due to buildings. In contrast, the digital twin approach avoids this issue by considering the RF environment, leading to better performance.

4.2 Case 2: Proactive Digital Twin Based Handover

First, we set up our custom scene in our DT framework. For our experiment, we consider **three transmitters(Tx)** and **one receiver(Rx)**. The receiver will have mobility and it's modeled using Gauss-Markov Mobility Model.

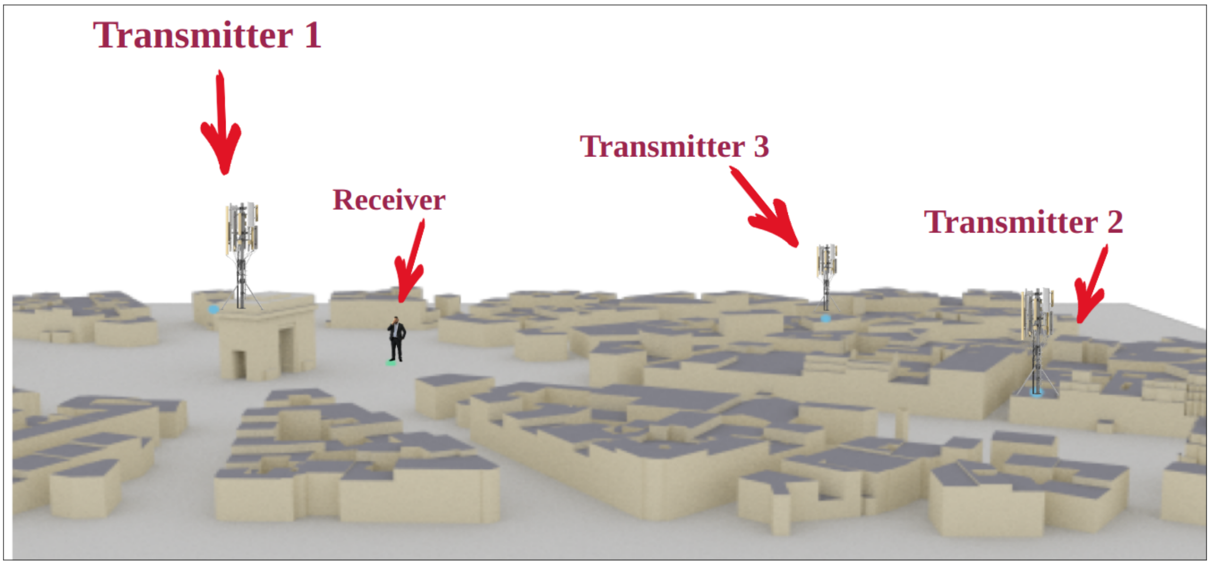

Figure 4.2: Set Up in the Digital Twin Framework

Now, we will generate **Coverage Maps**.

A coverage map assigns a metric, such as path gain, received signal strength (RSS), or signal-to-interference-plus-noise ratio (SINR), for a specific transmitter to every point on a plane. In other words, a given transmitter associates every point on a surface with the channel gain, RSS, or SINR, that a receiver with a specific orientation would observe at this point.

As soon as there are multiple transmitters in a scene, we can either visualize a metric for the specific transmitter or visualize the maximum metric across all transmitters. In this experiment, we choose **SINR metric** for our coverage maps.

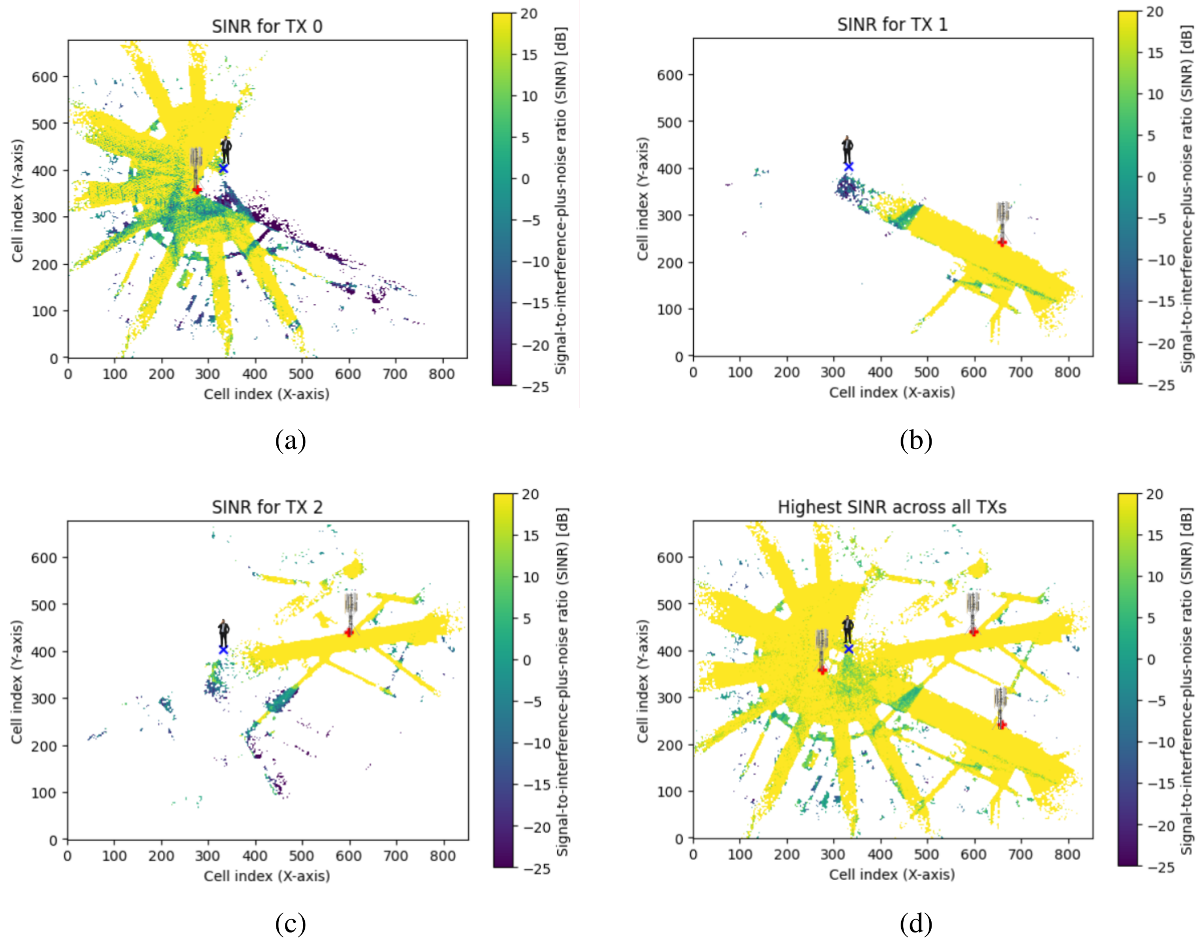

Figure 4.3: SINR Coverage Maps (a) SINR for First Transmitter, (b) SINR for Second Transmitter, (c) SINR for Third Transmitter, (d) SINR across all the transmitters

We can see from the fig. 4.3 of SINR Coverage Maps for the transmitters. The figures: fig. 4.3a, fig. 4.3b, fig. 4.3c illustrate the Signal-to-Interference-plus-Noise Ratio (SINR) for a specific transmitter. The SINR is represented in decibels (dB) and visualized using a color scheme that transitions from yellow to other colors, indicating high to low SINR values.

- **Yellow:** Represents the highest SINR values, indicating strong signal quality.
- **Other Colors (Green, Blue, Purple):** Represent progressively lower SINR values, with purple indicating the lowest SINR values.
- **Transmitter ('+') :** The **'+'** symbol indicates the location of the transmitter.
- **Receiver ('x')** : The **'x'** marks the location of the receiver.

The fig. 4.3d represents SINR for all the transmitters, present in the scene across the area.

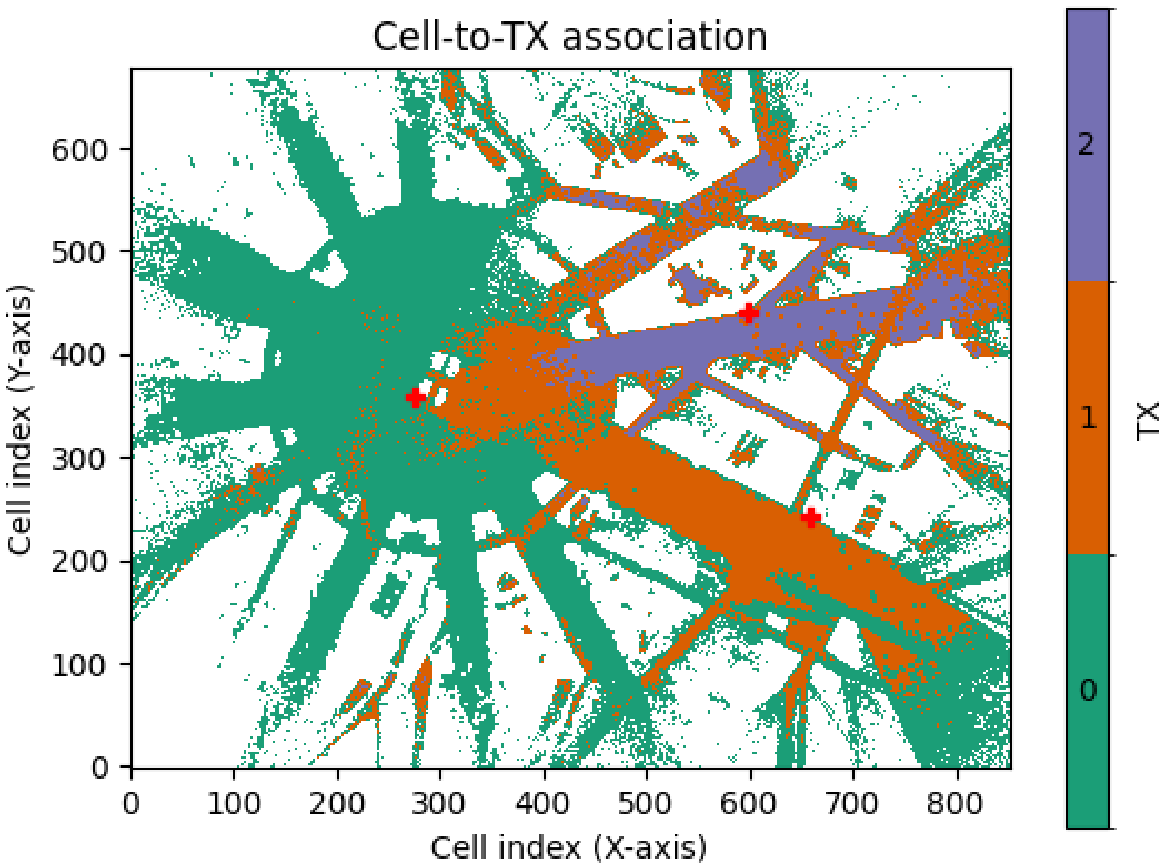

Figure 4.4: Transmitters Coverage

From fig. 4.4, we can also investigate which regions of a coverage map are "covered" by each transmitter, i.e., where a transmitter provides the strongest metric. This helps us to decide effectively for the proactive handover, which lacks in distance-based handover.

4.2.1 DT-Based Proactive Handover Procedure

The user (receiver) is moving around. When the user reaches a specific location, we calculate the SINR (Signal-to-Interference-plus-Noise Ratio) at that position. In simple terms, we can create coverage maps for all transmitters in the area and then calculate the SINR for each transmitter's map. The user will try to switch (handover) to another transmitter with a better signal, i.e., beginning transmission with the best transmitter (Providing best SINR).

We will provide **two examples** to support our claim.

Example 1

In a particular position, we first calculate the SINR values at that position for all the transmitters.

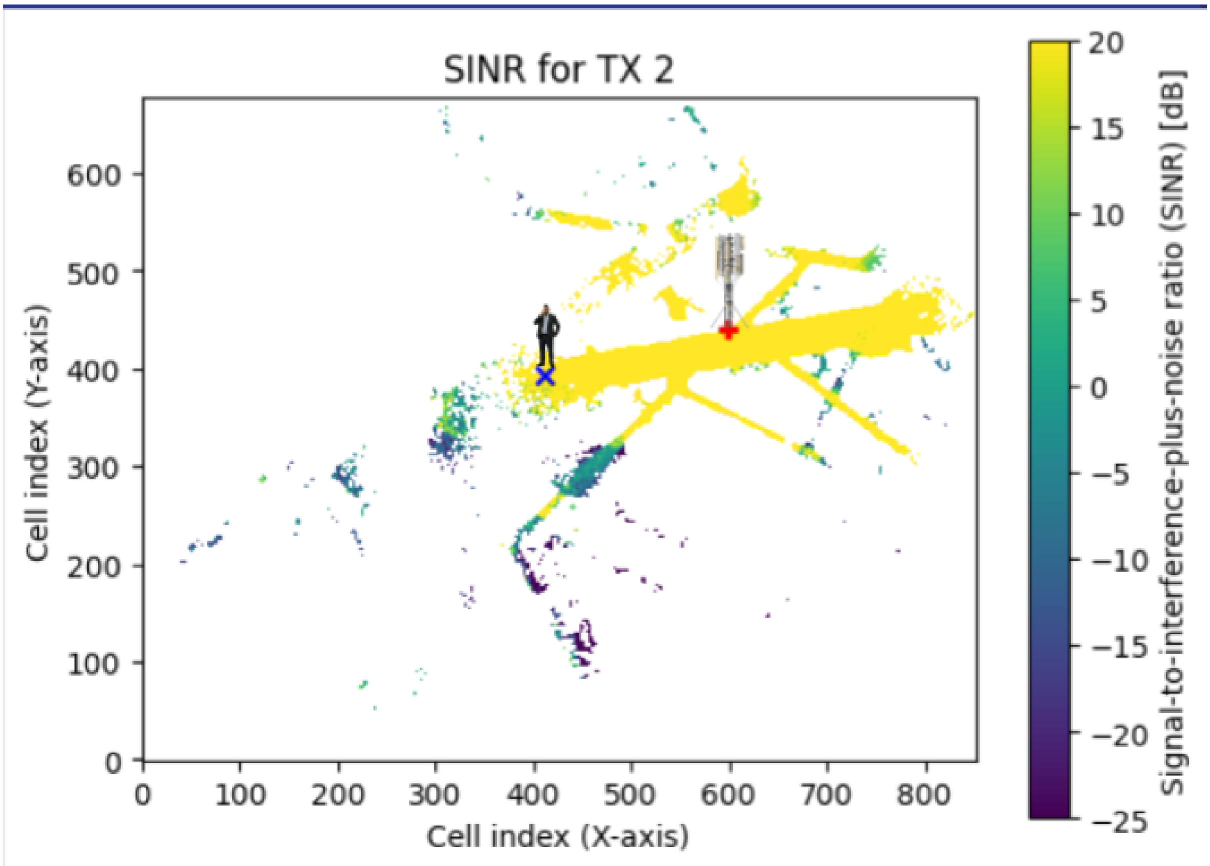

Figure 4.5: Best Transmitter Providing Best SINR

On that particular position, the SINR values are:

Table 4.1: SINR Values for Different Transmitters, Example 1

Transmitters	SINR Value
Tx 0	-inf
Tx 1	-inf
Tx 2	**61.64 dB**

As we can see, the best SINR value provided is **61.64 dB**, which is from **Tx 2**. So the transmission will be handed over to Tx 2

Why the other transmitters are not the best to start transmission?

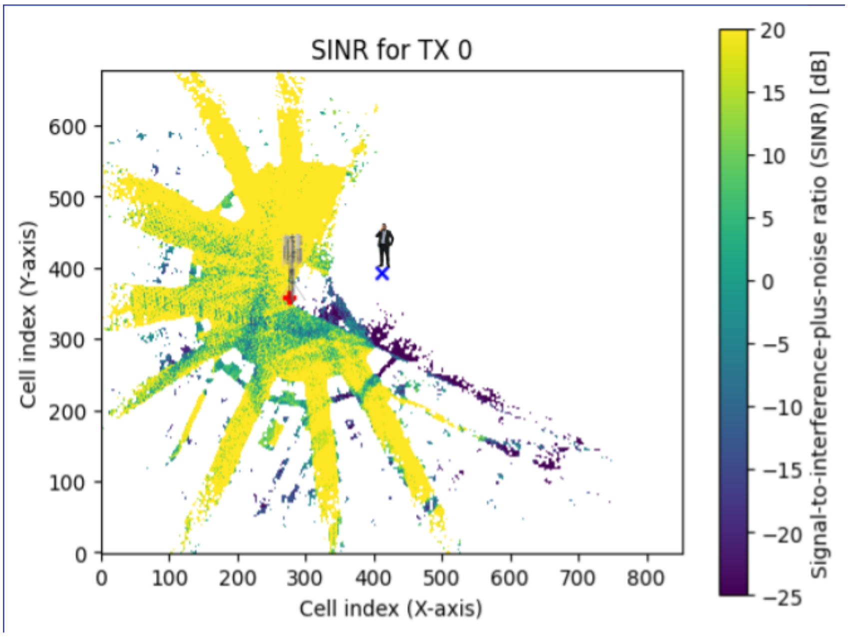

Figure 4.6: Why Tx 0 is not the best transmitter

As we can see, the receiver is not even in the coverage area of the Tx 0. As a result, we get -inf (very poor SINR) on that position from Tx 0.

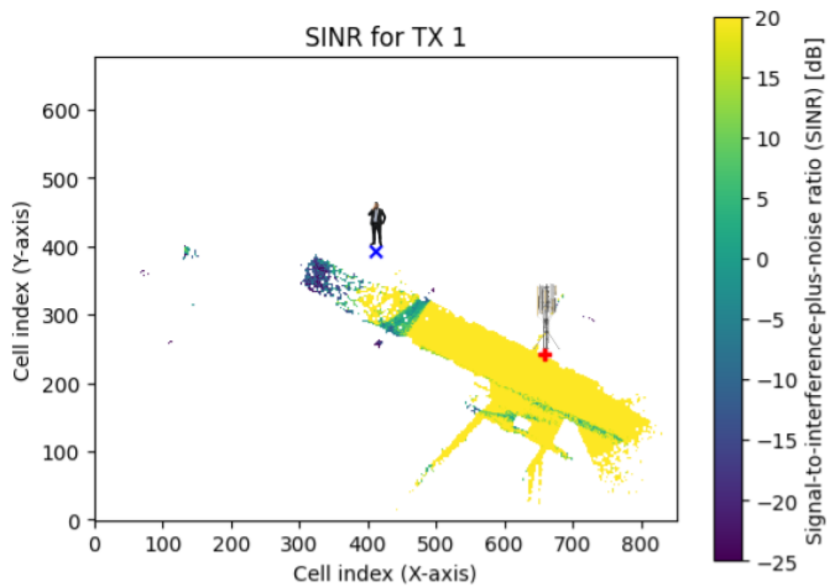

Figure 4.7: Why Tx 1 is not the best transmitter

Similarly, the receiver is not even in the coverage area of the Tx 1. As a result, we get -inf (very poor SINR) on that position from Tx 1.

Example 2

In another particular position, we first calculate the SINR values at that position for all the transmitters.

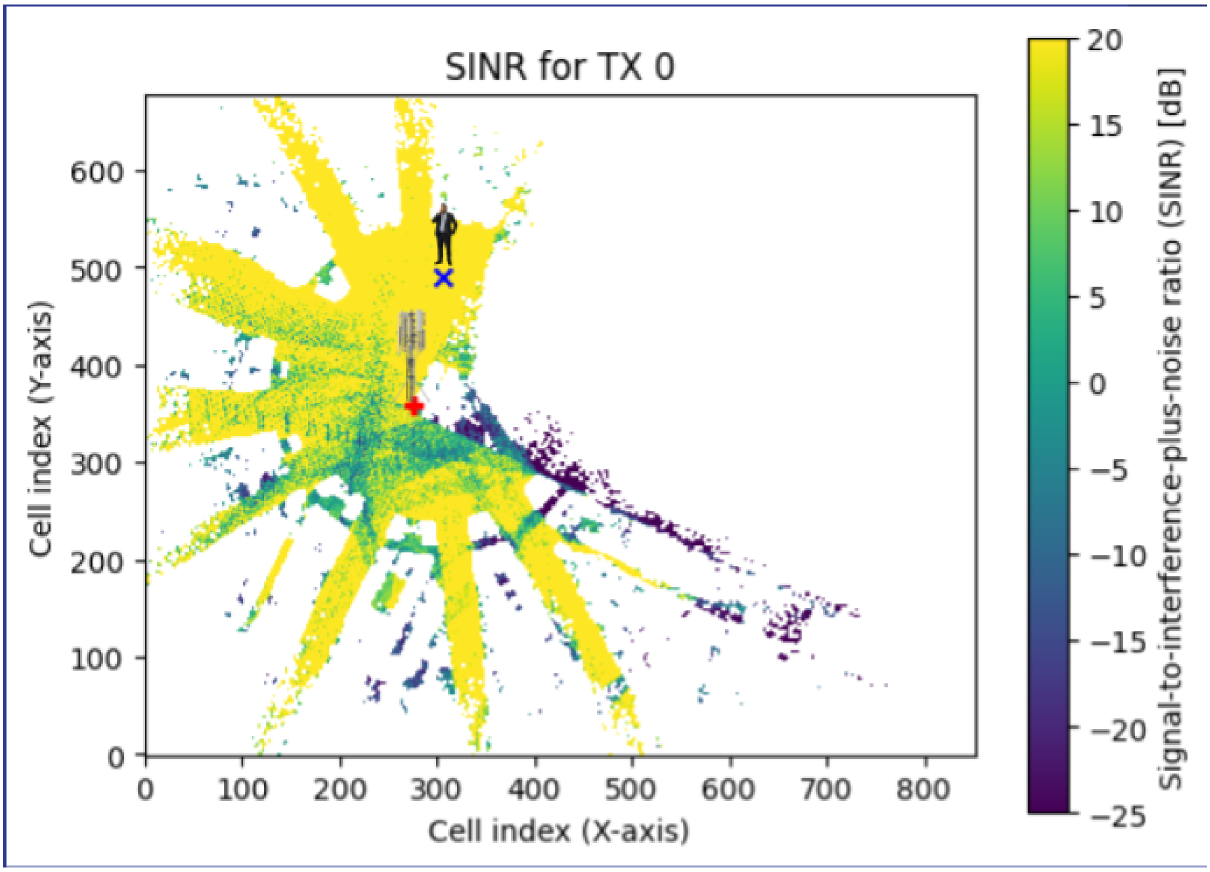

Figure 4.8: Best Transmitter Providing Best SINR

On that particular position, the SINR values are:

Table 4.2: SINR Values for Different Transmitters, Example 2

Transmitters	SINR Value
Tx 0	**27.68 dB**
Tx 1	-27.7 dB
Tx 2	-51.5 dB

As we can see, the best SINR value provided is **27.68 dB**, which is from Tx 0. So, the transmission will be handed over to Tx 0.

Why the other transmitters are not the best to start transmission?

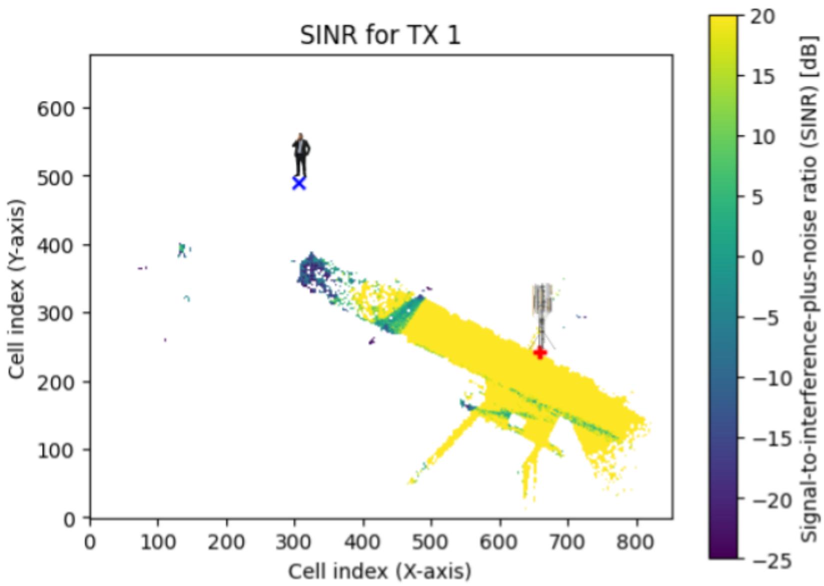

Figure 4.9: Why Tx 1 is not the best transmitter

As we can see, the receiver is not even in the coverage area of the Tx 1. As a result, we get **-27.7 dB** (very poor SINR) on that position from Tx 1.

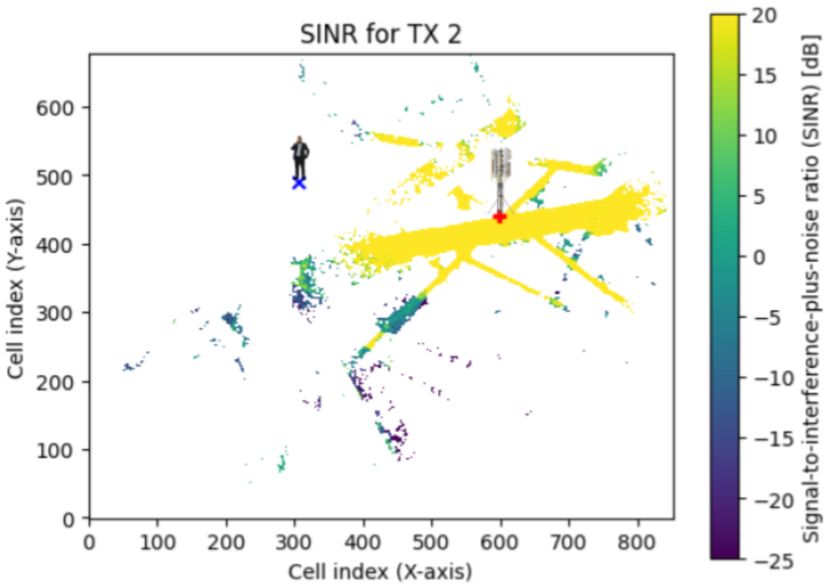

Figure 4.10: Why Tx 2 is not the best transmitter

Similarly, the receiver is not even in the coverage area of the Tx 2. As a result, we get **-51.5 dB** (very poor SINR) on that position from Tx 2.

4.3 Case 3: DT vs Distance Based Handover

The primary objective of this case is to conduct a comprehensive comparison between the performance of a **Digital Twin (DT)-based handover mechanism** and the traditional **distance-based handover** approach . The **distance-based method** primarily depends on the physical proximity between the user and base stations to determine when a handover should occur. While this method is straightforward, it often fails to consider critical environmental factors such as obstacles, signal interference, and other real-world conditions. As observed in **Case 1** , these limitations can lead to suboptimal handovers, resulting in poor network performance and user dissatisfaction.

In contrast, the **Digital Twin-based handover** employs advanced modeling and simulation techniques to predict and optimize handover decisions. By integrating real-time data and incorporating environmental awareness, the DT can more accurately assess the quality of potential connections. This enhanced accuracy ensures that handovers are made based on the most reliable and efficient options available, thereby significantly improving network performance and user experience. The benefits of this approach were clearly demonstrated in **Case 2** , where the DT-based handover resulted in smoother transitions between base stations and reduced instances of connectivity disruptions.

This case aims to demonstrate that the DT-based handover outperforms the distance-based approach and ensuring smoother transitions between base stations. The DT-based handover not only enhances the overall quality of service but also contributes to building more resilient and adaptable wireless networks. This comparative study underscores the transformative potential of Digital Twin technology in revolutionizing network management and optimizing user experiences in dynamic communication environments.

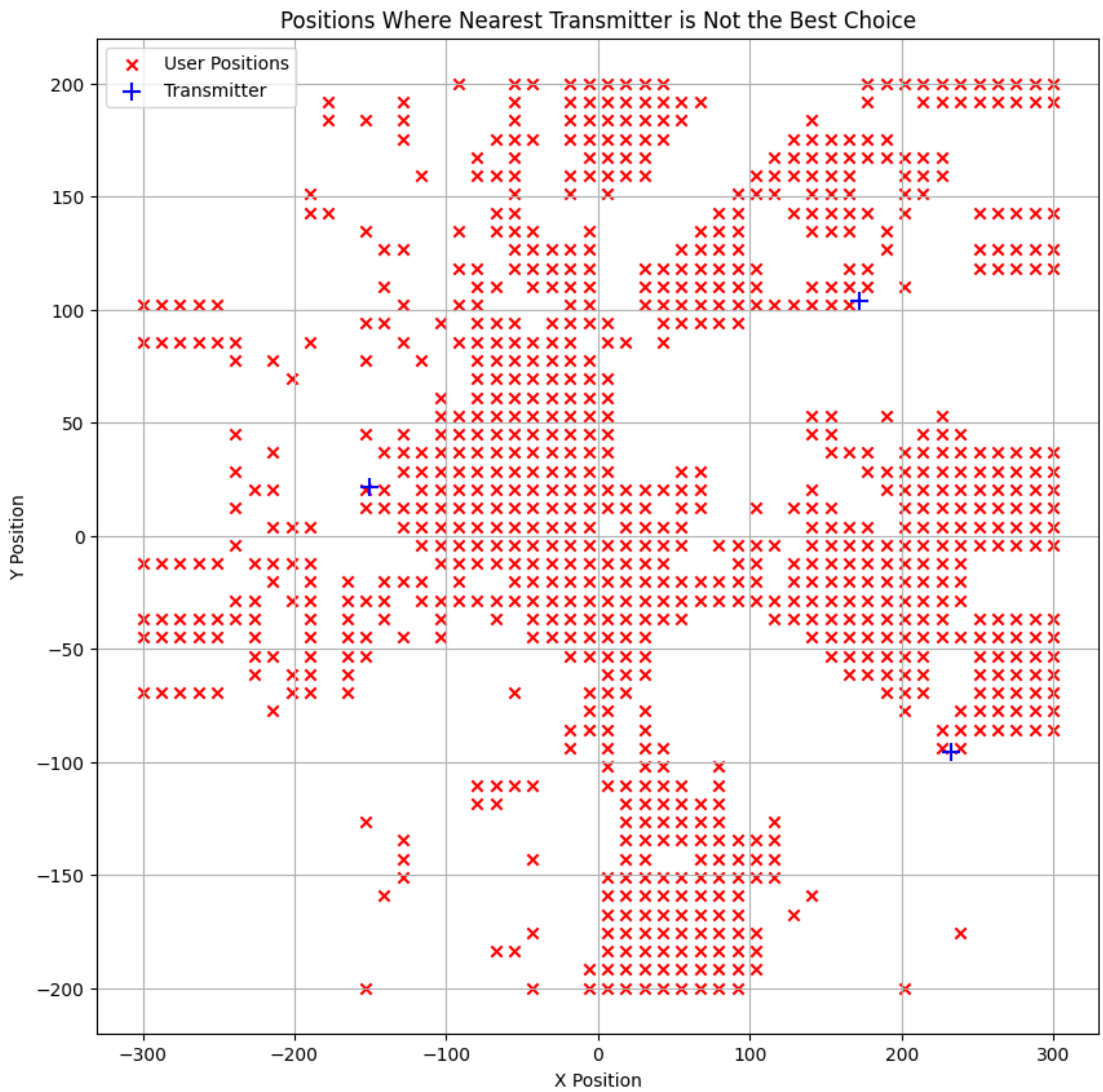

Figure 4.11: User Positions where the nearest transmitter is not the best choice

From the coverage map of our custom scene, we calculated the best positions for users. In fig. 4.11, **'X'** marks where users are located, and **'+'** shows the transmitters. This means that if a user is in any of the **'X'** positions, the nearest transmitter might not be the best choice.

For example, if we measure the SINR at these **'X'** positions, we may find that the closest transmitter gives poor SINR, while a farther one provides better results. This could happen because buildings or other obstacles block the signal from the nearest transmitter.

If we only rely on distance-based handover, which always chooses the nearest transmitter, it can lead to bad results. The system would pick a transmitter with a weak signal instead of a better one that's slightly farther away.

We will now see an example.

We randomly chose a **'X'** position and put a receiver on that position. Then, we calculated the SINR.

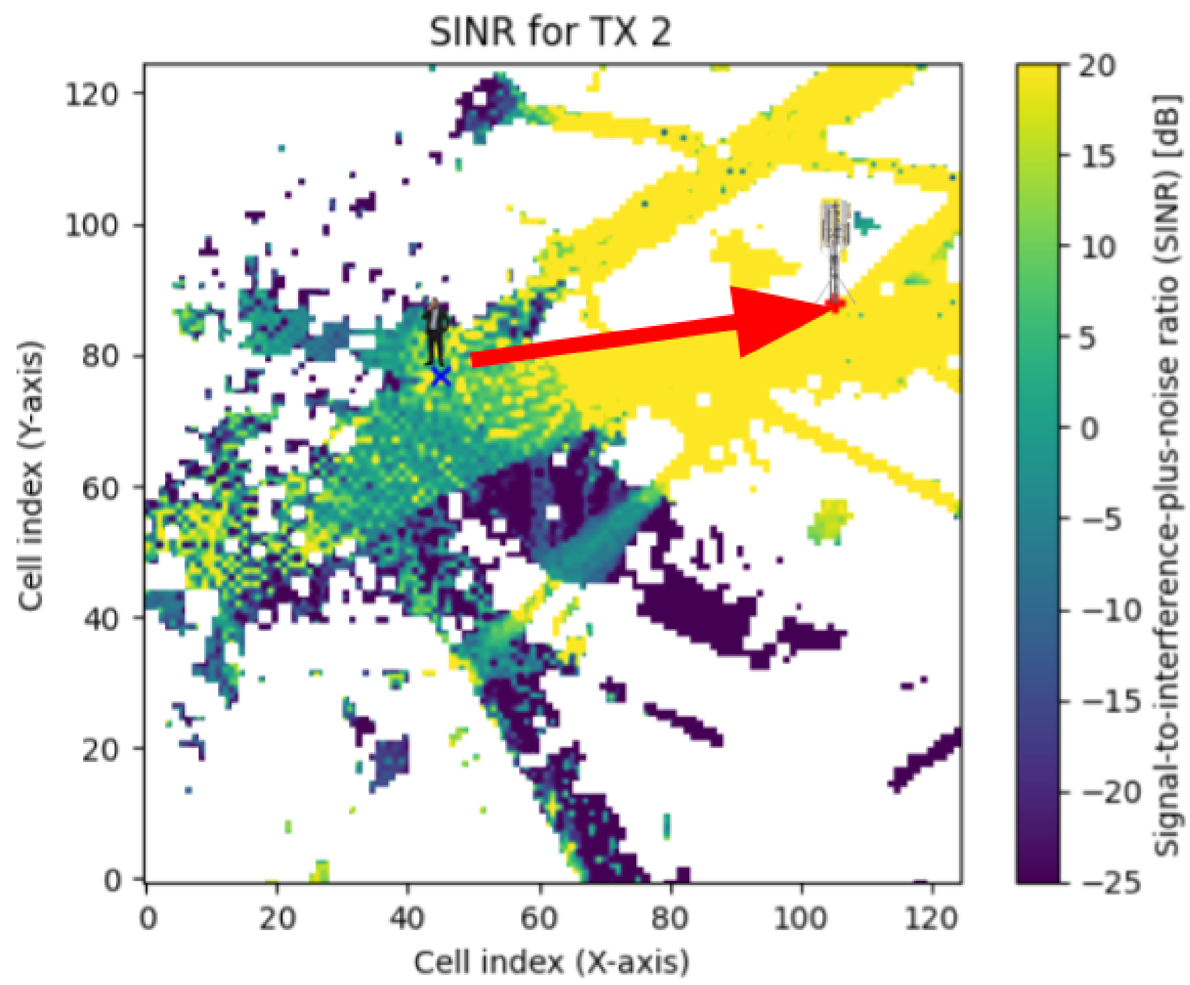

Figure 4.12: User with a Distant Transmitter

In the fig. 4.12, we can see that the user is in **Light Green** region of the transmitter.

After calculation, we got:

- **Distance:** The distance between the user and the transmitter 'Tx 2' is **338.6 Units**
- **SINR:** **5 dB** (We can also see that on the fig. 4.12 as the user is in the light green region which represents 5-10 dB SINR)

In the same location:

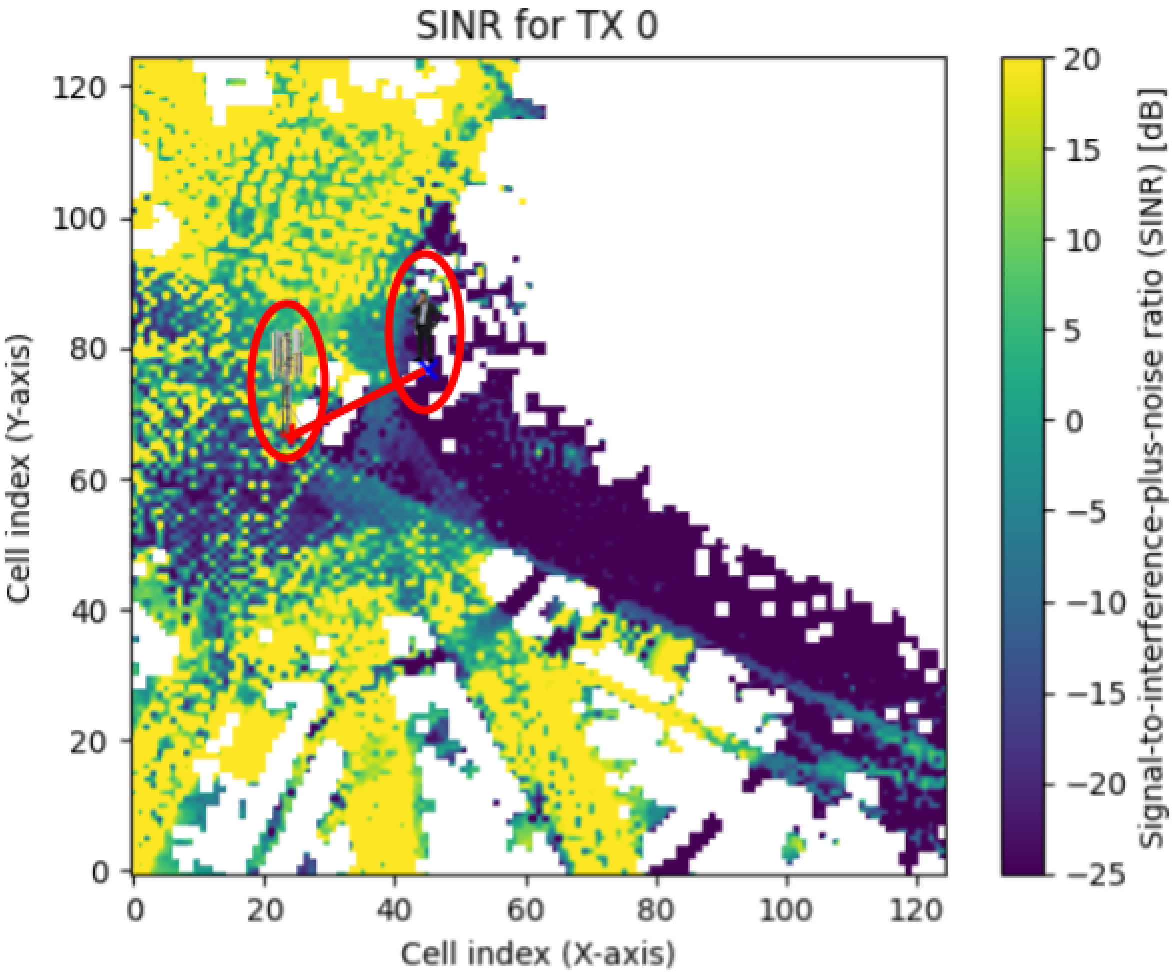

Figure 4.13: User with a Nearest Transmitter

In the fig. 4.13, we can see that the user is the **Deep Blue** region of the transmitter 'Tx 0'.

After calculation, we got:

- **Distance:** The distance between the user and the transmitter 'Tx 0' is **91.91 Units**
- **SINR:** **-25 dB** (We can also see that on the fig. 4.13 as the user is in the deep blue region which represents -20 to -25 dB SINR)

In summary, for that particular position:

Table 4.3: Transmitters' Information and SINR Values

Transmitters	Distance with the receiver on that particular position	SINR	Label
Tx 0	91.91 Units	-25 dB	Nearest Transmitter
Tx 1	338.6 Units	5 dB	Distant Transmitter

Distance-based handovers don't always give the best results. Sometimes, they can lead to poor signal quality. In these cases, using Digital Twin-based handovers is better because they provide a stronger and more reliable signal (SINR).

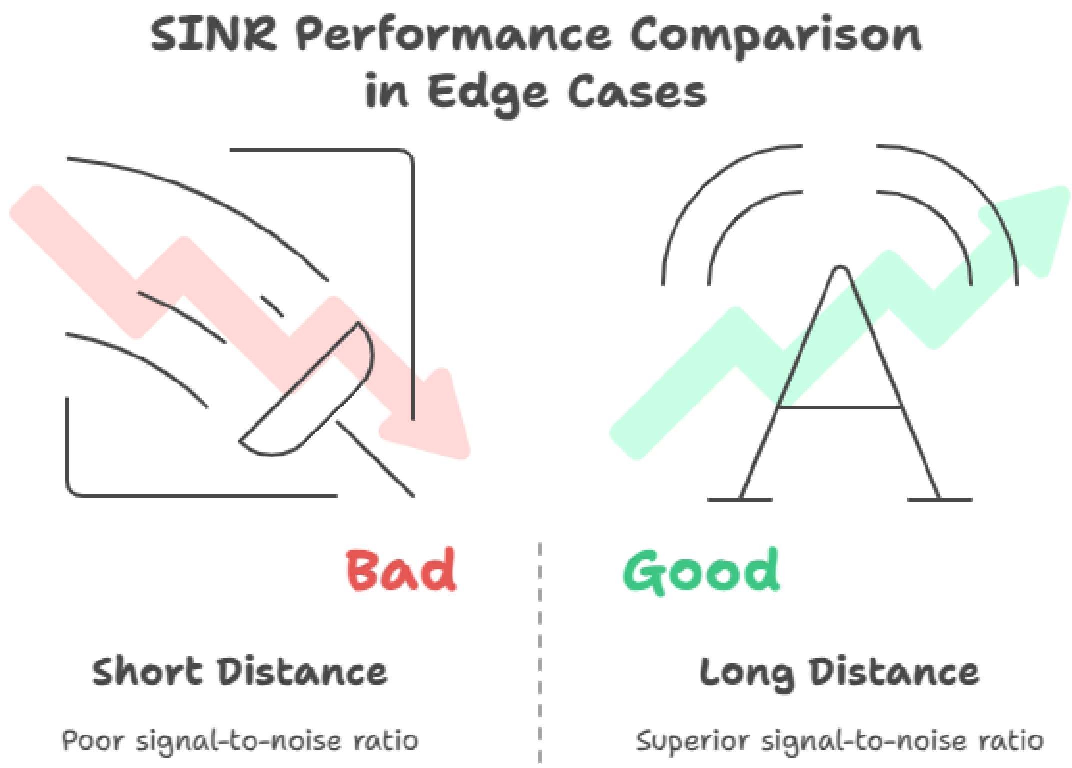

Figure 4.14: Why DT-based handovers are good in edge cases

Digital Twin has the RF environment awareness. As a result, it can pick up the best transmitter proactively, which is very useful in the edge cases as we have seen in table 4.3.

Chapter 5

Evaluation

5.1 Assessment of Issues

5.1.1 Societal Issues

The DT framework enhances connectivity, addressing societal needs by improving access in underserved areas, as shown by achieving good SINR in challenging scenarios.

5.1.2 Health and Safety Issues

The methodology ensures health and safety by optimizing EMF exposure (per International Commission on Non-Ionizing Radiation Protection (ICNIRP)) through power control and reliable handovers for critical communications.

5.1.3 Legal and Cultural Issues

Legal compliance with General Data Protection Regulation (GDPR) protects user data privacy, while cultural adaptability is supported by modeling diverse mobility patterns (e.g., urban vs. rural).

5.2 Evaluation of Environment and Sustainability

The DT reduces environmental impact via energy-efficient beamforming and adaptive power control, minimizing the ecological footprint of BS operations. It also supports disaster resilience, enhancing sustainability.

5.3 Ethical Issues

Ethical principles were upheld by ensuring data privacy through encryption, avoiding plagiarism, and maintaining transparency in research. The work aligns with professional engineering ethics and societal responsibility norms.

5.4 Future Works

The current study has laid a robust foundation for leveraging Digital Twin (DT) technology in wireless network management, particularly in optimizing handover processes and enhancing user experience. However, several avenues remain for further exploration to refine and expand the framework's capabilities.

A key area of future work involves comparing our DT-enabled handover results with commercially available handover protocols used by commercial telecom providers. This comparative analysis will benchmark the performance of our RF environment-aware handover strategy against industry standards, focusing on metrics such as Bit Error Rate (BER), Signal-to-Interference-plus-Noise Ratio (SINR), handover success rates, and latency. By evaluating our approach against protocols deployed by major telecom operators, we aim to identify strengths, limitations, and potential areas for integration, ensuring the DT framework aligns with real-world operational requirements.

Additionally, we plan to enhance the DT model by incorporating more complex user mobility patterns, such as those observed during large-scale events or in disaster scenarios, to further test its adaptability. The integration of non-RF sensors, such as LiDAR, will be explored to improve blockage prediction and optimize handover execution. We also intend to implement deep-learning models with higher accuracy (e.g., exceeding 90% [3]) for proactive handover strategies, predicting user positions and selecting optimal base stations (BSs) in advance. Furthermore, the adoption of multi-BS connectivity and advanced beamforming techniques, such as codebook-based beamforming, will be prioritized to enhance signal quality and reduce interference.

Finally, expanding the framework to support emerging technologies like 6G communication and integrating it with environmental monitoring systems will enable the DT to contribute to disaster resilience and sustainability goals. These enhancements will ensure the DT remains a scalable and impactful solution for future wireless networks.

References

[1] Hoydis, J., Cammerer, S., Aoudia, F.A., Vem, A., Binder, N., Marcus, G. and Keller, A. "SIONNA: An Open-Source Library for Next-Generation Physical Layer Research." *arXiv (Cornell University)*, 1 2022

[2] Freedman, A. "Network Digital Twins: the next revolution in network management."

[3] Vankayala, S.K., Gollapudi, S.K.S., Singh, S., Jain, B., Yoon, S. and Bashir, A.K. "Deep-Learning Based Proactive Handover for 5G/6G Mobile Networks using Wireless Information." *2022 IEEE Globecom Workshops (GC Wkshps)*, pp. 461–466, 12 2022

[4] Vivas, F.Y., Caicedo, O.M. and Nieves, J.C. "A Semantic and Knowledge-Based approach for handover management." *Sensors*, volume 21, no. 12:p. 4234, 6 2021

[5] Hanjri, A.E., Ouahab, I.B.A. and Haqiq, A. "Survey on Handover Techniques for Heterogeneous mobile networks." *Híradástechnika/Infocommunications journal*, volume 16, no. 3:pp. 64–74, 1 2024

[6] Liu, Y., Peng, Z., Zhang, Z., Yu, H. and Chen, M. "Digital Network Twins for next-generation Wireless: Creation, optimization, and challenges." *arXiv (Cornell University)*, 10 2024

[7] Verdecchia, R., Scommegna, L., Picano, B., Becattini, M. and Vicario, E. "Network Digital Twins: A Systematic Review." *IEEE Access*, volume 12:pp. 145400–145416, 1 2024

[8] Kaur, M.J., Mishra, V.P. and Maheshwari, P. *The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action*, 7 2019

[9] Gupta, M. and Khan, N. *Digital twin understanding, current progressions, and future perspectives*, 2 2023

[10] Perno, M., Hvam, L. and Haug, A. "Enablers and Barriers to the Implementation of digital twins in the process Industry: A Systematic literature review." *2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 959–964, 12 2020

[11] Gomaa, A. “Digital Twins for improving proactive maintenance management.” *Engineering Science*, volume 9, no. 3:pp. 60–70, 12 2024

[12] Khan, L.U., Han, Z., Saad, W., Hossain, E., Guizani, M. and Hong, C.S. “Digital Twin of Wireless Systems: Overview, taxonomy, challenges, and opportunities.” *IEEE Communications Surveys Tutorials*, volume 24, no. 4:pp. 2230–2254, 1 2022

[13] Kabashkin, I. “Digital Twin Framework for aircraft Lifecycle Management based on Data-Driven Models.” *Mathematics*, volume 12, no. 19:p. 2979, 9 2024

[14] Arin, I.A., Meyliana, N., Warnars, H.L.H.S. and Murad, D.F. “A Systematic Literature Review of Recent Trends and Challenges in Digital Twin Implementation.” *ICISS*, pp. 1–10, 9 2023

[15] “OpenStreetMap.”

[16] “OpenCelliD - largest open database of cell towers geolocation - by Unwired Labs.”

[17] Foundation, B. “blender.org - Home of the Blender project - Free and Open 3D Creation Software.”

[18] Developer, N. “Sionna RT: Scene Creation with Blender using OpenStreetMap.”, 3 2023

[19] “Ray Tracing — Sionna 0.19.2 documentation.”

[20] GeeksforGeeks,. “Outdoor Mobility Model — GaussMarkov.”, 8 2020

[21] “Simple Network Simulator (sim2net) — sim2net documentation.”, 12 2014

[22] Aghaei, F., Eldeeb, H.B., Bariah, L., Muhaidat, S. and Uysal, M. “Comparative characterization of indoor VLC and MMW communications via ray tracing simulations.” *IEEE Access*, volume 11:pp. 90345–90357, 1 2023

Appendix A

Codes

A.1 Thesis Code

We use this Python code for our thesis to generate the necessary results and plots.

```python
# -*- coding: utf-8 -*-
"""New Thesis Work Fresh Copy (Coverage Map).ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1UtFAFzsZ8jMemX56GVW7o45PBrjGZYxO

# Import Sionna & Others

---
"""

import os
gpu_num = 0 # Use "" to use the CPU
os.environ["CUDA_VISIBLE_DEVICES"] = f"{gpu_num}"
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

# Colab does currently not support the latest version of ipython.
# Thus, the preview does not work in Colab. However, whenever possible we
# strongly recommend to use the scene preview mode.
try: # detect if the notebook runs in Colab
    import google.colab
    colab_compat = True # deactivate preview
except:
    colab_compat = False
resolution = [480,320] # increase for higher quality of renderings

# Allows to exit cell execution in Jupyter
class ExitCell(Exception):
    def _render_traceback_(self):
        pass

# Import Sionna
try:
```

```
    import sionna
except ImportError as e:
    # Install Sionna if package is not already installed
    import os
    os.system("pip install sionna")
    import sionna

# Configure the notebook to use only a single GPU and allocate only as much memory as needed
# For more details, see https://www.tensorflow.org/guide/gpu
import tensorflow as tf
gpus = tf.config.list_physical_devices('GPU')
if gpus:
    try:
        tf.config.experimental.set_memory_growth(gpus[0], True)
    except RuntimeError as e:
        print(e)
# Avoid warnings from TensorFlow
tf.get_logger().setLevel('ERROR')

tf.random.set_seed(1) # Set global random seed for reproducibility

# Commented out IPython magic to ensure Python compatibility.
# %matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import time

# Import Sionna RT components
from sionna.rt import load_scene, Transmitter, Receiver, PlanarArray, Camera

# For link-level simulations
from sionna.channel import cir_to_ofdm_channel, subcarrier_frequencies, OFDMChannel,
    ApplyOFDMChannel, CIRDataset
from sionna.nr import PUSCHConfig, PUSCHTransmitter, PUSCHReceiver
from sionna.utils import compute_ber, ebnodb2no, PlotBER
from sionna.ofdm import KBestDetector, LinearDetector
from sionna.mimo import StreamManagement

"""# Loading Scene (Custom or Pre-Made)

---


"""

# Load integrated scene
scene = load_scene(sionna.rt.scene.etoile) # Here we loaded a pre-made scene

"""## Setting Up Camera

---


"""

# Create new camera with different configuration
scene.add(Camera("my_cam",
                 position=[0,0,1000],
                 orientation=[0,np.pi/2,-np.pi/2]))
```

```
scene.bandwidth=100e6
# Render scene with new camera*
scene.render("my_cam", resolution=resolution, num_samples=512); # Increase num_samples to
     increase image quality

"""## Setting Up Transmitters & Receivers
Here I considered **3 Transmitters** (Tx) and **1 Receiver** (Rx) for this simulation

---


"""

# Remove transmitters and receivers so that we can simulate it for multiple times
scene.remove('tx1')
scene.remove('tx2')
scene.remove('tx3')
scene.remove('rx')

# Configure antenna array for all transmitters
scene.tx_array = PlanarArray(num_rows=1,
                             num_cols=1,
                             vertical_spacing=0.5,
                             horizontal_spacing=0.5,
                             pattern="tr38901",
                             polarization="V")

# Configure antenna array for all receivers
scene.rx_array = PlanarArray(num_rows=1,
                             num_cols=1,
                             vertical_spacing=0.5,
                             horizontal_spacing=0.5,
                             pattern="dipole",
                             polarization="cross")

# Create transmitter
tx1 = Transmitter(name="tx1",
                 position=[-150.3, 21.63, 42.5])
tx2 = Transmitter(name="tx2",
                 position=[232.8, -95.5, 17])
tx3 = Transmitter(name="tx3",
                 position=[172.1, 103.7, 24])

# Add transmitter instance to scene
scene.add(tx1)
scene.add(tx2)
scene.add(tx3)

# Create a receiver
rx = Receiver(name="rx",
              position=[-94.5, 64.94, 22.5],
              orientation=[0,0,0])

# Add receiver instance to scene
scene.add(rx)

tx1.look_at(rx) # Transmitter points towards receiver
tx2.look_at(rx)
tx3.look_at(rx)
```

```

# Create new camera with different configuration
scene.add(Camera("my_cam",
                     position=[0,0,1000],
                     orientation=[0,np.pi/2,-np.pi/2]))
scene.bandwidth=100e6
scene.render("my_cam", resolution=resolution, num_samples=512);

"""Another Camera Angle:"""

scene.remove('my_cam1')
# Create new camera with different configuration
scene.add(Camera("my_cam1",
                     position=[50, -700, 150],
                     #orientation=[-np.pi/4, 0, 0],
                     look_at=[50, 50, 50]))
scene.bandwidth=100e6
scene.render("my_cam1", resolution=resolution, num_samples=512);

"""Another Camera Angle:"""

scene.remove('my_cam2')
# Adjust the camera for a closer and higher view
scene.add(Camera("my_cam2",
                 position=[50, -600, 200],  # Move closer and higher
                 look_at=[50, 50, 50]))  # Keep the look_at point centered

# Render the scene with the updated camera configuration
scene.render("my_cam2", resolution=resolution, num_samples=512)

"""## Computing SINR Using Coverage Maps!!
New feature of Sionna V0.19.0

---

A coverage map assigns a metric, such as path gain, received signal strength (RSS), or signal-to-interference-plus-noise ratio (SINR), for a specific transmitter to every point on a plane. In other words, for a given transmitter, it associates every point on a surface with the channel gain, RSS, or SINR, that a receiver with a specific orientation would observe at this point.

A coverage map depends on the transmit and receive arrays and their respective antenna patterns, the transmitter and receiver orientations, as well as the transmit precoding and receive combining vectors. Moreover, a coverage map is not continuous but discrete, as the plane must be quantized into small rectangular bins, which we refer to as cells.


---
As soon as there are multiple transmitters in a scene, we can either visualize a metric for specific transmitter or visualize the maximum matric across all transmitters. **The latter option is relevant if we want to inspect, e.g., the SINR across a large scene, assuming that a receiver always connects to the transmitter providing the best SINR**.
"""

# Compute the SINR map
cm_etoile = scene.coverage_map(
    max_depth=7,
    los=True,
    reflection=True,
```

```
    diffraction=True,
    cm_cell_size=(1, 1),
    num_samples=int(10e6),  # Decrease to reduce memory consumption
    num_runs=1)  # Increase to average over multiple coverage maps to reduce noise

# Show SINR for individual transmitters
# show_rx=True add the receiver in the plot
cm_etoile.show(metric="sinr", tx=0, vmin=-25, vmax=20,show_rx=True);
cm_etoile.show(metric="sinr", tx=1, vmin=-25, vmax=20,show_rx=True);
cm_etoile.show(metric="sinr", tx=2, vmin=-25, vmax=20,show_rx=True);

# Show maximum SINR across all transmitters
cm_etoile.show(metric="sinr", tx=None, vmin=-25, vmax=20,show_rx=True);

"""**It is also interesting to investigate which regions of a coverage map are    covered
    by each transmitter, i.e., where a transmitter provides the strongest metric. You can
    obtain this information either as a tensor from the class method `cell_to_tx()` or
    visualize it using `show_association()`.**"""

cm_etoile.show_association("sinr");

"""# Simulation

---

## Simulation 1: DT Based Handover




Beginning transmission with the receiver from the best transmitter (Providing best SINR)


---

### Logic behind the simulation:
Computing the SINR at a given position. In principle, you could compute coverage maps for all
    transmitters in the scene and then compute for each transmitter's SINR Map. Users would
    always seek to handover to another transmitter when the SINR is too low.

You would not even need to compute propagation paths with this approach and simply compute the
     coverage maps once.


---

**Creating Necessary Function:**
"""

'''
This approach:

1) Converts the world position to the local coordinate system of the coverage map.

2) Shifts the origin to match the coverage map's coordinate system.

3) Calculates the cell indices by dividing the local position by the cell size.

4) Ensures the indices are within the bounds of the coverage map.
```

```

5) Retrieves the SINR values for all transmitters at that cell.

Output:

This gives you the SINR values for all transmitters at the specified position,
which you can use for handover decisions or other analyses.

You're getting three SINR values because there are three transmitters in the scene.
Each value represents the SINR for a different transmitter at the given position.
'''
from sionna.rt.utils import rotation_matrix
def get_sinr_at_position(coverage_map, position):
    # Convert position to local coordinates relative to coverage map center
    local_pos = position[:2] - coverage_map._center[:2]

    # Get cell size and map size
    cell_size = coverage_map._cell_size
    map_size = coverage_map._size

    # Shift to coverage map coordinate system
    local_pos += map_size / 2

    # Calculate cell indices
    cell_indices = tf.cast(local_pos / cell_size, tf.int32)

    # Clip indices to valid range
    cell_indices = tf.clip_by_value(
        cell_indices,
        [0, 0],
        [tf.shape(coverage_map.sinr)[2] - 1, tf.shape(coverage_map.sinr)[1] - 1]
    )

    # Get SINR values for all transmitters at this cell
    # Shape: [num_tx]
    sinr_values = coverage_map.sinr[:, cell_indices[1], cell_indices[0]]

    # Convert to dB
    sinr_db = 10 * tf.math.log(sinr_values) / tf.math.log(10.0)

    return sinr_db.numpy()

"""### Showing best serving transmitters
Three Example Cases as there are three transmiiters:

**Best Serving Transmitter: 02**


---
"""

scene.remove('rx')
# Usage
position = [-14.5, 54.94, 42.5]  # Your specific position

# Create a receiver
rx = Receiver(name="rx",
              position=position,
              orientation=[0,0,0])
```

```

# Add receiver instance to scene
scene.add(rx)

sinr_at_position = get_sinr_at_position(cm_etoile, position)
print(f"SINR at position {position}: {sinr_at_position} dB")

best_tx_index = np.argmax(sinr_at_position)
best_sinr = sinr_at_position[best_tx_index]
print(f"Best transmitter is {best_tx_index} with SINR = {best_sinr} dB")

# Show SINR Coverage Map for the best transmitter
cm_etoile = scene.coverage_map(
    max_depth=7,
    los=True,
    reflection=True,
    diffraction=True,
    cm_cell_size=(1, 1),
    num_samples=int(10e6),  # Decrease to reduce memory consumption
    num_runs=1)  # Increase to average over multiple coverage maps to reduce noise
cm_etoile.show(metric="sinr", tx=int(best_tx_index), vmin=-25, vmax=20,show_rx=True);

"""Showing the other transmitters why they are not the best to start transmission:"""

cm_etoile.show(metric="sinr", tx=0, vmin=-25, vmax=20,show_rx=True);

cm_etoile.show(metric="sinr", tx=1, vmin=-25, vmax=20,show_rx=True);

"""**Best Serving Transmitter: 00**


---


"""

scene.remove('rx')
# Usage
position = [-120.3, 151.63, 42.5]  # Your specific position

# Create a receiver
rx = Receiver(name="rx",
              position=position,
              orientation=[0,0,0])

# Add receiver instance to scene
scene.add(rx)

sinr_at_position = get_sinr_at_position(cm_etoile, position)
print(f"SINR at position {position}: {sinr_at_position} dB")

best_tx_index = np.argmax(sinr_at_position)
best_sinr = sinr_at_position[best_tx_index]
print(f"Best transmitter is {best_tx_index} with SINR = {best_sinr} dB")

# Show SINR Coverage Map for the best transmitter
cm_etoile = scene.coverage_map(
    max_depth=7,
    los=True,
```

```
    reflection=True,
    diffraction=True,
    cm_cell_size=(1, 1),
    num_samples=int(10e6),  # Decrease to reduce memory consumption
    num_runs=1)  # Increase to average over multiple coverage maps to reduce noise
cm_etoile.show(metric="sinr", tx=int(best_tx_index), vmin=-25, vmax=20,show_rx=True);

"""Showing the other transmitters why they are not the best to start transmission:"""

cm_etoile.show(metric="sinr", tx=1, vmin=-25, vmax=20,show_rx=True);

cm_etoile.show(metric="sinr", tx=2, vmin=-25, vmax=20,show_rx=True)

"""**Best Serving Transmitter: 01**


---


"""

scene.remove('rx')
# Usage
position = [190.8, -145.5, 17]  # Your specific position

# Create a receiver
rx = Receiver(name="rx",
              position=position,
              orientation=[0,0,0])

# Add receiver instance to scene
scene.add(rx)

sinr_at_position = get_sinr_at_position(cm_etoile, position)
print(f"SINR at position {position}: {sinr_at_position} dB")

best_tx_index = np.argmax(sinr_at_position)
best_sinr = sinr_at_position[best_tx_index]
print(f"Best transmitter is {best_tx_index} with SINR = {best_sinr} dB")

# Show SINR Coverage Map for the best transmitter
cm_etoile = scene.coverage_map(
    max_depth=7,
    los=True,
    reflection=True,
    diffraction=True,
    cm_cell_size=(1, 1),
    num_samples=int(10e6),  # Decrease to reduce memory consumption
    num_runs=1)  # Increase to average over multiple coverage maps to reduce noise
cm_etoile.show(metric="sinr", tx=int(best_tx_index), vmin=-25, vmax=20,show_rx=True);

"""Showing the other transmitters why they are not the best to start transmission:"""

cm_etoile.show(metric="sinr", tx=0, vmin=-25, vmax=20,show_rx=True);

cm_etoile.show(metric="sinr", tx=2, vmin=-25, vmax=20,show_rx=True);

"""## Simulation 2: DT vs Distance Based Handover

```

```

---

### Logic Behind the simulation:
User having high BER/low SINR, even after handover to the nearest transmitter is that there is  some link blockage such as buildings  that result  in huge path loss and shadowing and thus high BER/low SINR. In contrast, another transmitter, despite having large distance, has clear path and thus providing  better BER/ High SINR.

Particularly, distance based handover is usually  employed as it is most simple. However this distance based handover does not have any knowledge of the RF environment. Thus, it may end up associating a user to near-by BS, which has high path loss due to buildings. In contrast the digital twin avoids the case thanks to its RF environment awareness.

We can compare SINR performance of two systems: RF environment  aware DT enabled handover  and  distance based handover. In the latter case, user will be alaways associated withe the nearest BS. We can show DT based handover achieves better SINR.
"""

'''
In this code, I'm trying to calculate the user positions in the scene where the nearest transmitter is not the best choice.
On that positions, distant transmitters provide better SINR than the nearest transmitter.
'''
import numpy as np
import matplotlib.pyplot as plt
from sionna.rt import load_scene, Transmitter, Receiver, PlanarArray

# Load the etoile scene
scene = load_scene(sionna.rt.scene.etoile)

# Add transmitters
tx_positions = [
    [-150.3, 21.63, 42.5],
    [232.8, -95.5, 17],
    [172.1, 103.7, 24]
]

for i, pos in enumerate(tx_positions):
    tx = Transmitter(name=f"tx{i+1}", position=pos, power_dbm=44)
    scene.add(tx)

# Configure antenna arrays
scene.tx_array = PlanarArray(num_rows=4, num_cols=4, vertical_spacing=0.5, horizontal_spacing=0.5, pattern="iso", polarization="V")
scene.rx_array = PlanarArray(num_rows=1, num_cols=1, vertical_spacing=0.5, horizontal_spacing=0.5, pattern="iso", polarization="V")

# Compute coverage map for the etoile scene
coverage_map = scene.coverage_map(
    num_samples=2e6,
    max_depth=5,
    los=True,
    reflection=True,
    diffraction=True,
    cm_cell_size=[4, 4],
    cm_orientation=[0, 0, 0],
    cm_center=[0, 0, 1.5],
    cm_size=[500, 500]
```

```
)

# Define a function to calculate distance
def calculate_distance(pos1, pos2):
    return np.linalg.norm(np.array(pos1) - np.array(pos2))

# Create a grid of potential user positions
x_range = np.linspace(-300, 300, 50)
y_range = np.linspace(-200, 200, 50)
grid_positions = [(x, y, 1.5) for x in x_range for y in y_range]

# Find positions where the nearest transmitter is not the best choice
selected_positions = []

for position in grid_positions:
    # Add receiver to the scene
    receiver = Receiver(name="rx", position=position)
    scene.add(receiver)

    # Calculate SINR values
    sinr_values = get_sinr_at_position(coverage_map, position)

    # Distance-based handover
    distances = [calculate_distance(position, tx_pos) for tx_pos in tx_positions]
    nearest_tx_index = np.argmin(distances)

    # DT-enabled handover
    best_tx_index = np.argmax(sinr_values)

    # Check if the nearest transmitter is not the best choice
    if nearest_tx_index != best_tx_index:
        selected_positions.append(position)
        print(f"Position: {position}")
        print(f"Nearest Transmitter: TX{nearest_tx_index+1}, SINR: {sinr_values[
    nearest_tx_index]:.2f} dB")
        print(f"Best Transmitter: TX{best_tx_index+1}, SINR: {sinr_values[best_tx_index]:.2f}
    dB\n")

    # Remove receiver from the scene
    scene.remove("rx")

# Visualize the selected positions
plt.figure(figsize=(10, 10))
plt.scatter([], [], color='red', marker='x', label='User Positions')
plt.scatter([], [], color='blue', marker='+', s=100, label='Transmitter')
for pos in selected_positions:
    plt.scatter(pos[0], pos[1], color='red', marker='x')
for tx_pos in tx_positions:
    plt.scatter(tx_pos[0], tx_pos[1], color='blue', marker='+', s=100)
plt.xlabel('X Position')
plt.ylabel('Y Position')
plt.title('Positions Where Nearest Transmitter is Not the Best Choice')
plt.legend()
plt.grid(True)
plt.show()

"""### Example Cases:

---
```

```python


"""

scene.remove('rx')
# Usage
position = (-128.57142857142858, 12.244897959183675, 1.5)  # Your specific position

# Create a receiver
rx = Receiver(name="rx",
              position=position,
              orientation=[0,0,0])

# Add receiver instance to scene
scene.add(rx)

sinr_at_position = get_sinr_at_position(coverage_map, position)
print(f"SINR at position {position}: {sinr_at_position} dB")

best_tx_index = np.argmax(sinr_at_position)
best_sinr = sinr_at_position[best_tx_index]
print(f"Best transmitter is {best_tx_index} with SINR = {best_sinr} dB")

# Show SINR Coverage Map for the best transmitter
coverage_map = scene.coverage_map(
    num_samples=2e6,
    max_depth=5,
    los=True,
    reflection=True,
    diffraction=True,
    cm_cell_size=[4, 4],
    cm_orientation=[0, 0, 0],
    cm_center=[0, 0, 1.5],
    cm_size=[500, 500]
)
coverage_map.show(metric="sinr", tx=int(best_tx_index), vmin=-25, vmax=20,show_rx=True);

"""Nearest transmitter is not the best because:"""

coverage_map.show(metric="sinr", tx=0, vmin=-25, vmax=20,show_rx=True)

"""Another Example:"""

scene.remove('rx')
# Usage
position = (-67.34693877551021, 61.22448979591837, 1.5)  # Your specific position

# Create a receiver
rx = Receiver(name="rx",
              position=position,
              orientation=[0,0,0])

# Add receiver instance to scene
scene.add(rx)

sinr_at_position = get_sinr_at_position(coverage_map, position)
print(f"SINR at position {position}: {sinr_at_position} dB")

best_tx_index = np.argmax(sinr_at_position)
```

```
best_sinr = sinr_at_position[best_tx_index]
print(f"Best transmitter is {best_tx_index} with SINR = {best_sinr} dB")

# Show SINR Coverage Map for the best transmitter
coverage_map = scene.coverage_map(
    num_samples=2e6,
    max_depth=5,
    los=True,
    reflection=True,
    diffraction=True,
    cm_cell_size=[4, 4],
    cm_orientation=[0, 0, 0],
    cm_center=[0, 0, 1.5],
    cm_size=[500, 500]
)
coverage_map.show(metric="sinr", tx=int(best_tx_index), vmin=-25, vmax=20,show_rx=True);

"""Nearest transmitter is not the best because:"""

coverage_map.show(metric="sinr", tx=0, vmin=-25, vmax=20,show_rx=True)
```

B.Sc.
Engg.
EEE
BUET

Proactive Handover in Multi-Cell Networks:
A Digital Twin Approach for Low-Complexity and Scalable Performance

Anindha Dhar
1906194

March
2025