Politecnico di Torino

DAUIN
Dipartimento di Automatica e Informatica

Exploring Device-to-Device (D2D) Communication in Cellular Technology: 5G and Beyond

Master Thesis

Exploring Device-to-Device (D2D) Communication in Cellular Technology: 5G and Beyond

Master Thesis
October, 2024

By
Pietro Marrone

Copyright: Reproduction of this publication in whole or in part must include the customary bibliographic citation, including author attribution, report title, etc.
Cover photo: ITEGRITI, 2023
Published by: Politecnico di Torino, Dipartimento di Automatica e Informatica, Corso Duca degli Abruzzi, 24 , 10129 Torino TO Italy
https://www.dauin.polito.it/

Approval

This thesis has been prepared over six months at the Sector of DTU Compute, Department of Electrical and Photonics Engineering, at the Technical University of Denmark, DTU, in partial fulfilment for the degree Master of Computer Engineering, MSc Eng. as an exchange student from Politecnico di Torino.

Pietro Marrone - s310561

....

Signature

....

Date

Abstract

This study explores the capabilities and performance of Device-to-Device (D2D) communication within the 5G network framework, focusing on the DECT NR+ standard. Developed by the European Telecommunications Standards Institute (ETSI), DECT NR+ aims to meet the stringent requirements of 5G technology. The hypothesis assumes that DECT NR+ can significantly enhance communication efficiency and reliability in D2D scenarios.

To test this hypothesis, hardware and software tools from Nordic Semiconductor was utilized, including the nRF9161 Development Kit, nRF Connect for Desktop and nRF Connect SDK. These tools facilitated the setup and evaluation of seamless communication between devices using the DECT NR+ standard.

Key performance metrics were systematically evaluated, including latency, coverage, Packet Success Rate (PSR), data rate, and power consumption. The results indicated that DECT NR+ provides robust data rate performance, achieving up to 2.2 Mbps, and high PSR in controlled environments up to a critical distance, beyond which PSR declined. The maximum observed range was 220 meters, which is notable but still less compared to other technologies as LTE-M and NB-IoT. Latency analysis revealed a linear relationship between Round Trip Time (RTT) and packet size, with higher Modulation and Coding Scheme (MCS) values improving latency performance. However, observed RTTs did not fully meet the Ultra-Reliable Low Latency Communication (URLLC) standards, suggesting the need for further optimization. Finally, Power consumption analysis showed that DECT NR+ demonstrated significantly lower power consumption compared to Wi-Fi, especially during idle and active operation, and was comparable to NB-IoT and LTE-M. Future implementations of the complete DECT NR+ stack and additional power-saving features are expected to enhance this efficiency further.

In conclusion, this thesis contributes to the ongoing development and optimization of DECT NR+ technology, underscoring its promise as a versatile and efficient solution for future wireless communication challenges. Future work will focus on addressing the identified optimization areas to further enhance DECT NR+ performance.

Acknowledgements

Pietro Marrone, MSc Computer Engineering, DTU
During the realisation of this master thesis at the Technical University of Denmark, DTU, I have been in contact with various collaborators. Therefore, I would like to express my sincere gratitude and appreciation to the following people for their cooperation and support. Their role as mentors and their support towards me, an exchange student, have been fundamental to the successful accomplishment of this project.

Michael Stübert Berger, Associate Professor, Department of Electrical and Photonics Engineering, Technical University of Denmark.

Claudio Ettore Casetti, Associate Professor, Department of Automatics and Computer Science, Politecnico di Torino.

Christian Kloch, IoT Tech Lead, FORCE Technology.

Radheshyam Singh, PhD student, Department of Electrical and Photonics Engineering, Technical University of Denmark.

For their close support, guidance and good spirit throughout the project.

To my family and friends, my greatest cheerleaders, for standing by me and making this incredible experience possible. Thank you for always being by my side.

Contents

1 Introduction

1.1 Background

1.1.1 The evolution of 5G

The advent of 5G technology marks a significant leap forward in wireless communication, promising unprecedented improvements in speed, capacity, and connectivity. Unlike its predecessor, 4G LTE, which primarily focuses on enhanced mobile broadband services, 5G aims to provide a versatile platform capable of supporting a diverse range of applications, including the Internet of Things (IoT), autonomous vehicles, smart cities and more.

5G promises to be much more than an incremental improvement over 4G LTE in at least three aspects: speed performance, projected to be about 600 times better than the current 4G LTE standard; network latency, enabling instantaneous and real-time communication across devices and applications; and capacity, leading to ubiquitous connectivity with an astonishing 100× increase in traffic capacity and network efficiency and thus significantly overcoming the traditional issues of band scarcity that plagued earlier generations of mobile connectivity [1].

But these are not the only advantages of 5G networks. 5G technology offers significantly wider bandwidth and greater flexibility regarding how bands are used. This means 5G networks can maintain stable connectivity for a far greater number of devices in a concentrated area. This is the primary way 5G is transforming the Internet of Things, opening the door to a revolution in the IoT world. In fact, there are already billions of IoT devices, but the wider bandwidth and more efficient spectrum usage of 5G networks will allow far more devices to operate in close proximity without interfering with one another [2].

5G is divided into three frequency bands (low, mid and high). Each band has different capabilities: the low band (less than 1 GHz) offers greater coverage but lower speeds, the mid band (1 GHz–6 GHz) provides a balance of both, and the high band (24 GHz–40 GHz) delivers higher speeds but with a smaller coverage radius. 5G can utilize more frequencies across all three bands to achieve optimal performance, even using multiple frequencies simultaneously [3]. In contrast, 4G/LTE cellular technology primarily operates within the 700 MHz to 900 MHz and 1.8 GHz to 2.5 GHz frequency bands [4]. The key to this versatility and enhanced performance lies in the new radio access technology known as 5G New Radio (NR).

However, because of this broader frequency use, 5G necessitate a greater number of base stations to ensure comprehensive coverage, and this can be seen as the most significant disadvantage of 5G. In fact, considering that 4G primarily operates below 3 GHz, it cannot support the vast number of end devices and the massive influx of data that 5G can handle. The 4G network simply is not designed for this level of connectivity [5].

1.1.2 5G NR

5G New Radio, or 5G NR, is the global standard for the air interface of 5G networks. Developed by the 3rd Generation Partnership Project (3GPP), 5G NR replaces the 4G LTE standards and enables a wide array of enhancements in wireless communication. 5G NR significantly improves speed and reduces latency compared to previous 4G standards, supporting the growing demand for high-speed, low-latency applications. Key improvements introduced by 5G NR include:

- Diverse Spectrum Utilization: Ranging from several hundred kilohertz to millimeter wave (mmWave) bands, enabling various use cases, cell sizes and data rates.
- Advanced Modulation and Coding: New orthogonal frequency-division multiplexing methods and channel-coding techniques.
- Frequency Reuse Algorithms: Enhanced algorithms for dense environments.
- Beam-forming: A signal processing technique to improve signal quality and coverage.
- Massive MIMO: Supports multiple input, multiple output capabilities to increase capacity and throughput.
- Ultra-low Latency: Slot time operations designed to deliver ultra-low-latency communications.

The benefits of 5G NR over the best LTE networks include larger network capacity, increased energy savings per device, shorter service creation time cycles, enhanced speed and data rates, improved efficiency in data sharing and better latency [6]. 5G NR plays a crucial role in the 5G network, primarily addressing the radio access technology component. It integrates the physical (PHY) and medium access control (MAC) layers of the OSI protocol stack, managing radio transmissions and channel sharing. These layers are vital for ensuring the high performance and capabilities of 5G networks.

1.2 Overview

The rapid evolution of cellular technology has paved the way for innovative communication paradigms, with Device-to-Device (D2D) communication emerging as a significant feature. Traditionally, mobile devices communicate via a centralized network infrastructure, such as a base station or a mobile switching center. However, D2D communication enables direct interaction between devices without traversing the core network, offering numerous advantages in terms of latency, throughput and energy efficiency.

This evolution is particularly significant in the context of 5G and beyond 5G (B5G) networks, where D2D communication optimizes the utilization of available network resources, reduces latency, enhances data transmission speed and increases overall network capacity. Mobile operators play a crucial role in driving D2D development by leveraging short-range communication data, enhancing proximity-based services and overall network performance. Current research focuses on various aspects of D2D communication, including the discovery process, mode selection methods, interference management and power allocation, especially in 5G technologies. As the industry moves beyond 5G, leveraging these advancements to address the growing demand for efficient and reliable communication systems is crucial, facilitating applications such as public safety, content distribution and the Internet of Things (IoT) [7].

The motivation behind this project is to explore the potential of D2D communication within the realm of 5G and future cellular technologies. By bypassing traditional network infrastructure, D2D communication can significantly enhance spectral efficiency and improve overall network performance. These benefits are particularly relevant in scenarios requiring high-speed and reliable communication, such as emergency response situations and dense urban environments. Understanding the challenges and solutions associated with D2D communication, such as interference management, resource allocation and security, is essential for its successful integration into existing and future cellular networks.

1.3 Objectives

In the context of Device-to-Device (D2D) communication, this thesis focuses on the DECT NR+ standard. Developed and standardized by the European Telecommunications Standards Institute (ETSI), DECT NR+ is a radio access technology designed to meet the technical requirements of 5G. Standards organizations like ETSI and the 3rd Generation Partnership Project (3GPP) play crucial roles in developing these technologies, ensuring they adhere to stringent performance and interoperability criteria.

Nordic Semiconductor is one of the companies that has invested in the DECT NR+ standard, underscoring its commitment to advancing D2D communication technologies. This investment includes becoming a full member of the DECT Forum, which highlights Nordic's active involvement in shaping the future of DECT NR+ technology. Nordic has developed several devices, including the nRF9161 DK, which supports not only the LTE-M and NB-IoT standards but also the new DECT NR+ standard.

The tools used in this project include both software and hardware components provided by Nordic Semiconductor. For the software part, nRF Connect for Desktop and nRF Connect SDK were utilized. These tools offer comprehensive development environments for building and testing applications on Nordic's hardware platforms. The nRF Connect SDK provides a robust software framework for developing complex wireless applications, while nRF Connect for Desktop facilitates easy configuration and management of connected devices.

For the hardware aspect, the nRF9161 Development Kit (DK) was used. This development kit offers a versatile platform for prototyping and testing DECT NR+ communication features in various applications. It supports a wide range of functionalities necessary for evaluating the performance of the communication link between devices.

To achieve seamless communication between devices using the DECT NR+ standard and evaluate its performance, several experimental methodologies were employed. These methodologies included controlled indoor and outdoor testing environments to measure key performance metrics such as latency, coverage, Packet Success Rate (PSR), data rate and power consumption:

1. **Latency Measurement**: Latency tests were conducted by sending and receiving packets between devices and recording the Round Trip Time (RTT). Different Modulation and Coding Schemes (MCS) were applied to understand their impact on latency. A linear relationship between packet size and RTT was observed, with higher MCS values improving latency. However, current latency levels suggest further optimization is needed to meet Ultra-Reliable Low Latency Communication (URLLC) standards.
2. **Coverage and PSR Testing**: Coverage and PSR were evaluated by placing devices at incremental distances and monitoring the success rate of packet transmissions. This helped identify the maximum effective range and the impact of environmental factors on communication quality. DECT NR+ demonstrated higher PSR in controlled indoor environments, and a noticeable decline beyond critical distances. The maximum effective range was observed to be around 220 meters.
3. **Data Rate Analysis**: Data rate tests involved transmitting data with different MCS values to determine the highest sustainable throughput. This provided insights into the efficiency of data transmission under various conditions. The highest data rate achieved was 2.2 Mbps, lower than the expected maximum of 3.4 Mbps. However,

robust performances are still achieved, making the standard suitable for various IoT applications.

4. **Power Consumption Assessment**: Power consumption was measured during different states of device operation, including idle, active, and transmission modes. Comparisons were made with other communication standards to highlight DECT NR+'s efficiency. In particular, DECT NR+ showed similar power consumption compared to technologies like NB-IoT and LTE-M and significantly better performance than Wi-Fi, making it ideal for applications requiring long battery life.

By systematically assessing these parameters, this project aims to provide a comprehensive understanding of DECT NR+'s potential in real-world applications and contribute to its further development and optimization.

1.4 Structure of the Thesis

This section provides an outline of the structure and content of the thesis. It highlights the main chapters and their subtopics, emphasizing the order in which the research was conducted, and the methodology used to address the research questions. The thesis follows this structure to ensure a thorough investigation of the subject matter:

Chapter 1: Introduction. The first chapter introduces the background of the thesis, focusing on 5G and 5G NR (New Radio). It provides an overview of the project, outlines the objectives and details the tools used along with the expected outcomes. This chapter also includes the structure of the report to guide the reader through the subsequent chapters.

Chapter 2: Related Work/State of the Art. The second chapter reviews existing literature on D2D communication, concentrating on its development, benefits, challenges and solutions. Initially, it discusses D2D communication in general terms, then delves deeper into the DECT NR+ standard. This review sets the foundation for understanding the current state of the technology and identifies gaps that this thesis aims to address.

Chapter 3: Experimental Setup. The third chapter explains the experimental setup, detailing the tools, configurations and logic used in the study. It includes descriptions of the software tools (nRF Connect for Desktop and nRF Connect SDK) and the hardware (nRF9161 DK and Power Profiler Kit II). This chapter ensures that the methodology is transparent and reproducible.

Chapter 4: Tests and Results. The fourth chapter presents the tests conducted and the results obtained. It evaluates the performance of the DECT NR+ standard in terms of key metrics such as latency, coverage, Packet Success Rate (PSR), data rate and power consumption. This chapter also provides a comprehensive analysis of the experimental findings.

Chapter 5: Conclusion and Future Work. The fifth chapter concludes the thesis, summarizing the key findings and their implications. It also discusses potential future work, suggesting areas for further research and development to enhance DECT NR+ communication technologies, particularly in the context of 5G and beyond.

2 State of the art

This chapter provides a comprehensive review of existing literature and the current state of the art in Device-to-Device (D2D) communication. Initially, the concept of D2D communication will be introduced, covering its definition, architecture, features, channels and resources used, as well as the problems and challenges it faces. This section aims to present the general communication paradigm and its envisioned implementation.

Following this general overview, the focus will shift to a specific standard: DECT NR+. DECT NR+ is an example of a concrete standard that has been approved and developed for D2D communication. By examining DECT NR+, this chapter will illustrate how D2D communication can be standardized and implemented in real-world scenarios, highlighting its practical applications and advancements. DECT NR+ will be the primary standard addressed throughout the remaining part of this thesis work.

2.1 Introduction to D2D Communication

2.1.1 Definition

Device-to-Device (D2D) communication is a paradigm in wireless networks where devices communicate directly with each other without routing data through the network infrastructure, such as base stations (BSs) or access points (APs), allowing you to bypass them [8]. This technology reduces the workload on the network infrastructure by enabling direct interaction between user equipment (UE), especially when they are in close proximity to each other.

From a technical perspective, exploiting the natural proximity of communicating devices may provide multiple performance benefits. First, D2D UE may enjoy high data rates and low end-to-end delay due to the short range direct communication. Second, it is more resource-efficient for proximate UE devices to communicate directly with each other than routing through an evolved Node B (eNB) and possibly the core network. In particular, compared to normal downlink/uplink cellular communication, direct communication saves energy and improves radio resource utilization and spectral efficiency. Third, switching from an infrastructure path to a direct path offloads cellular traffic, alleviating congestion and thus benefiting other non-D2D UE as well [9] [10].

In D2D communication, the mode selection technique allows the devices to switch from the infrastructure path to direct path easily. This reduces the congestion in the network. From an econometric point of view, D2D communication can play a big role in commercial applications, social networking applications, e-commerce, etc., where users can directly share the required information locally [11]. However, 3GPP Release 12 introduced D2D communication in the context of LTE-A architecture, and its major targeted application is public safety.

2.1.2 Frequencies used

D2D communication can operate using both licensed cellular spectrum and unlicensed spectrum; the two modes are known as in-band communication and out-band communication, respectively.

In-band Communication allows both D2D and cellular communication to share the entire licensed cellular spectrum. This mode, known as underlay D2D communication, can lead to increased interference between D2D and cellular users. To address this, overlay D2D communication has been proposed. In this mode, D2D users utilize a portion of cellular resources that are not assigned to regular cellular users, focusing on effective resource allocation to avoid spectrum wastage.

Out-band Communication operates using unlicensed spectrum, reducing the potential for interference with cellular communications [12].

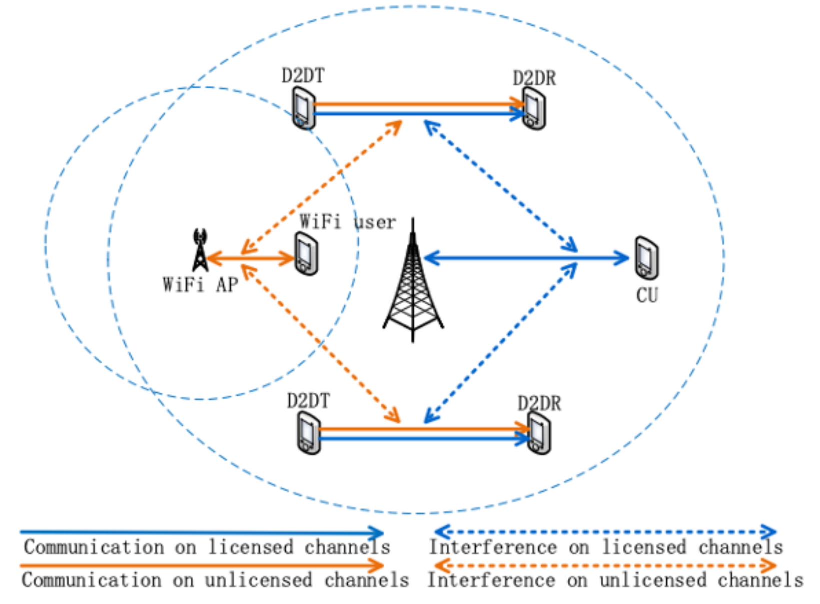


Figure 2.1: *The System Model of the D2D enabled 5G NR network (from [13]).*

As it is shown by the image above, in the 5G NR network, the cellular users can either transmit via the BS, i.e., in cellular mode (CM), or communicate directly with each other, i.e., in D2D mode. When users are working in the CM mode, they are named CM users (CU), and different licensed channels will be allocated to different users to avoid the co-channel interference. When two users work in D2D mode, they can reuse the licensed uplink channels and share the unlicensed channels with the WiFi APs. Therefore, it is named as D2D on unlicensed channels (D2D-U), where the transmitter and receiver are denoted as D2DT and D2DR, respectively [13].

2.2 Architecture and Technologies

As already said, the main goal of D2D communication is to reduce dependency on network infrastructure. This is supported by two key technologies: Proximity Services (ProSe) and Group Communication [14].

1. **Proximity Services (ProSe)**, introduced in 3GPP Release 12, enable devices in physical proximity to discover each other and communicate directly. ProSe encompasses D2D discovery and direct communication. During D2D discovery, a User Equipment (UE) finds another UE nearby, either through the Evolved Packet Core (EPC) or directly between the UEs. For direct communication, the UEs that have discovered each other can communicate directly via the LTE air interface without routing signals through the eNodeB (eNB) and the core network. This method reduces network congestion and allows devices to communicate even when network coverage is absent. Notably, device discovery is not always necessary for direct communication since it can also happen through broadcasting.

2. **Group Communication** is essential for meeting public safety needs, as it allows for services like one-to-many calling, which can quickly distribute messages to large groups. The 3GPP standardization efforts have significantly emphasized improving group communication techniques. In this type of communication, all user equipment (UE) in a group receives the same downlink stream, which helps to optimize overall resource use. To make group communication more efficient and adaptable, 3GPP has focused on enhancing broadcast and multicast techniques. This type of communication can be facilitated through Device-to-Device (D2D) communication or through existing or improved 3GPP multimedia broadcast/multicast services (MBMS) in LTE/LTE-A networks.

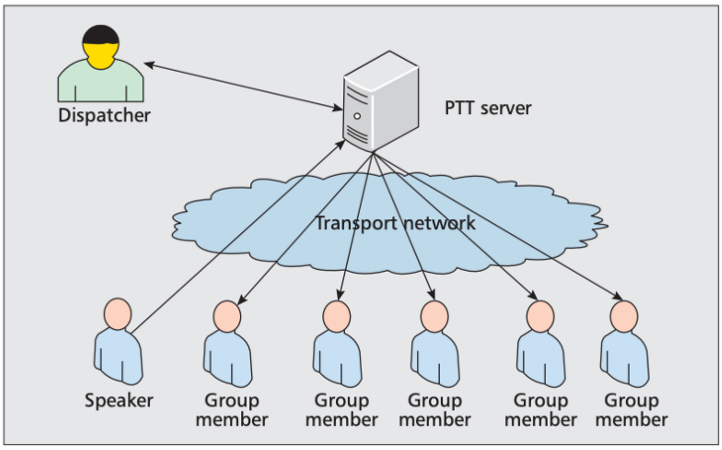


Figure 2.2: *An example of group communications in a public safety network (from [14]).*

2.2.1 D2D Architecture

The ProSe architecture includes several components:

- **ProSe App Server**: Provides ProSe capabilities for public safety and commercial use cases, directly communicating with applications in UEs.
- **ProSe UE App**: An application within the UE that utilizes ProSe capabilities.
- **ProSe Function**: Acts as a reference point for the ProSe App Server, EPC and UEs, handling verification, authorization and configuration of UEs. It also facilitates EPC-level discovery for direct communication.

A UE (user equipment) that wants to use ProSe must first contact the ProSe function through the logical interface named PC3 to get authorization and security parameters. After the discovery request and response message exchange via PC3 is completed, the UE can start the direct discovery process to find other UEs with ProSe capability in their proximity using the PC5 interface. When two (or more) ProSe-enabled UEs have discovered each other, they can start direct communication over the direct link between them. The physical interface between two ProSe UEs is called sidelink. The sidelink transmission is based on multicasting with no hybrid automatic repeat request (HARQ) feedback. Instead, each medium access control (MAC) protocol data unit (PDU) is re-transmitted three times with a different redundancy version for each transmission [15].

If a UE is out of the coverage area of eNB, but the signal strength is good enough for the UEs to communicate with each other, then they can exchange data directly among themselves by creating an ad hoc network; here, UEs communicate using a lower level interface called PC5ah [16].

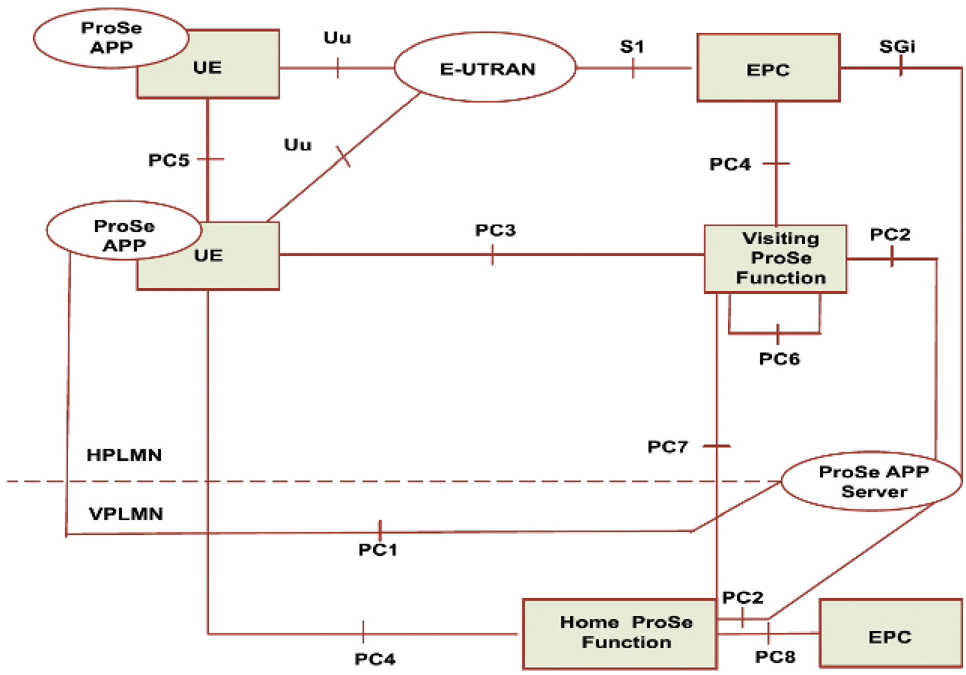


Figure 2.3: *D2D architecture in 3GPP specifications (from [12]).*

2.2.2 Scenarios for D2D Communication

D2D communication can be utilized in various scenarios including:

- in-coverage,
- out-of-coverage,
- partial coverage situations.

In-coverage refers to when UEs are within the eNB coverage area, out-of-coverage is when they are outside this area and partial coverage involves some UEs within and some outside the coverage area.

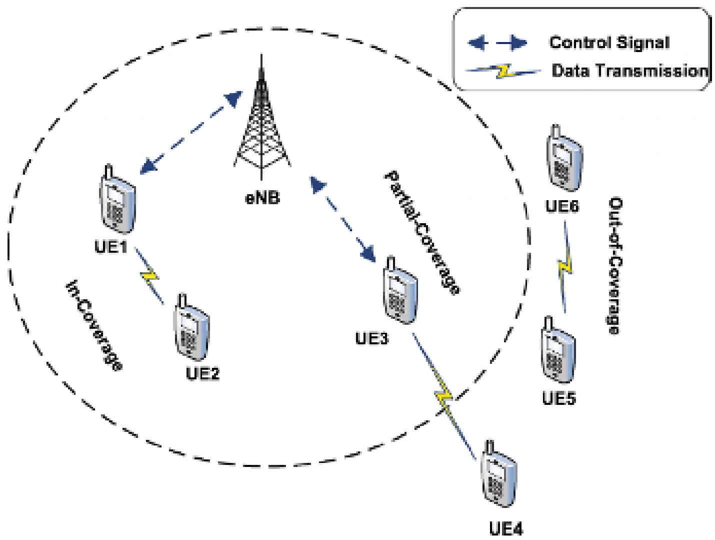


Figure 2.4: *D2D Communications scenarios supported in 3GPP Release 12 (from [12]).*

Figure below also gives an illustration of possible D2D use cases and the potential benefits.

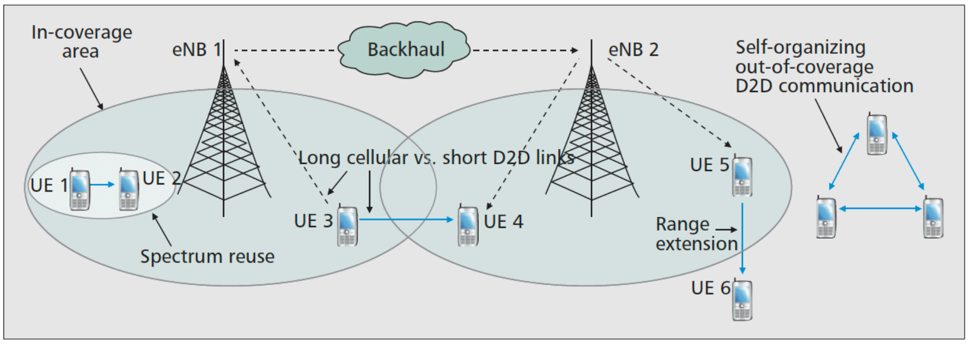


Figure 2.5: *Illustrations of possible D2D use cases and potential benefits (from [10]).*

2.2.3 Channels and Modulation techniques

In developing LTE standards, it was decided that User Equipments (UEs) would transmit on uplink LTE frequencies using SC-FDMA and OFDMA modulation and uplink resources [17]. Understanding the selection of these uplink resources involves introducing sidelink channels and resource pools. Release 12 of the LTE standard introduces sidelink as a new interface for D2D communication and discovery, incorporating transport and physical channels:

- **Physical Sidelink Shared Channel (PSSCH)**: Carries sidelink data for multiple UEs.
- **Physical Sidelink Control Channel (PSCCH)**: Carries sidelink control information (SCI) for PSSCH resource allocation.
- **Physical Sidelink Discovery Channel (PSDCH)**: Related to direct discovery services.
- **Physical Sidelink Broadcast Channel (PSBCH)**: Carries frame number, bandwidth and TDD (Time Division Duplex) UL/DL configuration.

Additionally, the sidelink physical signals defined include:

- **Sidelink synchronization signals**: Primary and secondary synchronization signals (PSS and SSS) are necessary for synchronizing UEs that are out of coverage and cannot use the primary and secondary synchronization signals emitted by the eNodeB. PSBCH occurs simultaneously with PSSS/SSSS (Primary and secondary Sidelink synchronization signals).

Resource Pools

A resource pool, defined by a subset of subframes and resource blocks within these subframes, allocates resources for sidelink data transmission and control. The PSCCH period ranges from 40 ms to 320 ms. According to [18], the resource pool is further detailed as follows:

- **Subframe pools**: Defined by a bitmap where a '1' indicates an available subframe for sidelink operation.
- **Resource block pools**: Defined by three parameters: prb-Num, prb-Start and prb-End, specifying the start, end and number of resource blocks available, as it can be seen in Figure 2.6.

Transmission and Modulation
Both the PSCCH and PSSCH utilize resources from designated resource pools. The PSSCH uses either QPSK or 16-QAM modulation with one of the allowed coding rates. The modulation and coding scheme (MCS) information is included in the sidelink control information carried by the PSCCH. Since PSSCH carries data for multiple UEs, a closed-loop HARQ scheme cannot be used; instead, transmissions are repeated three times, although individual UEs might decode a transport block in fewer transmissions.

The PSCCH and PSSCH are allocated to physical resources within resource pools, following two transmission modes specified in LTE Release 12:

- **Transmission mode 1 (network-directed)**: When the UE is within coverage, the eNodeB dynamically assigns resources for D2D transmission, ensuring no collision between sidelink and uplink transmissions or between sidelink transmissions.
- **Transmission mode 2 (UE-selected)**: The UE independently selects resources for transmission, applicable in both coverage and out-of-coverage scenarios, choosing resources randomly to reduce collision risk.

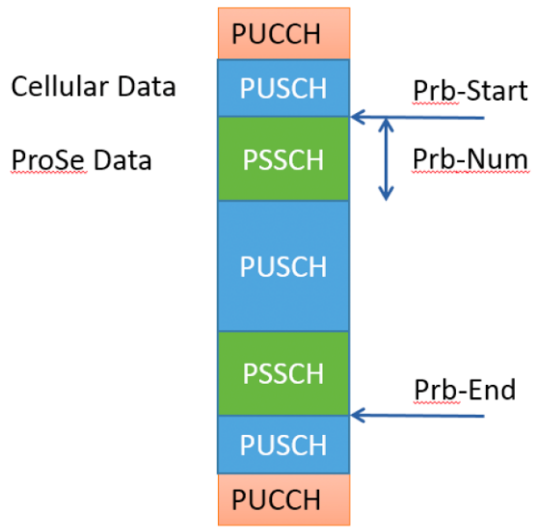


Figure 2.6: *Resource block pool in a subframe that is part of the subframe pool (from [18]).*

2.2.4 Features of D2D Communication

To facilitate efficient direct transmission between UEs, synchronization, discovery and direct communication are essential features in the 3GPP specifications:

- **D2D Synchronization** involves several steps:
  1. When a UE powers on, it synchronizes to a synchronization signal (PSS/SSS) from an eNB if detected.
  2. If no eNB signal is detected but signals from other UEs are found, it synchronizes to those signals.
  3. If no signals are detected at all, the UE becomes a D2D synchronization source itself, transmitting synchronization signals and PSBCH.

The first scenario happens when the UE is within the coverage area of an eNB, while the other scenarios occur when the UE is outside the eNB coverage area. In the first scenario, if a UE is synchronized with an eNB but detects a synchronization signal and PSBCH from another UE (indicated by the source type in the synchronization signal), it realizes that there is an out-of-coverage UE acting as a synchronization source. As a result, it becomes a D2D synchronization source itself, allowing the out-of-coverage UEs to synchronize with the eNB.

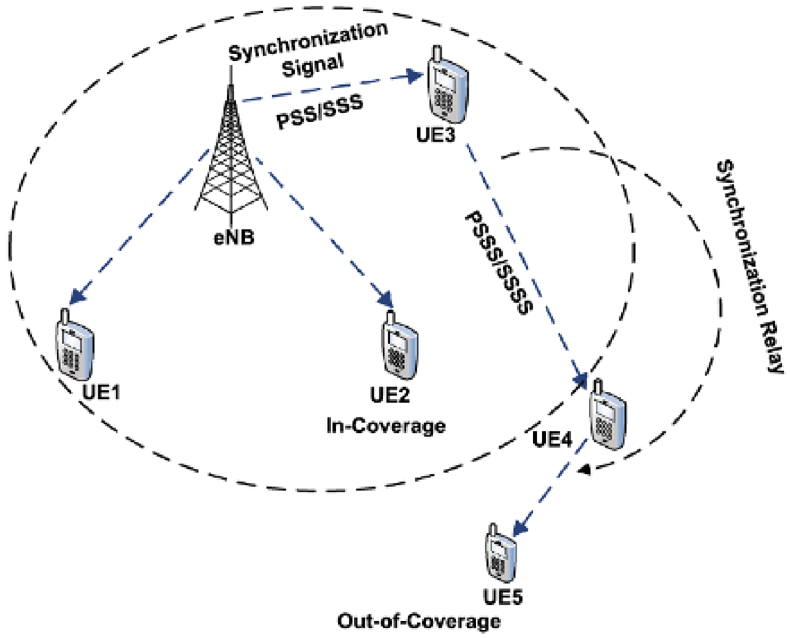


Figure 2.7: *D2D Synchronization (from [12]).*

- **D2D Discovery** is primarily an application layer procedure where specific messages are exchanged between UEs using the PSDCH at the physical layer. It can be network-assisted (EPC-level) or direct, where UEs physically close to each other detect and identify one another using E-UTRAN signals. Direct discovery has two models:

  1. Model A ("I am here"): The announcing UE periodically broadcasts discovery messages. The process involves monitoring and announcing. UEs broadcast information to discover nearby devices, which other UEs within proximity can use.

  2. Model B ("who is there?" / "are you there?"): The discovering UE broadcasts a message indicating the information it needs. Another UE that receives this message responds with the relevant information.

  Transmission and reception of discovery packets vary in complexity. Smaller packets are easier to transmit but offer limited information, while larger packets provide more information but are challenging to transmit periodically [19] [20] [21].

- **Direct Communication** involves exchanging control signals via the PSCCH, while data is transmitted using the PSSCH through the PC5 interface at Layer 1. Direct communication between UEs occurs through broadcast without feedback channels at Layer 1. At Layer 2, destination UE identifiers (UE ID) are used for unicast transmission, and Group ID is used for groupcast transmission [22].

2.3 D2D Use Cases and Challenges

Apart from public safety, the use cases for D2D communication are diverse:

- **IoT and Wearables**: IoT devices and wearables would especially benefit from short D2D links, improving power efficiency by connecting to a relay UE. This can be within or outside cell coverage or in coverage-enhanced mode, with relaying being bi- or uni-directional, as shown in Figure below. There are still several open issues regarding D2D and UE relaying for cellular IoT. For example, it is clear that, with UE relaying, the devices willing to operate as relays consume more power than the remote UEs [23].

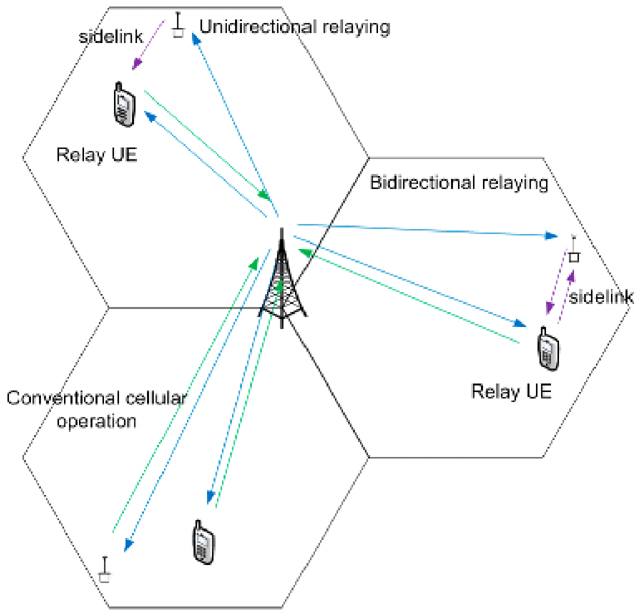


Figure 2.8: *D2D Relaying variants for cellular Internet of Things (IoT) devices (from [15]).*

- **Security in Data Communication**: D2D communication enhances security by enabling data transfer between trusted nearby devices, avoiding potentially insecure third-party databases.

- **Maritime Communication**: 3GPP Release 16's LTE-Maritime system provides seamless communication up to 100 kilometers offshore, supporting voice and data communication between vessels using D2D technology.

- **Vehicle-to-Vehicle (V2V) Communication**: Allows direct communication between vehicles, providing warnings about lane changes to prevent accidents. This extends to Vehicle-to-Everything (V2X) communication, including vehicle-to-pedestrian, vehicle-to-infrastructure and vehicle-to-network interactions, enabling advanced features like vehicle platooning and automated driving [24].

- **D2D Offloading**: Enables opportunistic data offloading, reducing overall network load and conserving spectrum resources. UEs with poor channel conditions can transfer data to neighboring UEs with better conditions, improving user experience and reducing network congestion [25].

- **Indoor Installation and Positioning**: Enhancements in D2D communication aim to improve location services and mission-critical voice services for LTE devices in indoor environments, addressing multi-path propagation challenges.

- **Multi-user Cooperative Communication (MUCC)**: in heterogeneous networks, a supporting user with strong signals helps a benefited user with weak signals, improving communication through LTE-A D2D [26].

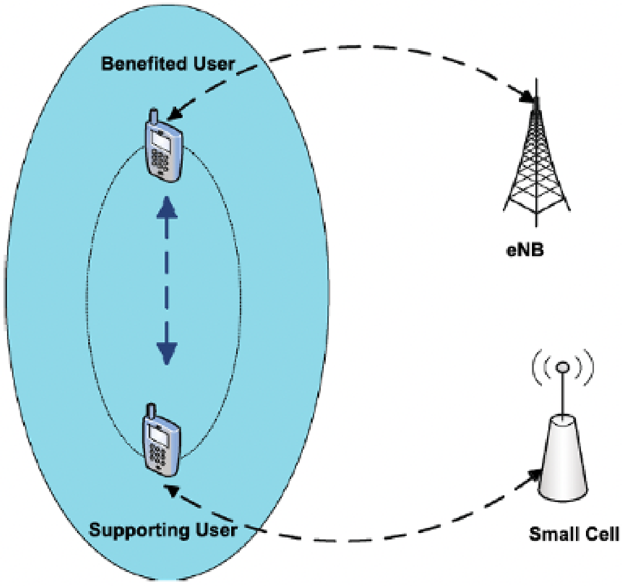


Figure 2.9: *MUCC communication (from [26]).*

2.3.1 Problems and Challenges

Device-to-Device (D2D) communication promises enhanced spectrum efficiency, increased data rates and improved coverage for 5G networks. However, several challenges need to be addressed for its successful integration into 5G networks:

- **User Mobility**: Managing user mobility involves efficient handoff selection, categorized into horizontal (within the same network) and vertical (across different networks) handoffs. Vertical handoffs in dense heterogeneous networks is influenced by controlling and signaling frequencies, which can deteriorate the signal-to-noise ratio (SNR) and performance. Developing standardized handover criteria is essential to minimize energy consumption and enhance device longevity [27].
- **Synchronization**: Achieving complete synchronization in in-coverage scenarios is difficult due to multiple D2D links and varying distances from eNBs, causing confusion over uplink and downlink timing. In out-of-coverage scenarios, periodic transmission of synchronization signals from UEs increases power consumption, making discovery more challenging if the target device is not synchronized.
- **Discovery Mechanisms**: Discovery signals can be sequence-based or packet-based, with complexities in centralized (EPC-level) or distributed (direct discovery) mechanisms. Continuous transmission of pilot signals in direct discovery consumes power and causes interference, necessitating standardized scheduling to avoid interference.
- **Hybrid Automatic Repeat Request (HARQ):** Combining ARQ and forward error correction [28], HARQ can be direct (between devices) or indirect (involving the eNB). Integrating HARQ into D2D communication can enhance robustness but increases network overhead.

- **Resource Allocation**: Effective resource management and reuse are crucial for network scalability, reduced power consumption and minimized interference. This is complex in dense networks with multiple base stations, requiring versatile resource allocation strategies.
- **Mode Selection**: D2D users choose between D2D mode and infrastructure mode based on channel quality and distance [29]. Frequent mode changes generate network overhead, making mode selection in dynamic conditions complex and necessitating further exploration.
- **Interference Management**: Interference occurs when D2D and cellular links share the same spectrum. Managing interference through power control, frequency selection and resource allocation is vital to ensure high communication quality and system efficiency [30].
- **Energy Efficiency**: to improve energy efficiency, techniques like power control, duty cycling and sleep mode can be used. Balancing communication quality and energy efficiency is crucial, as increasing power for better communication range raises energy consumption.
- **Security and Privacy**: Issues such as data theft, hacking and privacy violations are significant. Ensuring secure D2D communication requires robust key management, physical layer security and secure routing. Most existing proposals address specific issues but lack comprehensive protocols, underscoring the need for robust security protocols in 5G D2D communication [31].

2.4 Introduction to DECT NR+

2.4.1 A 5G technology

In the world of wireless communication technologies, Digital Enhanced Cordless Telecommunications New Radio (DECT-2020 NR), marketed as DECT NR+, represents a pivotal advancement ready to redefine the deployment and capabilities of local area wireless networks. Developed by the European Telecommunications Standards Institute (ETSI), DECT NR+ emerges as a robust standard specifically designed to meet the demanding requirements of Industrial Internet of Things (IIoT) applications. Its inclusion as part of the 5G standards in IMT-2020 technology recommendation made by the International Telecommunication Union Radio-communication Sector (ITU-R) [32] underscores its significance as a non-cellular 5G technology. Not relying on a cellular network, DECT NR+ is distinguished as the first and only standard approved as a 5G standard.

The 5G standard defines three separate use cases for 5G technologies to service: Enhanced Mobile Broadband (eMBB), Massive Machine Type Communication (mMTC) and Ultra-Reliable Low Latency Communication (URLLC). URLLC brings wireless technologies into mission-critical applications where failure is not an option, such as self-driving factory vehicles, high-speed robots working alongside human operators and critical infrastructure in buildings and cities. mMTC involves large networks with machine-type devices, connecting up to tens of billions of nodes that need to operate for many years from small batteries, transmitting small amounts of data. Typical use cases include collecting measurements from a vast number of sensors, such as smart metering, which requires a low-maintenance and low-cost autonomous network structure.

DECT NR+ is the first non-cellular radio standard to be recognized as a radio technology fulfilling the formal IMT-2020 5G requirements for both URLLC and mMTC use cases. This offering allows enterprise IoT customers to build their own low-cost private networks, filling a gap in the IoT ecosystem for large-scale machine-to-machine operations [33].

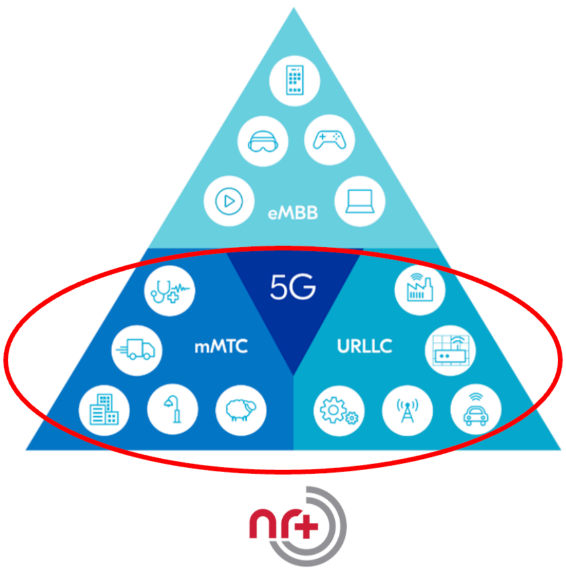


Figure 2.10: *DECT NR+ fulfills the requirements for both mMTC and URLLC*

Key features of DECT NR+ include highly reliable point-to-point and point-to-multipoint wireless connections, effectively serving as a valid alternative to traditional cable-based solutions [34]. This fits perfectly when it comes to set up large Internet of Things (IoT) networks in industries such as manufacturing, warehousing, energy and commercial buildings, where is crucial to have the ultra-high reliability and low latency of a wired (e.g., Ethernet) connection but without the physical constraints of the wires, even at a massive scale involving millions to billions of nodes.

DECT NR+ is built to be low-cost, autonomous, self-healing and decentralized, opening up many new opportunities for different industries. Its design emphasizes simplicity and accessibility, enabling organizations of various sizes to cost-effectively deploy and manage independent 5G wireless networks. By leveraging the license-exempt 1.9 GHz DECT band, DECT NR+ mitigates interference concerns while reducing operational costs associated with spectrum allocation and certification, making it an attractive proposition for medium-sized enterprises seeking sustainable connectivity solutions [35].

The inclusion of DECT NR+ in the IMT-2020 framework positions it uniquely within the 5G ecosystem, although it is distinct from conventional cellular and Wi-Fi technologies. Unlike traditional 5G networks that rely on infrastructure support from mobile network operators, DECT NR+ networks operate independently —meaning no base stations, no SIM cards and no subscriptions— offering inherent scalability and reliability through their self-organizing mesh architecture. This distinction enhances data security and operational autonomy, reduces dependency on external service providers and improves overall network efficiency by eliminating single points of failure [36]. However, even if DECT NR is not a cellular technology, its implementation shares fundamental similarities with cellular systems, including a robust radio physical layer with advanced modulation and coding schemes.

2.4.2 Frequency Used

DECT NR+ leverages the 1.9 GHz frequency band, significantly influencing its performance and suitability for various industrial applications. In Europe, the commonly used spectrum allocation is 1880 MHz to 1900 MHz, but bands 1900 MHz to 1920 MHz and 1910 MHz to 1930 MHz are also prevalent in many countries [37].

Benefits of the 1.9 GHz Frequency Band
One notable advantage of the 1.9 GHz frequency band is its relative lack of congestion. Unlike the heavily utilized 2.4 GHz band, the 1.9 GHz band experiences much lower levels of traffic, reducing the likelihood of interference and leading to more stable and reliable network performance. Additionally, the 1.9 GHz band offers superior coverage and penetration capabilities. Lower frequency bands like 1.9 GHz are better at penetrating obstacles such as walls and machinery, ensuring robust connectivity in industrial environments. Another significant advantage is that the 1.9 GHz band is license-exempt, allowing organizations to operate their DECT NR+ networks without securing costly and time-consuming licenses, reducing deployment costs and making it attractive for medium-sized enterprises and cost-sensitive deployments [38].

Disadvantages of the 1.9 GHz Frequency Band
Despite its benefits, the 1.9 GHz frequency band has drawbacks. Initial costs may be higher compared to the 2.4 GHz band due to the need for specialized equipment. Limited compatibility is another challenge: not all devices support this frequency, which can restrict connectivity options and require additional investment in compatible hardware. Higher power consumption is also a potential disadvantage, as operating at this frequency may require higher transmission power, increasing energy consumption — critical for battery-operated devices or energy-efficient applications. Additionally, even if the 1.9 GHz band supports higher bandwidth than many other frequencies, it still offers less bandwidth than the 2.4 GHz band, making the latter more suitable for applications requiring extensive data throughput.

2.5 Network topologies and Technologies used

DECT NR+ supports three primary network topologies: point-to-point, star and mesh. Each of these topologies serves different purposes and offers unique benefits, ensuring versatility and efficiency in various deployment scenarios.

2.5.1 Point-to-Point and Star Topologies

In point-to-point and star topologies, the network is structured simply and efficiently. Nodes with a backhaul to the internet are defined as sink nodes. These sink nodes act as gateways to the internet, selecting operating frequencies and initiating beacon transmissions to signal their connectivity to the external world. All other nodes are characterized as leaf nodes, which connect directly to the sink or sinks. This configuration ensures a centralized control model, facilitating efficient and reliable data transfer.

2.5.2 Mesh Topology

The mesh topology in DECT NR+ is more sophisticated and is defined as a partial mesh, clustered tree topology. This means devices are organized into individual cluster tree topologies, known as clusters. A cluster tree topology is essentially a star topology with branches. All clusters are interconnected to form the overall mesh network, described as partial mesh because not all nodes are connected to each other, unlike in a full-mesh topology [39].

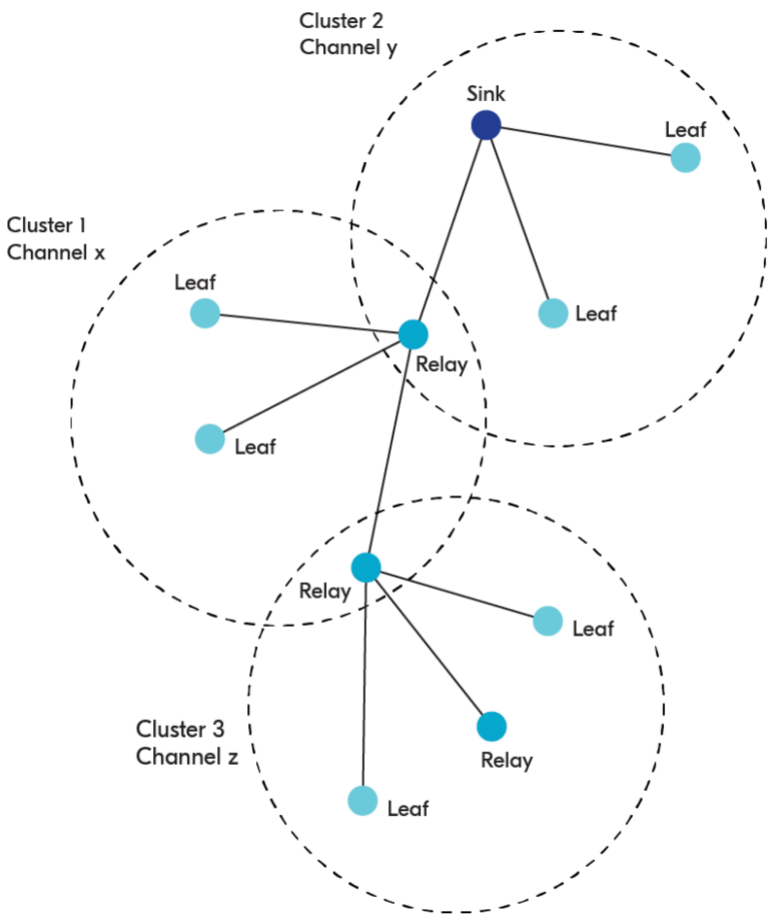


Figure 2.11: *Example of a DECT NR+ mesh network with one sink and three clusters operating on three different frequency channels. In cluster 3, note that the relay nodes do not necessarily have to be forwarding data - the role just indicates their ability to do so (from [39]).*

Roles of Nodes in Mesh Networks
In DECT NR+ mesh networks, nodes can take on various roles, including sink nodes, relay nodes and leaf nodes. Each role has specific responsibilities and functions within the network:

- **Sink Nodes**: These nodes serve as gateways to the internet. They are responsible for selecting operating frequencies and initiating beacon transmissions to signal connectivity. Networks can have multiple sinks to enhance redundancy and reliability.
- **Relay Nodes**: These nodes extend the network by routing messages between clusters and to leaf devices within their cluster. It is worth noting that clusters can also use different channels within the specified operation frequency, so relay nodes can continuously scan frequency channels to find the one with the least interference, instructing other nodes in the cluster to switch channels accordingly.
- **Leaf Nodes**: Positioned at the network's periphery, leaf nodes are the outermost points of the network and can only send data.

Scalability and Efficiency
DECT NR+ mesh networks are designed to support massive machine-type communication (mMTC) use cases. The system is scalable to a very high number of devices, with routing based on cost value, eliminating the need for centralized routing tables. Key features that support scalability and efficiency include:

- **Autonomous Decision-Making**: All radio devices can route data, with decisions based on local conditions. Devices take local decisions on radio resources, such as Hybrid ARQ and modulation and coding schemes.
- **Dynamic Mode Switching**: Radio devices can switch between sink, relay or leaf mode, based on local decisions.
- **No Central Coordinators**: This enables massive network scalability, as there is no need for central coordination.
- **Multi-Channel Operation**: Radio devices can operate on multiple radio channels, further enhancing performance and reducing interference.

Self-Organizing and Self-Healing
The mesh topology in DECT NR+ supports a self-organizing and self-healing network. Device roles are appointed autonomously based on the network, and each device decides its next hop individually based on available routes toward the sink, forming clusters autonomously. Furthermore, each node can function as an access point with a direct connection to the internet, and nodes can change roles based on the needs of the network, enabling it to adjust dynamically to congestions. This means that each radio device can act as a node transmitting a message, as a node forwarding any message from another radio device or as a node being the destination of a message. Each radio device can communicate directly (device to device) or, if not in range, indirectly - via other radio devices establishing a communication route - with each other which minimizes the probability of outage. This property eliminates single points of failure in the network and autonomously resolves high-traffic situations that can occur in dense IoT networks.

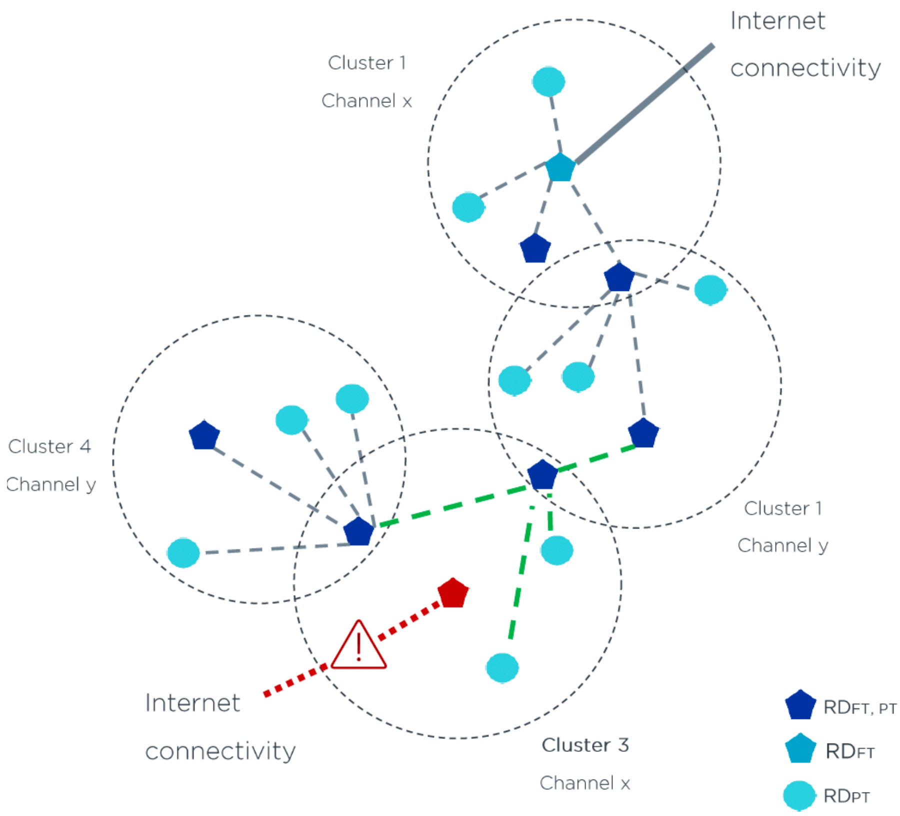


Figure 2.12: *Upon disconnection of the Sink Radio Device, the RDs automatically re-route to the top Sink node, ensuring seamless communication without application layer intervention. In FT mode, the RD provides connection info (corresponds to Sink), while in PT mode RDs select FT mode RDs for association (corresponds to Leaf). RD in FT and PT modes corresponds to a Relay node (from [40]).*

Formation of Clustered Tree Topology
The formation of a clustered tree topology involves the following steps:

1. **Sink Node Initialization**: An RD (Radio Device) with internet connectivity, operating in sink mode, selects an operating frequency and initiates beacon transmissions to signal its route to the external world. This enables other RDs to detect and associate with it.

2. **Beacon Detection and Association**: RDs detecting a beacon from another RD evaluate the connection based on the information and signal quality. Based on this evaluation, RDs make an independent decision on which node to associate with.

3. **Autonomous Association**: RDs monitor their neighborhood and may autonomously initiate an association process towards another RD based on routing cost, continuously optimizing the network.

2.5.3 Goals and Technologies of DECT NR+

The DECT NR+ standard is designed to address the evolving needs of modern industrial and IoT applications by focusing on several key objectives:

- Scalability, supporting networks from a few devices to a million within a km^2, accommodating dense environments without performance loss.

- Power saving, extending battery life for devices operating long-term without frequent replacements.

- Low latency, ensuring end-to-end latency of less then 50 ms on the application layer and less then 1 ms on the radio interface for efficient data exchange in high-demand scenarios.
- Reliability, ensuring that data is transmitted accurately and consistently, with an ultra-reliability requirement of over 99.99% packet delivery, making it suitable for mission-critical applications.

The techniques utilized to achieve these four objectives will now be examined. Additionally, the DECT NR+ standard promises to impressive indoor wireless coverage and an outdoor range of up to 2 km, categorizing it as a Wireless Neighborhood Area Network (WNAN) rather than a Wide Area Network (WAN) like cellular IoT. Initial products can achieve data rates of up to 3.4 Mb/sec, with the standard enabling much higher data rates for future designs [41].

Scalability
DECT NR+ achieves scalability through several key mechanisms, enabling the creation of highly scalable and dense networks. Its autonomous and decentralized nature allows for efficient network management and expansion without centralized control. A major contributor to this scalability is the construction of DECT NR+ IDs. The 32-bit network IDs can create up to 16.7 million unique global networks and support 256 overlapping networks within the same radio area, allowing multiple networks to coexist without interference. Additionally, the 48-bit radio device IDs accommodate up to 4 billion individual devices in a single network and 65 thousand within radio communication distance, supporting vast numbers of devices in dense environments such as industrial settings and smart cities.

DECT NR+ also incorporates advanced techniques from the cellular world to maximize bandwidth utilization. It employs Cyclic Prefix Orthogonal Frequency Division Multiplexing (CP-OFDM) combined with Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA). These techniques enable efficient bandwidth utilization by overlapping channel divisions, thereby saving bandwidth and allowing for higher data rates.

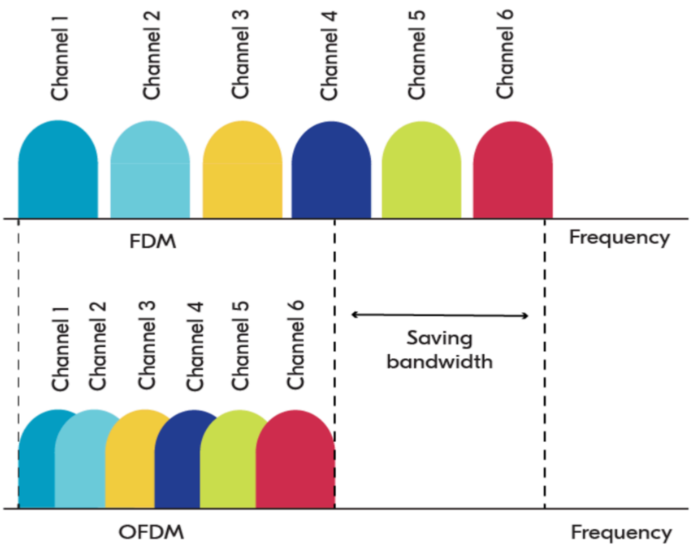


Figure 2.13: *Illustration of bandwidth efficiency achieved through OFDM technique ([39]).*

OFDM is a modulation technique that transmits high data rates by dividing the available spectrum into multiple orthogonal subcarriers, each carrying a portion of the data. This division allows efficient use of the spectrum, minimizing interference and maximizing data rates. FDMA assigns different users separate frequency bands within the available spectrum, while TDMA divides the channel into time slots, allocating one or more slots to each user for transmission. The combination of FDMA and TDMA in DECT NR+ ensures non-overlapping channels in the frequency domain and non-overlapping transmission slots in the time domain, providing efficient bandwidth utilization and reliable data transmission.

This combination allows DECT NR+ to use minimal bandwidth while maintaining reliable data transmission, even if the network scales up to accommodate more devices. So, while OFDM itself is not a multiple access scheme, it is used as the modulation technique within the DECT-2020 system, which employs a combination of TDMA and FDMA for multiple access.

Reliability

One of the primary methods DECT NR+ employs to achieve high reliability is the hybrid automatic repeat request (HARQ) methodology. HARQ combines forward error correction (FEC) and automatic repeat request (ARQ) error-control techniques to enhance the reliability of packet transmissions. When a packet is only partially received or not received at all, the receiver buffers the received portion and notifies the transmitter to resend the missing data. This process continues until the complete packet is successfully reconstructed. Since HARQ operates at the physical layer, higher protocol layers remain unaffected by re-transmissions, which conserves resources and improves overall system efficiency.

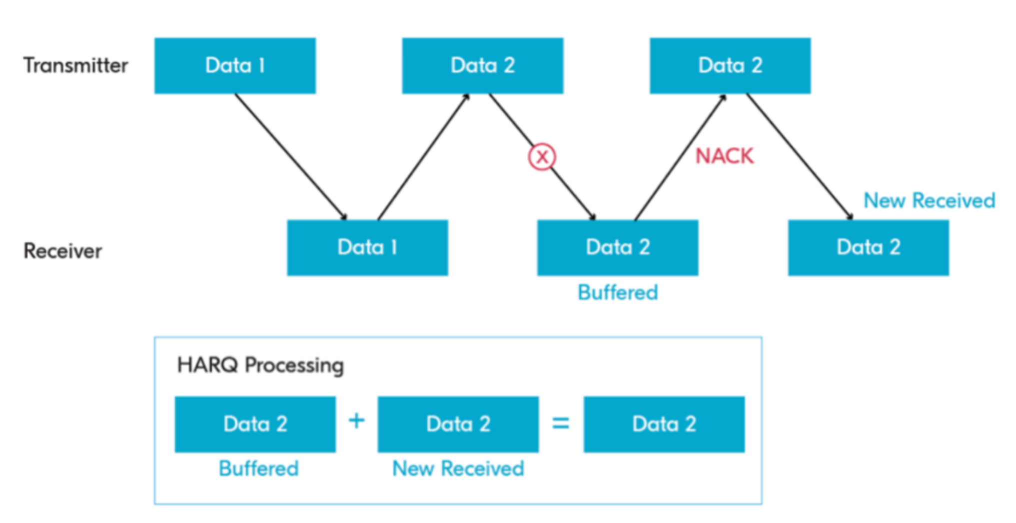


Figure 2.14: *Illustration of the HARQ Protocol: The transmitter sends DATA 1, which is correctly received, prompting an ACK. When DATA 2 is sent, the receiver does not receive the entire data and buffers what it has. It sends a NACK to the transmitter, which then re-sends the missing data to complete the transmission (from [39]).*

Additionally, DECT NR+ uses relay nodes to scan frequency channels and detect the ones with the least interference. Relay nodes can then instruct other nodes in their cluster to switch channels, optimizing interference avoidance. This capability ensures that the network maintains reliable and low-latency communication, even in environments with high device density.

These techniques provide ultra-reliable communication, fulfilling the requirement of over 99.99% packet delivery. This high level of reliability ensures that DECT NR+ can support mission-critical applications where failure is not an option.

Power Saving
DECT NR+ incorporates several innovative features designed to achieve significant power savings, ensuring long battery life for devices. One of the key power-saving mechanisms in DECT NR+ is the management of radio usage by relay devices within their clusters. Relay devices coordinate with leaf nodes, informing them when uplink data can be forwarded. This scheduling allows relay devices to enter sleep mode outside of these designated periods, reducing overall power consumption. Leaf nodes are informed via beacon messages sent by relay devices, indicating when they need to listen for downlink data. These beacon periods can vary from 10 milliseconds up to 32 seconds, providing the flexibility to balance between low latency and extended sleep periods for power savings.

Furthermore, DECT NR+ integrates many core features, such as routing and re-transmitting packets, directly into the radio stack level. This approach enables multi-core systems to maintain reliable radio communications without the need to wake up additional cores. By keeping the device's power usage to a minimum, this feature significantly contributes to energy efficiency.

Additionally, DECT NR+ devices utilize autonomous dynamic power control of transmissions, varying from -40 dBm up to +23 dBm. This power control limits overhearing in dense networks and achieves low power consumption by adjusting the transmission power based on the proximity of the receiving device. This means that if the receiving device is nearby, there is no need for high transmission power, further enhancing the overall energy efficiency of the network.

Low Latency
DECT NR+ meets ultra-low latency requirements through several advanced techniques: one of them consists in the supporting for multiple input and multiple output (MIMO) operations within the same 10 ms radio frame, facilitating high-speed communication. Each 10 ms frame is divided into 24 time slots, each with a duration of approximately 0.41667 ms. These slots can be aggregated or subdivided into sub-slots, allowing for efficient multiplexing of data, control signals and synchronization information.

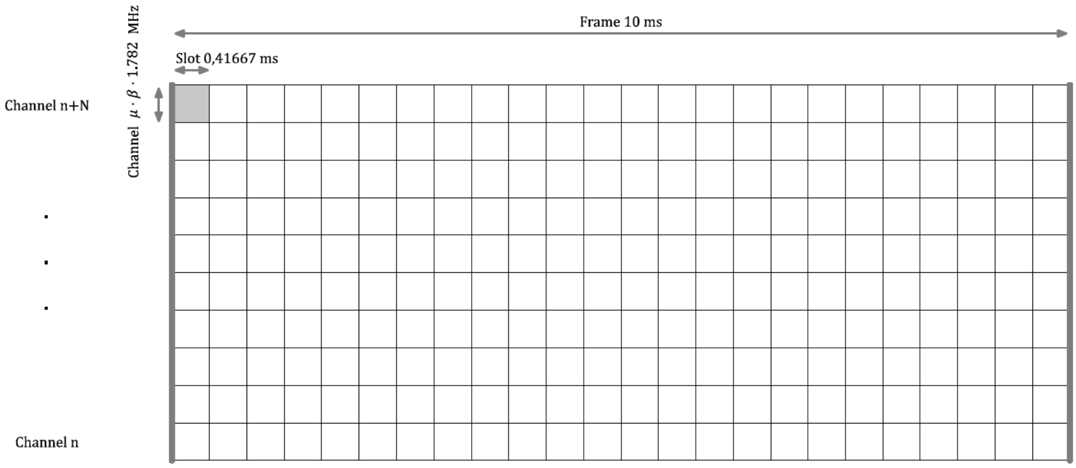


Figure 2.15: *DECT-2020 frame structure (from [42]).*

This structure enables multiple devices to transmit and receive data within the same frame without interference. The low mode-change time between receiving and transmitting modes is a key factor in achieving this goal, with less than 1 ms radio layer latency in a single link. In mesh topologies, DECT NR+ maintains low latency across radio links, minimizing the end-to-end latency throughout the network.

2.6 Network Protocol Layers

2.6.1 Pysical Layer

The physical layer in DECT NR+ is responsible for the transmission of MAC PDUs (Protocol Data Units) through the Physical Control Channel (PCC) and the Physical Data Channel (PDC). These channels are defined by how they transfer information over the radio interface within a single transmission packet. The physical layer carries out several crucial functions to ensure reliable data transport. It detects errors on the physical channels and indicates these errors to higher layers, and performs Forward Error Correction (FEC) encoding and decoding. Hybrid ARQ soft-combining, rate matching of coded data to physical channels and mapping of coded data onto physical channels are also managed by this layer.

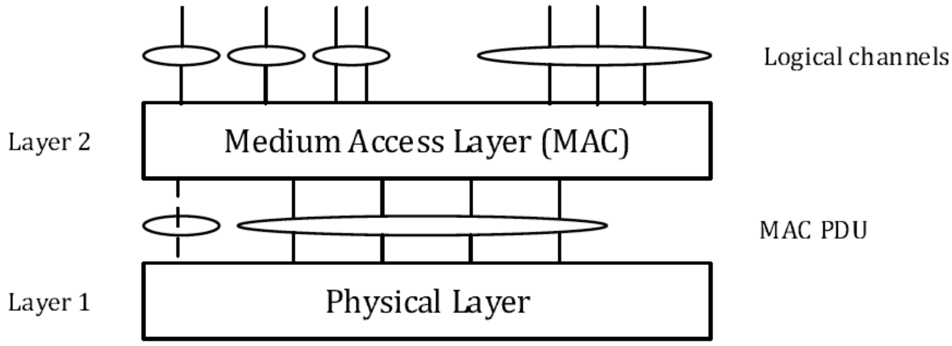


Figure 2.16: *Illustration of the DECT-2020 radio interface protocol architecture highlighting the Physical layer (PHY) and its interface with the Medium Access Control (MAC) layer. The circles between various layers and sub-layers represent Service Access Points (SAPs) (from [42]).*

Modulation and demodulation, frequency and time synchronization and radio characteristics measurements are key functions that facilitate efficient data transmission. The physical layer also employs advanced techniques such as Multiple Input Multiple Output (MIMO) antenna processing, transmit diversity (TX diversity) and beamforming to enhance performance [42].

The physical channels, PCC and PDC, utilize various modulation schemes including BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM and 1024-QAM. The multiple access scheme is based on Time Division Duplex (TDD) combined with Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA). This scheme ensures non-overlapping channels in the frequency domain and non-overlapping transmission slots in the time domain. The modulation within transmitted packets is Orthogonal Frequency Division Multiplexing (OFDM) with a Cyclic Prefix (CP).

DECT-2020 transmission packets consist of a Synchronization Training Field (STF), a Data Field (DF) and a Guard Interval (GI). The STF creates a repetitive time domain pattern for receiver gain, timing and frequency acquisition. The DF carries the Demodulation Reference Signal (DRS), PCC and PDC, while the GI at the end of the packet facilitates transmission-reception turnaround and prevents overlapping transmissions from adjacent TDMA timeslots. It separates different transmissions, ensuring clear and efficient communication.

The PCC identifies whether the packet is a Beacon or a Data transmission packet, and the PDC carries the actual data.

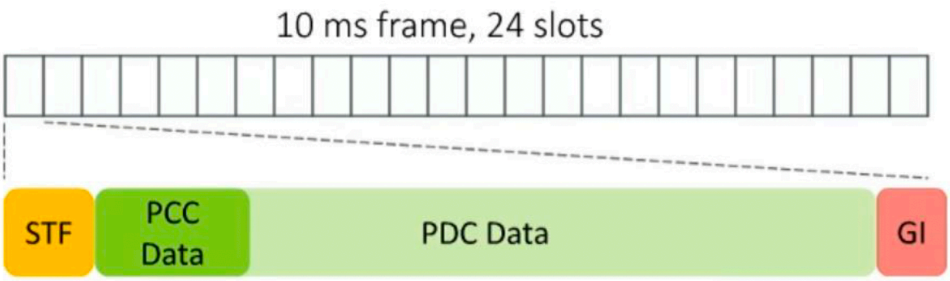


Figure 2.17: *Message Structure in the Physical Layer (from [40]).*

2.6.2 MAC Layer

The MAC layer in DECT NR+ manages several essential functions to ensure efficient communication. It handles dedicated traffic channels (DTCH) and multicast/broadcast traffic channels (MTCH) for higher layer data transfer. The MAC layer processes various transport channels, including paging and broadcast (PCH/BCH), dedicated (DCH) and random access channels (RACH). These transport channels are mapped to the Physical Data Channel (PDC) within physical layer packets. The MAC layer also provides information on used spatial streams (data streams transmitted and received) and the contents of the Physical Layer Control Field to the physical layer, which is mapped to the Physical Control Channel (PCC) of the physical layer packet.

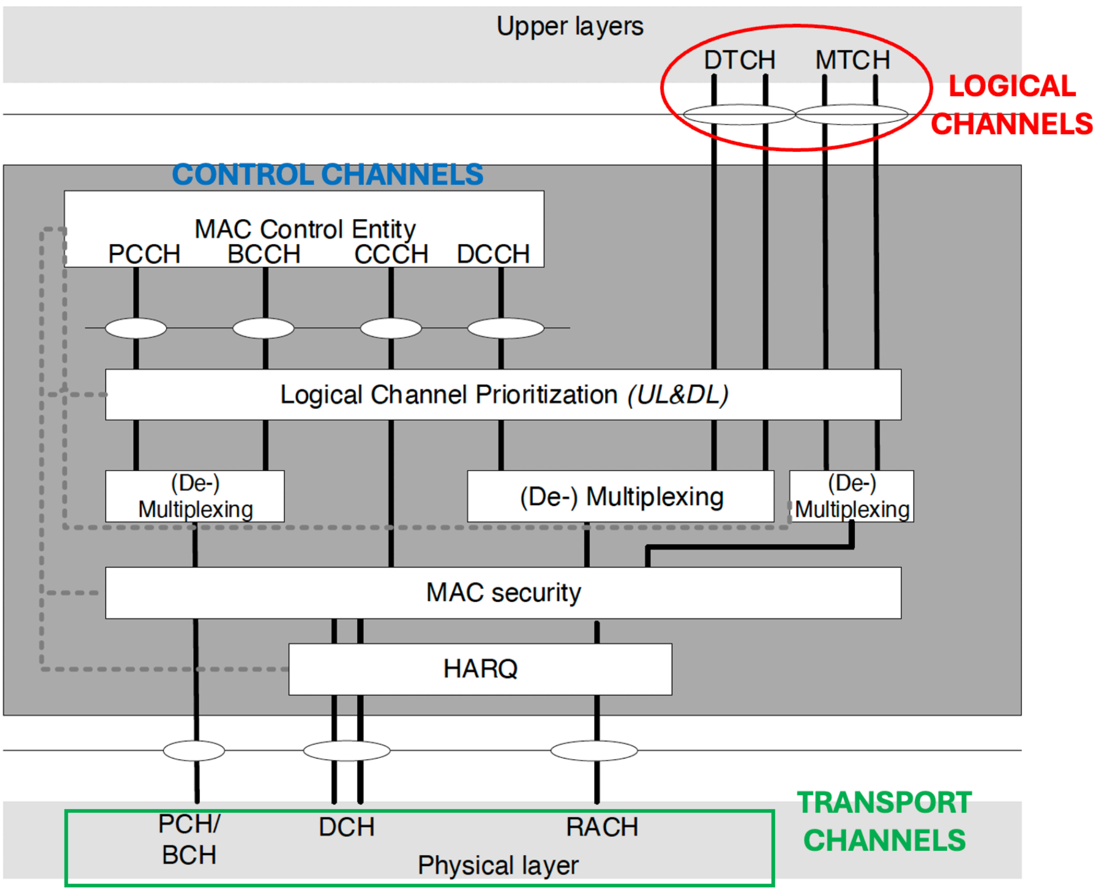


Figure 2.18: *MAC structure overview*

The MAC layer uses a 32-bit network ID, with the first 24 bits uniquely identifying a DECT-2020 network and the last 8 bits selected locally to minimize collision with other networks. This ID is periodically transmitted in a beacon frame, allowing other devices to detect the network. The MAC layer offers services to upper layers, including data transfer and radio resource allocation, and relies on the physical layer for data transfer services and measurements. It supports various control functions, such as local radio control, paging transmission control, broadcast control, random access control, beacon scanning control and connection configuration control. These functions are crucial for managing radio resources, controlling paging and broadcast transmissions, scanning operations and configuring connections.

In terms of transmission functions, the MAC layer handles paging and broadcast signaling, control signaling, radio resource management, logical channel prioritization and mapping between logical and transport channels. It also manages the multiplexing and demultiplexing of MAC SDUs (Service Data Units) to and from transport channels, error correction through hybrid automatic repeat request (HARQ), and provides security through integrity protection and ciphering [43]. Relay devices, or parent devices, are responsible for radio management within their clusters, using beacons to control radio resources. These beacons inform child devices when it is safe to send and receive traffic. A RD in FT mode (Fixed Terminal, analogous to a router) transmits periodic beacons to inform PTs (Portable Terminals or leaf devices) of their random access allocations, which can lead to scheduled allocations managed by the FT. Beacons also establish a common time base between the FT and PT MAC layers. DECT NR+ employs a listen-before-talk mechanism within random access to prevent transmission collisions [44].

2.6.3 DLC Layer

The Data Link Control (DLC) layer is responsible for the routing and transport service selection for the MAC layer. It provides essential segmentation and packet routing functions, ensuring that data is efficiently and accurately forwarded through the network. Routing within the DLC layer operates in two primary modes: upward routing and downward flooding. In upward routing, packets are forwarded to the parent device until they reach the sink. For downward flooding, packets from the sink are forwarded to all child relays, and relays continue forwarding to their child relays. If a relay has the destination device as a direct child, it sends the packet only to that leaf (see Figure 2.19). This flooding mechanism, which includes a hop limit for node-to-node communication, ensures resilience to network changes, allowing leaf devices to move to other relays instantaneously. The DLC architecture features a single routing service entity in each Radio Device (RD). Each RD operating in Portable Terminal (PT) mode has a single DLC entity set for its radio link with a Fixed Terminal (FT) mode RD. For an RD operating in both FT and PT modes (Relay node) within a mesh cluster, it maintains one DLC entity set for each RD that has associated with it. Additionally, RDs have a DLC entity set for broadcast communication, enabling comprehensive and flexible network connectivity [45].

(a) *Upward Routing* (b) *Downward Routing*

Figure 2.19: *DLC Network Routing (from [44]).*

2.6.4 CVG Layer

The Convergence (CVG) layer offers a suite of optional services along with a transparent procedure that bypasses these services if necessary. It plays a crucial role in identifying application layer data and is vital for application multiplexing, particularly when multiple applications utilize the stack simultaneously. Supporting endpoint multiplexing, it is possible to direct traffic for different applications.

The CVG layer provides adaptation functions that bridge the application layer protocols and the NR+ radio interface. It includes security services with features like ciphering and integrity protection, as well as transmission services such as segmentation and reassembly, re-transmission, flow control, lifetime control, duplicate removal and delivery order services [45].

In relay nodes, the CVG layer is not engaged, as the relaying is managed by the lower layers, bypassing the need for higher layer processing.

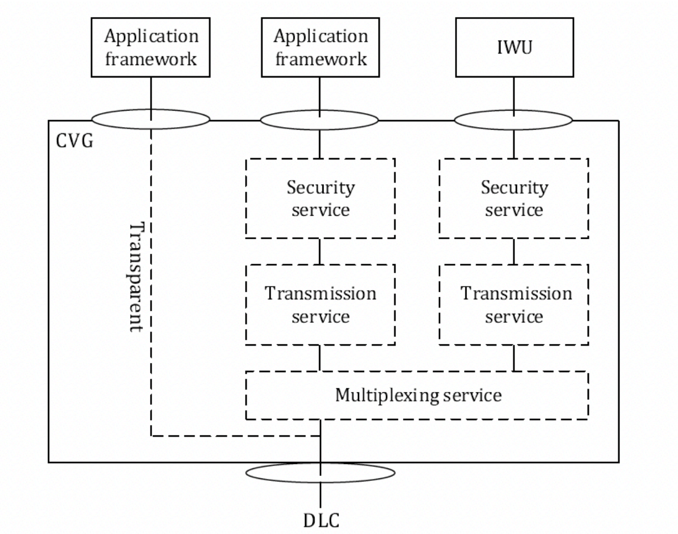


Figure 2.20: *CVG architecture (from [45]).*

2.6.5 Architecture Overall

Figure 2.21 depict an example of protocol architecture exploiting mesh topology and routing with IPv6 and non-IPv6 applications.

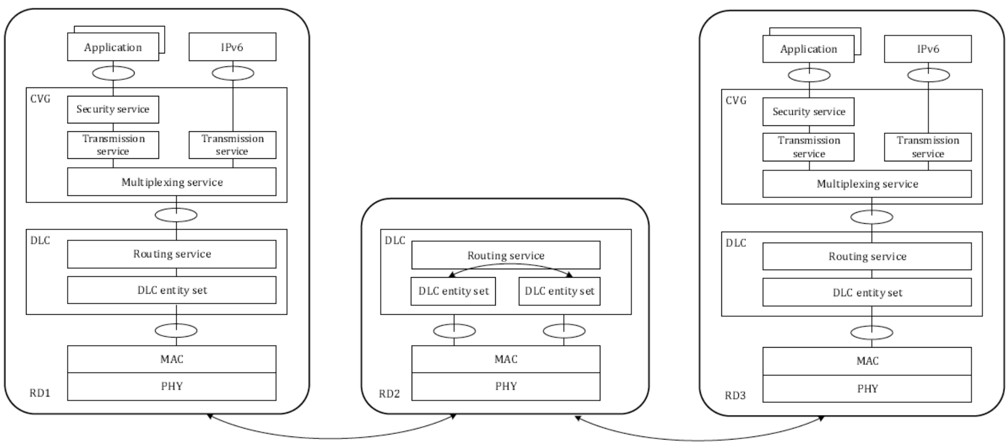


Figure 2.21: *Illustration of overall protocol architecture for mesh networking ([45]).*

2.7 DECT NR+ Use Cases

DECT NR+ is well-suited for a variety of applications across different sectors. All the analyzed features that the standard promises to fulfill make it a versatile solution for modern industrial and IoT applications. The following section explore the key use cases where DECT NR+ can be effectively applied.

- **Smart Industries / Asset Tracking**: DECT NR+ is at the forefront of revolutionizing industrial environments through its advanced capabilities in asset tracking and Industrial IoT (IIoT). In today's smart industries, the seamless wireless tracking and locating of tools, equipment and inventory are indispensable. The dense network topology of DECT NR+ efficiently supports numerous nodes, enabling comprehensive and cost-effective asset tracking. Furthermore, its low latency feature ensures real-time monitoring of critical assets, proactively enhancing worker safety by preventing hazardous situations. This technology plays a pivotal role in Industry 4.0, the era of industrial automation characterized by ultra-reliable low-latency IoT solutions [40].

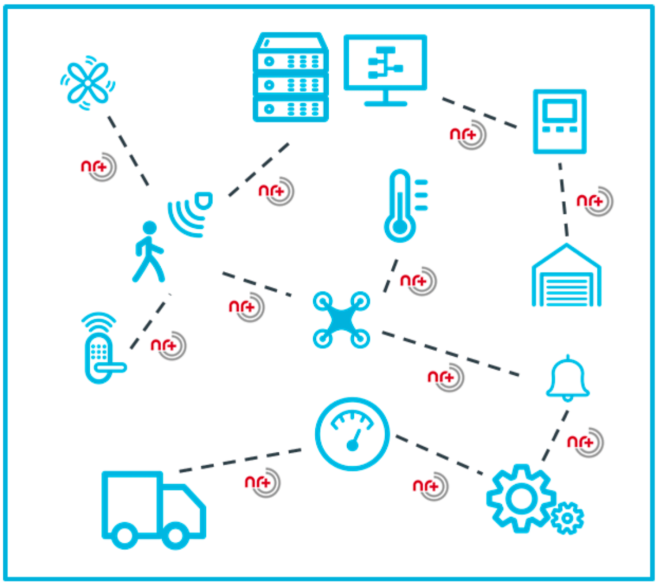


Figure 2.22: *Example of DECT NR+ network topology supporting real-time asset tracking and monitoring in smart industrial environments (from [40]).*

- **Smart Cities**: DECT NR+ excels in smart city applications, providing real-time monitoring and control for urban environments like airports, campuses and retail centers. Its ability to support diverse applications with minimal deployment costs and robust performance under high traffic conditions makes it an ideal solution for these settings.

  For example, smart street lights benefit from DECT NR+ with dynamic light control, significantly reducing energy consumption, while smart energy storage solutions balance energy supply and demand to improve overall energy management. Additionally, real-time diagnostics enable continuous monitoring for prompt issue resolution and optimal operation.

  Beyond street lighting, DECT NR+ supports other essential smart city applications including environmental monitoring devices and surveillance cameras, all of which can be integrated into a scalable city-wide network infrastructure. IoT in smart cities combats climate change and enhances urban living: traffic management systems, for example, reduce congestion and emissions by optimizing traffic flow. Additionally, parking space management solutions help efficiently locate available parking, minimizing search time and reducing vehicle emissions.

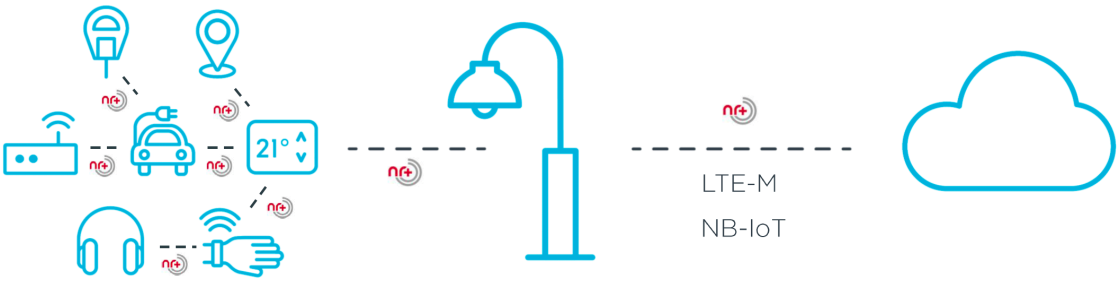


Figure 2.23: *DECT NR+ enabling real-time monitoring and control in smart cities, enhancing urban living with applications like dynamic street lighting and traffic management (from [40]).*

- **Pro-Audio Applications**: DECT NR+ brings significant advancements to professional audio applications, improving indoor radio performance, correcting errors effectively and supporting more users. These enhancements are crucial for PMSE (Programme Making and Special Events) microphones used in touring bands, recording studios, theaters and broadcasting. They also strengthen PMSE intercom systems, allowing for more users and flexible frequency management, perfect for dynamic environments like sports events and music festivals [46].

3 Experimental Setup

This chapter outlines the experimental setup used in the study, detailing the hardware and software tools employed, as well as the developed applications that enable device communication. The hardware used in the experiments includes the nRF9161 Development Kit (DK) and the Power Profiler Kit II, both developed by Nordic Semiconductor. These devices provide a robust platform for testing and development.

The software tools include nRF Connect for Desktop, nRF Connect SDK and Visual Studio Code (VS Code) with the nRF Connect for VS Code extension pack, also developed by Nordic Semiconductor. These tools are essential for configuring and programming the hardware.

Additionally, the chapter describes two applications: the Broadcast Application, a C-programmed application enabling simple broadcast of DECT NR+ messages for testing, and the Modem Shell Application, a *.hex* file functioning as a modem shell for the device, used for various purposes such as ping-based range tests and single link throughput testing. This setup ensures a transparent and reproducible methodology for the study.

3.1 Hardware Components

3.1.1 nRF9161 Development Kit

The nRF9161 Development Kit (DK) is a innovative hardware platform designed for developing applications on the nRF9161 System in Package (SiP). Notably, the nRF9161 is the world's first device to support DECT NR+, showcasing its advanced capabilities in the field of wireless communication. This development kit includes all essential circuitry, such as antennas and a Subscriber Identity Module (SIM) card holder, and provides developers access to all I/O pins and relevant module interfaces, making it a comprehensive tool for development and testing.

The nRF9161 DK features antenna interfaces for both Long-Term Evolution (LTE) and Global Navigation Satellite System (GNSS). Importantly, DECT NR+ utilizes the same onboard antenna as LTE, ensuring seamless integration and operation across different communication protocols [47].

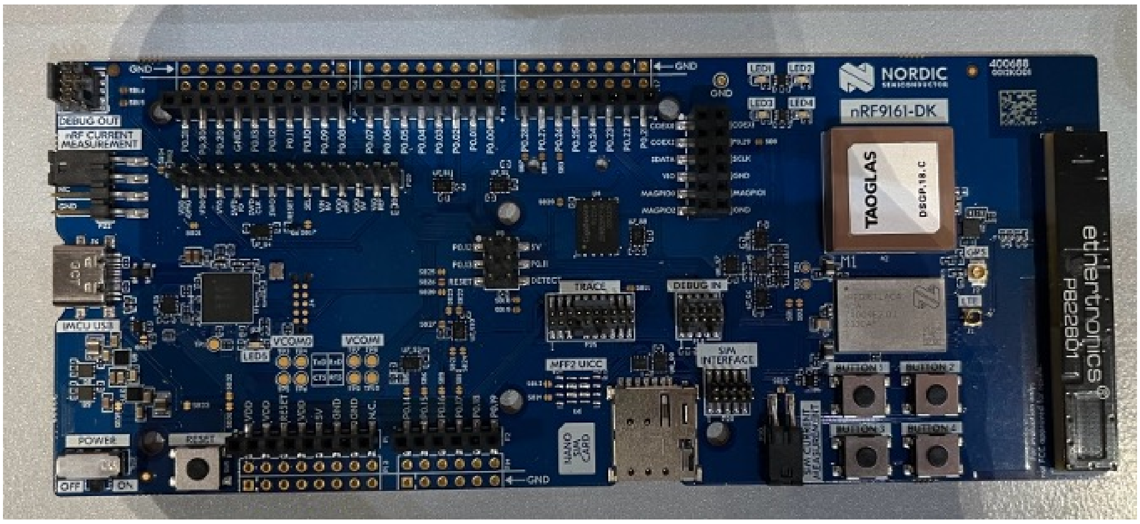


Figure 3.1: *the nRF9161 DK.*

Diving deeper into the nRF9161 System-in-Package (SiP), it is a low-power IoT solution that integrates an Arm® Cortex®-M33 processor with advanced security features, a range of peripherals and a Low-Power Wide-Area (LPWA) network processor. This LPWA network processor is versatile, capable of operating as a 5G DECT NR+ device or as an LTE modem, depending on the installed firmware. This ability further enhances its versatility and applicability in diverse IoT scenarios. Key features of the SiP also include support for the 1.9 GHz NR+ band and scalability to 3 Mbps bandwidth [48].

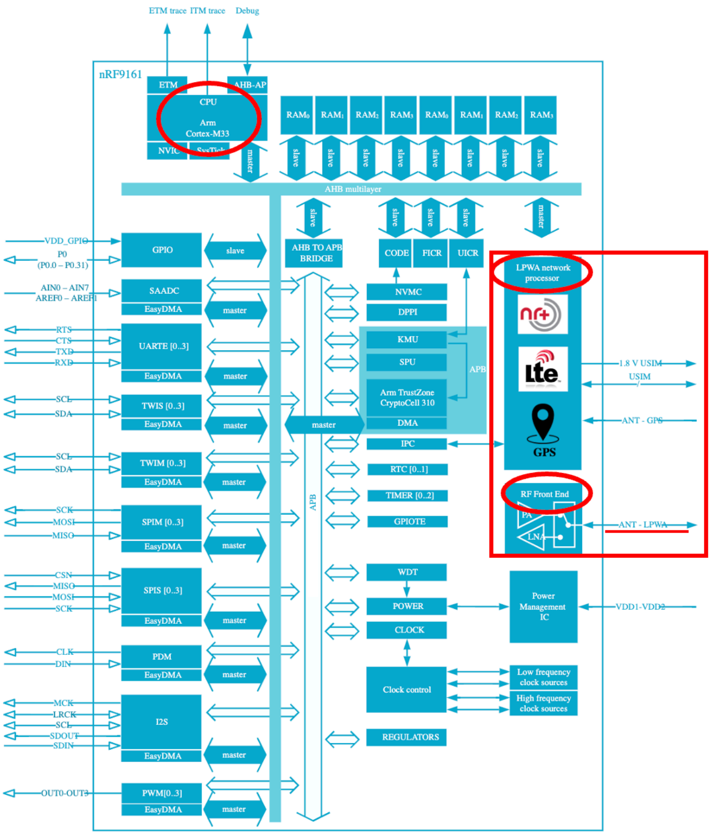


Figure 3.2: *Block Diagram of the nRF9161 SiP (from [49]).*

Application and LPWA network domains interact through the inter-processor communication (IPC) mechanism. The application processor is the system master and is responsible for starting and stopping the NR+ device. The LPWA network processor enables the clocks and power required for its own operation [50].

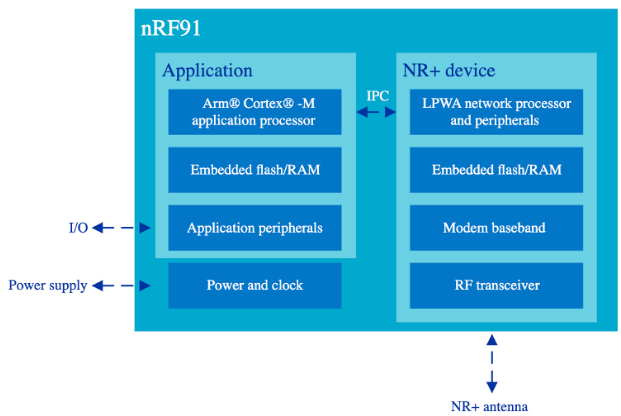


Figure 3.3: *nRF9161 NR+ device functional overview (from [50]).*

3.1.2 Power Profiler Kit II

The Power Profiler Kit II (PPK2) is a tool for real-time power consumption measurement, indispensable for evaluating the power efficiency of IoT devices. It can either supply power to an external board or function as an Ampere Meter, measuring current from 500 nA to 1 A. The kit is compatible with nRF91 Series Development Kits (DKs) and includes a desktop application for detailed measurement analysis. Key features of the PPK2 include:

- Variable power supply voltage (0.8 V to 5.0 V);
- Maximum current measurement of 1 A;
- High accuracy measurement down to approximately 200 nA;
- 100 kS/s sampling speed;
- USB communication for easy integration with other applications;
- Real-time current measurement display [51].

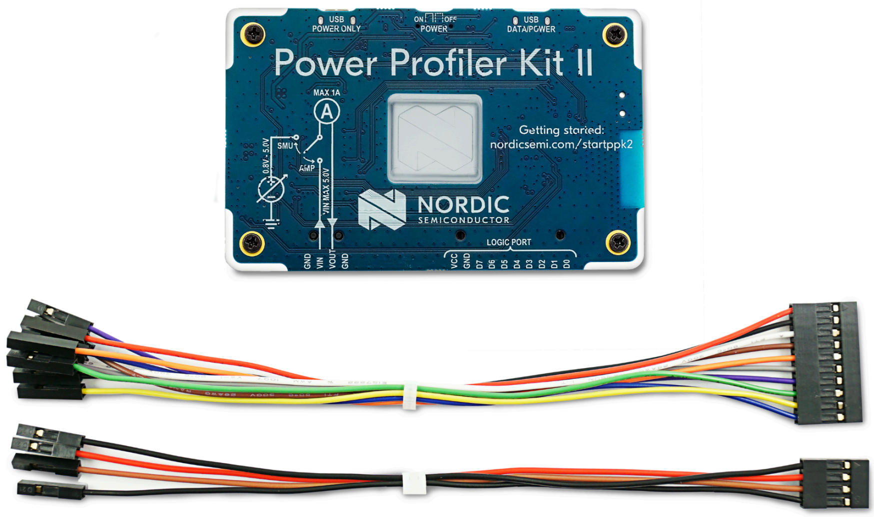


Figure 3.4: *PPK2 board and cables.*

Configuration setup for measuring current on the DK

In this project, the PPK2 was utilized to measure the power consumption performance of the nRF9161 DK. The goal was to verify the actual power dissipation of the device compared to traditional wireless communication methods. The PPK2 was used in Ampere meter mode to measure the current on the nRF9161 DK while allowing the DK to be powered from its USB. Specifically, the nRF9161 DK and the PPK2 were connected as follows:

- PPK2 VIN (Power input for the Ampere meter) to P22 VDD_5V (Power source for the DK);
- PPK2 VOUT (Power output for the Ampere meter) to P22 VDD_nRF (Power drain for the DK);
- PPK2 GND to P22_GND;

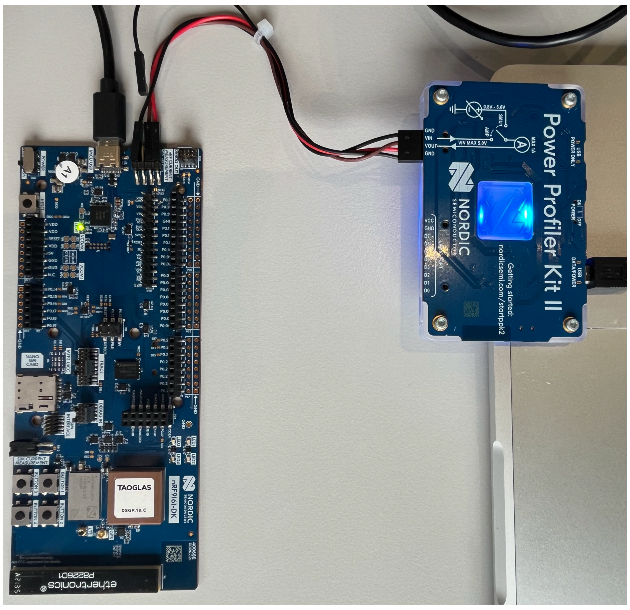


Figure 3.5: *PPK2 and nRF9161 DK connection.*

After connecting the DK and the PPK2 to a computer using a micro-USB cable, the “Power Profiler App” from “nRF Connect for Desktop” (discussed in the next section) was installed to visualize the results. In the Power Profiler App, the Ampere Meter mode was selected to perform the necessary measurements.

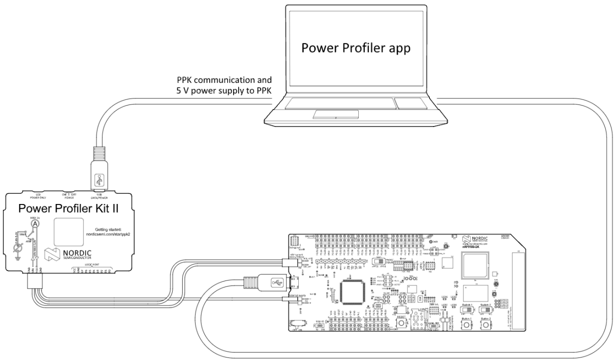


Figure 3.6: *Typical configuration for measuring current on the DK (from [52]).*

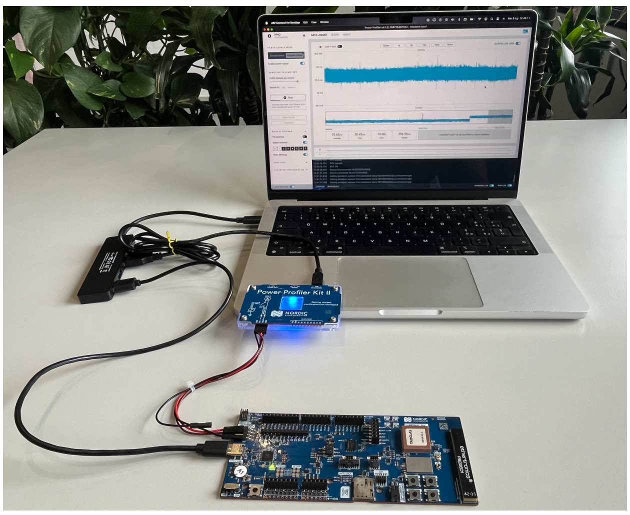


Figure 3.7: *Configuration of my workstation.*

3.2 Software Tools

3.2.1 nRF Connect SDK

The nRF Connect SDK is a comprehensive development environment provided by Nordic Semiconductor, essential for developing applications on the nRF9161 System in Package (SiP). It offers developers an extensible framework for building size-optimized software for memory-constrained devices as well as powerful and complex software for more advanced devices and applications [53].

The application processor of the nRF9161 DK runs user-developed applications, while the modem core runs pre-compiled modem firmware binaries provided by Nordic Semiconductor. To facilitate communication between the application and the modem, the nRF Connect SDK includes the Modem library. This library provides a standard set of function calls that applications use to interface with the DECT NR+ modem, allowing for smooth and efficient communication between the two cores.

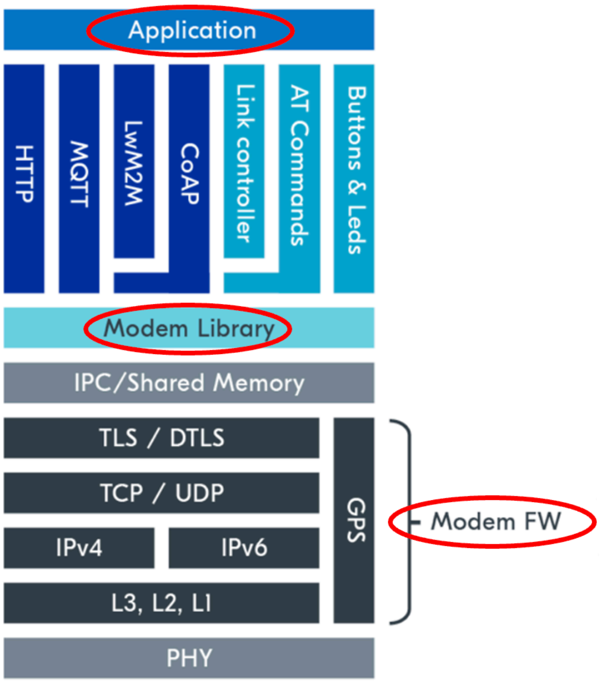


Figure 3.8: *Communication stack of the nRF91 Series SiP (from [54]).*

The communication stack of the nRF91 Series SiP, as illustrated above, depicts the modem firmware running on the modem core, with the Modem library providing the necessary interface. At the top of this stack resides the user-developed cellular application, which communicates through these layers to interact with the modem firmware. For the specific case of DECT NR+, the Modem library is part of the SDK from release version v2.7.0, which was released in June 2024.

3.2.2 nRF Connect for VS Code extension pack

The nRF Connect for VS Code extension pack is an essential tool for developers working with the nRF Connect SDK. This extension integrates seamlessly with Visual Studio Code (VS Code), enabling a streamlined development experience for creating applications on Nordic Semiconductor devices. In particular, this extension provides an easier, user-friendly graphical user interface (GUI) for building and configuring embedded systems projects based on the nRF Connect SDK. It integrates all necessary development tools into a single program, enabling developers to manage project files, connect devices to a serial port and build and customize applications within the same Visual Studio Code window [55]. Key features of the nRF Connect for VS Code extension pack include:

- Unified Interface: Simplifies workflow by enabling creation, building, flashing and debugging of applications within a single interface.
- SDK Management: Easily manage nRF Connect SDK versions, including installation, uninstallation and setting the active version.
- Multi-Application Development: Develop multiple applications simultaneously without needing a multi-root workspace.
- Device-tree Support: Provides syntax validation, highlighting, code completion and the ability to add required properties for accurate hardware configuration.
- Memory Report: Analyzes and identifies memory distribution, aiding in efficient memory management.
- Built-In Terminal: Connect to devices via serial ports or Real-Time Transfer (RTT), facilitating easy device communication.

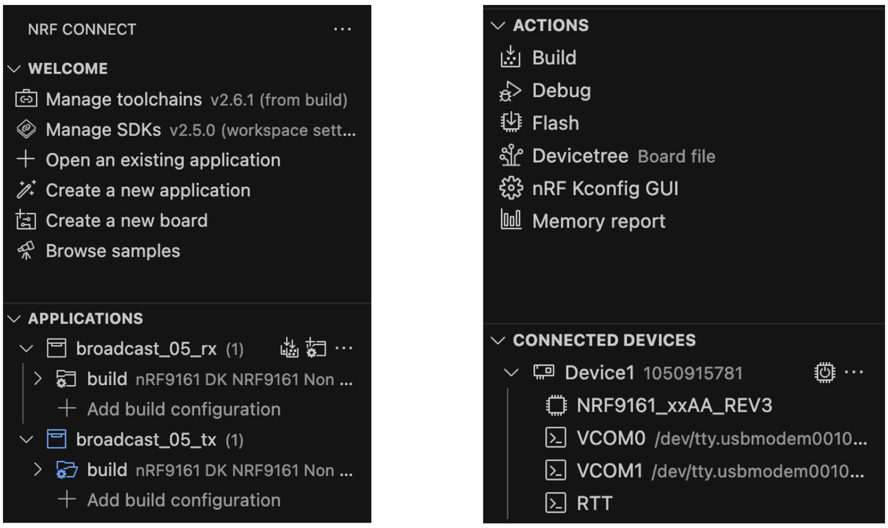


Figure 3.9: The nRF Connect for VS Code extension pack dropdown menu. The left panel shows options for managing SDKs and supports multiple applications, *broadcast_05_rx* and *tx* in this case, with their respective build configurations. The right panel provides options to build, debug and flash applications, access the Device-tree Visual Editor, view memory reports and connect to the device's serial ports.

After creating an application, it is essential to set up a build configuration. This step compiles and links the application components, preparing it for deployment and execution. After this, the specific app (if multiple apps are open) must be selected and flashed onto the device (nRF9161 DK), in order to run it. If multiple devices are connected to the computer, the target device must be chosen for flashing. The Broadcast application (discussed later) is developed in C and managed through VS Code and Nordic's extension. It comprises two applications, *broadcast_05_rx* and *broadcast_05_tx*. One is flashed onto a device functioning as the transmitter, while the other onto a device functioning as the receiver.

3.2.3 nRF Connect for Desktop

nRF Connect for Desktop is a versatile, cross-platform tool framework designed to support development on Nordic Semiconductor's nRF devices. This powerful software suite includes a variety of applications that assist in testing, monitoring, measuring, optimizing and programming embedded systems projects. For this thesis project, three key applications within nRF Connect for Desktop were utilized: Programmer, Serial Terminal and the Power Profiler.

Programmer

The nRF Connect Programmer is an essential application used for programming firmware onto Nordic Semiconductor devices. The Programmer app's main window displays both the memory layout of the device and the file you want to work with, providing a user-friendly interface for managing firmware updates. When the Programmer app is started, the main window opens with the File Memory Layout and Device Memory Layout panels empty.

- **File Memory Layout Panel**: This panel displays the memory layout for files added to the Programmer app using the "Add file" option. Once a file is added, it can be programmed onto the device. The firmware image file, typically a HEX file or a ZIP file, needs to be selected.

- **Device Memory Layout Panel**: This panel shows the memory layout of the device. It is possible to read the name, address range and size of a memory section by hovering the mouse cursor over it. This feature becomes available after loading a memory layout.

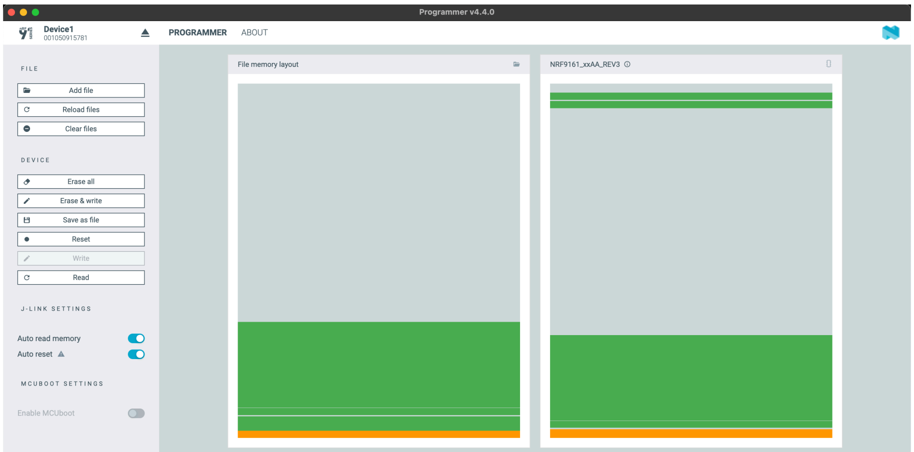


Figure 3.10: Example of the Programmer app. The File Memory Layout panel (on the left) shows the layout of the file *mosh-fmac-dect-0.1.8-nrf9161dk-uart-mdm-traces.hex*, while the Device Memory Layout panel (on the right) displays the memory content of the selected device, programmed to operate with the Modem Shell Application. The orange rectangle corresponds to the MBR section, while the green one to the Application section.

For this thesis project, in order to use the nRF9161DK with both the two applications, different firmware files were needed:

- **Broadcast Application**: The file programmed onto the device for this application is *mfw-nr+_nrf91x1_1.0.0.zip*. After adding this file and performing the "Write" operation to load it into the device's memory, the next step is to "Flash" the C-programmed application using the "nRF Connect for VS Code" extension, as detailed in the previous subsection.

- **Modem Shell Application**: The files used for this application are the modem firmware *mfw-nr+_nrf91x1_1.0.0-46.prealpha.zip* together with the *mosh-fmac-dect-0.1.8-nrf 9161dk-uart-mdm-traces.hex*. Once these files are added and written to the device's memory, no additional operations are required to program the device. The device will be ready to operate with all the functionalities provided by the application, which can be accessed through the Serial Terminal app (to be discussed next).

It is important to note that all these files were developed by Nordic Semiconductor and provided for this thesis project.

Serial Terminal
The nRF Connect Serial Terminal is a versatile, cross-platform terminal emulator designed for serial port communications with Nordic Semiconductor devices using Universal Asynchronous Receiver/Transmitter (UART) protocols. This tool is essential for configuring, monitoring and interacting with virtual serial ports on Nordic devices [56]. To use the Serial Terminal, after connecting the device to the computer using a USB cable, click the "Connect to port" button to establish a connection with the selected serial port.

The Serial Terminal supports both logging output and console input:

- Logging Output: Particularly useful for applications like the Broadcast Application, where monitoring the device's output is crucial for debugging and performance assessment.
- Console Input: Extensively used in the Modem Shell Application, allowing for real-time interaction and debugging by inputting commands directly into the console.

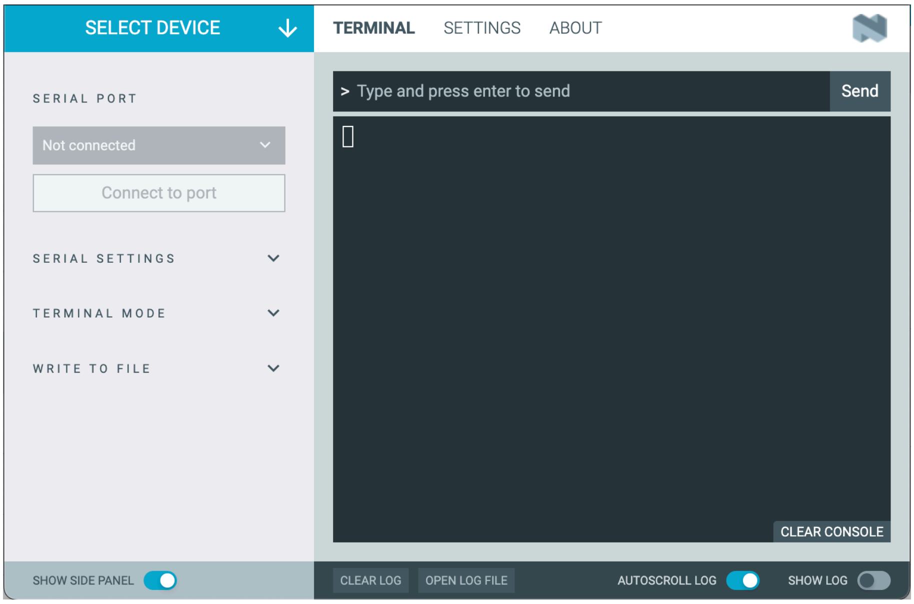


Figure 3.11: *Display of the Serial Terminal App.*

Power Profiler
The nRF Connect Power Profiler is an app used to communicate with the Power Profiler Kit II (PPK2) to obtain real-time current measurements of the nRF9161. You can select one of the power supply modes for the PPK2, depending on your hardware setup:

- Source meter: In this mode, the Device Under Test (DUT) is supplied power by the PPK2.
- Ampere meter: In this mode, the DUT is supplied power from an external source.

For the power consumption measurements of the device, the Ampere meter mode was used. As shown in the figure below, once the PPK2 is selected (connected to the computer and to the nRF9161 as detailed in the HW Components section, see Figure 3.6 and 3.7), it is possible to select the number of samples per second to be recorded. The slider allows selecting different values up to 100,000 samples per second (100 kHz). It can also be specified how long the sampling should last (by default, it is set to run indefinitely).

Figure 3.12: *Display of the Power Profiler App.*

After pressing Start, the sampling continues until the specified sampling time is reached. If set to run indefinitely, sampling will last until manually stopped. The window displays values for the measurement from start to finish, including the average current measured, the maximum current measured, the duration of the measurement and the total charge.

3.3 Broadcast Application

The Broadcast Application involves a straightforward broadcast of DECT NR+ messages, intended for initial testing. This setup requires two nRF9161 Development Kits (DKs): one for transmitting a counter value, needing only power and printing progress information, and the other for receiving the counter value and requiring a terminal for printout.

Nordic Semiconductor has developed specialized firmware for operating the Low Power Wide Area (LPWA) network processor using the DECT NR+ standard. This firmware implements the physical layer (PHY) of the NR+ radio protocol stack, adhering to ETSI specifications.

Currently, Nordic Semiconductor's implementation covers only the PHY layer of the DECT NR+ protocol, including basic transmission (TX) and reception (RX) operations. Networking, meshing, MAC layer access control and external communication features are not yet implemented and thus are not available for this application. The DECT NR+ PHY firmware is a variant of the nRF91x1 firmware but is specifically optimized for a different radio technology. Unlike the nRF91x1 firmware, which supports cellular operations and the Global Navigation Satellite System (GNSS), the DECT NR+ PHY firmware does not include these features [57]. This distinction highlights the specialized nature of the DECT NR+ PHY firmware, which is dedicated to NR+ radio protocol stack operations.

The firmware used to operate the Broadcast application is the *mfw-nr+_nrf91x1_1.0.0.zip* file. As already mentioned, this file should be loaded into the "Programmer" app and overwritten in the device's memory. Afterward, the developed C application should be flashed using Visual Studio Code (VS Code) along with the VS Code extension pack.

3.3.1 How the application works

The DECT NR+ PHY firmware operates through the DECT PHY interface, which schedules radio operations executed by the DECT NR+ physical layer's scheduler. Given the nature of a radio scheduler, it allows radio operations to be executed at a specific future time. Therefore, all radio operations within the DECT PHY interface are asynchronous, with their completion signaled to the application via callbacks. Notably, all scheduling is final, meaning that once an operation is scheduled, it cannot be unscheduled. These operations can either be scheduled to execute at a future time or immediately if the radio is currently idle. Each radio operation is associated with an application-defined handle, which identifies the operation and is returned by the callback signaling the completion of the operation. This handle remains opaque to all underlying layers, including the DECT PHY interface [57].

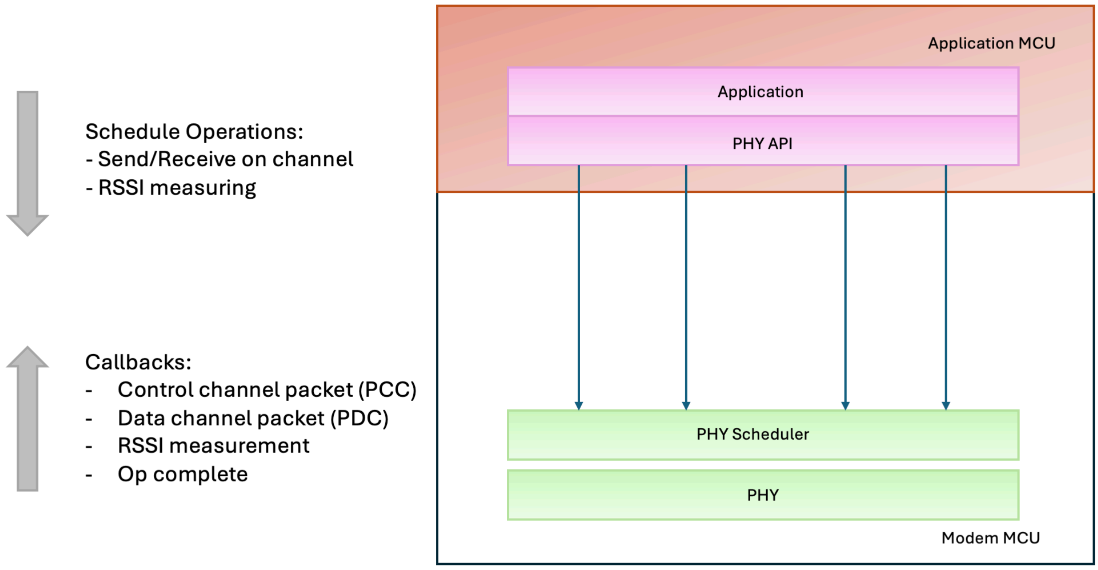


Figure 3.13: *Communication between the PHY API and the PHY Scheduler. The image shows the "scheduled operations", which are sent by the application through the PHY interface and received by the modem, which will schedule them for execution. Once the modem completes an operation, it invokes the corresponding callback to notify the application that the results are available for further processing.*

The DECT physical layer (PHY) interface in the Modem library is used to control the DECT NR+ PHY in the DECT NR+ PHY firmware. The DECT PHY interface offers three primary radio operations:

- Transmission (TX)
- Reception (RX)
- RSSI Measurement

Each of these operations can be performed with various parameters. The completion of a radio operation is signaled to the application by invoking the `nrf_modem_dect_phy_callbacks.op_complete` callback function. This callback receives the same handle specified by the application when the operation was scheduled.

Transmission
The application schedules a transmission by calling the `nrf_modem_dect_phy_tx()` function. This function is called by the first device, which is responsible for transmission. The operation is asynchronous, and the successful completion of the `nrf_modem_dect_phy_tx()` function only indicates that the request was sent to the modem. The result of the operation, whether successful or not, is signaled to the application via the `nrf_modem_dect_phy_callbacks.op_complete` callback function. Any errors occurring during the scheduling or execution of the operation are also returned through this callback.

Reception
The application schedules a reception by calling the `nrf_modem_dect_phy_rx()` function, which is used by the second device, responsible for receiving. Similar to transmission, this operation is asynchronous, and the completion of the `nrf_modem_dect_phy_rx()` function indicates that the request was sent to the modem. During reception, data received on the Physical Control Channel (PCC) and the Physical Data Channel (PDC) is forwarded to the application via the `nrf_modem_dect_phy_callbacks.pcc` and `nrf_modem_dect_phy_callbacks.pdc` callback functions, respectively. These callbacks handle incoming data on the respective channels. Additionally, any CRC errors on the PCC and PDC are reported to the application through the `nrf_modem_dect_phy_callbacks.pcc_crc_err` and `nrf_modem_dect_phy_callbacks.pdc_crc_err` callbacks. Upon completion of the operation, its result is signaled to the application through the `nrf_modem_dect_phy_callbacks.op_complete` callback function, similar to the transmission process. Any errors occurring during the scheduling or execution of the operation are also communicated via this callback.

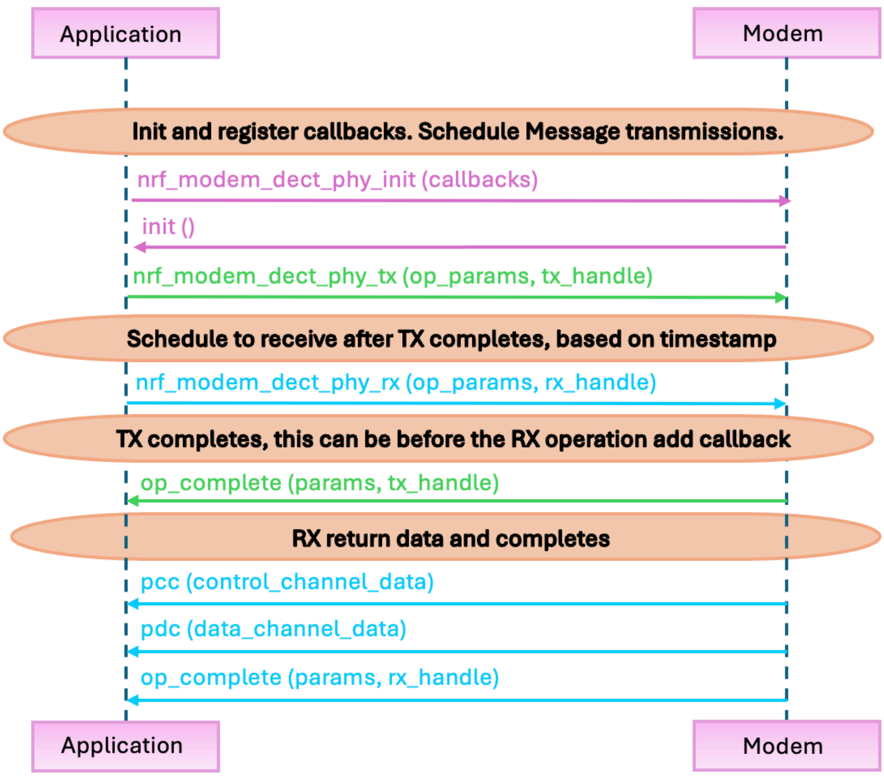


Figure 3.14: *Simple data message transmit and receive. The functions in green concern the first device (transmitter). The ones in blue concern the second device (receiver).*

3.3.2 What the application does

As already mentioned, the application performs a simple broadcast of DECT NR+ messages, primarily for initial testing purposes. This setup requires two nRF9161 DKs, where one device acts as the transmitter (TX) and the other as the receiver (RX). The first device transmits a counter value and only needs power to print progress information, while the second one receives this value and requires a terminal for printout.

To implement this functionality, two distinct applications were created: *broadcast_05_tx* for the transmitter and *broadcast_05_rx* for the receiver. Once started, the TX begins sending packets on a hard-coded channel, and it is programmed to transmit four packets per second. The receiver, on the other hand, listens on the same channel. To stop the connection between the devices, a button on the DK can be pressed. If the button on the transmitter is pressed, it stops sending messages. If the button on the receiver is pressed, it stops receiving messages. Simple statistics about the communication are provided at the end when a button is pressed on the receiver.

The RX accepts the first received counter value and then detects if there are any missed messages between receptions, incrementing a counter for missed messages if any are detected. Additionally, the RX calculates CRC errors using the *pdc_crc_err callback* and prints basic data for each received message to show progress.

The data structures, functions and callbacks that will be present are defined in the Modem Library, all provided by Nordic Semiconductor. Let's now have a brief look into the code of both the applications:

Variable Definitions and Communication Configuration

```
int EXIT = 0; // exit flag, set by button press

int previous_received = -1; // previous received message number
int missing_errors = 0;
int crc_errors = 0;
int received_ok = 0;
float rssi_average = 0;
int n = 0; // number of received messages

int txHandle = 1;
int rxHandle = 31400;

K_SEM_DEFINE(modem, 1, 1);
```

Listing 3.1: *Definition of the main variables, including State Flags, Statistics Collection and Semaphore for API Calls.*

- The `int EXIT = 0;` variable is used in the `button_handler` function to set an exit flag when a button is pressed:

  ```
  void button_handler(uint32_t button_state, uint32_t has_changed)
  {
      EXIT=1;
  }
  ```

  This function is initialized in the main function using:

  ```
  dk_buttons_init(button_handler);
  ```

  This initialization ensures that when any button is pressed, the `EXIT` flag is set to 1. Since the main loop condition (for both TX and RX) is `while(EXIT == 0)` (will be seen later), pressing any button will stop the device.

- The variables `previous_received, missing_errors, crc_errors, received_ok, rssi_average, n` are used for statistics collection. They are modified within callbacks such as `pdc, pdc_crc_err and pcc_crc_err` and are utilized for the final display of the communication performance. A capture of the communication output will be provided later to demonstrate this.

- The variables `txHandle` and `rxHandle` are used in the functions `modem_tx` and `modem_rx` respectively (will be seen later), serving as identifiers for API calls related to transmission and reception operations. `txHandle` acts as an identifier for transmission operations within the `modem_tx` function. It distinguishes individual transmission operations, incremented after each transmission. When the value reaches 30000, it wraps back to 1 to maintain a valid range of identifiers. `rxHandle` serves as an identifier for reception operations within the `modem_rx` function. This ensures unique tracking of reception operations.

- `K_SEM_DEFINE(modem, 1, 1)` defines a semaphore called *modem* with a maximum count of 1 and an initial count of 1. The semaphore ensures that only one asynchronous operation is performed at a time. The third argument specifies the initial count, which is the number of units available when the semaphore is created. Here, the initial count is 1, meaning the semaphore is initially available for use.

The following code defines some important parameters and data buffers used in both the transmission and reception processes:

```
#define MCS  1
#define POWER  0xb
#define DATA_LEN 4
uint8_t _txData[DATA_LEN];  //data to be sent
uint8_t _rxData[DATA_LEN];  //received data

#define CARRIER 1677
```

Listing 3.2: *Definition of the Modulation and Coding Scheme (MCS), transmission power, data length, data buffers and carrier frequency.*

- MCS stands for “Modulation and Coding Scheme” and is a parameter that defines the modulation and coding method used to transmit data in a wireless communication system. In simpler terms, it represents the technique used to convert digital data into radio signals that can be transmitted over a radio channel. The MCS impacts how much data can be fit into subslots/slots. Here, it is hardcoded to MCS 1.

- `#define POWER 0xb` defines the transmission power level. The value 0xb corresponds to a power level of 13 dBm.

- `#define DATA_LEN 4` defines the length of the data to be sent or received. The length depends on the MCS defined above.

- The `_txData` and `_rxData` arrays store the data to be transmitted and the data that has been received.

- `#define CARRIER 1677` defines the carrier frequency used for the transmission and reception.

Callbacks definition

The following callbacks have been previously defined in the system architecture (Figure 3.14) when explaining the overall application functionality. These callbacks handle various events and data received during radio operations.

The op_complete callback function is invoked to signal the completion of a radio operation to the application. This callback receives the handle specified by the application when the operation was scheduled, along with the status, temperature and time. It also releases the semaphore to indicate that the operation has completed.

```
void op_complete(const uint64_t *time, int16_t temperature, enum
    nrf_modem_dect_phy_err err, uint32_t handle)
{
  LOG_DBG("operation_complete_cb Status %d, Temp %d, Handle %d", err,
      temperature, handle);
  k_sem_give(&modem); //releases the semaphore, operation is completed.
  return;
}
```

Listing 3.3: *The op_complete callback function signals the completion of a radio operation and releases the semaphore.*

The pcc callback has also been defined earlier in the architecture discussion (Figure 3.14), where the reception process was explained. During reception, data received on the Physical Control Channel (PCC) is forwarded to the application via the pcc callback function. This callback handles incoming PCC data and logs relevant information such as the header status and RSSI.

```
void pcc(
  const uint64_t *time,
  const struct nrf_modem_dect_phy_rx_pcc_status *status,
  const union nrf_modem_dect_phy_hdr *hdr)
{
  LOG_DBG("pcc_cb phy_header_valid %d rssi_2 %d", status->header_status,
      status->rssi_2);
  return;
}
```

Listing 3.4: *The pcc callback function handles data received on the Physical Control Channel (PCC), logging the header status and RSSI.*

However, the most important and well-defined callback is the pdc callback. It is responsible for handling data received on the Physical Data Channel (PDC).

```
void pdc(
  const uint64_t *time,
  const struct nrf_modem_dect_phy_rx_pdc_status *status,
  const void *data, uint32_t len)
{
  int rx_counter=0;
  int32_t rx_rssi=calcRSSI(status->rssi_2, 1);

  memcpy(&_rxData, data, 4);

  rx_counter = (_rxData[0]<<24) + (_rxData[1]<<16) + (_rxData[2]<<8) + (
      _rxData[3]);
  received_ok++;
  if(previous_received>0 && rx_counter != previous_received+1){
    missing_errors=missing_errors+(rx_counter-previous_received)-1; }
  previous_received=rx_counter;

  LOG_INF("RECEIVED DATA, %d, rssi_2, %d, missed/crc errors, %d", rx_counter,
      rx_rssi, missing_errors);
}
```

Listing 3.5: *The pdc callback function handles data received on the Physical Data Channel (PDC), processes the data, updates statistics and logs the received message information.*

When data is received, this callback extracts the first 4 bytes and stores them in the `_rxData` buffer (with the `memcpy` function). It then extract the received counter value from the buffer, which is converted to an integer value `rx_counter` that is used for tracking the received message number. If there are any missed messages, it updates the error count. Additionally, it logs the received data, RSSI and the number of missed and CRC errors for every received message.

The `LOG_INF` function from the logger module is used to log messages to the console. Unlike the printk() function, which waits until all bytes of the message are sent, the logger module adds messages to a dedicated buffer. Each log message contains a source ID, timestamp and severity level. The logger module is thread-safe and minimizes the time needed to log the message. There are *four severity levels* available in the system; the LOG_INF level is the third one and prints informational messages that require no action. The timestamp format is hh:mm:ss.ms,us and represents the uptime since boot, as measured by the system's hardware clock [58].

To conclude the explanation of the callbacks used in the application, it's essential to understand their configuration and management through a specific structure. The `nrf_modem_dect_phy_callbacks` structure is used to define and register the callback functions that handle all the events and operations in the DECT physical layer. Here is the structure definition:

```c
struct nrf_modem_dect_phy_callbacks dect_cb_config = {
    .init = init,  //DECT PHY initialization callback.
    .op_complete = op_complete,
    .rssi = rssi,  //Callback to receive RSSI measurements.
    .rx_stop = rx_stop, //Callback for nrf_modem_dect_phy_rx_stop.
    .pcc = pcc, //Callback for control channel reception.
    .pcc_crc_err = pcc_crc_err, //Callback for CRC failures on the PCC.
    .pdc = pdc,  //Callback for data channel reception.
    .pdc_crc_err = pdc_crc_err,  //Callback for CRC failures on the PDC.
    .link_config = link_config,
    .time_get = time_get,
    .capability_get = capability_get,
    .deinit=deinit
};
```

Listing 3.6: *Structure defining callbacks for DECT PHY operations*

This structure contains function pointers to callbacks responsible for initialization, data reception and other critical aspects of DECT system operations. These callbacks are then registered using `nrf_modem_dect_phy_callback_set(&dect_cb_config);` in the main() function. Here is how the callbacks are registered:

```c
err = nrf_modem_dect_phy_callback_set(&dect_cb_config);
  if (err != 0)
  {
    printk("ERROR settings callbacks %d\n", err);
  }
```

Listing 3.7: *Snippet extracted from the main() function. It sets the DECT PHY callbacks and if an error occurs during callback registration, an error message is printed.*

By setting the modem callbacks for PHY operations before attempting other operations, including PHY initialization with `nrf_modem_dect_phy_init()`, the system ensures proper handling of all DECT PHY-related events and data. This setup is critical for the seamless operation and management of the DECT NR+ communication system.

Message Transmission

The `modem_tx(i)` function is only present in the *broadcast_05_tx* application: it serves as the main function responsible for message transmission as it calls the `nrf_modem_dect_phy_tx` function, indicating that the transmission request was sent to the modem (Figure 3.14). The primary purpose of the `modem_tx(i)` function is to prepare and initiate the transmission of a message using the specified parameters, and then calls the `nrf_modem_dect_phy_tx` function to start the operation.

```c
void modem_tx(uint32_t i)
{
  uint8_t tmp[DATA_LEN];
  tmp[0] = (i >> 24) & 0xff;
  tmp[1] = (i >> 16) & 0xff;
  tmp[2] = (i >> 8) & 0xff;
  tmp[3] = (i) & 0xff;

  memcpy(_txData, tmp, sizeof(tmp));
  struct phy_ctrl_field_common header = {
      .header_format = (uint8_t)0x0,  //short header format, broadcast
      .packet_length_type =  (uint8_t)0x0, //length given in subslots
      .packet_length = (uint8_t) 0x01,  //1 subslot with MCS = 1
      .short_network_id = (uint8_t)(0x0a & 0xff),
      .transmitter_id_hi = (uint8_t) (0x0101 >> 8), //made up transmitter ID
      .transmitter_id_lo = (uint8_t) (0x0101 & 0xff),
      .transmit_power = POWER,
      .reserved = 0,
      .df_mcs = MCS
  };
  memcpy(&phyHeader.type_1, &header, 5);

  // Setup the nrf_modem_dect_phy_operation_tx
  struct nrf_modem_dect_phy_tx_params txOpsParams;
  txOpsParams.start_time = 0; //send counter value immediately
  txOpsParams.handle = txHandle;
  txOpsParams.network_id = 0x0a;
  txOpsParams.phy_type = 0;
  txOpsParams.lbt_rssi_threshold_max = 0;
  txOpsParams.carrier = CARRIER;
  txOpsParams.lbt_period = 0;
  txOpsParams.phy_header = &phyHeader;
  txOpsParams.data = _txData;
  txOpsParams.data_size = sizeof(tmp);

  k_sem_take(&modem, K_FOREVER);

  int err=nrf_modem_dect_phy_tx(&txOpsParams);
  if(err!=0) LOG_ERR("TX FAIL %d", err);
  if(txHandle==30000)txHandle=1;
  else txHandle++;
}
```

Listing 3.8: *Function responsible for preparing and initiating the transmission of a message using the specified parameters.*

The function performs the following steps:

- Copies an integer `i` (passed as an input parameter to `modem_tx`) into a `uint8_t` array `tmp` with a length of `DATA_LEN`. The array `tmp` is filled by extracting each byte from the integer `i`.
- Copies the contents of the `tmp` array to the `_txData` array using `memcpy`.

- Sets up the parameters of the PHY header in the `header` structure, which includes fields such as `header_format`, `packet_length_type`, `packet_length`, `df_mcs` and so on.
- Initializes the `txOpsParams` structure with parameters for the transmission operation, including `start_time`, `handle`, `network_id`, `phy_type`, `lbt_rssi_threshold_max`, `carrier`, `lbt_period`, `phy_header`, `data` and `data_size`.
- Initializes the `txOpsParams` structure with parameters for the transmission operation. The `txOpsParams` structure (or its pointer) will then be passed as an argument to the `nrf_modem_dect_phy_tx` function.
- Calls `k_sem_take(&modem, K_FOREVER)` to wait for the semaphore before scheduling the transmission.
- Calls `nrf_modem_dect_phy_tx` to schedule a data transmission with the given parameters and logs an error if the transmission fails. The `txHandle` is incremented and reset if it reaches 30000 (see Listing 3.1).

The function `modem_tx(i)` is called in the `main()` function of the *broadcast_05_tx* application. The `main()` function is as follows:

```
int main(void)
  {
  uint32_t i =1;  //send an increasing counter, start from 1
  printk("START DECT SAMPLE\n");
  dk_buttons_init(button_handler);
  printk("Buttons init\t");
  dk_leds_init();
  printk("leds init\t");

  nrf_modem_lib_init();

  k_sem_take(&modem, K_FOREVER);
  int err=0;

  err=nrf_modem_dect_phy_callback_set(&dect_cb_config);
  if(err!=0) {
    printk("ERROR settings callbacks %d\n",err);
  }

  err=nrf_modem_dect_phy_init(&init_params);
  if(err!=0) {
    printk("ERROR initializing modem PHY %d\n",err);
    return -1;
  }
  printk("DECT init started\n");

  printk("Transmitting on channel %d \n", CARRIER);

  dk_set_led_on(1);
  printk("TX ROLE\n");

  while(EXIT==0){
    //TX ROLE, LOOP
    k_msleep(250); //4 messages every second

    modem_tx(i);
    LOG_INF("TX %d\n", i);

    i++;
```

```
    if (i == INT32_MAX)
    { i = 0;
      LOG_INF("Sent integer value looped"); }
  }

  k_msleep(1000);  //messages may be in logging pipeline, wait a sec
  printk("Button pressed... End of transmission\n");
  printk("****************************************\n");
  printk("Transmitted messages %d\n", i);

  return 0;
}
```

Listing 3.9: The `main()` function of the *broadcast_05_tx* application.

Initially, several components such as buttons and LEDs are initialized. Also the DECT PHY interface in the Modem library requires explicit initialization. But, before initializing the DECT PHY interface of the Modem library, the application must:

- Initialize the Modem library by calling the `nrf_modem_lib_init()` function, which also turns on the modem core.
- Register callbacks for DECT PHY operations by calling the `nrf_modem_dect_phy_callback_set()` function (as explained in Listing 3.7).

Afterwards, the application can initialize the DECT PHY interface by calling the `nrf_modem_dect_phy_init()` function. Upon successful initialization, both the DECT PHY interface and DECT NR+ physical layer in the modem are ready for operation. The function then enters a `while` loop, where:

- it's called the `modem_tx(i)` to transmit the message, logging the transmission count.
- it's incremented the counter `i`. If `i` reaches `INT32_MAX`, it resets to 0 and logs this event.

After exiting the loop, the function waits for 1 second to ensure all messages are processed, prints the end of transmission message and returns the total number of transmitted messages.

Message Reception

The `modem_rx(rxMode, time_s)` function is only present in the *broadcast_05_rx* application: it serves as the main function responsible for message reception as it calls `nrf_modem_dect_phy_rx`, indicating that the reception request was sent to the modem (Figure 3.14). The primary purpose of `modem_rx` is to prepare and initiate the reception of messages using the specified parameters, and then calls the `nrf_modem_dect_phy_rx` function to start the operation.

```
void modem_rx(uint32_t rxMode, int time_s)
{
  // Setup the nrf_modem_dect_phy_operation_rx
  struct nrf_modem_dect_phy_rx_params rxOpsParams={0};
  rxOpsParams.start_time = 0; //start immediately
  rxOpsParams.handle = rxHandle;
  rxOpsParams.network_id=0;
  rxOpsParams.mode = rxMode;
  rxOpsParams.link_id = NRF_MODEM_DECT_PHY_LINK_UNSPECIFIED;
  rxOpsParams.rssi_level = -60;
  rxOpsParams.carrier = CARRIER;
  rxOpsParams.duration = time_s*69120*1000;
  rxOpsParams.filter.short_network_id = (uint8_t)(0x0a);
```

```c
    rxOpsParams.filter.is_short_network_id_used = 1;
    rxOpsParams.filter.receiver_identity = 0;

    k_sem_take(&modem, K_FOREVER);

    int err=nrf_modem_dect_phy_rx(&rxOpsParams);
    if(err!=0) LOG_ERR("RX FAIL %d", err);
    if(rxHandle==65000)rxHandle=31400;
    else rxHandle++;
}
```

Listing 3.10: *Function responsible for preparing and initiating the reception of messages using the specified parameters.*

The function performs the following steps:

- Sets up the parameters for the receiver operation in the `rxOpsParams` structure, including fields such as `start_time`, `handle` (associated to rxHandle, see 3.1), `carrier` and so on.
- Takes the semaphore using `k_sem_take(&modem, K_FOREVER)`, blocking until it is available. This ensures that only one asynchronous operation is performed at a time.
- Schedules a reception by calling `nrf_modem_dect_phy_rx` with the given parameters and logs an error if the reception fails. The `rxHandle` is incremented and reset if it reaches 65000.

Incoming data received on the physical layer control and data channels is sent to the `pcc` and `pdc` callbacks.

The main function of the *broadcast_05_rx* application is quite similar to the one of the *broadcast_05_tx* application. The initialization part is the same, with the only differences being in the while loop and the console printout of variables for performance evaluation. Here the application starts listening for incoming messages on the specified channel, repeatedly calling the `modem_rx` function.

```c
while (EXIT == 0)
{
  // RX ROLE, LOOP
  dk_set_led_on(DK_LED1);
  dk_set_led_on(DK_LED2);

  // loop RX mode
  modem_rx(NRF_MODEM_DECT_PHY_RX_MODE_SINGLE_SHOT, 2);
}

// messages may be in logging pipeline, wait a sec
k_msleep(1000);
printk("Button pressed... End of reception\n");
printk("****************************************\n");
// CRC error causes a missed message, so missing errors includes also CRC
printk("Received messages %d\n", received_ok);
printk("Missed messages %d\n", (missing_errors - crc_errors));
printk("CRC errors  %d\n", crc_errors);
printk("RSSI_2 AVG  (rounded) %d\n", (int)rssi_average);
printk("***************EXIT*************************\n");

return 0;
```

Listing 3.11: Snippet of the `main()` function of the *broadcast_05_rx* application.

Final Output
Here it can be seen the Serial Terminal screens of the two devices, one in transmission and the other in reception.

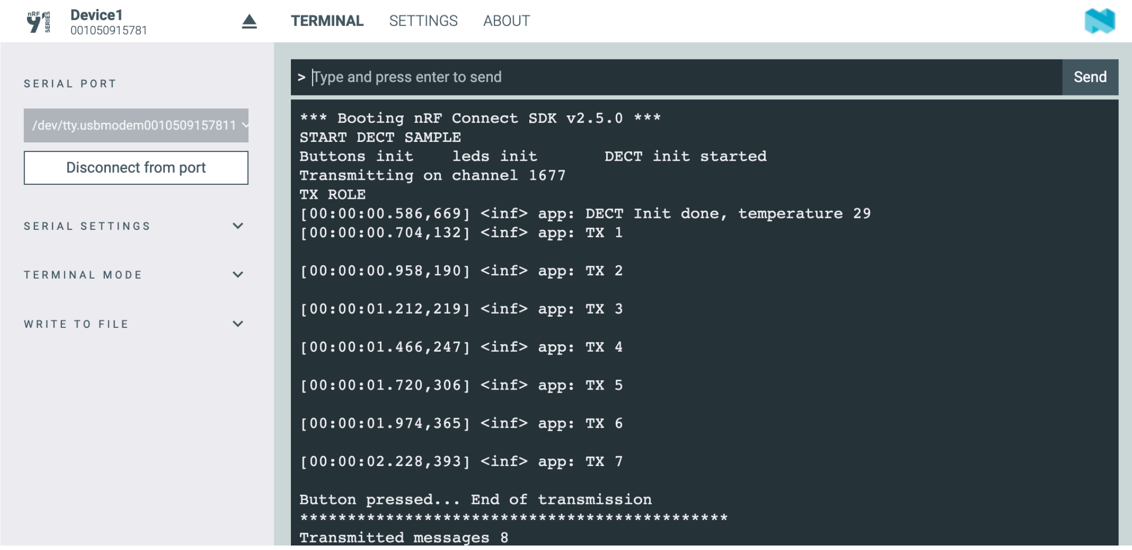


Figure 3.15: Output of the *broadcast_05_tx* application, showing the transmission of messages.

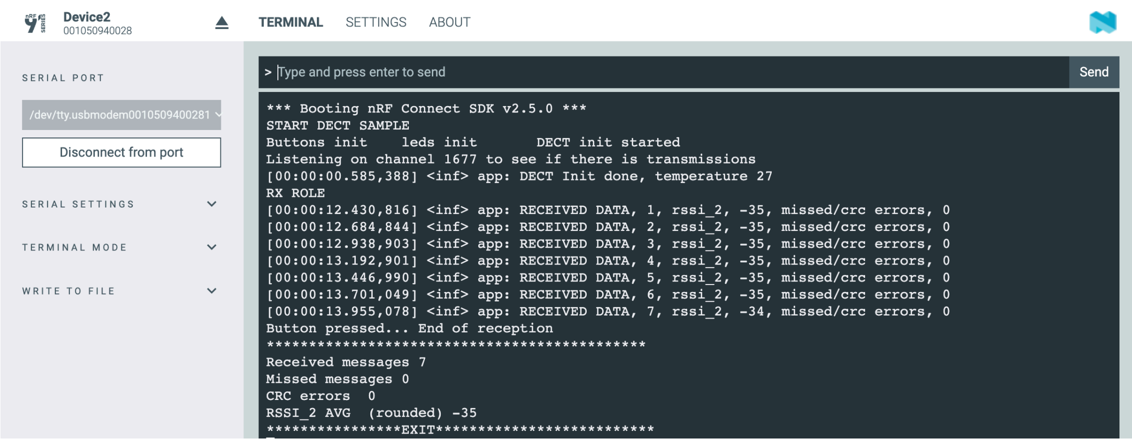


Figure 3.16: Output of the *broadcast_05_rx* application, showing the reception of messages.

3.4 Modem Shell Application

The Modem Shell Application, once the *.hex* file is flashed into the device's memory, enables various commands to be executed via the Serial Terminal. This application facilitates the configuration and management of the DECT NR+ network by providing a suite of commands, each serving a specific function.

Before continuing, however, it is important to clarify the differences between the two applications. Although it is possible to define and use functionalities above the physical layer with the Modem Shell Application —such as setting the device's role (sink, relay, or client) or using the ping or perf command— both the Broadcast and the Modem Shell App. ultimately use the same DECT NR+ PHY layer on the modem core (the only layer of the DECT NR+ stack implemented so far). The .hex application employs proprietary message formats, executing higher operations within the application itself, but not implementing the standard DECT NR+ MAC or higher layers. This is achieved by allowing the devices to perform basic PHY layer operations: listening when configured as a sink, and sending using the PHY TX when configured as a client. The relay device combines these operations, listening and forwarding any received messages.

Among the commands provided by the Application, the `dect sett` command enables the user to set various device parameters, such as the role of the device (e.g., sink, relay, client), transmission power, Modulation and Coding Scheme (MCS), the beacon interval and many others. This flexibility is crucial for adapting the device to different network configurations and performance requirements.

Another important command is `dect rssi_scan`, which scans all available channels (from 1657 to 1677) to identify their current status. The scan results categorize channels as:

- **Free**: Indicating minimal interference from other devices, making these channels optimal for initiating transmissions and establishing connections.
- **Possible**: Suggesting a moderate level of interference, where the channel may be shared with other devices but is not heavily congested.
- **Busy**: Signifying significant activity from other devices, rendering these channels unreliable for initiating new transmissions.

Furthermore, the `dect beacon_start` command initiates the transmission of beacon messages on a specified channel, facilitating network synchronization and device discovery.

The versatility of the Modem Shell Application (guaranteed by the above-mentioned commands and others that will be mentioned later) allows it to be used in various scenarios, creating multiple use cases. In the context of this thesis project, three primary use cases were explored:

- **DECT NR+ Network Creation**: A DECT NR+ network was established with three interconnected devices functioning as sink, relay and client (leaf). This setup demonstrated the ability of the network to support multiple roles and maintain reliable communication paths among the devices.
- **Throughput Performance Testing**: The `dect perf` command was employed to measure the data rate of the communication between two devices. This testing provided valuable insights into the network's throughput performance, highlighting the efficiency and reliability of data transmission under various configurations and conditions.
- **Ping Command Testing**: The `dect ping` command was used to evaluate the network's latency and responsiveness. By sending ping messages between devices, the round-trip time (RTT) was measured, offering a clear understanding of the network's latency characteristics and its suitability for time-sensitive applications.

3.4.1 DECT NR+ Network Creation

In this scenario, a mesh topology was created using three nRF9161 DKs, configured as a sink, a relay and a leaf, respectively (as it was shown in Figure 2.11). Here, the process of creating a DECT NR+ network with synchronization and device discovery capabilities is illustrated.

Sink Configuration
The sink device is configured with the command:

```
dect sett -c 0 --b_dl_channel 0 --b_ul_channel 0 --b_name mosh_sink -s yes
```

This command sets the beacon broadcast channel, beacon downlink (DL) channel and beacon uplink (UL) channel to the first free channel. It also sets the beacon broadcast name to *"mosh_sink"* and enables the sink mode. The sink starts sending beacons on a specific channel with:

```
dect beacon_start
```

This initiates beacon transmission on the chosen channel, which will be referred to as "X" from now on.

Relay Configuration
The relay device is set up similarly with the following command:

```
dect sett -c 0 --b_dl_channel 0 --b_ul_channel 0 --b_name relay_beacon
```

This sets the relay's beacon broadcast name to *"relay_beacon"* and selects the first free channel for beacon, DL and UL transmissions. To synchronize with the sink and to gather information about it, the relay scans for the sink beacon:

```
dect beacon_scan -c X
```

Now the beacon broadcast from *"mosh_sink"* is stored in the relay's neighbor cache. After detecting the sink's beacon, the relay starts its own beacon transmission on another channel, which will be referred to as "Y", and adjusts its timing with *"mosh_sink"* beacon:

```
dect beacon_start -s mosh_sink
```

Finally, the relay enables synchronization and background scanning towards the sink:

```
dect beacon_relay -b mosh_sink
```

Returning to the sink console, once the channel on which the relay is transmitting beacon messages has been identified, the same procedure executed on the relay is repeated. The device scan for the relay beacon to gather information about it:

```
dect beacon_scan -c Y
```

This stores the beacon broadcast from *"relay_beacon"* into the sink's neighbor cache. Next, enable synchronization and background scanning towards the relay:

```
dect beacon_relay -b relay_beacon
```

In case synchronization needs to be stopped, it can be done with:

```
dect beacon_relay -stop
```

This ensures that the sink and relay devices are synchronized and can communicate effectively, completing the network setup process.

Client/Sender Configuration
The client device is again initialized with the `dect sett` command, where various parameters are defined, such as the interval duration for synch/background scanning towards the relay (set to be more frequent in order to detect better relay channel changes) and for the keepalive messages of a connected client. Then the client scans for the relay beacon:

```
dect beacon_scan -c Y
```

This command stores the beacon broadcast from *"relay_beacon"* into the client's neighbor cache, allowing the client to recognize the relay. The client device can then perform two primary operations:

- **Send data to the Internet/Sink via the relay**: This is accomplished by transmitting data using the following command:

  ```
  dect client_rach_tx -b relay_beacon -d TAPARRA! -i
  ```

  Here, *"TAPARRA!"* represents the data being sent. The relay handles the forwarding of the data to the sink, ensuring that the messages reach their final destination.

- **Enable connected client mode towards the relay**: This mode maintains synchronization with the relay beacon in the background, ensuring consistent communication. The client can then send data at regular intervals (every 10 secs in this scenario), including additional information such as the modem temperature (formatted in JSON), with the relay forwarding the data to the sink as required:

  ```
  dect client_connect relay_beacon
  dect client_send -d TAPARRA! -i -t 10 -c
  ```

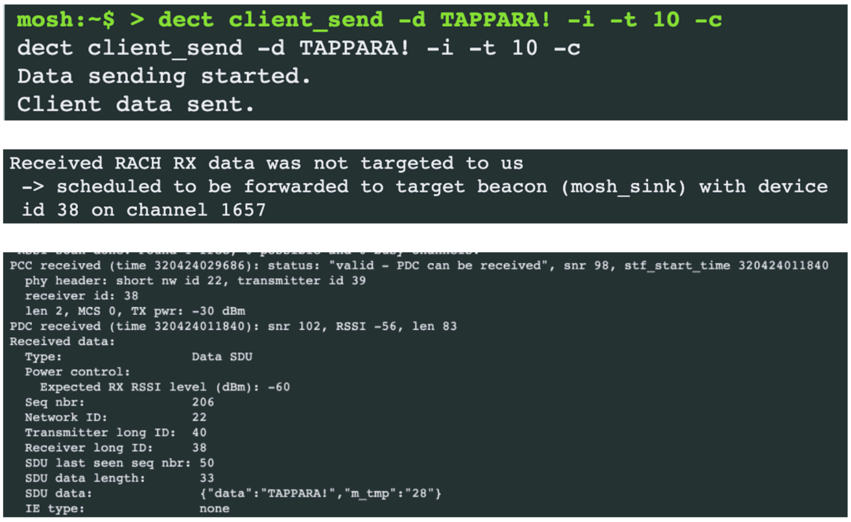


Figure 3.17: *Screenshots of the three device's consoles: the Client (at the top), the Relay (in the center), the Sink (at the bottom).*

To summarize the entire setup, it can be said that the process involves several key steps: both the sink and relay continuously transmit beacon messages using the *dect beacon_start* command, ensuring their presence is broadcasted on the network. They select a free channel for sending these beacons, facilitating clear and uninterrupted communication. Device discovery is accomplished by executing the *dect beacon_scan* command on the specified channel, allowing the relay to detect the sink and vice versa. Once a device is discovered, its information is stored in the neighbor cache of the other DK. Additionally, when the relay begins its beacon transmission, it adjusts its timing to align with the sink's beacon channel, ensuring synchronization. Following the initial scan, background scanning and synchronization with the discovered device are maintained using the *dect beacon_relay -b 'name'* command.

Through these configurations and commands, a DECT NR+ network is established, enabling seamless communication and message forwarding from the client to the sink via the relay.

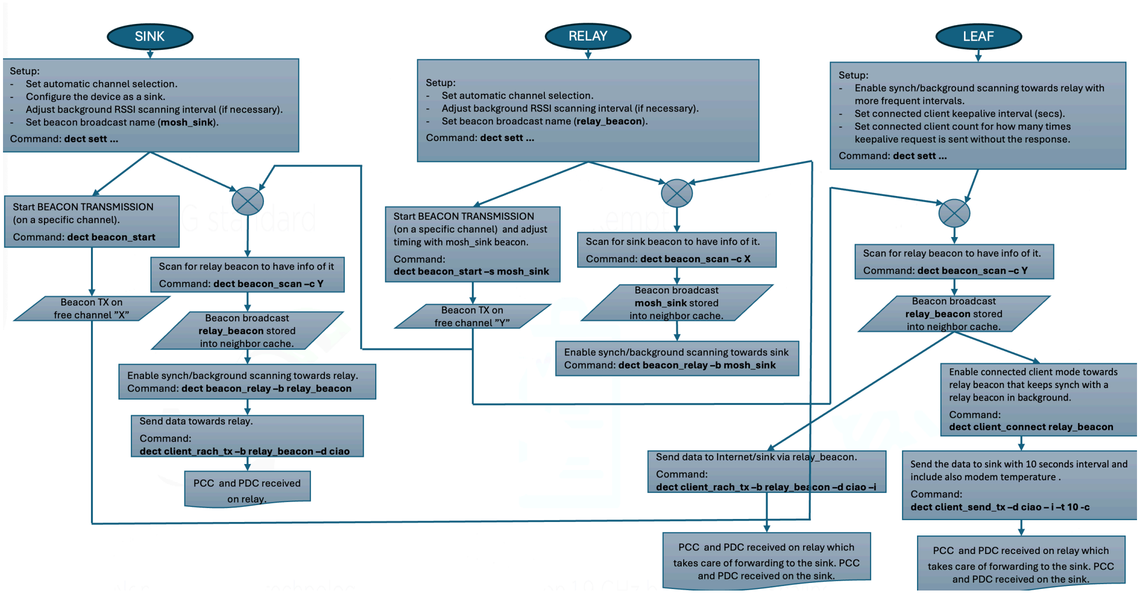


Figure 3.18: *Flow diagram illustrating the step-by-step commands used to configure the sink, relay and client devices in a DECT NR+ network setup.*

3.4.2 Throughput Performance Testing

In this scenario, a point-to-point topology was established using two nRF9161 DKs, configured as a client and a server respectively, to assess the throughput performance of the DECT NR+ network. The evaluation was conducted using the `dect perf` command, which provides comprehensive metrics on data rate and efficiency during communication between the devices.

Server Configuration

The server initiates by scanning a selected channel to determine its suitability for communication:

```
dect rssi_scan -c 1672
```

Subsequently, the server begins continuous reception on the chosen channel using:

```
dect perf -s -t -1 --channel 1672
```

The `-t -1` flag signifies continuous RX mode, enabling the server to remain actively listening without time constraints.

Client Configuration

The client, on the other hand, focuses on achieving maximum throughput over a defined period:

```
dect perf -c -t 12 --channel 1672
```

Here, the `-t 12` parameter specifies a transmission duration of 12 seconds, during which the client sends data at its maximum capacity.

The `dect perf` command offers extensive configuration options to tailor the testing environment. Parameters such as Modulation and Coding Scheme (MCS), transmission power, packet slot count and duration can be adjusted to optimize performance under varying network conditions. Moreover, advanced options including Hybrid Automatic Repeat Request (HARQ) parameters can be adjusted for specific reliability requirements.

3.4.3 Ping Command Testing

In this scenario, a point-to-point topology was established using two nRF9161 DKs, where one device acted as a client and the other as a server. The objective was to evaluate the latency performance of the DECT NR+ network by assessing network responsiveness using the `dect ping` command. This command measures the round-trip time (RTT) between devices by displaying the RTT upon receiving the ping response following each ping request.

Server Configuration

To begin the evaluation, the server first scans a designated channel to assess its suitability for communication:

```
dect rssi_scan -c 1677
```

Subsequently, the server initializes the ping server on the specified channel. This command sets up the server to respond to ping requests:

```
dect ping -s --channel 1677
```

Client Configuration

The client initiates basic ping testing to measure network latency:

```
dect ping -c --channel 1677
```

This command prompts the client to send ping requests to the server on channel 1677, capturing RTT metrics for analysis. The `dect ping` command offers various configuration parameters to customize testing conditions. For instance, it allows adjustment of Modulation and Coding Scheme (MCS), transmission power, packet slot count, and duration of packet transmission. Additionally, specific parameters such as the number of ping requests sent or the interval between successive transmissions can be set for specific requirements.

4 Test and Results

This chapter outlines the methodology used to conduct the tests, outlining the results obtained (using the applications explained in the previous chapter) and evaluating the performance of the DECT NR+ standard in terms of key metrics such as latency, coverage, Packet Success Rate (PSR), data rate and power consumption. Here it is provided a comprehensive analysis of the experimental findings, offering insights into the capabilities and limitations of the DECT NR+ standard.

4.1 COVERAGE - PSR

4.1.1 Experimental methodology

To thoroughly evaluate the coverage and Packet Success Rate (PSR) of the DECT NR+ standard, a series of measurements were conducted in both outdoor and indoor environments. The outdoor tests took place in the park behind DTU Building 358 [59], while the indoor tests were conducted in the 101 Building, in particular in the Canteen and in the Library [60]. This variety of environments allowed for a comprehensive assessment of the standard's performance under different conditions. The experimental setup included the ability to adjust various transmission power (TX power) levels using the Broadcast application (which was used for this test, Section 3.3), changing the value of the macro ”POWER” (Listing 3.2). According to [43], it is possible to set the following TX levels:

Bit field	TX Power [dBm]
0000	-40
0001	-30
0010	-20
0011	-16
0100	-12
0101	-8
0110	-4
0111	0
1000	4
1001	7
1010	10
1011	13
1100	16
1101	19
1110	21
1111	23

Figure 4.1: *Different Transmit Power values available.*

The TX power levels tested were:

- 0x0 = -40 dBm (Minimum TX power)
- 0x4 = -12 dBm
- 0x7 = 0 dBm
- 0xb = 13 dBm
- 0xd = 19 dBm (Hardware maximum TX power)

According to Nordic's guidelines, it is important to note that any value higher than 0xd will still result in an actual transmission power of 19 dBm. During the measurements, the TX device was kept stationary while the RX device was moved to various distances to simulate different coverage scenarios. It was also measured the quality of the communication when the RX was moving from the maximum reached distance to the position of the transmitter (and vice versa).

Figure 4.2: *Transmitter device, kept fixed in the same position.*

For each measurement, the PSR was calculated after establishing a connection where exactly 1000 messages were exchanged. The PSR was determined by the ratio of correctly received messages to the total number of messages sent, including missed messages and those with CRC (Cyclic Redundancy Check) errors. Several important considerations were taken into account during the measurements:

- All measurements were conducted without any obstacles between the two devices to ensure clear line-of-sight conditions.
- Various physical obstructions significantly impacted performance. For instance, if a person stood in front of the device, the receiver stopped receiving signals entirely.
- Placing the receiver on the keyboard of the computer caused the device to cease functioning, due to interference. In this scenario, it failed to establish any connection and could not display any messages, whether they were reception confirmations or error notifications. However, when the device was positioned very close (within a few centimeters) to the computer, it operated normally. The problem was specific to the device being placed directly on top of the keyboard. For this reason, all measurements were considered with the device placed 30-40 cm away from the computer to avoid interference.

This setup ensured that the tests were conducted under consistent and controlled conditions, allowing for a reliable evaluation of the DECT NR+ standard's performance across different power levels and environments. The following sections will present the detailed results of these experiments.

4.1.2 Power = 0x0 (-40 dBm)

INDOOR Measurements

Distance	PSR
10cm	100 %
20cm	58.2 %
30cm	0 %
>30cm	No Connection established

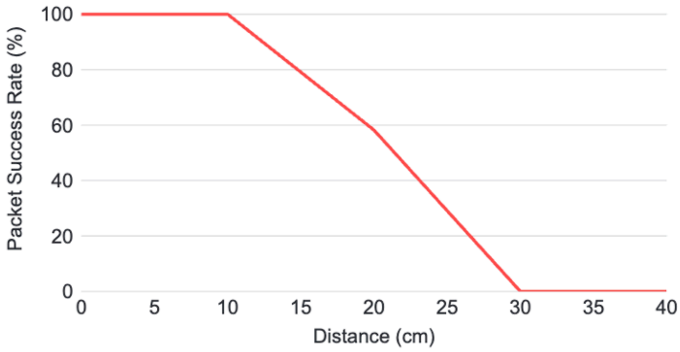


Figure 4.3: *PSR with TX Power = 0x0 (-40dBm, Minimum Pwr) in an Indoor environment*

OUTDOOR Measurements

Distance	PSR
10cm	100 %
20cm	0.5 %
30cm	0 %
>30cm	No Connection established

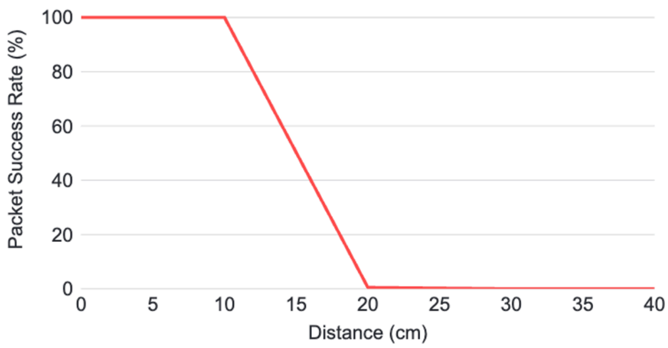


Figure 4.4: *PSR with TX Power = 0x0 (-40dBm, Minimum Pwr) in an Outdoor environment*

INDOOR/OUTDOOR comparison

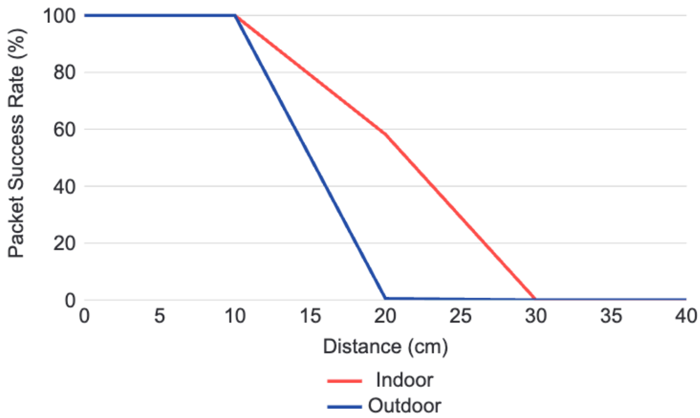


Figure 4.5: *PSR Comparison with TX Power = 0x0 in Indoor/Outdoor scenarios*

With such low transmission power levels, even minor changes in the device's orientation can significantly impact signal reception. For instance, if the transmitter (TX) and receiver (RX) are perfectly aligned, the RX will successfully receive the signal. However, if the TX is slightly tilted downwards, the RX may lose the signal and stop functioning. Regardless of the position of the devices, with a power of -40 dBm it is not possible to establish a connection beyond 30 cm distance.

4.1.3 Power = 0x4 (-12dBm)

INDOOR Measurements

Distance	PSR
1m	100 %
10m	100 %
15m	95.5 %
20m	83.2 %
30m	0 %
>30m	No connection established
1m-30m	39 %
30m-1m	54.2 %

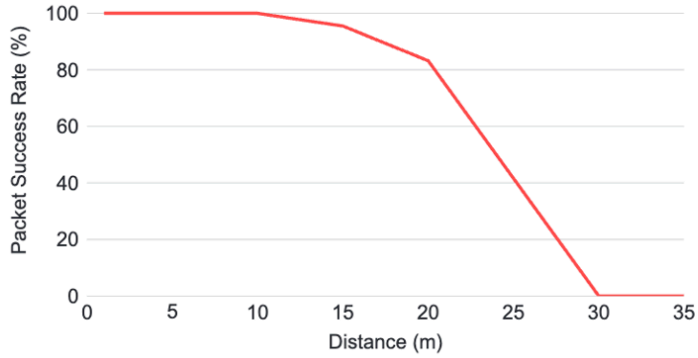


Figure 4.6: *PSR with TX Power = 0x4 (-12dBm) in an Indoor environment*

PSR maintains nearly 100% up to around 20 meters, after which it experiences a rapid decline, reaching 0% at about 30 meters.

OUTDOOR Measurements

Distance	PSR
1m	100 %
10m	100 %
15m	0 %
>15m	No Connection established
1m-15m	45 %
15m-1m	52 %

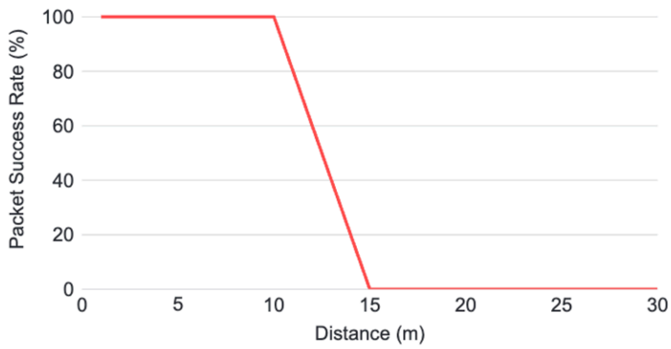


Figure 1.5: PSR measurements with TX Power = 0x4 (-12dBm)

Figure 4.7: *PSR with TX Power = 0x4 (-12dBm) in an Outdoor environment*

Similar to the indoor scenario, the PSR is high up to around 10 meters and then quickly falls to 0% by 15 meters.

INDOOR/OUTDOOR comparison

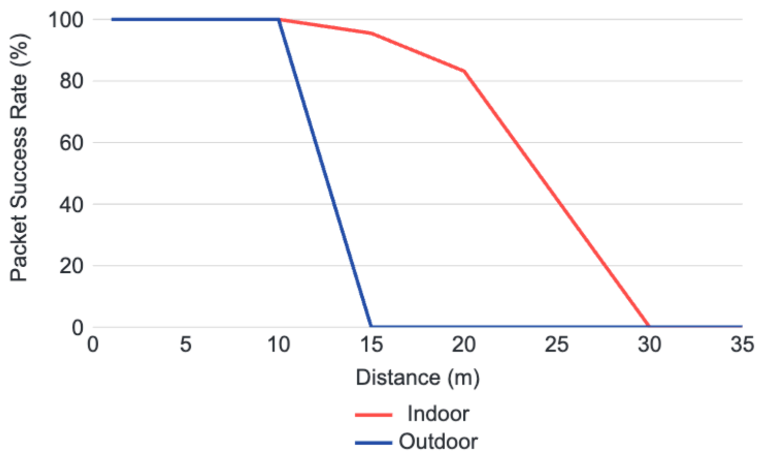


Figure 4.8: *PSR Comparison with TX Power = 0x4 in Indoor/Outdoor scenarios*

4.1.4 Power = 0x7 (0dBm)

INDOOR Measurements

Distance	PSR
1m	100 %
10m	100 %
20m	99.6 %
40m	99.9 %
50m	98 %
60m	19 %
>60m	No connection established
1m-60m	56.5 %
60m-1m	66.9 %

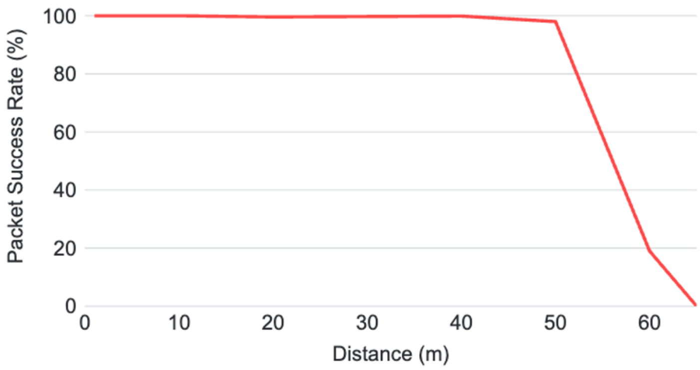


Figure 4.9: *PSR with TX Power = 0x7 (0dBm) in an Indoor environment*

PSR remains high up to approximately 50 meters, but then drops sharply to 0% by around 65 meters.

OUTDOOR Measurements

Distance	PSR
1m	100 %
10m	100 %
20m	100 %
40m	91.6 %
50m	62.4 %
60m	2.8 %
>60m	No Connection established
1m-60m	65.9 %
60m-1m	83.5 %

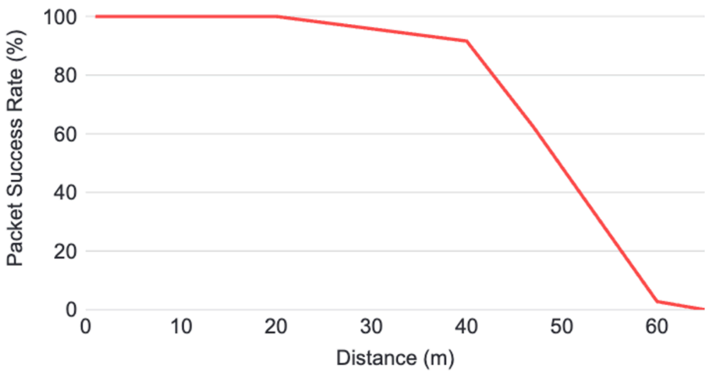


Figure 4.10: *PSR with TX Power = 0x7 (0dBm) in an Outdoor environment*

At 0 dBm, the PSR holds steady until about 40 meters, then rapidly decreases, hitting 0% by around 60 meters.

INDOOR/OUTDOOR comparison

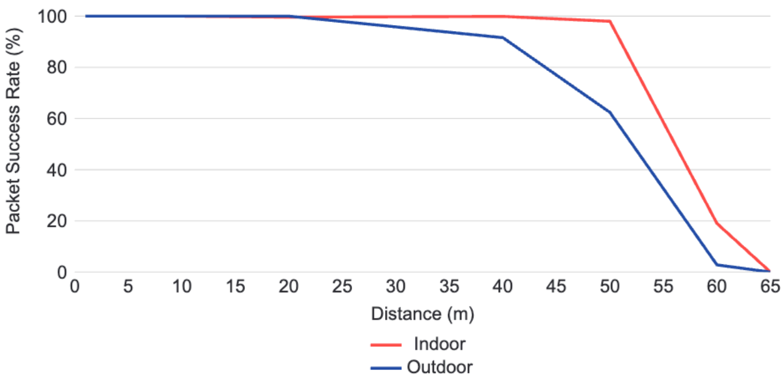


Figure 4.11: *PSR Comparison with TX Power = 0x7 in Indoor/Outdoor scenarios*

4.1.5 Power = 0xb (13dBm)

INDOOR Measurements

Distance	PSR
1m	100 %
10m	100 %
20m	100 %
40m	98.2 %
60m	99.3 %
80m	95.2 %
100m	100 %
>100m	-
1m-60m	99.9 %
60m-1m	97.4 %
1m-80m	84.2 %
80m-1m	90.3 %
1m-120m	-
120m-1m	-

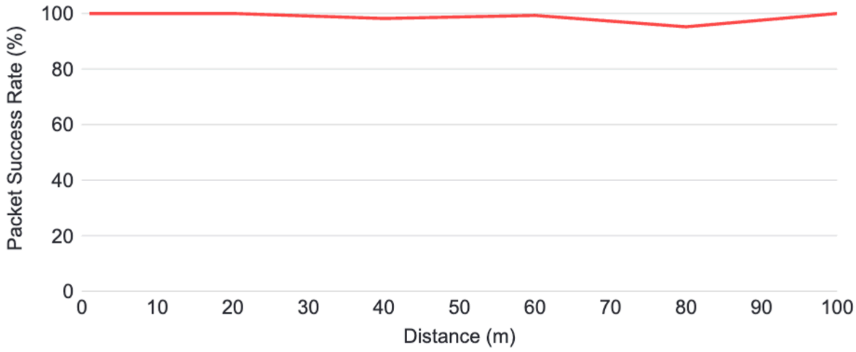


Figure 4.12: *PSR with TX Power = 0xb (13dBm) in an Indoor environment*

PSR remains near 100% for distances up to 100 meters, indicating excellent performance over the measured range. After 100m, it was not possible to continue the measurements cause it was not found an indoor place (with no obstacles) larger than 100m .

OUTDOOR Measurements

Distance	PSR
1m	100 %
10m	100 %
20m	100 %
40m	100 %
60m	98.8 %
80m	92.6 %
100m	98.4 %
>100m	0
1m-100m	55.3 %
100m-1m	74.8 %

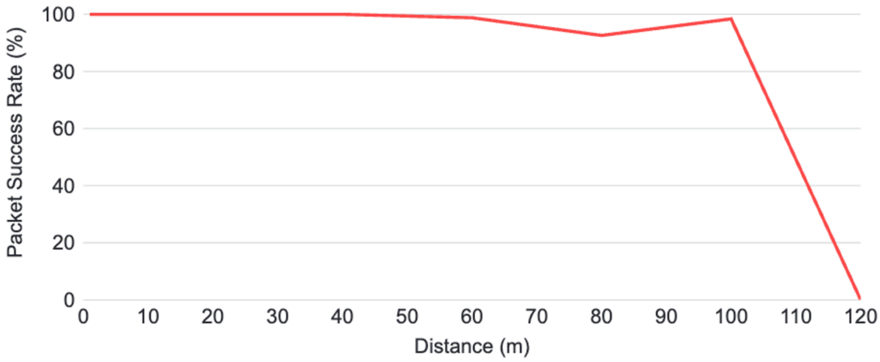


Figure 4.13: *PSR with TX Power = 0xb (13dBm) in an Outdoor environment*

PSR remains high up to around 100 meters, but then starts to decline, dropping to 0% just beyond 120 meters.

INDOOR/OUTDOOR comparison

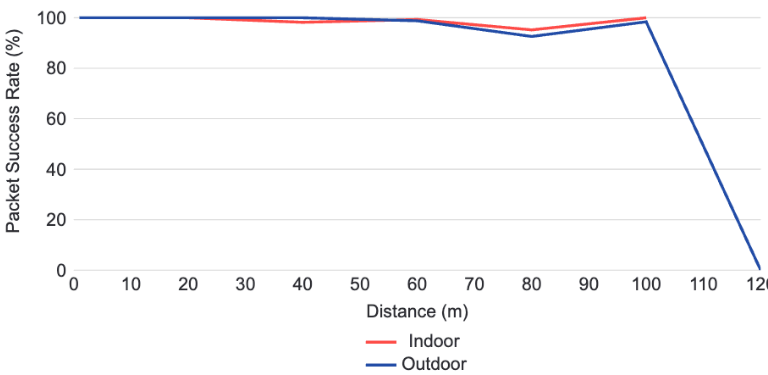


Figure 4.14: *PSR Comparison with TX Power = 0xb in Indoor/Outdoor scenarios*

4.1.6 Power = 0xd (19dBm)

INDOOR Measurements

Distance	PSR
1m	100 %
10m	100 %
20m	100 %
40m	100 %
60m	100 %
80m	99.9 %
100m	100 %
120m	-
150m	-
200m	-
>200m	-
1m-200m	-
200m-1m	-

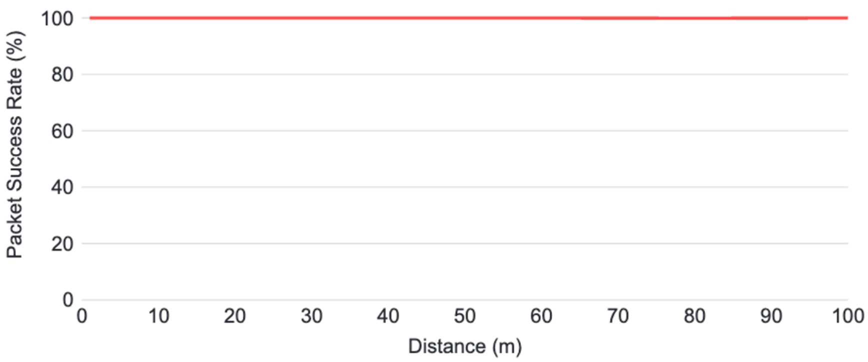


Figure 4.15: *PSR with TX Power = 0xd (19dBm, HW Maximum Power) in an Indoor environment*

PSR remains near 100% for distances up to 100 meters, indicating excellent performance over the measured range. After 100m, it was not possible to continue the measurements cause it was not found an indoor place (with no obstacles) larger than 100m .

OUTDOOR Measurements

Distance	PSR
1m	100 %
10m	100 %
20m	100 %
40m	100 %
60m	100 %
80m	100 %
100m	100 %
120m	95.1 %
150m	100 %
200m	7.4 %
>200m	0 %
1m-200m	69.0 %
200m-1m	69.3 %

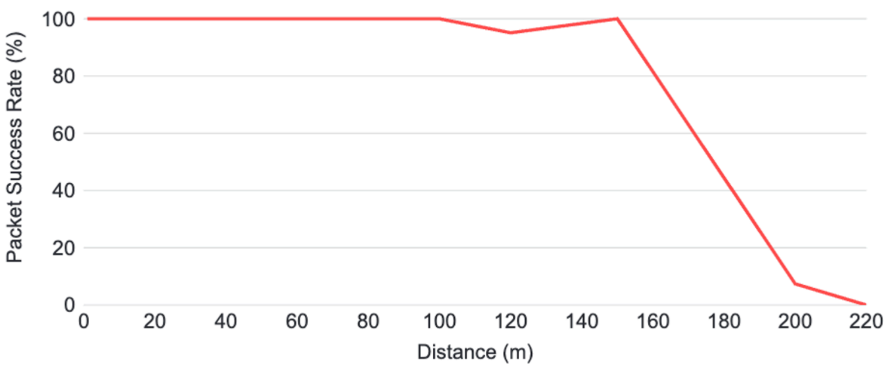


Figure 4.16: *PSR with TX Power = 0xd (19dBm, HW Maximum Power) in an Outdoor environment*

PSR is nearly 100% up to around 160 meters, after which it falls off, reaching 0% at about 220 meters.

INDOOR/OUTDOOR comparison

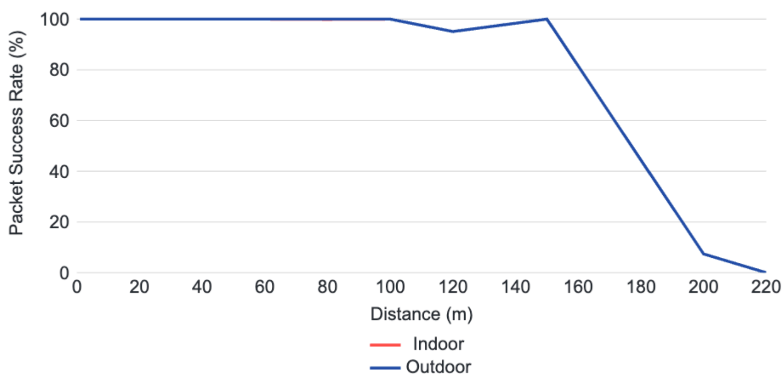


Figure 4.17: *PSR Comparison with TX Power = 0xd in Indoor/Outdoor scenarios*

4.1.7 Considerations on Indoor and Outdoor performances

Despite the absence of results for the 0xb and 0xd power levels, the overall performance metrics indicate higher efficiency in indoor environments compared to outdoor settings. Several factors could have a contribution to the enhanced performance observed in indoor environments:

- **Controlled Environment:** Indoor settings provide a more controlled environment, particularly in the absence of external elements such as wind, which can affect signal strength and stability by moving the USB-C cable and other equipment.
- **Signal Reflection and Multi-path Propagation:** Indoor environments benefit from signal reflections and multi-path propagation. These phenomena occur when signals bounce off walls and other surfaces, creating multiple paths for the signal to reach the receiver. This can enhance the overall signal strength.
- **Signal Directionality:** Indoors, signals are often “channeled” or “directed” towards the receiver due to the presence of walls and other structures. This directionality helps maintain a stronger and more focused signal path. In contrast, outdoor environments allow signals to propagate more diffusely, spreading out in all directions. This widespread propagation results in reduced signal strength over distance as the energy disperses into the broader environment.

4.1.8 Comparison of Device Movement Performance

Table 4.1: *Performance Comparison of the Device in movement (for both Indoor and Outdoor scenarios)*

POWER (dBm)	1-MAX Indoor	MAX-1 Indoor	1-MAX Outdoor	MAX-1 Outdoor
0x4	39.0 %	54.2 %	45.0 %	52.0 %
0x7	56.5 %	66.9 %	65.9 %	83.5 %
0xb	-	-	55.3 %	74.8 %
0xd	-	-	69.0 %	69.3 %

1. **Higher Stability in MAX-1 Measurements:** Measurements taken while moving from the maximum distance towards the transmitter (MAX-1) consistently show higher performance compared to those taken while moving from close proximity to the maximum distance (1-MAX). This suggests that the device performs better when it starts from the farthest point and moves into a region with a more stable and stronger signal. In contrast, starting from a close, stable position and moving towards a more distant, unstable position likely results in deteriorating signal quality and reduced performance.
2. **Impact of Power on Stability:** Connection stability improves with higher transmission power. Data for the 0x4 power level shows the lowest performance, indicating that lower power settings are less effective in maintaining a stable connection during movement. As transmission power increases to 0x7, performance metrics improve, suggesting that higher power levels enhance the device’s ability to maintain a stable connection while in motion.
3. **Performance Ceiling:** The packet success rate (PSR) generally does not exceed 70-75% in most cases, except for the MAX-1 outdoor measurement at power level 0x7, which reaches 83.5%. This ceiling suggests inherent limitations in the device’s communication performance while moving, possibly due to environmental factors, device sensitivity, or signal interference.

4.1.9 Comparative Analysis

INDOOR Measurements

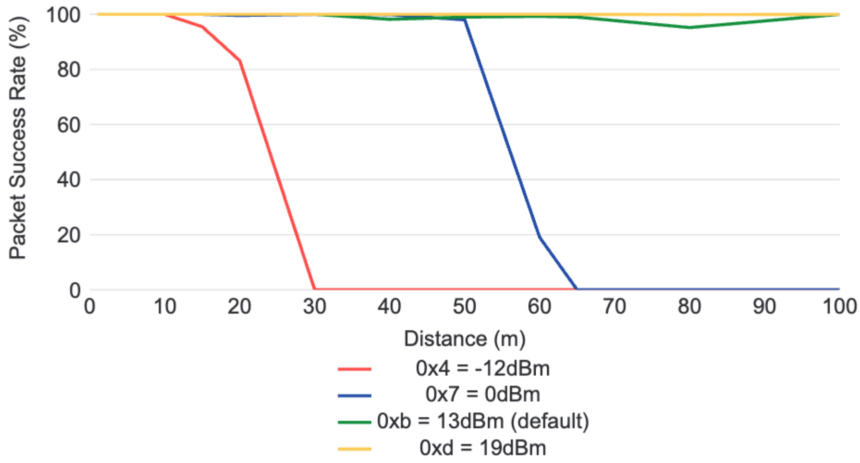


Figure 4.18: *PSR Comparison with different PW TX values*

OUTDOOR Measurements

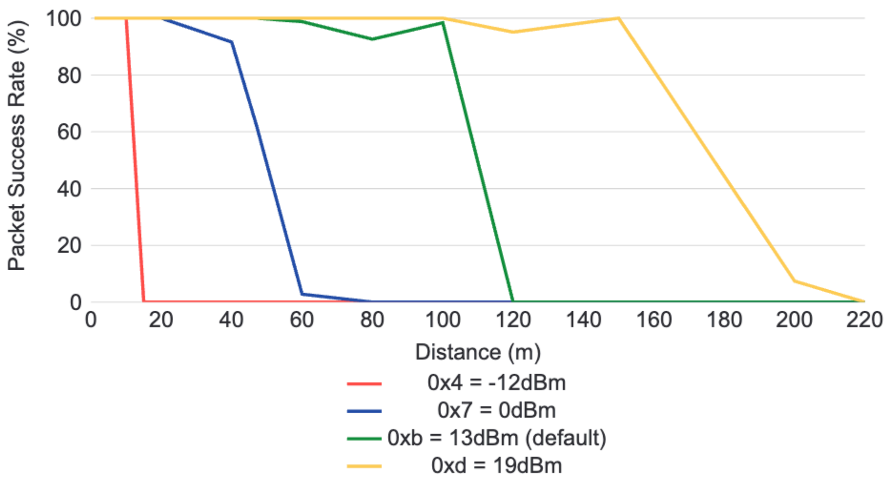


Figure 4.19: *PSR Comparison with different PW TX values*

Across all power levels and in both indoor and outdoor environments, the Packet Success Rate (PSR) demonstrates excellent performance up to a certain critical distance. Beyond this critical distance, there is a sharp drop in PSR, indicating a sudden and significant decline in communication quality.

One possible reason for this performance drop at specific distances could be related to the length of the packets being transmitted. In this experiment, each packet contains 32 bits (4 bytes), represented by a single uint32_t variable. For future research, it would be interesting to analyze how performance varies with different packet lengths. As the amount of data (and thus the number of bits) transmitted increases, the likelihood of encountering a bit error also increases, potentially degrading the PSR.

The maximum distance achieved during these measurements, with the highest transmission power, was 220 meters. This is a commendable result, especially when considering that Wi-Fi routers operating on the 2.4 GHz band typically reach up to 90 meters (approximately 300 feet) outdoors [61]. However, DECT NR+ and Nordic Semiconductor promise greater capabilities. Nordic Semiconductor has acknowledged that they have encountered issues with the 61 Development Kit on radio sensitivity, resulting in a range that is significantly less than what they have observed with their lab device. They are investigating the issue, but do not have clear data at the moment.

Despite these challenges, it is important to note that the maximum coverage offered by DECT NR+ will likely be less than other IoT technologies such as LTE-M or NB-IoT. For example, LTE-M and NB-IoT transmission range, in terms of maximum distance between the tower and the IoT device, is approximately one kilometer in urban areas and approximately 10 kilometers in rural areas [62] [63].

4.2 Data Rate

4.2.1 Experimental methodology

To evaluate the data rate performance of the DECT NR+ standard, a series of measurements were conducted in indoor environments, specifically in the 101 Building Canteen and Library. The network utilized the Modem Shell Application to facilitate communication between the devices, in particular the *Throughput Performance Testing* use case (Subsection 3.4.2) was used. The experiments were taken with a fixed transmission power of 13 dBm for the client device, with the devices positioned at a fixed distance of 30-35 meters, ensuring excellent coverage performance given the TX power level.

In the experimental setup, one device always acted as a Server while the other acted as a Client. The Modem Shell Application facilitated the ability to set various Modulation and Coding Scheme (MCS) values ranging from 0 to 4. For each MCS value tested, the number of slots that make up a packet was also increased, allowing for larger amounts of data to be transmitted within the same packet. In fact, the slot number directly influences the size of each packet in terms of bytes. All measurements were conducted over a fixed 12-second interval where the client device transmitted data continuously to the server. The server displayed the amount of data correctly received relative to the total bytes sent by the client, alongside the corresponding data rate from the server's perspective.

It's important to note that, for a given number of slots per packet, the number of packets sent remained constant, regardless of the MCS. For instance, when the number of slots per packet was set to 1, the client consistently sent 7490 packets, both with MCS = 0 or with MCS = 4. Similarly, with 2 slots per packet, the client sent 7217 packets, and so forth. The variation in MCS affected only the amount of data transmitted within each packet, not the total number of packets sent.

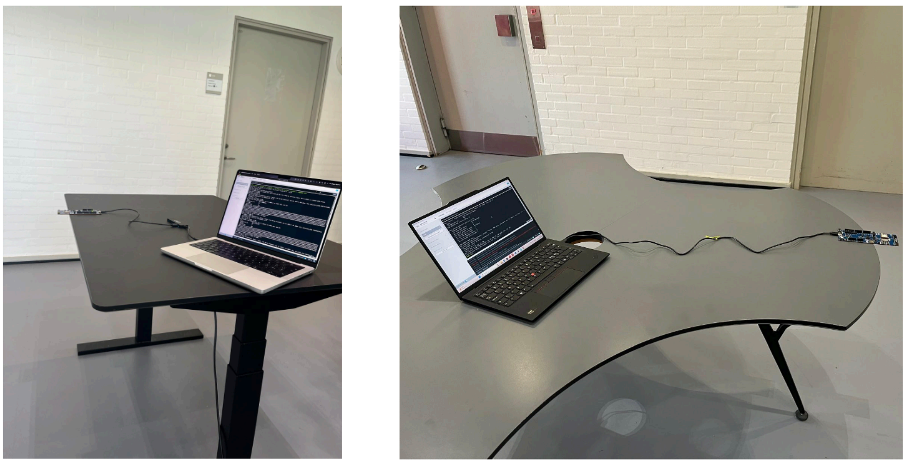


Figure 4.20: *The client (on the left) and server (on the right) devices connected to the computer and positioned on a table, ready for the data rate performance tests.*

The maximum number of slots per packet changed based on the MCS being considered. For example, with MCS values of 0 and 1, the maximum number of slots was 8, while with MCS value 4, the maximum number of slots was 4. The amount of bytes that can be sent in each packet varied according to the number of slots in the packet, as illustrated in the following figure.

MSC / Subslot-index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MCS 0	0	136	264	400	536	664	792	920	1064	1192	1320	1448	1576	1704	1864	1992
MCS 1	32	296	552	824	1096	1352	1608	1864	2104	2360	2616	2872	3128	3384	3704	3960
MCS 2	56	456	856	1256	1640	2024	2360	2744	3192	3576	3960	4320	4768	5152	5536	–
MCS 3	88	616	1128	1672	2168	2680	3192	3704	4256	4768	5280	–	–	–	–	–
MCS 4	144	936	1736	2488	3256	4024	4832	5600	–	–	–	–	–	–	–	–

Figure 4.21: *Bits per subslot index with given modulation scheme (from [57])*

In this table, the amount of data is expressed based on MCS/Subslot. However, the result is the same because each slot that can be added or removed consists of 2 subslots. The documentation uses subslots, whereas this application uses slots.

The objective of these measurements was to observe how the communication data rate changes by varying two parameters:

- MCS (from 0 to 4)
- Number of slots in a packet

For every MCS value a table displays, with a specific number of slots, the total bytes sent, the bytes per packet, number of packets sent and the data rates for both the client and server. This setup ensured a systematic approach to testing the data rate capabilities of the DECT NR+ standard, providing valuable insights into its performance under different configurations and conditions. The following sections will present the detailed results of these experiments.

4.2.2 MCS = 0

Slots	Tot Byte	Byte/packet	N. Packets	Client DR (Kbps)	Server DR (Kbps)
1	127 330	17	7490	84.89	84.78
2	360 850	50	7217	240.57	240.43
3	479 325	83	5775	319.55	319.28
4	553 840	115	4816	369.20	368.67
5	615 370	149	4130	410.21	409.43
6	653 772	181	3612	435.81	435.21
7	684 369	213	3213	456.21	455.29
8	719 859	249	2891	479.83	479.03

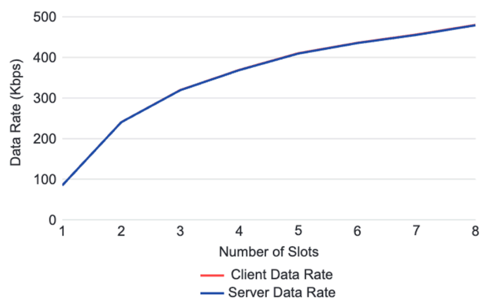


Figure 4.22: *Data Rate (Kbps) vs. Number of Slots for Client and Server using MCS = 0.*

4.2.3 MCS = 1

Slots	Tot Byte	Byte/packet	N. Packets	Client DR (Kbps)	Server DR (Kbps)
1	277 130	37	7490	184.75	184.47
2	743 351	103	7217	495.55	495.10
3	975 975	169	5775	650.65	649.49
4	1 122 128	233	4816	747.98	746.70
5	1 218 350	295	4130	812.17	810.56
6	1 296 708	359	3612	864.33	862.97
7	1 359 099	423	3213	906.02	903.56
8	1 431 045	495	2891	953.79	951.55

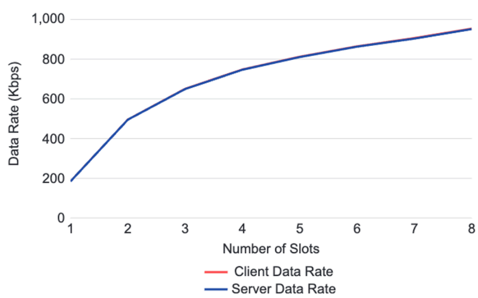


Figure 4.23: *Data Rate (Kbps) vs. Number of Slots for Client and Server using MCS = 1.*

4.2.4 MCS = 2

Slots	Tot Byte	Byte/packet	N. Packets	Client DR (Kbps)	Server DR (Kbps)
1	426 930	57	7490	284.60	284.00
2	1 133 069	157	7217	755.38	754.52
3	1 461 075	253	5775	974.02	972.73
4	1 651 888	343	4816	1101.14	1098.92
5	1 846 110	447	4130	1230.60	1226.80
6	1 950 480	540	3612	1300.03	1296.80
7	2 069 172	644	3212	1379.40	1375.65

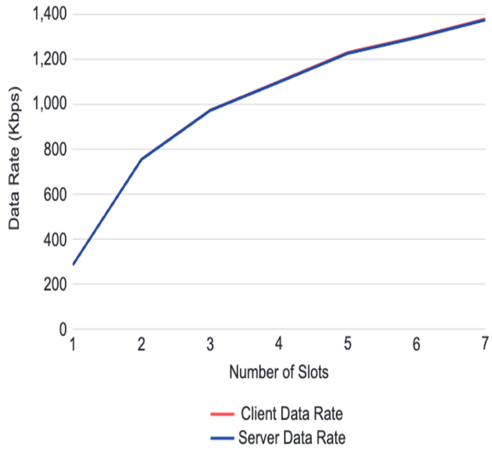


Figure 4.24: *Data Rate (Kbps) vs. Number of Slots for Client and Server using MCS = 2.*

4.2.5 MCS = 3

Slots	Tot Byte	Byte/packet	N. Packets	Client DR (Kbps)	Server DR (Kbps)
1	576 730	77	7490	384.48	383.73
2	1 508 353	209	7217	1005.57	1003.40
3	1 934 625	335	5775	1289.64	1288.01
4	2 229 808	463	4816	1486.41	1482.15
5	2 461 480	596	4130	1640.85	1636.52

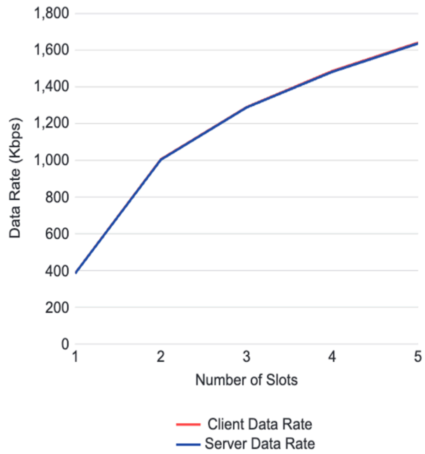


Figure 4.25: *Data Rate (Kbps) vs. Number of Slots for Client and Server using MCS = 3.*

4.2.6 MCS = 4

Slots	Tot Byte	Byte/packet	N. Packets	Client DR (Kbps)	Server DR (Kbps)
1	876 330	117	7490	584.20	583.38
2	2 244 487	311	7217	1496.32	1494.71
3	2 904 825	503	5775	1936.39	1932.93
4	3 371 200	700	4816	2247.28	2242.98

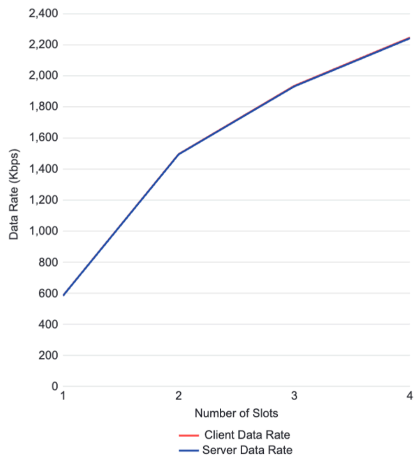


Figure 4.26: *Data Rate (Kbps) vs. Number of Slots for Client and Server using MCS = 4.*

4.2.7 Comparative Analysis

MCS Value	Max Data Rate (Client, Kbps)	Max Data Rate (Server, Kbps)
0	479.83	479.03
1	953.79	951.55
2	1379.40	1375.65
3	1640.85	1636.52
4	2247.28	2242.98

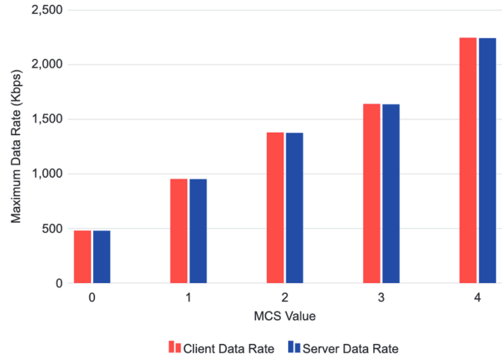


Figure 4.27: *Table and Bar Graphs for Maximum Data Rate Achieved by Each MCS Value for Both Client and Server.*

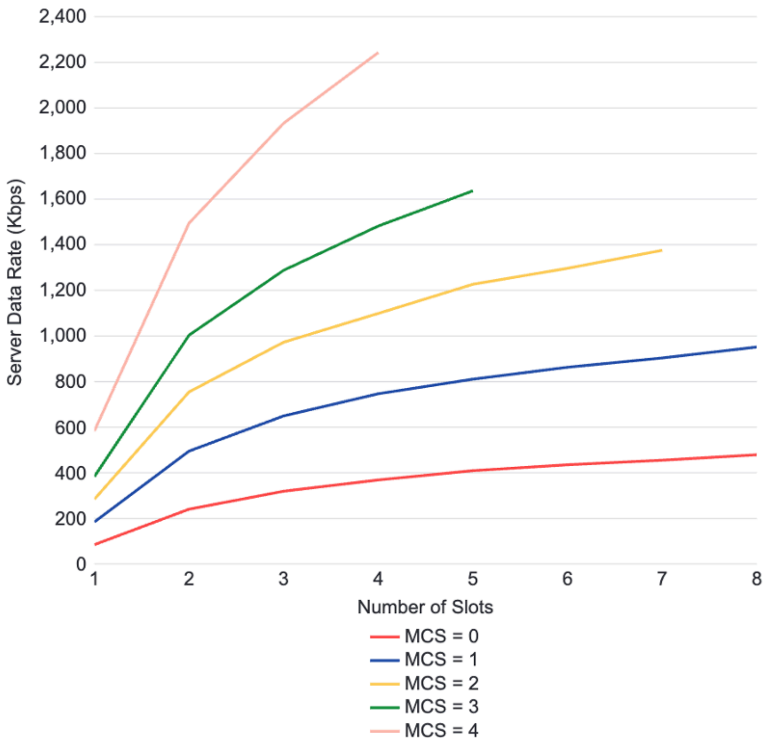


Figure 4.28: *Comparison of Server Data Rate (Kbps) vs. Number of Slots for different MCS values. Each line represents the data rate performance for a specific MCS value (0 to 4) as the number of slots per packet increases from 1 to 8.*

The comparison of data rates across different MCS values clearly indicates that higher MCS values provide better performance (Figure 4.27). The maximum data rates achieved for each MCS value illustrate the potential of DECT NR+ in various configurations:

- MCS = 0: Provides the lowest data rate, suitable for applications with minimal data requirements.
- MCS = 1-2: Offer moderate data rates, suitable for applications with moderate data requirements.
- MCS = 3-4: Provide the highest data rates, suitable for high-data-demand applications.

The comparison graph (Figure 4.28) also highlights that increasing the number of slots in a packet (thus increasing the quantity of Bytes sent in a single packet) enhances the data rate significantly, especially for higher MCS values (where for the same number of slots more data is sent in a packet). This suggests that DECT NR+ can be effectively optimized for different data requirements by adjusting the MCS and number of slots.

Disadvantages of Increasing the MCS

While increasing the Modulation and Coding Scheme (MCS) values significantly enhances data rates, it is essential to consider the potential disadvantages associated with higher MCS values:

- **Increased Bit Error Rate (BER)**: Higher MCS values use more complex modulation schemes that are more susceptible to noise and interference. This can lead to an increased bit error rate, potentially reducing the reliability of the communication.
- **Reduced Range**: Higher MCS values typically require a stronger signal, which can limit the effective communication range. This means that while data rates are higher, the devices may need to be closer to maintain a stable connection.
- **Higher Power Consumption**: Operating at higher MCS values often requires more power to maintain signal quality and manage increased processing demands. This can be a significant disadvantage in battery-powered IoT devices where energy efficiency is critical.
- **Complexity and Cost**: Implementing higher MCS values may require more sophisticated and costly hardware, increasing the overall system cost. This can be a barrier for low-cost IoT applications.

Balancing Performance and Practicality

After evaluating the potential disadvantages of increasing the Modulation and Coding Scheme (MCS) values, it is crucial to balance the need for higher data rates with practical considerations such as reliability, environmental adaptability and security.

Balancing higher data rates with the reliability of the communication link is vital. Applications should be evaluated to determine the optimal MCS value that provides sufficient data throughput while maintaining an acceptable error rate and communication range. This balance ensures that the system remains robust and dependable, even in varying conditions.

The adaptability of the DECT NR+ standard to different environmental conditions also requires further exploration. Performance in high-interference environments or areas with significant physical obstructions should be studied to ensure robust communication. Understanding how the standard behaves in real-world scenarios with diverse challenges will help in fine-tuning its deployment strategies.

Security is an additional concern, especially with higher data rates and more complex modulation schemes. Ensuring secure data transmission, particularly at higher MCS values, is critical to protect against potential vulnerabilities. Robust security measures must be adopted to safeguard data integrity and privacy.

However, it is noteworthy that during the measurements conducted in this study, no degradation in performance concerning data reception accuracy was observed. The data rate on the client side (indicating the amount of data sent by the first device) was nearly identical to the data rate on the server side (indicating the amount of data correctly received by the second device), regardless of the MCS used. This is evidenced by the fact that the data rate curves for the client and server displayed on the graphs for each MCS value overlap and are indistinguishable from one another. This consistency in performance suggests that, under the tested conditions, DECT NR+ maintains a high level of reliability in data transmission even at higher MCS values.

Future research should explore under what conditions significant performance degradation in data reception accuracy might occur with increasing MCS. Identifying these conditions would provide valuable insights into the operational limits of DECT NR+ and help in developing strategies to mitigate potential issues, ensuring robust and reliable communication across various scenarios. Several other areas can further enhance the performance and applicability of the DECT NR+ standard:

- **Dynamic MCS Adaptation**: Developing algorithms for dynamic adaptation of MCS values based on real-time environmental conditions and communication quality could optimize performance. This would allow the system to automatically adjust for the best balance between data rate and reliability.
- **Extended Range Testing**: Conducting more extensive range testing in various real-world scenarios, including urban, suburban and rural environments, would provide a more comprehensive understanding of the standard's capabilities and limitations.
- **Energy Efficiency Studies**: Investigating the energy efficiency of different MCS values and slot configurations can help design low-power IoT solutions that do not compromise on performance.

Evaluation of Nordic's Key Feature: Data Rate up to 3.4 Mbps
Nordic Semiconductor highlights that the DECT NR+ standard supports data rates up to 3.4 Mbps, depending on the modulation scheme used [64]. However, the maximum data rate observed during the experiments did not fully achieve this value. The closest results (2.2Mbps), achieved with MCS 4, showed a significant increase in data rate but still fell short of the 3.4 Mbps mark. Several factors, including hardware limitations, environmental conditions and potential interference, could contribute to this discrepancy.

Despite not reaching the peak value, the observed data rates still demonstrated robust performance, especially with higher MCS settings. The experimental data suggests that while the theoretical maximum data rate of 3.4 Mbps is an attractive feature, real-world applications often face numerous practical challenges that can affect performance. The DECT NR+ standard, even without reaching the absolute peak rate, still offers competitive and reliable data transmission capabilities suitable for various IoT applications.

4.3 Latency

4.3.1 Experimental methodology

The experimental setup for measuring latency was designed to closely mirror that used for evaluating data rates. Measurements were conducted in indoor environments (101 Building Canteen and Library) with a fixed transmission power of 13 dBm for the client device. The devices were positioned at a fixed distance of 30-35 meters, ensuring excellent coverage performance considering the TX power level. In this setup, one device always acted as the Client, initiating communication by sending ping requests, while the other acted as the Server, responding with ping replies. The Modem Shell Application was used to establish these communications (in particular the *Ping Command Testing* use case, Subsection 3.4.3) across varying Modulation and Coding Scheme (MCS) values ranging from 0 to 4. For each MCS value, the number of slots that make up a packet was varied, thereby changing the total amount of data carried in each ping message. (in the same way as shown in Figure 4.21, like the data rate measurements).

The objective was to observe how communication latency changes by varying:

- the MCS (from 0 to 4);
- the Number of slots in a packet.

Latency was evaluated using the Round Trip Time (RTT) metric obtained from ping messages. RTT measures the time taken for a signal to go from the client to the server and back to the client. For every MCS value a table displays, with a specific number of slots, the total amount of bytes per ping message, payload length and RTT needed to send a ping request and wait for a ping response. Measurements were taken as follows: for each pair (MCS, n_slot), a ping request was sent, and the RTT was measured upon receiving the ping response. This process was repeated three times for each pair (MCS, n_slot). The RTT shown for each pair is the arithmetic mean of the three RTT measurements. The following sections will present the detailed results of these experiments.

4.3.2 MCS = 0

Slots	Byte/packet	Payload Length (B)	RTT (msec)
1	17	9	73
2	50	42	76
3	83	75	80
4	115	107	83
5	149	141	87
6	181	173	91
7	213	205	95
8	249	241	99

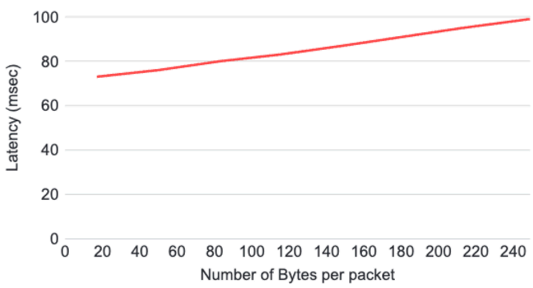


Figure 4.29: *RTT (Round Trip Time) vs. Number of Bytes per packet using MCS = 0.*

4.3.3 MCS = 1

Slots	Byte/packet	Payload Length (B)	RTT (msec)
1	37	29	75
2	103	95	82
3	169	161	89
4	233	225	96
5	295	287	102
6	359	351	109
7	423	415	116
8	495	487	125

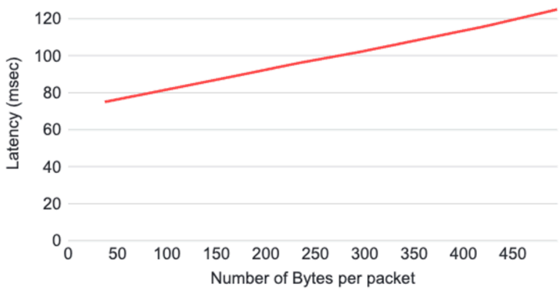


Figure 4.30: *RTT (Round Trip Time) vs. Number of Bytes per packet using MCS = 1.*

4.3.4 MCS = 2

Slots	Byte/packet	Payload Length (B)	RTT (msec)
1	57	49	77
2	157	149	87
3	253	245	97
4	343	335	107
5	447	439	118
6	540	532	128
7	644	636	139

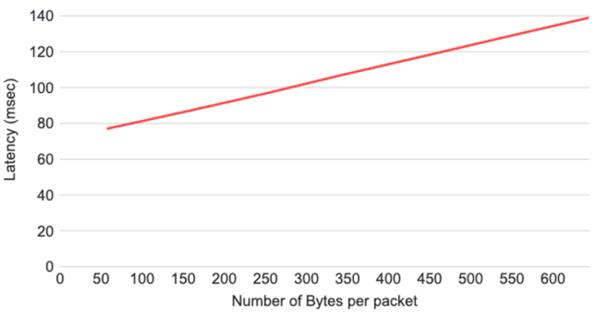


Figure 4.31: *RTT (Round Trip Time) vs. Number of Bytes per packet using MCS = 2.*

4.3.5 MCS = 3

Slots	Byte/packet	Payload Length (B)	RTT (msec)
1	77	69	79
2	209	201	92
3	335	327	106
4	463	455	120
5	596	588	133

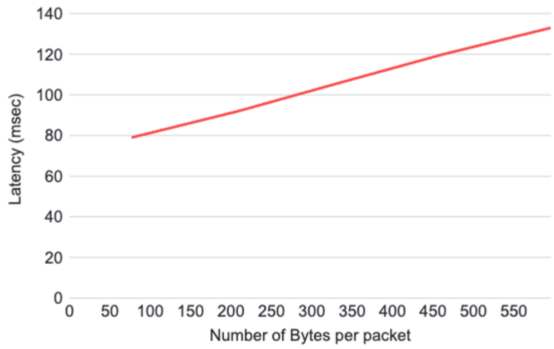


Figure 4.32: *RTT (Round Trip Time) vs. Number of Bytes per packet using MCS = 3.*

4.3.6 MCS = 4

Slots	Byte/packet	Payload Length (B)	RTT (msec)
1	117	109	83
2	311	303	103
3	503	495	123
4	700	692	143

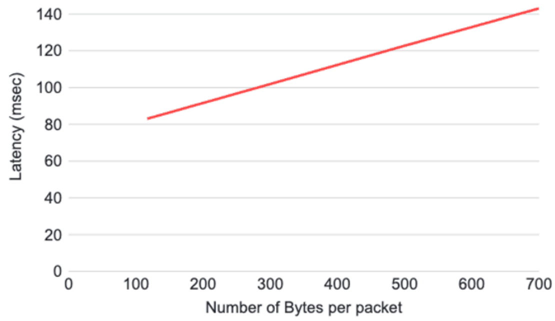


Figure 4.33: *RTT (Round Trip Time) vs. Number of Bytes per packet using MCS = 4.*

4.3.7 Comparative Analysis

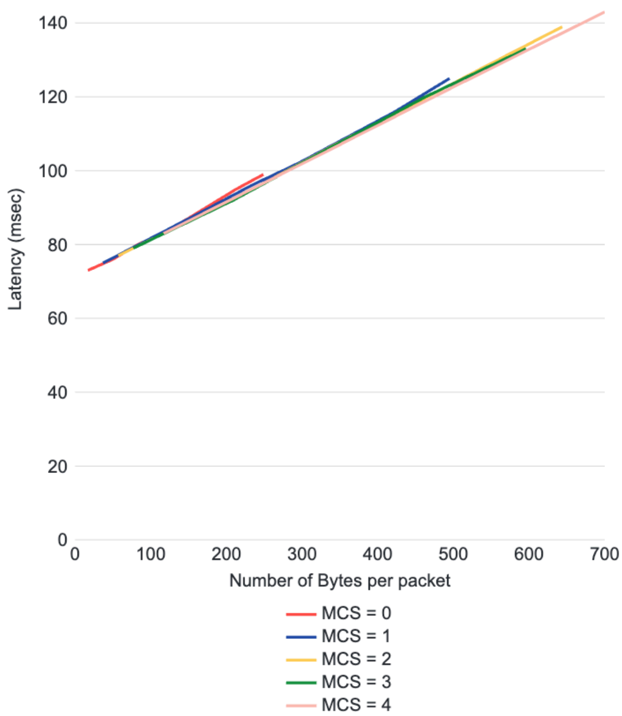


Figure 4.34: *Comparison of RTT (msec) vs. Number of Bytes per packet for different MCS values. Each line represents the RTT performance for a specific MCS value (0 to 4) as the number of bytes per packet increases.*

An interesting observation from the latency measurements is the linear behavior exhibited by RTT for each MCS value. The RTT increases linearly with the number of bytes per packet, as demonstrated in Figures 4.29 through 4.33 for MCS values 0 to 4. This linear relationship allows for a straightforward calculation of the slope (m) of the regression line representing RTT as a function of the number of bytes per packet. The slope indicates the rate at which RTT increases with increasing packet size, providing a measure of latency performance.

The slope of the regression line can be calculated using the linear regression formula for the slope (m) of a line passing through a series of points (x_i, y_i):

$$m = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

where n is the number of points. Applying this formula to the RTT data, the slopes for each MCS value were determined as follows:

- MCS = 0: $m = 0.11348459127172401$
- MCS = 1: $m = 0.107799174729572$
- MCS = 2: $m = 0.10620516293926952$
- MCS = 3: $m = 0.10524140409728046$
- MCS = 4: $m = 0.10303709946052833$

These results confirm the expected trend: as the MCS value increases, the slope decreases. A smaller slope indicates that the same amount of data can be transmitted in less time, thereby improving latency performance. This reduction in the slope with higher MCS values is consistent with improved efficiency and faster data transmission capabilities.

Implications of Slope Values

The positive and less-than-one nature of these slopes indicates that, for a given MCS, larger packets are sent proportionally faster than smaller packets. For instance, with MCS = 1, a ping message of 103 bytes corresponds to an RTT of 82 ms, whereas a ping message of 233 bytes (more than double the size) corresponds to an RTT of 96 ms (only a 14 ms increase, which is a 17% increase in time). This efficiency in handling larger packets highlights the advantage of higher MCS values in optimizing data transmission times.

MCS Value	Max Byte/packet	Max RTT (msec)
0	249	99
1	495	125
2	644	139
3	596	133
4	700	143

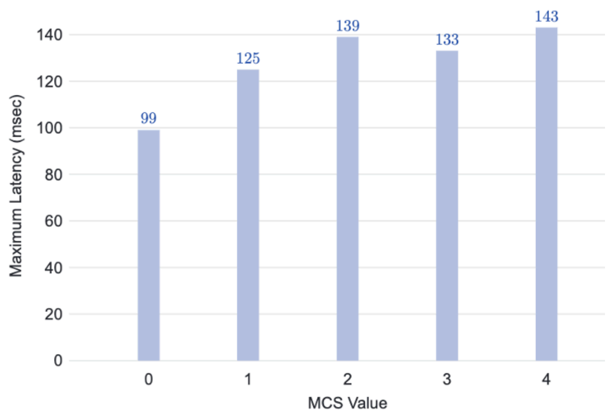


Figure 4.35: *Table and Bar Graphs for Maximum Latency achieved by Each MCS Value for the Client.*

Compliance with URLLC Requirements
For DECT NR+ to meet Ultra-Reliable Low Latency Communication (URLLC) requirements, it must ensure end-to-end low latency (<50 ms on the application layer, <1 ms on the radio interface) [39]. In our experiments, the RTTs observed were higher than the threshold, particularly at lower MCS values. It is important to note that the ping message communication used in the experiments can be considered at the application layer. The higher-than-expected RTTs suggest that while DECT NR+ performs reliably, there may be challenges in meeting the stringent URLLC latency requirements under certain conditions.

Further considerations on the results could focus on optimizing system parameters, exploring the impact of different environmental conditions on latency, and developing advanced algorithms for dynamic MCS adaptation to achieve better latency performance. The robustness of DECT NR+ in maintaining low latency under various scenarios should be further investigated to ensure it meets the stringent requirements of URLLC applications.

4.4 Power Consumption

4.4.1 Experimental methodology

To assess the power consumption of devices within a DECT NR+ network, an experimental set-up was configured with three devices designated as Sink, Relay and Client/Leaf. The network utilized the Modem Shell Application to facilitate communication between the devices. In particular, for this experiment, the *DECT NR+ Network Creation* use case (Subsection 3.4.1) was used. Apart from the three nRF9161 DKs, The Power Profiler Kit II (in order to measure the power consumption of each device accurately, Subsection 3.1.2) was also essential for the experimental set-up.

All three devices were configured with a fixed TX Power of 13 dBm and MCS set to 0. The devices were strategically positioned at a fixed distance to ensure reliable communication and effective measurement of power consumption. The distances were set as follows:

- The Client/Leaf was placed 30 meters from the Relay.
- The Relay was placed 30 meters from the Sink.
- The total distance between the Client/Leaf and the Sink was 60 meters.

In this configuration, the Client and Sink exchanged messages while the Relay continuously managed the forwarding of these messages. The choice of a 20-30 meter distance between devices was based on ensuring excellent coverage performance, given the fixed transmission power of 13 dBm. This power level was selected to strike an optimal balance between coverage and performance without leading to excessive power consumption. Using a high transmission power, such as 19 dBm, could result in increased power dissipation by the devices, which is not desirable for power-saving objectives. The 13 dBm transmission power provides a compromise that maintains good performance while managing power consumption effectively.

The primary objective of this experiment was to observe and compare the power consumption of the three devices within the DECT NR+ network. Special attention was given to the Relay device, as it was expected to have higher power consumption compared to the Client due to its role in continuously forwarding messages. By analyzing the power consumption data obtained from the Power Profiler Kit II, the experiment aimed to provide insights into the power-saving capabilities of DECT NR+ and identify any potential areas for optimization in terms of power usage across different network roles and configurations.

4.4.2 Sink Device Power Consumption Analysis

In this section is analyzed the power consumption of the Sink device within the DECT NR+ network based on the experimental data presented in Table 4.2. This analysis covers different operational phases including Idle, RSSI Scanning, Sending Beacons and Channel Scanning.

Table 4.2: *Power Consumption of Sink Device in Various Operational Phases*

Action	**Avg. Current (mA)**	**Max Current (mA)**	**Avg. Pwr (W)**	**Time (s)**	**Charge (C)**
Idle	33.0	35.5	0.165	60.00	1.98
RSSI Scanning	36.8	40.2	0.184	42.21	1.55
Sending Beacons	51.6	107.8	0.258	20.00	1.03
Channel Scanning	61.0	64.2	0.305	4.00	0.24

The *Idle* phase refers to the period when the device is powered on and initialized but no data transfer occurs between devices, nor are any channel scans performed. During this phase, the only command executed is `dect sett`, which is used to configure specific device parameters. The messages displayed in the Serial Terminal during this phase are merely confirmations of the successful changes made by the `dect sett` command. In this phase, the device's average current consumption is 33.0 mA, with occasional peaks reaching 35.5 mA. These peaks correspond to the output confirmations of parameter changes. Additionally, for the purpose of analysis, a fixed time interval of 60 seconds was arbitrarily chosen to measure the duration of this phase across all three devices being analyzed.

The *RSSI Scanning* phase begins after executing the `dect beacon_start` command. This phase involves scanning all available channels (from 1657 to 1677) to determine their status (Free, Possible, or Busy). The device will then start sending beacon messages on the first channel declared as free. Each channel is scanned for 2.01 seconds (as shown also with the RSSI scan duration per channel, Figure 4.34), resulting in a total scanning time of approximately 42.21 seconds for all 21 channels. During this phase, the device's average current consumption is 36.8 mA, with a noticeable dip to 31.7 mA when the scan results are printed to the console. The average power during this phase is 0.184 W, resulting in a charge consumption of 1.55 C.

Following the *RSSI scanning*, the device enters the *Sending Beacons* phase, still initiated by the `dect beacon_start` command, where beacon messages are transmitted on the first available free channel. The device's current consumption exhibits peaks of up to 107.8 mA, and these peaks directly correspond to the transmission of beacon messages, as evidenced by the 2-second interval between peaks (matching the configured beacon interval of 2000 ms, as can be seen in Figure 4.34).

The entire phase lasts approximately 20 seconds, as set by the `dect sett --b_rx_rssi_scan_bg_interval 20` command, which determines the periodic background RSSI scanning interval. After the 20-second beacon transmission period, the device resumes scanning some or all transmission channels for a variable duration. In fact, as mentioned in Chapter 2, within a DECT NR+ network nodes continuously scan frequency channels to find the one with the least interference, even while sending beacon messages. During this scanning period, the device's average current consumption is approximately 51-52 mA, with peaks reaching 62-63 mA. Following the scan, the device resumes beacon transmission for another 20 seconds, and this cycle continues. Thus, after the `dect beacon_start` command is issued, considering the initial RSSI scan, the repeated beacon transmission phases and the variable channel scanning periods, the device demonstrates an overall performance with an average current consumption of approximately 49.1 mA and peak consumption around 107 mA.

Finally, the *Channel Scanning* phase is triggered by the `dect beacon_scan` command, during which the Sink device scans the specific channel used by the Relay for beacon transmissions in order to discover it and obtain information about it. This phase lasts approximately 4 seconds, with the device consuming an average current of 61.0 mA and a maximum of 64.2 mA. The average power during this phase is 0.305 W, resulting in a total charge consumption of 0.24 C.

Following the discovery of the relay, additional commands such as `dect beacon_relay -b relay_beacon` will be issued on the sink side to enable synchronization and background scanning towards the relay. However, during this phase, as well as during the reception of periodic messages sent by the client and forwarded by the relay to the sink, the device's performance remains unaffected. There are no recorded changes in current consumption compared to the one observed during the beacon transmission and channel scanning phases.

The average power dissipation is calculated using the formula $P = IV$, where the nRF9161 DK is powered by a USB connector providing 5V voltage [65]. Additionally, all specified parameter values can be found in the image below, including the Beacon interval (set to 2000 ms), RSSI scan duration per channel (set to 2010 ms), RSSI background scanning interval (set to 20 seconds) and the conditions required for a channel to be considered free, possible or busy based on its RSSI value.

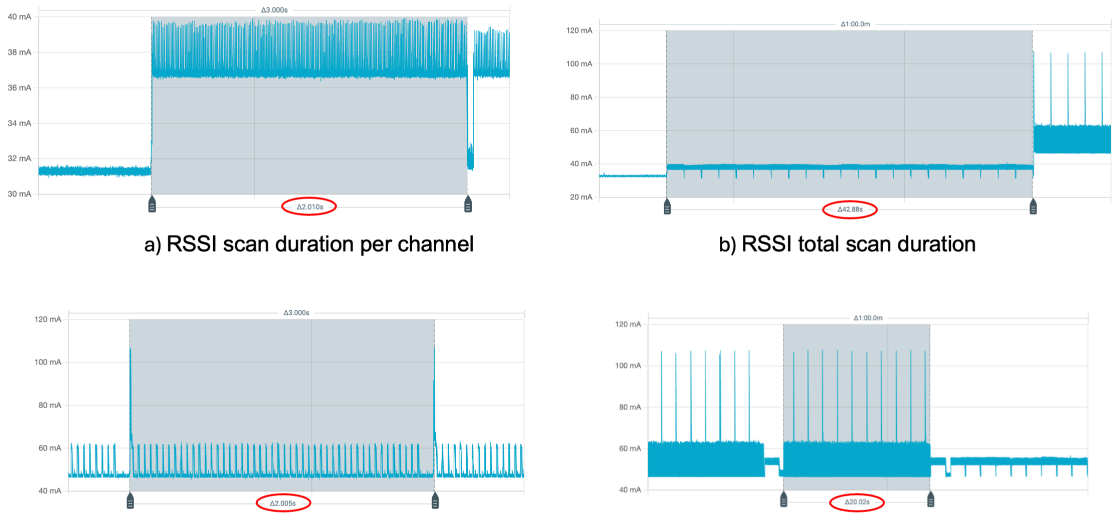
```
mosh:~$ > dect sett --read
dect sett --read
Common settings:
  network id..................................22
  transmitter id..............................40
  band number.................................1
Scheduler common settings:
  scheduling offset/delay (us)..................10000
Beacon settings:
  name........................................mosh_sink
  interval (msecs)............................2000
  is a sink...................................yes
  channel.....................................0
  ch access: RSSI scan (msecs) per ch...........2010
  ch access: RSSI scan busy (dBm):..............signal level > -72
  ch access: RSSI scan possible (dBm):..........-72 >= signal level > -85
  ch access: RSSI scan free (dBm):..............signal level <= -85
  ch access: RSSI background scanning interval...20 seconds
```

Figure 4.36: *Serial Terminal output showing the configured parameters for the sink device.*

4.4.3 Relay Device Power Consumption Analysis

Table 4.3: *Power Consumption of Relay Device in Various Operational Phases*

Action	Avg. Current (mA)	Max Current (mA)	Avg. Pwr (W)	Time (s)	Charge (C)
Idle	32.8	35.7	0.164	60.00	1.97
Channel Scanning	44.8	48.2	0.224	4.00	0.18
RSSI Scanning	36.9	40.4	0.185	42.21	1.56
Sending Beacons	49.2	107.6	0.246	20.00	0.98
Client Connected	49.7	181.3	0.249	20.00	0.99
Forwarding to Sink	54.1	181.1	0.271	20.00	1.08



a) RSSI scan duration per channel

b) RSSI total scan duration

c) Beacon interval

d) Total beacon transmission interval

Figure 4.37: *Visualisation of the current consumption results, validating the accuracy of manually set parameters in four different cases: (a) RSSI scan duration per channel, (b) RSSI total scan duration, (c) Beacon interval and (d) RSSI background scanning interval (total beacon transmission interval).*

The first four phases of the relay device's operation closely resemble those of the sink device, with one notable difference. During the *Channel Scanning* phase, the relay scans the channel used by the sink for beacon transmissions to discover and obtain information about the sink. This scanning occurs before beacon transmission (triggered by the `dect beacon_start` command) and, despite having the same 4-second duration, consumes less current, averaging 44.8 mA and peaking at 48.2 mA. In addition to these four phases, two new moments can be identified during the operation of the relay.

When the client connects to the relay, initiated by the `dect client_connect relay_beacon` command, there is a continuous exchange of keepalive messages between the client and the relay. During this phase, the average current consumption is 49.7 mA, only slightly higher than during the beacon transmission phase (49.2 mA). This indicates that the device's performance is minimally impacted as it now handles additional message exchanges. However, the current peaks reach up to 181.3 mA, corresponding to the relay sending keepalive messages to the client. The interval between these peaks is approximately 8 seconds, matching the keepalive interval set by the command `dect sett --c_keepalive_secs 8`. The duration of this phase is set to 20 seconds, aligned with the beacon transmission period.

In the final phase, the relay forwards periodic messages from the client to the sink. The average current consumption during this phase increases to 54.1 mA, with peaks reaching 182.1 mA. These peaks occur both when sending keepalive messages and when forwarding messages to the sink.

It is noteworthy that the power required for transmitting different types of messages varies significantly. Typically, sending a beacon message results in a current peak of around 107-108 mA, whereas sending a standard message can cause peaks of 181-182 mA. This discrepancy can be attributed to the different structures and byte lengths of these messages. Beacon messages are generally used for synchronization and broadcasting network information, present at the PHY (Layer 1) and MAC (Layer 2) layers, and do not carry user data (SDU = 0), unlike other types of messages that do.

4.4.4 Client Device Power Consumption Analysis

The power consumption results for the client device align closely with those observed for the other two devices, the sink and the relay, as presented in Table 4.4:

Table 4.4: *Power Consumption of Client Device in Various Operational Phases*

Action	Avg. Current (mA)	Max Current (mA)	Avg. Pwr (W)	Time (s)	Charge (C)
Idle	32.8	35.4	0.164	60.00	1.97
Channel Scanning	44.8	48.3	0.224	4.00	0.18
Client Connected	48.6	179.0	0.243	60.00	2.92
Sending Messages	49.5	181.8	0.248	60.00	2.97

During the *Channel Scanning* phase, initiated by the `dect beacon_scan` command, the power consumption is almost identical to that of the relay, averaging 44.8 mA with a peak of 48.3 mA.

In the phase where the client connects to the relay, the current peaks at approximately 179 mA every 8 seconds, corresponding to the keepalive messages. This pattern is again very similar to what was observed with the relay device. Additionally, during the message-sending phase, the current peaks at around 182 mA, aligning with the power consumption characteristics noted for message transmissions destined for the sink.

4.4.5 Comparative Analysis

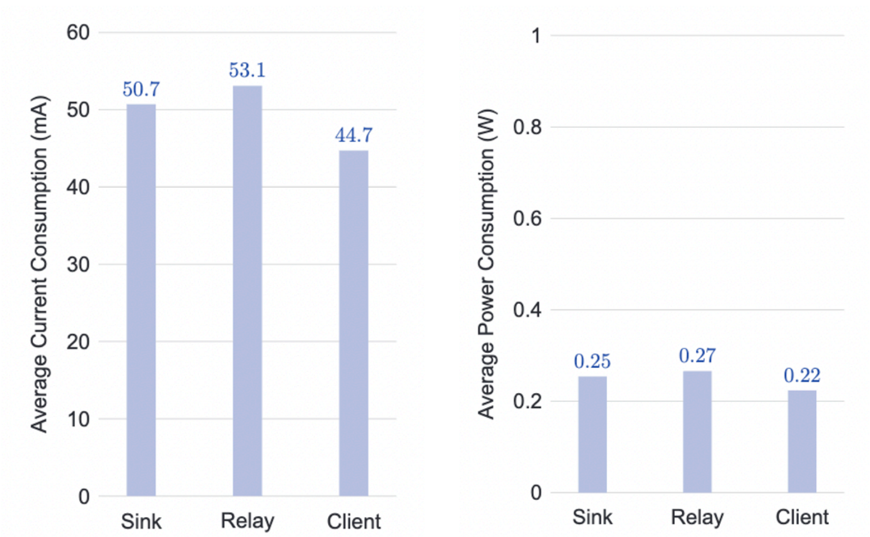


Figure 4.38: *Average current (in mA) and Power (in W) consumption of the Sink, Relay and Client devices during the entire duration of the experiment.*

In this section, we present a comparative analysis of the average current and power consumption of the three devices (Sink, Relay and Client) throughout the experiment. The experiment lasted for 10 minutes during which all the commands analyzed earlier were executed and the devices operated as described.

Figure 4.38 includes two bar graphs. The first graph illustrates the average current consumption (in mA) of the Sink, Relay and Client devices. The Sink consumed 50.7 mA, the Relay 53.1 mA and the Client 44.7 mA. The second graph depicts the average power consumption (in W) of the three devices, with the Sink consuming 0.25 W, the Relay 0.27 W and the Client 0.22 W.

From these graphs, it is evident that the Relay device consumes more power compared to the Sink and Client devices. This observation aligns with our expectations due to the Relay's role in continuously forwarding messages, as discussed in Chapter 2 (Section 2.3 - IoT and Wearables). The higher power consumption of the Relay is justified by its critical function in the network. The Relay not only transmits its own beacons but also forwards messages between the Client and Sink, requiring more energy to maintain these additional communications. This operational overhead results in increased average power consumption, making the Relay the most energy-intensive device among the three.

Power Consumption Comparison with Other Technologies

In order to have a better understanding of the obtained results, let's now compare the power consumption of the DECT NR+ network with other prevalent technologies such as Wi-Fi (IEEE 802.11), NB-IoT and LTE-M.

Wi-Fi routers, on average, consume between 5 and 20 watts of electricity, depending on the model and usage patterns [66]. Several studies have corroborated these findings. For instance, in [67] it was analyzed the average power consumption of three different access points (APs) in a private network setup where one station functioned as a server and another as a client. The power consumption recorded for the three APs was 9W, 6W and 6W respectively. In [68] it was done the same for a specific AP across different Wi-Fi standards (b/a/g/n/ac), to determine the most energy-efficient version. The results ranged between 3.5W and 4.3W during active usage and about 2.3W to 2.4W during idle periods.

Although these studies present varied results, it is evident that the average power consumption in Wi-Fi technology is significantly higher compared to the one in DECT NR+ network. During the experiments, the DECT NR+ devices demonstrated an idle power consumption of approximately 0.165W and an average power consumption of around 0.2-0.3W during active operation. This substantial difference highlights the energy efficiency of DECT NR+ technology, which can be attributed to various power-saving techniques employed by the standard (as explained in Section 2.5.3) such as routing and retransmitting packets directly into the radio stack level, and autonomous dynamic power control of transmissions. It is also fair to underline that the power-saving feature which involves relay devices sending beacon messages to manage radio resources and inform leaf nodes when to listen or not, is not available because it belongs to the MAC layer, which has not yet been implemented. Consequently, in the current application, beacons are used solely for device discovery and synchronization, without additional information such as those related to sleep mode.

While Wi-Fi is not designed with stringent power-saving features, NB-IoT and LTE-M are explicitly developed for power-efficient communication, meeting the needs of IoT requirements. These technologies are tailored for the so called ”constrained devices”, that rely on battery power without a permanent power source. Since these sensors usually only need to transmit small measurement of data such as temperatures or coordinates in intervals rather that delivering a constant data stream, they should be able to live for as long as possible on one battery charge. And the reason is really simple: it is time-consuming and expensive to replace the batteries on a regular basis after only a short runtime.

NB-IoT and LTE-M leverage Power Saving Mode (PSM) and extended Discontinuous Reception (eDRX) to achieve low power consumption. PSM puts the device into a deep sleep state to conserve energy, while eDRX allows developers to define the intervals at which the device switches from idle to receive mode, ensuring it remains reachable from the outside [69].

In [70], a deeper look in the power consumption of devices connected to NB-IoT and LTE-M networks is presented. The study analyzed three different scenarios: network authentication and actual data transfer using HTTP and MQTT application protocols. The power consumption during idle phases was approximately 0.16-0.18W, aligning closely with our DECT NR+ results. During network authentication, the power consumption was 0.33W for NB-IoT and 0.39W for LTE-M; for data transfer phases, instead, power consumption ranged between 0.4W and 0.5W, with a peak of 0.69W for data download using HTTP over LTE-M. Obviously, these findings confirm that also NB-IoT and LTE-M consume significantly less power than traditional cellular networks.

The power consumption of DECT NR+ settles therefore in the same range as NB-IoT and LTE-M, emerging consequentially as a highly efficient alternative, suitable for applications necessitating long battery life and continuous operation such as industrial monitoring, smart home devices and other IoT applications. The ability to maintain low power consumption without compromising performance underscores the effectiveness of DECT NR+ in various deployment scenarios.

It is noteworthy that DECT NR+ does not use any features comparable to PSM (Power Saving Mode) or eDRX (extended Discontinuous Reception). According to Nordic Semiconductor, the modem firmware will be improved in the future to enter a power-saving state if the scheduled operations have sufficiently long gaps. Therefore, while the current power consumption performance is already impressive, it is expected to improve further.

However, it is essential to consider that once the full DECT NR+ stack, including higher layers like the MAC, DLC and CVG, will be fully implemented, there may be an increase in power consumption due to additional overhead. The implementation of these higher layers introduces more complex protocols and control mechanisms that require additional processing power. Despite this, the physical layer is already highly optimized, and future power-saving features will likely balance out the additional consumption.

Future work could focus on exploring the full potential of the standard once the sleep mode and other power-saving features are implemented. It will be interesting to study in detail the power consumption of a DECT NR+ network when the entire stack is fully available, providing a comprehensive understanding of its efficiency and performance in various real-world scenarios.

An additional comparison of current consumption during message transmission further highlights the differences between DECT NR+ and these other two technologies. Using the Online Power Profiler by Nordic Semiconductor [71], it was estimated that uploading a 10B message in an LTE-M network consumes approximately 40.8mA over 27.2ms, and 35.3mA over 956.1ms in an NB-IoT network. These measurements were taken with a TX power of 10dBm. The substantial time difference is due to the differing data rates of the technologies: LTE-M supports up to 300/375 kbps, while NB-IoT supports up to 30/60 kbps [72]. With a TX power of 23dBm, LTE-M consumes 67.9mA, whereas NB-IoT remains at 35.3mA . In comparison, DECT NR+ reaches peaks of 180-181mA during message transmission with the same data length and a TX power of 13dBm. This higher current consumption aligns with significantly shorter transmission times, supported by data rates reaching up to 2.3Mbps, as demonstrated in previous sections.

Overall, the comparative analysis clearly demonstrates that DECT NR+ offers an alternative energy-efficient solution compared to NB-IoT and LTE-M and a way more optimal option than Wi-Fi, making it a compelling choice for modern network applications that prioritize power efficiency and sustainability.

5 Conclusion and Future Work

This thesis has explored the capabilities and performance of Device-to-Device (D2D) communication within the context of 5G and beyond, focusing on the DECT NR+ standard. Developed by the European Telecommunications Standards Institute (ETSI), DECT NR+ is designed to meet the stringent requirements of 5G technology. Through a series of detailed experiments and analyses, this study provides valuable insights into the practical applications and limitations of DECT NR+ in real-world scenarios.

5.1 Key Findings

1. **Distance and Coverage**: The performance analysis of DECT NR+ in both indoor and outdoor environments revealed a higher efficiency indoors, that can be attributed to factors as controlled conditions, multipath propagation and signal directionality. The Packet Success Rate (PSR) was excellent up to a critical distance, beyond which it sharply declined. This drop can also be likely related to packet length, suggesting that longer packets increase the likelihood of bit errors, thereby reducing PSR. The maximum observed range was 220 meters, which is impressive but still less compared to other IoT technologies like LTE-M or NB-IoT. Nordic Semiconductor's acknowledgment of issues with their nRF9161 Development Kit results in a range significantly less than the one expected (around 600m), but also suggests that there is room for improvement in radio sensitivity, which could potentially extend the effective range of DECT NR+.
2. **Data Rate Performance**: Higher Modulation and Coding Scheme (MCS) values were shown to improve data rates significantly. The study found no degradation in data reception accuracy with increased MCS values, indicating reliable data transmission. The highest data rate observed was 2.2 Mbps, short of the theoretical 3.4 Mbps supported by DECT NR+. Factors such as hardware limitations and environmental conditions likely contributed to this discrepancy. Despite this, DECT NR+ still offers robust performance, making it a competitive choice for various IoT applications.
3. **Latency Performance**: The analysis of latency revealed a linear relationship between Round Trip Time (RTT) and packet size, with higher MCS values leading to better latency performance. However, the observed RTTs exceeded the stringent requirements of Ultra-Reliable Low Latency Communication (URLLC), particularly at lower MCS values. This suggests that while DECT NR+ is reliable, meeting URLLC standards may require further optimization. Investigating system parameters, environmental impacts and dynamic MCS adaptation could help reduce latency and ensure compliance with URLLC requirements.
4. **Power Consumption Performance**: The analysis indicated that the Relay device consumes more power than the Sink and Client devices due to its role in continuously forwarding messages. When compared with other technologies like Wi-Fi, DECT NR+ demonstrated significantly lower power consumption, especially during idle and active operation. The similarity with NB-IoT and LTE-M showcases DECT NR+ as a highly efficient alternative for applications requiring long battery life and continuous operation. Future implementations of the complete DECT NR+ stack and additional power-saving features are expected to enhance this efficiency further, making it an even more attractive option for various IoT applications.

5.2 Future Work

Building on the findings of this thesis, several areas for future research and development are suggested to enhance DECT NR+ communication technologies:

1. **Packet Length Analysis**: One possible reason for performance drops at specific distances could be related to the length of the packets being transmitted. In this experiment, each packet contained 32 bits (4 bytes), represented by a single *uint32_t* variable. For future research, it would be interesting to analyze how performance varies with different packet lengths. As the amount of data (and thus the number of bits) transmitted increases, the likelihood of encountering a bit error also increases, potentially degrading the Packet Success Rate (PSR). Additionally, exploring how different packet lengths impact transmission efficiency and reliability could provide deeper insights into optimizing DECT NR+ for various applications.
2. **Maximum Indoor Coverage Exploration**: Another area worth investigating is the maximum indoor coverage supported by the DECT NR+ standard. If an indoor environment can be found that meets the requirements of the experiment conducted, testing the coverage beyond 100 meters would be valuable. Alternatively, if direct testing is not feasible, estimating the maximum coverage using more available data could provide insights into the potential range of DECT NR+. This would help in a better understanding of the limits and capabilities of the technology.
3. **Dynamic MCS Adaptation**: Developing algorithms for dynamic MCS adaptation based on real-time environmental conditions and communication quality could optimize performance. This would allow the system to balance data rate and reliability more effectively, while maintaining an acceptable error rate and communication range.
4. **Extended Range Testing**: Conducting more extensive range testing in diverse real-world scenarios, including urban, suburban, rural environments and areas with significant physical obstructions, would provide a more comprehensive understanding of DECT NR+'s capabilities and limitations. This could lead to more robust deployment strategies and improved performance in different settings.
5. **Compliance with URLLC Requirements**: To meet the stringent URLLC requirements, further optimization of system parameters and exploration of advanced algorithms for latency reduction are necessary. Ensuring DECT NR+ can maintain low latency under various scenarios will be crucial for its application in ultra-reliable and low-latency communication environments.
6. **Implementation of Power-Saving Features**: DECT NR+ currently lacks power-saving features like PSM (Power Saving Mode) and eDRX (extended Discontinuous Reception). Nordic Semiconductor plans to enhance the modem firmware to enable power-saving states during idle periods. Although the current power consumption is already low, these improvements are expected to further enhance efficiency. However, the full implementation of the DECT NR+ stack may increase power consumption due to additional overhead. Future research should investigate the detailed power consumption of a fully implemented DECT NR+ network, focusing on the impact of the complete stack on energy efficiency and performance in various scenarios.

By addressing these areas, future research can significantly enhance the performance, reliability, and applicability of DECT NR+ communication technologies, ensuring they remain competitive in the evolving landscape of 5G and beyond.

Bibliography

[1] M. A. Rossi. “The advent of 5G and the non-discrimination principle”. In: *Telecommunications Policy* 46 (2022), p. 102279. DOI: 10.1016/j.telpol.2021.102279.

[2] EMnify. *What Is 5G? Everything You Need to Know*. (accessed: 10.06.2024). 2024. URL: https://www.emnify.com/iot-glossary/5g.

[3] Celona. *5G Bands Explained: How They Work & Why They Matter*. (accessed: 10.06.2024). 2024. URL: https://www.celona.io/5g-lan/5g-bands.

[4] Arrow Electronics. *5G frequency bands: 3 primary 5G bands and when to use them*. (accessed: 10.06.2024). 2024. URL: https://www.arrow.com/en/research-and-events/articles/5g-frequency-bands-explained.

[5] Celona. *4G versus 5G – A paradigm shift in mobile communications?* (accessed: 10.06.2024). 2024. URL: https://www.emnify.com/blog/4g-versus-5g-connectivity.

[6] F. Rinaldi, A. Raschella, and S. Pizzi. “5G NR system design: a concise survey of key features and capabilities”. In: *Wireless Networks* 27 (2021), pp. 5173–5188.

[7] O. Muhammad et al. “A Comprehensive Review of D2D Communication in 5G and B5G Networks”. In: *LC International Journal of STEM* 4.3 (Sept. 2023), pp. 1–23.

[8] D. P. Hadyanto, D. S. Novariana, and A. Wulandari. “Device-to-Device Communication in 5G”. In: *Advances in Engineering Research*. Vol. 209. 2021, pp. 651–654.

[9] X. Lin et al. “Modeling, Analysis and Optimization of Multicast Device-to-Device Transmissions”. In: *submitted to IEEE Trans. Wireless Commun.* (2013). arXiv: arXiv:1309.1518 [cs.NI].

[10] X. Lin et al. “An Overview of 3GPP Device-to-Device Proximity Services”. In: *IEEE Communications Magazine* 52.4 (2014), pp. 40–48.

[11] X. Lin, J. G. Andrews, and A. Ghosh. “A comprehensive framework for device-to-device communications in cellular networks”. In: *arXiv preprint arXiv:1305.4219* (2013).

[12] U. N. Kar and D. K. Sanyal. “A Critical Review of 3GPP Standardization of Device-to-Device Communication in Cellular Networks”. In: *SN Computer Science* 1.37 (2020), pp. 1–18. DOI: 10.1007/s42979-019-0045-5.

[13] R. Yin et al. “Decentralized Radio Resource Adaptation in D2D-U Networks”. In: *IEEE Internet of Things Journal* 7.18 (2020), pp. 1–14. DOI: 10.1109/JIOT.2020.3016019.

[14] T. Doumi et al. “LTE for public safety networks”. In: *IEEE Commun Mag* 51.2 (2013), pp. 106–112.

[15] M. Höyhtyä, O. Apilo, and M. Lasanen. “Review of Latest Advances in 3GPP Standardization: D2D Communication in 5G Systems and Its Energy Consumption Models”. In: *Future Internet* 10.1 (2018), p. 3.

[16] S. Y. Lien et al. “3GPP device-to-device communications for beyond 4G cellular networks”. In: *IEEE Commun Mag* 54.3 (2016), pp. 29–35.

[17] PhD Direction. *Device-to-Device Communication in 5G Matlab*. (accessed: 10.06.2024). 2024. URL: https://www.phddirection.com/device-to-device-communication-in-5g-matlab/.

[18] MathWorks. *Device-to-Device (D2D) Communication for Public Safety in LTE*. (accessed: 10.06.2024). 2016. URL: https://www.mathworks.com/content/dam/mathworks/tag-team/Objects/d/90538_95058v00_LTE_Application_Note.pdf.

[19] ETSI 3GPP. *Universal mobile telecommunications system (UMTS); LTE; proximity-based services (ProSe); (Release 13) Stage 2*. Tech. rep. TS 23.303 V 13.3.0. 3GPP, 2016.

[20] L. Oscar et al. *D6.2.2 standardization and communication activity report (release b) WP 6-Dissemination, standardization, and exploitation*. Tech. rep. (accessed: 15.06.2024). FP7, 2015. URL: http://www.fp7moto.eu/wp-content/uploads/2015/12/D6.2.2..pdf.

[21] K. J. Zou et al. "Proximity discovery for device-to-device communications over a cellular network". In: *IEEE Commun Mag* 52.6 (2014), pp. 98–107.

[22] 3GPP. *Overview of 3GPP Release 12 V0.2.0 (2015-09)*. Tech. rep. 3GPP, 2015.

[23] 3rd Generation Partnership Project (3GPP). *RP-170295 Study on Further Enhancements to LTE Device to Device, UE to Network Relays for IoT and Wearables*. Report. 3GPP, 2017. URL: http://www.3gpp.org/ftp/tsg_ran/TSG_RAN/TSGR_75/Docs/RP-170295.zip (visited on 06/14/2024).

[24] ETSI 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception; (Release 14), TS 36.102 V14.5.0*. Tech. rep. 2017.

[25] Alexander Pyattaev et al. "3GPP LTE Traffic Offloading onto WiFi Direct". In: *Proceedings of the Wireless Communications and Networking Conference Workshops (WCNCW), 2013 IEEE*. IEEE. 2013, pp. 135–140.

[26] Shahid Mumtaz and Jonathan Rodriguez. *Smart Device to Smart Device Communication*. Berlin: Springer, 2014.

[27] Fiza Jameel et al. "A Survey of Device-to-Device Communications: Research Issues and Challenges". In: *IEEE Communications Surveys & Tutorials* 20.3 (2018), pp. 2133–2168.

[28] Shahid Mumtaz, Kazi Mohammad Shahidul Huq, and Jonathan Rodriguez. "Direct Mobile-to-Mobile Communication: Paradigm for 5G". In: *IEEE Wireless Communications* 21.5 (2014), pp. 14–23.

[29] Preeti Gandotra and Rakesh Kumar Jha. "Device-to-Device Communication in Cellular Networks: A Survey". In: *Journal of Network and Computer Applications* 71 (2016), pp. 99–117.

[30] Jaganathan Logeshwaran, Nallasamy Shanmugasundaram, and Jaime Lloret. "Energy-efficient resource allocation model for device-to-device communication in 5G wireless personal area networks". In: *International Journal of Communication Systems* 36.13 (2023). R E S E A R CH AR T I C L E, e5524. DOI: 10.1002/dac.5524. URL: https://onlinelibrary.wiley.com/doi/10.1002/dac.5524.

[31] Michael Haus et al. "Security and Privacy in Device-to-Device (D2D) Communication: A Review". In: *IEEE Communications Surveys & Tutorials* 19.2 (2017), pp. 1054–1079.

[32] Dominique Bonte, Michael Larner, and Adhish Luitel. *DECT-2020 NR: A New 5G IoT Standard*. Tech. rep. ABI Research, 2021.

[33] Nordic Semiconductor ASA. *What Is DECT-2020 New Radio (NR) and How Big a Deal Is It?* (accessed: 24.06.2024). 2020. URL: https://blog.nordicsemi.com/getconnected/what-is-dect-2020-new-radio-nr-and-how-big-a-deal-is-it.

[34] ETSI. *DECT-2020 New Radio (NR); Part 1: Overview; Release 1*. Tech. rep. TS 103 636-1 V1.1.1. European Telecommunications Standards Institute, 2024. URL: https://www.etsi.org/deliver/etsi_ts/103600_103699/10363601/01.01.01_60/ts_10363601v010101p.pdf.

[35] M. D. Pérez-Guirao, T. Weisshaupt, and A. Wilzeck. *DECT NR+: Unveiling the Essentials of a New Non-Cellular 5G Standard for Verticals*. Tech. rep. Ostfalia HAW, Wirepas Oy, Sennheiser electronic GmbH & Co. KG, 2022.
[36] 5G Technology World. *The First Non-Cellular 5G Standard: DECT NR*. (accessed: 23.06.2024). 2024. URL: https://www.5gtechnologyworld.com/the-first-non-cellular-5g-standard-dect-nr/.
[37] ETSI. *DECT*. (accessed: 23.06.2024). 2024. URL: https://www.etsi.org/technologies/dect.
[38] Samson Technologies. *Exploring the Benefits of the 1.9 GHz Band*. (accessed: 23.06.2024). 2024. URL: https://samsontech.com/blog/exploring-the-benefits-of-the-19-ghz-band/.
[39] Nordic Semiconductor. *DECT NR: A Technical Dive into Non-Cellular 5G*. (accessed: 24.06.2024). 2024. URL: https://devzone.nordicsemi.com/nordic/nordic-blog/b/blog/posts/dect-nr-a-technical-dive-into-non-cellular-5g#mcetoc_1h19479t81.
[40] Nordic Semiconductor. *Introduction to DECT NR+*. Nordic Tech Webinar, 15 June 2022. 2022. URL: https://devzone.nordicsemi.com/cfs-file/__key/communityserver-discussions-components-files/4/Introduction-to-DECT-NR_2B00_.pdf.
[41] Nordic Semiconductor. *DECT NR*. (accessed: 10.06.2024). 2024. URL: https://www.nordicsemi.com/Products/Wireless/DECT-NR?.
[42] ETSI. *DECT-2020 New Radio (NR); Part 3: Physical layer; Release 1*. Technical Specification ETSI TS 103 636-3 V1.3.1 (2021-12). Sophia Antipolis Cedex, France: ETSI, 2021. URL: https://www.etsi.org/standards-search.
[43] ETSI. *DECT-2020 New Radio (NR): Part 4 - MAC Layer*. Version V1.4.1. Technical Specification. 2023.
[44] Nordic Semiconductor. *Understanding DECT NR+: The first non-cellular 5G standard*. Nordic Tech Webinar, 3 May 2023. 2023. URL: https://devzone.nordicsemi.com/cfs-file/__key/communityserver-discussions-components-files/4/Understanding-DECT-NR_2B00_-webinar.pdf.
[45] ETSI. *DECT-2020 New Radio (NR): Part 5 - DLC and Convergence Layers*. Version V1.4.1. Technical Specification. 2023.
[46] DECT. *Introduction to the standard*. Webinar. 2023. URL: https://www.dect.org/userfiles/Public/DF_DECT%20NRplus.pdf.
[47] Nordic Semiconductor. *nRF9161 DK User Guide*. (accessed: 09.07.2024). 2024. URL: https://docs.nordicsemi.com/bundle/ug_nrf9161_dk/page/UG/nrf91_DK/intro.html.
[48] Nordic Semiconductor. *nRF9161 Product Information*. (accessed: 09.07.2024). 2024. URL: https://www.nordicsemi.com/Products/nRF9161.
[49] Nordic Semiconductor. *nRF9161 Documentation - Block Diagram*. (accessed: 09.07.2024). 2024. URL: https://docs.nordicsemi.com/bundle/ps_nrf9161/page/blockdiagram.html.
[50] Nordic Semiconductor. *DECT Page for nRF9161*. (accessed: 09.07.2024). 2024. URL: https://docs.nordicsemi.com/bundle/ps_nrf9161/page/dect.html.
[51] Nordic Semiconductor. *PPK User Guide Introduction*. (accessed: 09.07.2024). 2024. URL: https://docs.nordicsemi.com/bundle/ug_ppk2/page/UG/ppk/PPK_user_guide_Intro.html.
[52] Nordic Semiconductor ASA. *PPK User Guide - Quick Start*. (accessed: 24.06.2024). 2024. URL: https://docs.nordicsemi.com/bundle/ug_ppk2/page/UG/ppk/PPK_user_guide_Quick_start.html.
[53] Nordic Semiconductor. *nRF Connect SDK*. (accessed: 09.07.2024). 2024. URL: https://www.nordicsemi.com/Products/Development-software/nRF-Connect-

SDK?utm_term=sdk%20for%20matter&utm_campaign=M_ALL_GE_US&utm_source=adwords&utm_medium=ppc&hsa_tgt=kwd-1944774636920&hsa_grp=144720084600&hsa_src=g&hsa_net=adwords&hsa_mt=b&hsa_ver=3&hsa_ad=680036204613&hsa_acc=1116845495&hsa_kw=sdk%20for%20matter&hsa_cam=18545141408&gad_source=1&gclid=Cj0KCQjwv7O0BhDwARIsAC0sjWMf7RJnaci5kgd6IAiPu9zzcJtV
62p7d_cXVPHL_JxJxJ2caAhb1EALw_wcB.

[54] Nordic Semiconductor. *Lesson 1 - nRF91 Series SiP*. (accessed: 09.07.2024). 2024. URL: https://academy.nordicsemi.com/courses/cellular-iot-fundamentals/lessons/lesson-1-cellular-fundamentals/topic/lesson-1-nrf91-series-sip/.

[55] Nordic Semiconductor. *nRF Connect for Visual Studio Code*. (accessed: 10.07.2024). 2024. URL: https://docs.nordicsemi.com/bundle/nrf-connect-vscode/page/index.html.

[56] Nordic Semiconductor. *nRF Connect Serial Terminal*. (accessed: 09.07.2024). 2024. URL: https://docs.nordicsemi.com/bundle/nrf-connect-serial-terminal/page/index.html.

[57] Nordic Semiconductor. *nRF Modem: DECT PHY*. (accessed: 09.07.2024). 2024. URL: https://docs.nordicsemi.com/bundle/ncs-latest/page/nrfxlib/nrf_modem/doc/dectphy.html.

[58] Nordic Semiconductor. *Logger Module*. (accessed: 09.07.2024). 2024. URL: https://academy.nordicsemi.com/courses/nrf-connect-sdk-fundamentals/lessons/lesson-3-printing-messages-to-console-and-logging/topic/logger-module/.

[59] Google Maps. *Nørgaards Have*. (accessed: 13.07.2024). 2024. URL: https://maps.app.goo.gl/4R4DYo4niZcxNYcH7.

[60] Google Maps. *Nørgaards Have*. (accessed: 13.07.2024). 2024. URL: https://maps.app.goo.gl/wrem4JzeDNrfJPCy5.

[61] Life Wire. *What Is the Range of a Typical Wi-Fi Network?* (accessed: 14.07.2024). 2024. URL: https://www.lifewire.com/range-of-typical-wifi-network-816564#:~:text=A%20general%20rule%20of%20thumb,one%2Dthird%20of%20these%20distances..

[62] AWS. *LTE-M*. (accessed: 14.07.2024). 2024. URL: https://docs.aws.amazon.com/whitepapers/latest/implementing-lpwan-solutions-with-aws/lte-m.html#:~:text=NB%2DIoT%20devices.-,Range%20and%20coverage,10%20km%20in%20rural%20areas..

[63] AWS. *NB-IoT*. (accessed: 14.07.2024). 2024. URL: https://docs.aws.amazon.com/whitepapers/latest/implementing-lpwan-solutions-with-aws/nb-iot.html#:~:text=or%20cell%20tower.-,Range%20and%20coverage,NB%2DIoT%20supports%20indoor%20coverage..

[64] Nordic Semiconductor. *Differences between DECT and DECT NR+*. (accessed: 27.07.2024). 2024. URL: https://docs.nordicsemi.com/bundle/ncs-latest/page/nrf/protocols/dect/index.html#differences_between_dect_and_dect_nr.

[65] Nordic Semiconductor. *Power supply*. (accessed: 27.07.2024). 2024. URL: https://docs.nordicsemi.com/bundle/ug_nrf9161_dk/page/UG/nrf91_DK/hw_description/nrf9161_power_supply.html.

[66] energysage. *How many watts does a wifi router use?* (accessed: 28.07.2024). 2023. URL: https://www.energysage.com/electricity/house-watts/how-many-watts-does-a-wi-fi-router-use/.

[67] Anthony Murabito. *A Comparison of Efficiency, Throughput, and Energy Requirements of Wireless Access Points*. Tech. rep. (accessed: 28.07.2024). University of New Hampshire InterOperability Laboratory, 2009.

[68] Gunnar P. Noordbruis. *Energy efficient WLAN using WiFi standards b/a/g/n/ac on an Archer C7 AC1750 access point*. Bachelor thesis. First supervisor/assessor: dr. B.E.

van Gastel. Second assessor: prof. dr. M.C.J.D. van Eekelen. Computing Science: Radboud University, June 2020.
[69] 1NCE. *PSM and eDRX*. (accessed: 28.07.2024). 2024. URL: https://1nce.com/en-us/resources/news/blog/psm-and-edrx.
[70] Viktoria Vomhoff et al. “NB-IoT vs. LTE-M: Measurement Study of the Energy Consumption of LPWAN Technologies”. In: *Second International Workshop on Green and Sustainable Networking (GreenNet), IEEE ICC 2023*. IEEE. IEEE, 2023, pp. 403–408. DOI: 10.1109/ICCWKSHO57.2023.10283595.
[71] Nordic Semiconductor. *Online Power Profiler for LTE*. (accessed: 28.07.2024). 2024. URL: https://devzone.nordicsemi.com/power/w/opp/3/online-power-profiler-for-lte.
[72] Nordic Semiconductor. *LTE-M and NB-IoT*. (accessed: 28.07.2024). 2024. URL: https://academy.nordicsemi.com/courses/cellular-iot-fundamentals/lessons/lesson-1-cellular-fundamentals/topic/lesson-1-lte-m-and-nb-iot/.

Politecnico
di Torino

Corso Duca degli Abruzzi, 24
10129 Torino TO
Tlf. 011 090102

https://www.dauin.polito.it/