Theo Gasteiger, BSc

Improving the Performance of Bluetooth Low Energy for Motorcycle Intercom Systems

Master's Thesis
to achieve the university degree of
Master of Science
Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor
Assoc.Prof. Dr. Carlo Alberto Boano

Advisor
Dr.techn. Michael Spörk

Institute for Technical Informatics

Graz, October 2022

Theo Gasteiger, BSc

Optimierung von Bluetooth Low Energy für Motorrad Kommunikationsgeräte

Masterarbeit
zur Erlangung des akademischen Grades eines
Diplom-Ingenieur
Masterstudium: Computer Science

eingereicht an der

Technische Universität Graz

Betreuer
Assoc.Prof. Dr. Carlo Alberto Boano
Dr.techn. Michael Spörk

Institut für Technische Informatik

Graz, Oktober 2022

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

Datum   Unterschrift

Abstract

Although motorcycle intercom systems have become a staple in motorcycle touring as well as leisure riding, problems like unreliability or unsatisfactory audio quality still exist after decades of development. Given that most motorcycle intercom systems require prolonged operation on a constrained energy budget, manufacturers had to choose between reliability and audio quality or battery life, which resulted in poor performance and disappointing user experience. Bluetooth Low Energy, especially Bluetooth LE Audio and broadcast isochronous streams promise to be a viable option to tackle these challenges, as they offer high quality, reliable audio transmission to multiple receiver devices simultaneously.

This thesis presents an experimental campaign determining the most influential parameters on broadcast isochronous streams in terms of throughput, link quality, power consumption, as well as latency. Furthermore, a dynamic parameter adaptation algorithm is presented, which adjusts two isochronous parameters dynamically in order to achieve an optimal performance while minimizing the current consumption. Finally, an evaluation is conducted to highlight the improvements of such an algorithm, especially in comparison to naïve algorithms or a static parameter setting. This evaluation shows that a dynamic parameter adaptation algorithm can outperform all other approaches in respect to throughput, link quality as well as transmitter current consumption by adjusting the transmission power and retransmission number dynamically. Such dynamic adaptation algorithm can decrease the transmitter current consumption by up to 49%, at the cost of approximately 30% increase of the receiver current consumption.

Kurzfassung

Bereits seit Jahrzehnten sind Motorrad Kommunikationsgeräte im Bereich von Motorradreisen oder Gruppenfahrten nicht mehr wegzudenken. Allerdings blieben bekannte Probleme wie eine mangelnde Zuverlässigkeit oder unzufriedenstellende Sprachqualität weitgehend ungelöst. Weiters mussten sich die Hersteller oftmals entscheiden, den Fokus auf Zuverlässigkeit und Tonqualität oder Akku Laufzeit zu legen. Bluetooth Low Energy, im speziellen Bluetooth LE Audio mit broadcast isochronous streams, stellt eine mögliche Lösung für diese Probleme dar.

Diese Masterarbeit präsentiert eine experimentelle Vermessung der einflussreichsten Parameter auf broadcast isochronous streams in Bezug auf Durchsatz, Verbindungsqualität, Stromverbrauch sowie Latenz. Weiters wird ein dynamischer Parameter Adaptierungsalgorithmus vorgestellt, welcher die optimalen isochronen Parameter dynamisch auswählt, um die höchste Performanz mit minimalem Stromverbrauch zu erzielen. Schlussendlich wird eine Evaluierung durchgeführt, welche die Überlegenheit dieses dynamischen Algorithmus im Verglich zu naiven Adaptierungsalgorithmen, in Bezug auf Durchsatz, Verbindungsqualität und Sender-Stromverbrauch verdeutlicht. Diese Evaluierung zeigt, dass ein dynamischer Algorithmus den Sender-Stromverbrauch um bis zu 49% senken kann, unter der Erhöhung des Empfänger-Stromverbrauchs um 30%.

Acknowledgments

This master thesis was written during the year 2022 at the Institute for Technical Informatics at Graz University of Technology.

First and foremost, I want to thank my supervisors Carlo Alberto Boano and Michael Spörk for their excellent support throughout the course of this thesis. Their constant feedback has contributed significantly to the realization of this work and I am grateful that they always took time out of their busy schedule to support my research. Moreover, I want to thank Markus Schuß for his help during this work.

I also want to thank my family, especially my parents for their backing, education and support. Finally, I want to thank my partner Sandra for her constant motivation and understanding throughout my studies. Without her I would certainly not be at this point in my life.

Danksagung

Diese Diplomarbeit wurde im Jahr 2022 am Institut für Technische Informatik an der Technischen Universität Graz durchgeführt.

Zuallererst möchte ich mich bei meinen Betreuern Carlo Alberto Boano und Michael Spörk für deren hervorragende Unterstützung während der Durchführung dieser Arbeit bedanken. Ihr hilfreiches Feedback hat wesentlich zur Umsetzung beigetragen und ich bin beiden ausgesprochen dankbar, dass sie sich dafür regelmäßig Zeit genommen haben. Weiters möchte ich mich bei Markus Schuß für seine Hilfe und Unterstützung bedanken.

Ich möchte mich außerdem bei meiner Familie, insbesondere meinen Eltern für deren Unterstützung, Freiheit in der Wahl meiner Ausbildung und Rückhalt bedanken. Abschließend möchte ich mich bei meiner Partnerin Sandra für ihre Motivation und Verständnis bedanken. Ohne ihre Unterstützung wäre ich heute nicht an diesem Punkt in meinem Leben.

Contents

List of Figures

List of Tables

1 Introduction

The Internet of Things (IoT), a network of everyday objects connected over different communication mechanisms, has received a lot of attention over the past years, not only in the field of home automation (which has become a staple in the realm of IoT), but also in areas such as health care, fitness, as well as industrial applications. Consequently, a lot of different application requirements have emerged over the years. Although this diversity makes it hard to extract common requirements, most IoT devices are expected to operate for several years on constrained power sources such as batteries.

The requirement of a prolonged operation on a constrained energy budget has triggered the design of low-power communication technologies such as Bluetooth Low Energy (BLE). BLE was explicitly designed with the aim to provide short range communication in applications in which data is transferred infrequently and power consumption is important [1]. However, BLE can be used for many more use cases than those including an infrequent data transmission. One of the most important advancements of BLE is, indeed, Bluetooth LE Audio, a mechanism to transmit audio streams over BLE.

LE Audio was developed to provide audio streaming with low latency and great perceived audio quality whilst keeping power consumption at a minimum. Besides all of these features, LE Audio also allows for different stream topologies such as broadcast streams. These broadcast streams allow multiple receiver devices that are in transmission range of a sender, to pick up the audio stream without the establishment of a connection, which reduces overhead and increases the system's efficiency. This revolutionary technology is based on so-called *isochronous channels*, which allow for contemporaneous audio reception on all receiver devices.

Isochronous channels have the potential to be applied to several IoT applications, such as monitoring systems or smart TVs. In such applications, synchronous data transmission to multiple receiver devices is essential. One particular use case where isochronous channels are potentially introducing high benefits is a motorcycle intercom system, which is the focus of this thesis.

1.1 Problem Statement

Over the years, a multitude of different technologies for motorcycle intercom systems have emerged. In the early days, motorcycle manufacturers like Honda included citizens band (CB) radios on their flagship models but, as time went on, more and more manufacturers started to move towards smaller, more integrated systems, disconnected from the motorcycle itself. These systems are typically mounted on the helmet itself and powered by batteries. This provides a lot of flexibility, especially considering that riders can switch between motorcycles and still use the same intercom system.

These disconnected intercom systems use various different technologies such as Bluetooth Classic, Bluetooth Low Energy, or proprietary communication standards operating in the 2.4 GHz ISM band. However, all those solutions have a number of deficiencies. One of the biggest challenges is reliability: the connection between motorcycle intercom systems often terminates suddenly, or the audio stream itself cuts out due to the presence of obstacles and buildings between the devices or when their distance increases above a certain limit. Furthermore, the audio quality often degrades as the distance increases, and the initial pairing of these devices can be cumbersome and complicated [5]. For example, the initial pairing process between more than two devices can often only be achieved via a smartphone application or dedicated pairing modes. This can take up to several minutes. Additionally, such intercom systems can only operate for a few hours until they need to be recharged. Depending on the tour, this might not be enough for a whole day of riding.

All those limitations inspired me to investigate BLE in more detail, and to determine the benefits of Bluetooth LE Audio in motorcycle intercom systems. Especially a broadcast topology could be seen as an alternative to existing unicast or mesh intercommunication approaches. Therefore, the overall goal of this thesis is to create a mechanism that demonstrates the feasibility of Bluetooth LE Audio for motorcycle intercom systems, focusing on retaining throughput while minimizing the power consumption over broadcast isochronous streams.

However, one of the biggest challenges in this regard is the dynamic adaptation of the Bluetooth LE Audio parameters. These parameters are crucial for power efficiency as well as robustness and range, and are designed to be set up during the first establishment of the stream. Therefore, the first step is the conduction of multiple experiments to determine the optimal parameters, as well as their impact on communication performance as a function of distance. This step is crucial, given that broadcast isochronous streams are at their infancy and scientific work is very limited at this point in time. Afterwards, the gathered knowledge will be used to design and implement the above mentioned adaptation mechanism, which itself is challenging given that the broadcaster typically has no information about the link quality. Finally, the developed adaptation mechanism should be carefully evaluated in terms of reliability, robustness and power consumption in a real-world scenario.

Another key challenge is the hard- and software support for broadcast isochronous streams. Although multiple BLE controllers are available, these are either in an early experimental state or tailored to one particular hardware system, causing unreliable transmissions or lacking necessary features.

1.2 Contributions

This thesis investigates the feasibility of Bluetooth LE Audio, and, specifically, broadcast isochronous streams for motorcycle intercom systems.

Parameter exploration. The first contribution is an experimental campaign to discover the influence of key parameters (detailed in Section 4.3) on the communication performance of broadcast isochronous streams. This includes an analysis of the impact of such parameters on link quality, throughput, power consumption and latency as well as the investigation of the trade-off between costs and benefits. Furthermore, multiple field tests were conducted to explore the behavior of these parameters over distance in a real-world scenario.

Dynamic parameter adaptation mechanism. With the knowledge gained throughout the parameter exploration phase, a dynamic parameter adaptation mechanism was designed. This mechanism adapts the retransmission number as well as the transmission power dynamically, and ensures that the communication between two devices is always at an optimal level in regards to link quality and throughput while reducing the power consumption as much as possible.

Experimental evaluation. To illustrate the improvements introduced by the developed adaptation mechanism, an evaluation against two naïve algorithms has been conducted. This evaluation highlights the shortcomings of traditional adaptation mechanisms and the practical implications of a dynamic parameter adaptation algorithm. Especially in the realm of transmitter current consumption, a dynamic parameter adaptation algorithm is able to outperform both naïve adaptation algorithms by up to 20% in terms of transmitter current consumption.

It is important to highlight that the discovery of the parameters as well as the implementation and evaluation of the adaptation algorithm is executed on Nordic nRF52840 DK boards with the Zephyr controller. Initially, the Nordic nRF5340 DK with the closed-source nRF5340 Audio controller [24] was the hardware of choice, but due to the restricted nature of this controller, the dynamic retransmission adaptation, required for the experimental evaluation, would not have been possible.

1.3 Outline

Chapter 2 provides an overview of Bluetooth Classic as well as Bluetooth Low Energy. Additionally, the development of audio transmission over Bluetooth Classic and Bluetooth Low Energy is presented, and an introduction to audio codecs is given. The Zephyr operating system and the hardware used throughout the thesis is illustrated in detail. Chapter 3 lists existing approaches for audio transmission over Bluetooth Low Energy as well as approaches for parameter adaptation. In Chapter 4, the influence of different broadcast isochronous stream parameters is experimentally determined. Additionally, in Chapter 5, an adaptation algorithm, based on the gathered knowledge is presented, and evaluated in Chapter 6. Finally, Chapter 7 completes the thesis with a conclusion and an outlook on future work in relation to this thesis.

2 Background

This chapter provides an overview of Bluetooth Classic, Bluetooth Low Energy (BLE) as well as the hardware used throughout the thesis.

Sections 2.1 and 2.2 introduce Bluetooth Classic, the motivation and possible implementations in regards to wireless audio transmission. Section 2.3 introduces Bluetooth Low Energy as well as the differences between Bluetooth Classic and BLE. More details on the mechanisms of BLE in regards to audio applications can be found in Section 2.4. Since audio applications often correlate with audio codecs, Section 2.5 provides further information about these. The last two sections, Section 2.6 and 2.7, enumerate the used hardware as well as provide an overview on Zephyr, the real-time operating system used throughout the thesis.

2.1 Bluetooth Classic

Bluetooth Classic refers to the original Bluetooth specification and its subsequent specification adaptations up to the introduction of Bluetooth Low Energy. Bluetooth Classic was developed by the Bluetooth SIG (Special Interest Group) in the late 1990s with the goal to enable wireless communication between various devices [12].

Bluetooth Classic uses the 2.4 GHz ISM radio band[1], divided into 79 different channels, spaced 1 MHz apart [12]. This results in a frequency range from 2402 MHz to 2480 MHz. Furthermore, the data transmission on these channels is carried out via a frequency hopping algorithm to avoid interference with other devices, using the same frequency band [12].

Additionally, Bluetooth Classic uses two modes for data transmission:

- Basic Rate (BR);
- Enhanced Data Rate (EDR).

Both modes use distinct modulation techniques, where BR can achieve a data transmission up to 1 Mb/s and EDR up to 3 Mb/s [12].

Furthermore, Bluetooth Classic implements so-called piconets. These are networks of up to eight nodes, consisting of one master node and up to seven slave nodes. Slave nodes can also be shared across piconets, creating

[1] 2.4 GHz frequency band reserved for industrial, scientific and medical purposes.

a so-called scatternet [12]. It is important to highlight that all nodes in a piconet operate on the same frequency hopping sequence [12].

2.2 Bluetooth Classic Audio

Four use cases were initially proposed to enable audio transmission over Bluetooth Classic (also called Bluetooth Classic Audio):

- Wireless headset;
- Intercom;
- Cordless telephony;
- Dial-up networking.

Over the years, the intercom and cordless telephony use cases failed because of potential revenue loss for cellular operators, and the dial-up networking use case was superseded by Wi-Fi due to high GSM costs [10].

As more and more people started replacing physical mediums for audio distribution with services like Napster or iTunes in the early 2000s, the need for wireless audio transmission also started to increase. This increase led to the development of the Advanced Audio Distribution Profile (A2DP), which became the most used profile for wireless audio transmission over Bluetooth Classic in practice [10].

At first, adoption was very slow due to the fact that consumers did not want to replace corded headphones with wireless ones but this changed drastically with the introduction of Spotify in 2006. The combination of Spotify and the introduction of the first iPhone showed consumers that smartphones can be used for much more than phone calls, which also led to wide adoption of Bluetooth headphones [10]. This correlation can be seen in Figure 2.1. Even though Bluetooth Classic had become mainstream by then, the two main standards for audio transmission were only designed for specific use cases. A2DP allowed for high quality unicast music streaming without a return audio path and the Hands-free Profile (HFP) allowed "low latency, bidirectional, mono voice transmission"[10]. Neither of those standards were designed with the goal of multiple receiver nodes in mind. This was one of the main reasons why, for example, in stereo earbuds, multiple proprietary solutions to this problem emerged over time.

Furthermore, users started swapping between applications more and more as smartphones became a commodity. This led to multi-profile problems, considering A2DP and HFP were two dedicated audio profiles. All these

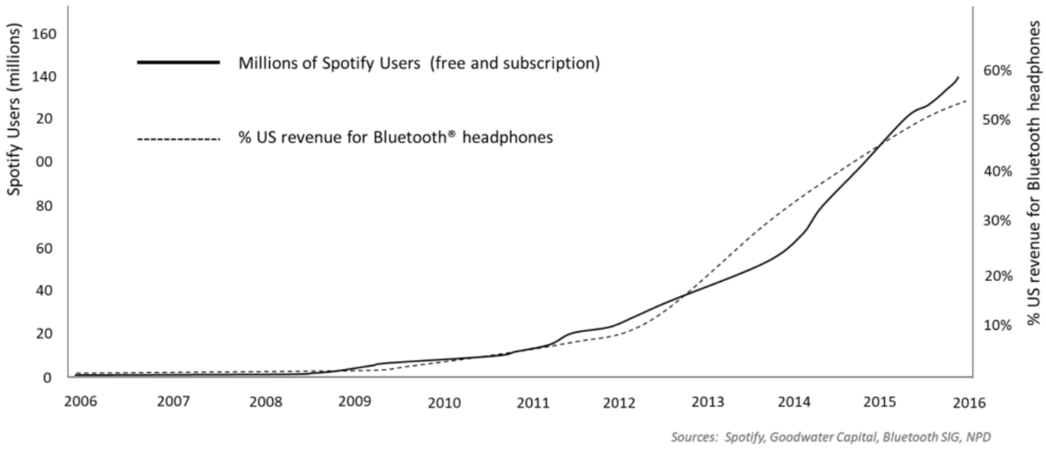

Figure 2.1: "The growth of Spotify subscriptions and Bluetooth® headphones"[10].

problems as well as the steady increase in ubiquitous computing led to the development of Bluetooth LE Audio [10].

2.3 Bluetooth Low Energy

Bluetooth Low Energy (BLE) refers to the redesigned version of Bluetooth Classic with the aim to provide short range communication in applications where data is transferred infrequently and power consumption is important [1]. It was introduced in 2010 with the Bluetooth version 4.0 and is also known as Bluetooth LE or Bluetooth Smart.

BLE uses the 2.4 GHz ISM radio band, but, in contrast to Bluetooth Classic, divides it into 40 different channels, spaced 2 MHz apart [1].

Up to the introduction of Bluetooth 5, BLE only supported one physical mode (PHY): the so-called 1M mode. This mode supports data transmissions of up to 1 Mbps and is mandatory for all BLE devices to ensure backwards compatibility [9].

With the introduction of Bluetooth 5, BLE was able to support two additional physical modes:

- 2M;
- Coded (S2 and S8).

The 2M PHY can achieve data transmissions of up to 2 Mbps due to an increased symbol rate, whereas the coded PHY can achieve a superior range because of an increased number of symbols per bit and the usage of forward error correction.

It is important to highlight that the two modes of operation implemented in the coded PHY S2 and S8 correspond to the number of symbols per bit and are related to the range as well as the throughput. S2 achieves a shorter range than S8 but with a throughput of up to 500 kbps. S8 achieves the greatest range of all BLE physical modes but can only achieve a throughput of up to 125 kbps [9].

In contrast to Bluetooth Classic, BLE also supports a broadcast as well as a mesh topology. The broadcast topology refers to a one-to-many, whereas the mesh topology refers to a many-to-many communication mechanism.

Additionally, BLE uses three dedicated channels (37, 38, 39), the so-called primary advertising channels, to advertise information that can be received and processed by other devices. This information can, for example, be used to initiate a connection or to request further data reception without a connection over the remaining channels (secondary advertisements) [1].

Focusing more on the actual data transmission, one first has to take a look at the architecture of BLE illustrated in Figure 2.2. This illustration divides the architecture into three layers: application, host and controller. The application layer represents the actual application, the host refers to multiple protocols needed for interoperable communication between devices, and the controller refers to the link and physical layer, which is responsible for the actual data transmission.

To ensure interoperable data transmission, BLE defines multiple host layer protocols such as the Attribute Protocol (ATT) and the Generic Attribute Profile (GATT). ATT defines how data is structured and exposed using attributes, while GATT defines services including their characteristics. Both ATT and GATT are necessary to ensure standardized communication and allow receiver devices to identify the sending device including the data they are exposing. This data exposure is achieved via so-called profiles, services, and characteristics.

Furthermore, it is important to differentiate between Bluetooth Classic and BLE since devices implementing these two specifications are not compatible with each other. This differentiation is illustrated in Figure 2.3.

This is the main reason why many smartphones nowadays implement both specifications, which is called dual mode.

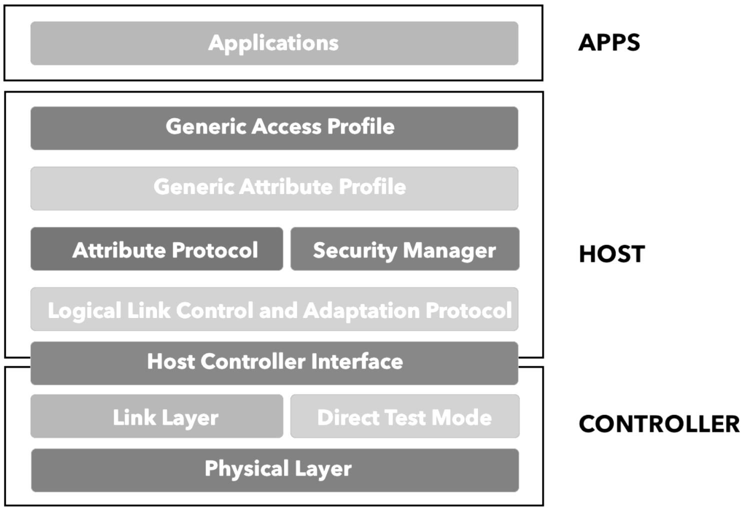

Figure 2.2: Bluetooth Low Energy Architecture [1].

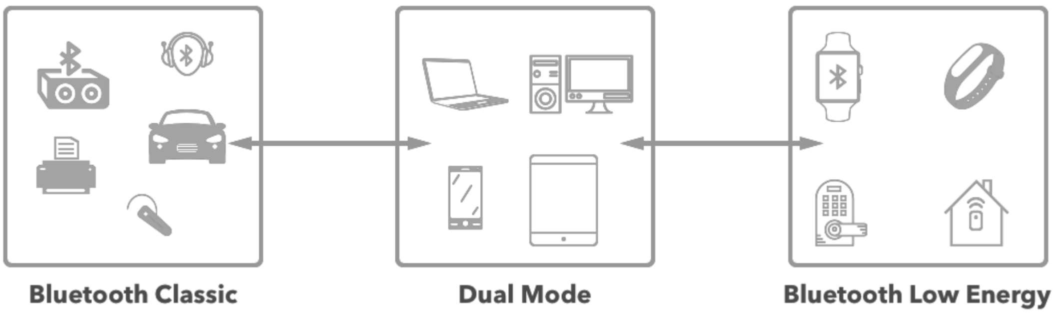

Figure 2.3: "Types of Bluetooth devices"[1].

2.4 Bluetooth LE Audio

Audio over Bluetooth Low Energy, also called Bluetooth LE Audio is the next generation of Bluetooth Audio. It supports not only all requirements of A2DP and HFP, but has also evolved into a powerful toolkit for years to come.

Initially, the Bluetooth LE Audio development was mainly driven by requirements from the hearing aid industry. Later on, basic telephony with separate left and right audio streams including return streams or low latency audio transmission from TVs also became part of the LE Audio standard.

Thanks to the latter, users should be able to select between broadcast audio streams or maintain connections to multiple other Bluetooth devices whilst saving as much energy as possible to extend battery life. Additionally, devices like hearing aids have constrained user interfaces, which led to the requirement that users should also be able to control them on other devices like smartphones or remote controls [10].

As the implementation of all those requirements on top of the existing Bluetooth Audio specification would have involved too many compromises, the implementation on top of the Bluetooth 4.1 specification ultimately broke the backwards compatibility with Audio over Bluetooth Classic [10].

One important feature added to LE Audio are isochronous channels, which are carrying the actual unidirectional or bidirectional audio streams alongside an existing asynchronous connection-oriented logical channel that is used for configuration and control of the audio stream [10].

LE Audio also allows for broadcast audio streams that, in contrast to unicast streams, allow sender devices to announce their presence via advertisements and multiple receiver nodes can scan for those and join the stream. These advertisements are so called Extended Advertisements (EA) and Periodic Advertisements (PA). These allow scanning devices to obtain details on how to connect, what is supported, and what the corresponding hopping sequence is [10]. Additionally, Periodic Advertising Synchronisation Transfer (PAST), another feature in LE Audio, helps external devices finding a broadcast stream, which, therefore, allows to extend the battery life of a scanning device.

To be able to handle 10 ms frames, which are recommended for the LC3 audio codec alongside 7.5 ms frames of legacy codecs, LE Audio makes use of the Isochronous Adaptation Layer (ISOAL), which is responsible for the bidirectional conversion of Service Data Units (SDUs) and Protocol Data Units (PDUs).

Finally, the Enhanced Attribute Protocol (EATT) has been added to the Core version 5.2, which allows "multiple instances of the ATT protocol to run concurrently"[10].

All of the aforementioned improvements suggest that Bluetooth LE Audio is "able to support the audio applications we have today as well as new audio applications which are yet to come"[10].

2.4.1 Architecture

As illustrated in Figure 2.4, Bluetooth LE Audio uses a multi layer structure with LE Audio specification blocks [10].

In general, one can distinguish between host and controller, where the controller refers to the link layer and is responsible for the actual transmission of Bluetooth packets and the host refers to all layers above.

One of the most prominent additions is the Generic Audio Framework (GAF), which acts as an audio middleware and includes all features that are likely to be used by multiple audio applications. Such features can be, for example, the Audio Stream Control Service (ASCS), which is responsible for the exposure of codec settings, or the Broadcast Audio Scan Service (BASS), which specifies how an external device can support the discovery of broadcast streams.

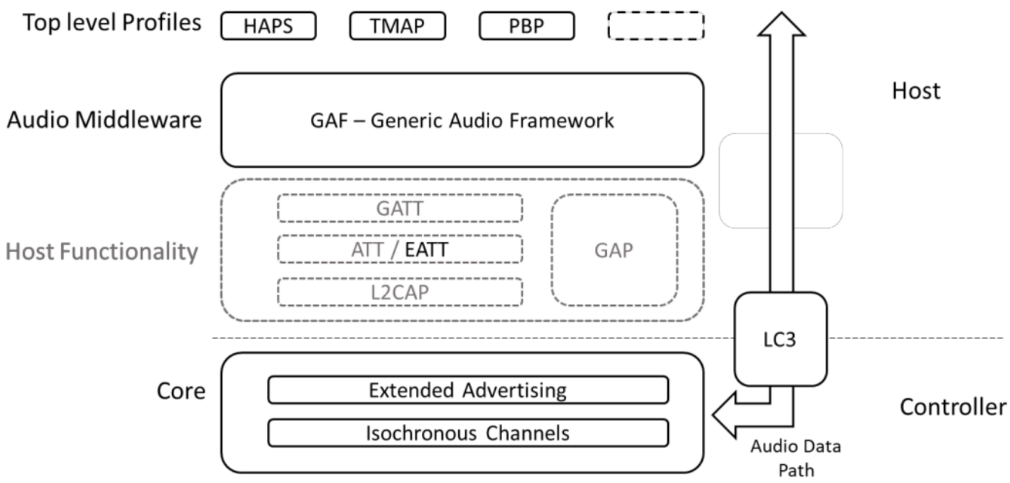

Figure 2.4: Bluetooth LE Audio architecture [10].

All features that are classified as optional in the GAF, are specified in the top level profiles. These profiles provide additional functionality for specific audio applications and are built on top of GAF features [10]. The Hearing Access Profile and Service (HAPS) and the Telephony and Media Audio Profile (TMAP) are two prominent examples.

Furthermore, as shown on the right side of Figure 2.4, LE Audio also introduces the Low Complexity Communications Codec, also referred to as LC3. This codec is used to encode and decode audio data in an efficient way and provides "excellent performance for telephony speech, wideband and super-wideband speech and high-quality audio"[10]. It is important to note that LC3 must be supported by every LE Audio product to ensure interoperability.

2.4.2 Topologies

Bluetooth LE Audio is built on the asymmetry property of Bluetooth Low Energy, in which one central device can connect to multiple peripheral devices, which can be more energy preserving. Depending on the type of isochronous stream, peripheral devices may also take part in the configuration of the stream itself [10]. With Bluetooth LE Audio, one can not only create the same topologies as with HFP and A2DP, but also multi acceptor setups as illustrated in Figure 2.5.

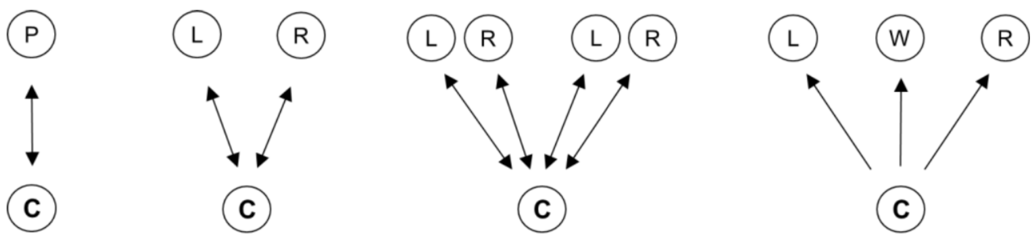

Figure 2.5: Unicast audio topologies supported in Bluetooth LE Audio [10].

Furthermore, Bluetooth LE Audio also allows for the creation of broadcast streams where a single broadcast device can broadcast data to multiple receiver nodes as shown in Figure 2.6. This topology allows for some practical use cases such as the transmission of "simultaneous audio streams in different languages"[10].

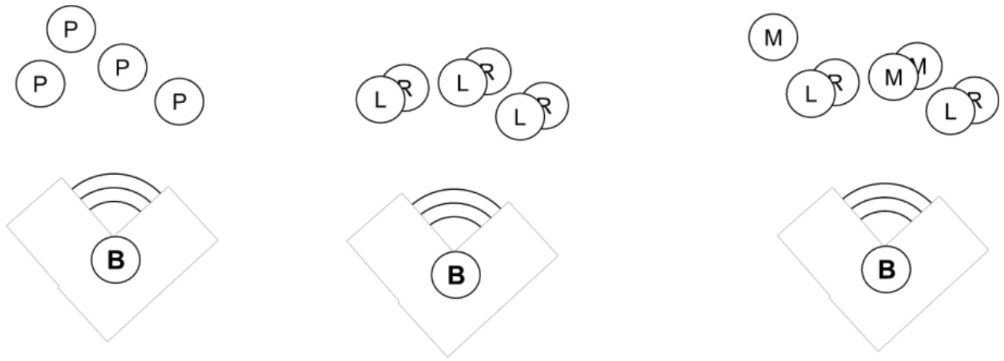

Figure 2.6: Broadcast audio topologies supported in Bluetooth LE Audio [10].

2.4.3 Isochronous Streams

To be able to create a Bluetooth Audio system that can not only cope with today's requirements but also with future ones, Bluetooth LE Audio has introduced the concept of a separate audio and control plane. This separation allows the transmission of pure audio streams over isochronous channels while an existing Asynchronous Connection-Less (ACL) channel is used to configure and control the stream.

Furthermore, until the introduction of LE Audio, Bluetooth Audio only allowed point-to-point transmission of data, which led to the development of various proprietary extentions to allow, for example, true wireless earbuds. Isochronous streams not only provide a solution to this problem but also allow for a multitude of receiver devices and streams.

In general, isochronous streams can be differentiated into unicast streams and broadcast streams. Unicast streams, also called Connected Isochronous Streams (CIS), are very similar to Bluetooth Classic Audio due to their point-to-point topology. Every sent packet is acknowledged by the receiver and the ACL connection is maintained throughout the CIS lifetime. In contrast, broadcast streams, also called Broadcast Isochronous Streams (BIS) are used for the audio transmission to multiple receiver nodes at once. Since the sending node, also referred to as the broadcaster, has no knowledge of the receiver nodes in the simplest form of a broadcast stream, there is also no need for an ACL connection [10].

Initially, the development of broadcast streams was driven by requirements of the hearing aid industry. Until the introduction of LE Audio, Telecoil, an inductive hearing aid technology introduced in 1947, was the standard used for broadcasting audio to hearing aids [17]. Telecoil was mostly used in public settings, so that users can pick up audio if their physical location is near a Telecoil broadcaster. However, due to the high costs, low audio quality and lack of interoperability (i.e. the need for hearing aids to be compatible with any smartphone or TV), the hearing aid industry was looking for a notable successor [10].

2.4.4 Broadcast Isochronous Streams

Broadcast Isochronous Streams (BIS) allow any receiver device within range of the broadcasting device to pick up the broadcasted audio streams. Unlike Bluetooth Classic Audio, these broadcast streams are one-way streams, meaning that packets are not acknowledged [10]. This fundamental technology introduces multiple real-world advantages. One of these advantages is that the link budget is often asymmetric (due to the fact that transmitter nodes are often mains powered), which results in a far greater range given that broadcasting can be done at the maximum power setting allowed [10].

As illustrated in Figure 2.7, the general structure of a BIS PDU consists of a 16 bit header, 32 bit optional Message Integrity Check (MIC) and up to 251 octet payload part.

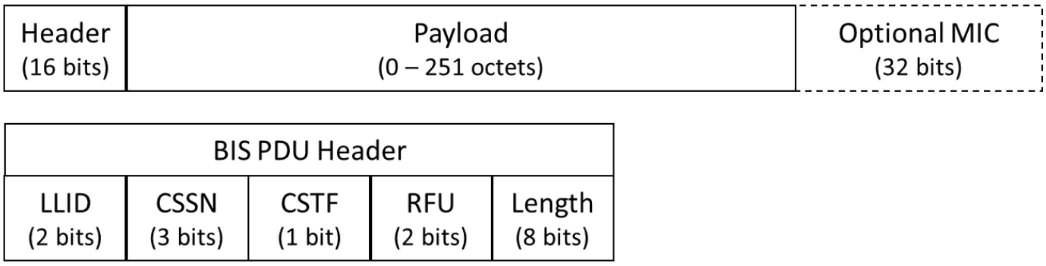

Figure 2.7: "Isochronous PDU and header for Broadcast"[10].

Taking a closer look at the header itself, one can determine the LLID (Link Layer ID) field, responsible for the indication of the content type (framed or unframed) of the PDU [2]. Additionally, the CSSN (Control Subevent Sequence Number) and CSTF (Control Subevent Transmission Flag) fields indicate the presence of a Control Subevent and the "Length field indicates the size, in octets, of the payload and MIC, if included"[2]. The RFU field in the PDU header is an acronym for "Reserved for future use"[2] and, therefore, shall not be used.

The 32 bit optional MIC shall be included in encrypted broadcast isochronous streams and is calculated using AES-CMAC [2].

In contrast to connected isochronous streams, broadcast isochronous streams are only transmitting unidirectional data. To be able to propagate information such as changes to the hopping sequence, regardless of the unidirectional nature, broadcast isochronous streams can transmit control subevents after the final subevent of the last BIS [10]. It is important to point out that control subevents do not need to be included in each Broadcast Isochronous Group (BIG; consisting of one or more broadcast isochronous Streams) event, but rather occasionally when information dissemination to all nodes is necessary [10].

As illustrated in Figure 2.8, BIG events are spaced ISO_Interval apart, where each BIG event can consist of multiple BIS events BIS_Spacing apart. Depending on the Sub_Interval and BIS_Spacing, the structure of the broadcast isochronous stream can either be sequential or interleaved.

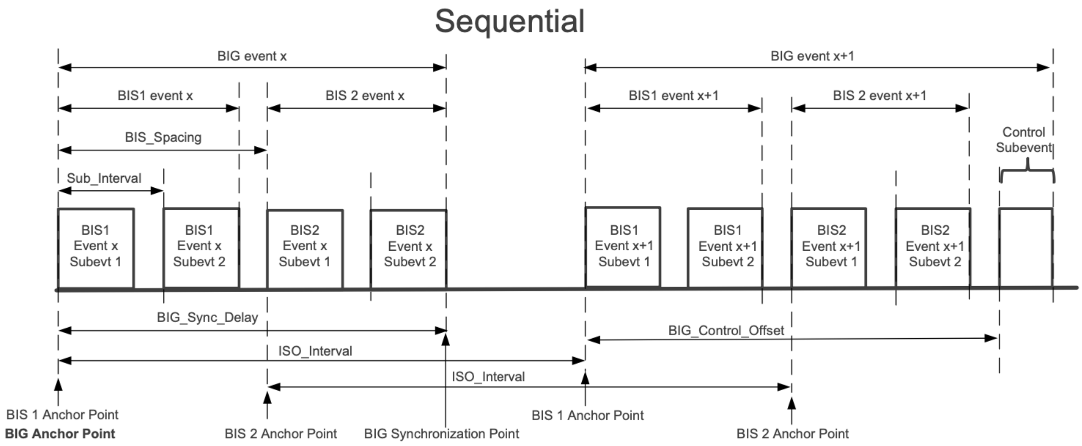

Figure 2.8: Sequential Broadcast Isochronous Stream Structure [2].

Additionally, one can determine that the Number of Subevents (NSE) of Figure 2.8 is two and the Control Subevent is not included in the NSE calculation.

Another key difference to connected isochronous streams is that broadcast isochronous streams have no idea if the packets have been received or not, due to the missing acknowledgments. To combat this limitation, broadcast isochronous streams introduce the concept of retransmission, where packets can be retransmitted to provide robustness. Furthermore, receiver nodes can turn off their radio as soon as a packet has been received until the next scheduled BIS [10]. This additionally supports the asymmetry property of BLE, since broadcast nodes are always transmitting whilst receiver nodes can sleep under certain conditions. This is especially important in energy-constrained environments.

Diving deeper into the robustness properties of broadcast isochronous streams, one can see that the core specification denoted the concept of a group count. This group count basically is the ratio between the number of subevents and the burst number and is used to increase diversity in the transmission scheme [10].

Two examples of group count usage can be seen in Figure 2.9. Each sample consists of six subevents (NSE) and a Burst Number (BN) of three and two respectively. These parameters led to the group counts of two and three given the calculation with the formula $GC = \frac{NSE}{BN}$.

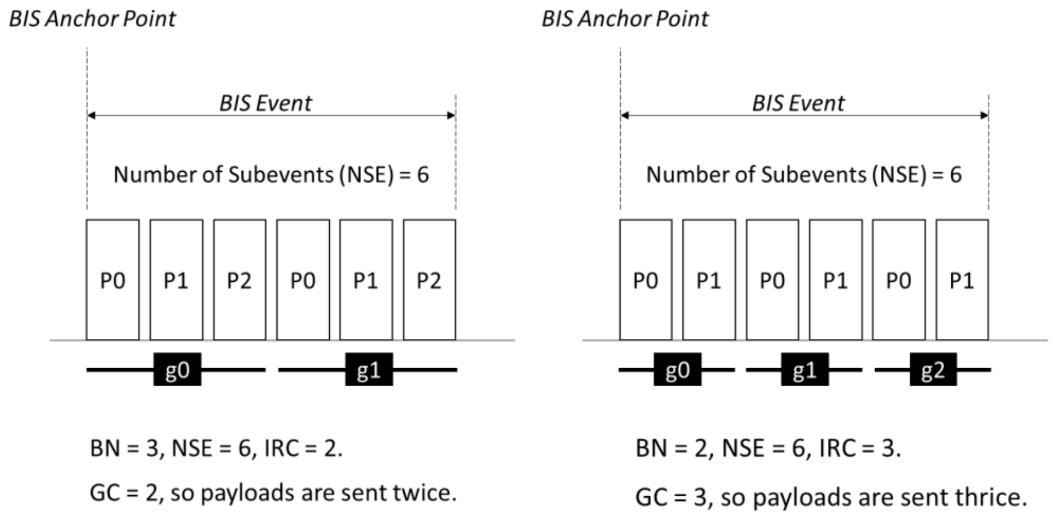

Figure 2.9: "The effect of Group Count"[10].

As stated previously, the number of subevents as well as the burst number is used to increase the diversity of packet transmissions, meaning increasing the chance that every receiver node receives the packet. In addition to the concept of group count, this is achieved with the introduction of the Immediate Repetition Count (IRC) and Pre-Transmission Offset (PTO).

IRC basically defines "the number of Groups which carry data associated with the current event"[10] and the PTO allows "early transmission of packets in the hope that a receiving device can receive audio data packets early"[10]. Therefore, these parameters can "determine which payloads are sent in each transmission slot"[10]. According to the Core Specification, this classification is given by the following rules:

- If g $<$ IRC, then group g shall contain the data associated with the current BIS event [2].
- If g $\geq$ IRC, then group g shall contain the data associated with the future BIS event that is PTO $\times$ (g - IRC + 1) BIS events after the current BIS event [2].

In general, some subevents will get associated with the current BIS event and some with future BIS events. Figure 2.10 illustrates this relationship, where five subevents belong to the current BIS event and three to future BIS events. A practical example of the usage of these parameters can be seen in Figure 2.11, in which the relationship between the number of subevents and the burst number results in a group count of five. Given the IRC of three, and the actual group number g_n, one can conclude that the first three groups belong to the current BIS event x and the remaining two groups belong to the BIS events $x + 2$ and $x + 4$. Therefore, this example spreads the transmitted data

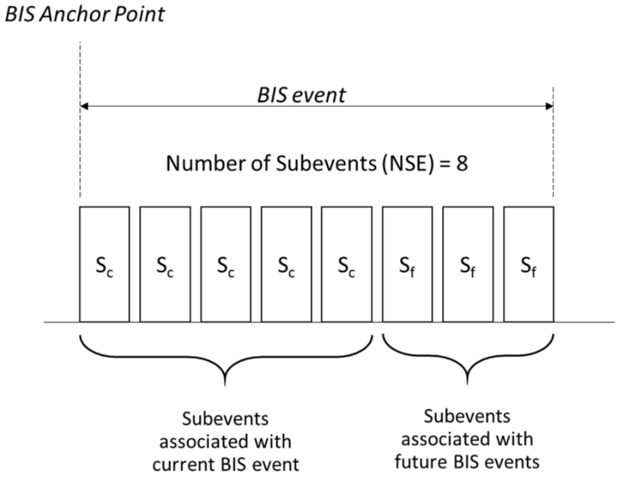

Figure 2.10: Future subevents [10].

out over five BIS events, resulting in a total latency of $4 * \text{ISO\_Interval}$ [10].

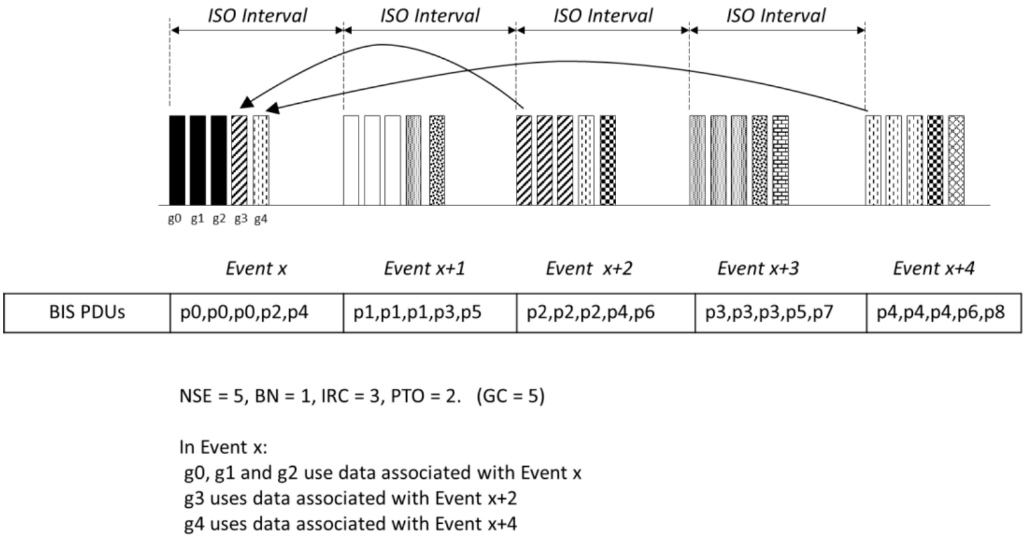

Figure 2.11: Example on the effects of IRC and PTO [10].

Another important aspect of a broadcast isochronous stream is its ability to synchronize its BIG events. This aspect is, for instance, needed in true wireless earbuds, where users have one Bluetooth device for the left ear and one for the right ear. Both devices must render the audio at the exact same point in time to mitigate the interpretation of a moving audio source by the human brain [10]. This effect is caused by the ability of the human's head to detect even slight differences in the arrival time. For example, a rotation of the human's head by 10 degree is equivalent to over $70\ \mu s$ difference in arrival time [10].

To be able to achieve such synchronization, the core specification defines a BIG synchronisation point, which "coincides with the end of the transmission of the audio data"[10], as illustrated in Figure 2.8 [2]. This point, also called the SDU synchronization reference, is a fixed point in time at which every receiver node knows that every other receiver node has received the data.

Since audio data often involves further processing before it can be rendered, this point in time can be used as a reference point, at which a so-called presentation delay can be applied. This presentation delay defines the point in time at which the data has to be rendered, as illustrated in Figure 2.12 [10].

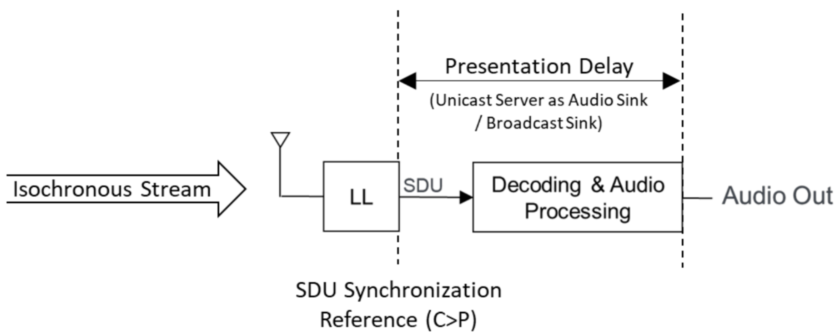

Figure 2.12: "Presentation Delay for rendering on an Acceptor"[10].

Due to the definition of broadcast isochronous streams, "the Initiator needs to make a judgement of what value of presentation delay will be acceptable to all potential broadcast sinks, based solely on its application"[10]. Although all the nodes must support values above 40 ms, it is not recommended due to the fact that such high presentation delay may introduce echo when ambient sounds are present [10]. Unfortunately, the core specification does not define how a receiver node should handle values outside its range. However, in practice, top level profiles like, for example, TMAP may specify the presentation delay value indirectly.

2.5 Audio Codecs

Audio transmission over a wireless connection is always a balancing act between audio quality and latency. Packet loss caused by interference may introduce gaps to the audio stream itself and packet retransmissions provide redundancy but increase the overall latency at the same time. To be able to increase the retransmissions while keeping the latency increase at a minimum, audio codecs may be a viable option.

Up to the invention of the CD, audio transmission was not only susceptible to interference but "a scratch on a record or wax cylinder, or stretching on a tape"[10] may cause permanent loss or distortion of the audio data itself.

This changed with the introduction of pulse code modulation (PCM) [10]. PCM samples audio signals at higher frequencies than humans can hear and converts them into digital signals. The higher the sampling frequency, the closer the decoded audio output will be compared to the original input. However, due to the 44.1 kHz sampling rate, 16 bit depth and lack of compression, PCM generates an output of 800 kbps [10]. Therefore, a standard CD could only store approximately 74 minutes of music. [10].

In contrast, MP3, one of the most influential audio codecs of all time, developed by the Frauenhofer institute in the early 1990s, was able to reduce the file size to 25%-95% of its original size [10]. The usage of perceptual coding, a technique which compares audio data with knowledge of what a human can actually hear, was a game changer for the audio industry and allowed for music sharing services such as Napster. Additionally, due to the compression of MP3, robust wireless audio transmission with multiple retransmissions became feasible [10].

In general, one can think of the audio transmission latency in terms of all of its elements from the actual audio input until the audio output. Figure 2.13 illustrates these elements, starting with the audio input (which needs to be sampled and encoded) meaning that a set of consecutive samples, also called a frame, gets interpreted and encoded according to the selected audio codec [10]. Afterwards, this encoded audio data will get transmitted, which causes a certain transport latency depending on the number of retransmissions and the actual transmission time. This transport latency can be in the "range from a few milliseconds to several tens of milliseconds"[10]. Finally, at the receiver side, the encoded audio data gets decoded to retrieve the actual audio data. This decoding, depending on the actual codec, is usually quicker and requires less power than the encoding since most codecs are designed in an asymmetric fashion [10].

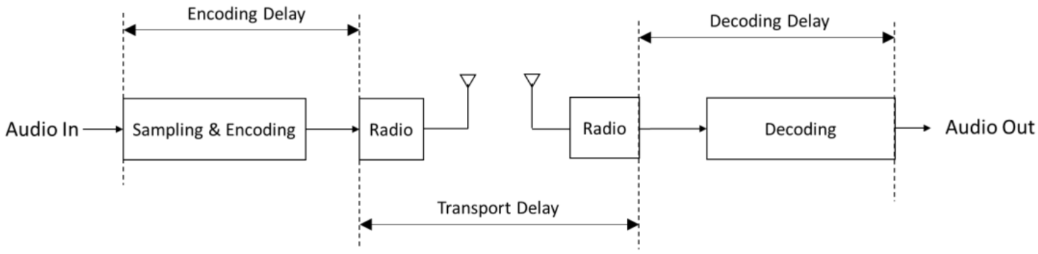

Figure 2.13: Audio transmission elements [10].

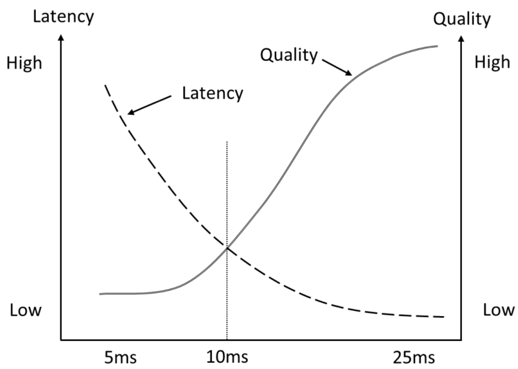

Figure 2.14: Audio codec latency/quality tradeoff [10].

Depending on the actual codec itself, encoding may use different frame sizes. As illustrated in Figure 2.14, the selection of the ideal frame size can be tricky since a smaller frame size leads to a smaller amount of available audio samples to encode, which, therefore, limits the amount of information fed to the encoder. This might be a problem if the encoder uses, for instance, perceptual coding techniques: these require a certain amount of information to work properly, and hence affect the overall audio quality [10]. Respectively, the higher the frame size, the longer the codec needs to wait per audio frame, which increases the audio quality but also increases the overall latency. A good compromise between latency and quality can typically be found in the range of a 10 ms frame size, which is also the reason why 10 ms has become the de facto industry standard regarding the frame size [10].

Furthermore, over the past decades, multiple audio codecs have been developed and used for wireless audio transmission. Continuous Variable Slope Delta modulation (CVSD) was one of the first codecs and was used in the hands-free profile of Bluetooth Classic. CVSD provided a very low latency, which was ideal for telephony applications. In contrast, Sub Band Coding (SBC), a codec based on psychoacoustic modeling, used in the A2DP profile of Bluetooth Classic provided high quality audio at the expense of a higher latency.

Over time, more and more codecs were developed and used in practice, one of the reasons being the A2DP specification, which permitted the usage of alternative codecs. Some of these codecs were, for instance, AAC, ATRAC, OPUS, or EVS.

While some of these codecs are targeted towards very high quality, some interesting developments also happened in the field of low bitrate codecs. Zeghidour et al. in their paper on *SoundStream: An End-to-End Neural Audio Codec* conducted a *MUSHRA*[2] test to assess the subjective sound quality over multiple low bitrate audio codecs [30]. This test shows that *SoundStream*, *EVS*, and *Opus* can achieve a *MUSHRA* score over 60 with bit rates limited to 12 kbps as shown in Figure 2.15.

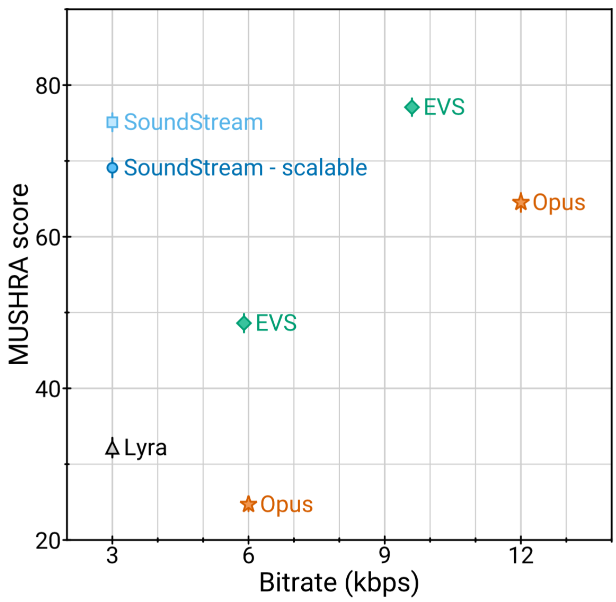

Figure 2.15: Low bit rate audio codec comparison [30].

All of these requirements forced the Bluetooth SIG to create a new audio codec, which could be used universally to support applications that need a low latency (and, therefore, a low bitrate) and support applications that need the highest audio quality possible (while keeping the latency at a reasonable

[2]MUSHRA is the acronym for "MUlti Stimulus test with Hidden Reference and Anchor" [29] and refers to a method for subjective audio quality assessment.

level). Additionally, due to the often constrained nature of Bluetooth products, the new codec should also be very efficient to save as much energy as possible.

Finally, after years of development, the Low Complexity Communications Codec, also called LC3, was introduced by the Bluetooth SIG.

2.5.1 Low Complexity Communications Codec

The Low Complexity Communications Codec (LC3) is not only "one of the most advanced audio codecs available today"[10], but it also provides a wide range of possible settings to accommodate for both low latency and high quality applications, as it can be seen in Figure 2.16.

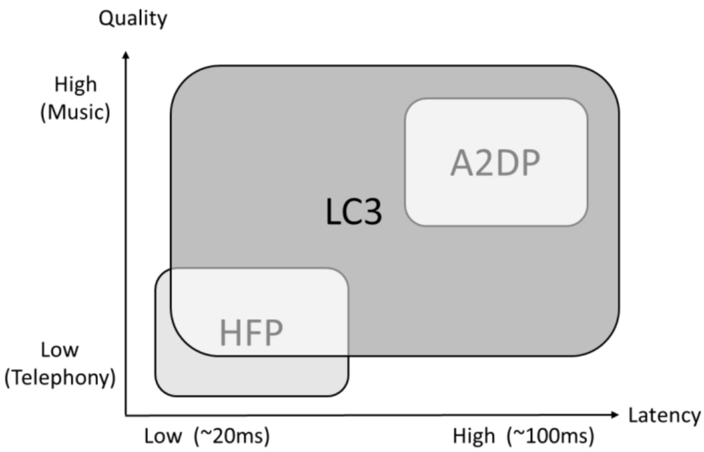

Figure 2.16: LC3 application range [10].

LC3 is optimized for a 10 ms frame size, but also provides compatibility with the 7.5 ms frame size of Bluetooth Classic Audio applications [10][16]. It also supports other specific combinations, which are detailed in Table 2.1.

During the development of LC3, multiple subjective audio quality tests were conducted to quantify its performance against legacy codecs. These quality tests revealed that, for example, LC3 provided equivalent or better audio quality compared to SBC at half the bitrate [10]. This not only reduces the transport delay, given a smaller chosen packet size, which allows for significant power saving but also allows for the implementation of additional functionality on devices that previously might have needed the full bandwidth just for the audio transmission itself [10].

Feature	Supported Range
Frame duration	10 ms (10.88 ms @ 44.1 kHz) and 7.5 ms (8.163 ms @ 44.1 kHz)
Look ahead delay	2.5 ms (2.72 ms @ 44.1 kHz) for 10 ms frame duration 4 ms (4.35 ms @ 44.1 kHz) for 7.5 ms frame duration
Total algorithmic delay	12.5 ms (13.6 ms @ 44.1 kHz) for 10 ms frame duration 11.5 ms (12.52 ms @ 44.1 kHz) for 7.5 ms frame duration
Supported sampling rates	8, 16(HA-SQ), 24(HA-HQ), 32, 44.1, and 48 kHz
Supported bitrate	20–400 bytes per frame and audio channel
Supported bits per audio sample	No restriction by the algorithm; however, optimized for 16-, 24-, and 32-bit depth input. See the limitation described in Section 3.2.3.

Table 2.1: Main features of LC3 [16].

Another very important feature in LC3s is the usage of Packet Loss Concealment (PLC). Due to the fact that Bluetooth shares the same 2.4 GHz frequency spectrum as Wi-Fi, packets will inevitably get lost. Unfortunately, lost packets in the realm of an audio stream are definitely noticeable by users. Over time, multiple concealment approaches have been used in practice, such as repetition of the previous frame. However, these approaches often have significant downsides and mostly break if multiple consecutive packets are lost [10]. PLC, in general, is a mechanism that tries to mitigate these shortcomings by predicting the lost audio data [16].

Sampling Rate	Description
8 kHz	Suitable for voice of telephony quality.
16 kHz	Higher quality voice. Adequate for voice recognition applications.
24 kHz	Adequate for music where there are imperfect listening conditions, such as background noise, or where a listener has any hearing impairment. Listeners are likely to detect a difference from higher sampling rates if they are concentrating on the audio stream in noise-free surroundings.
32 kHz	Most users will not detect a difference from the original when listening with any background noise.
48 kHz	Users cannot detect a difference to the original.

Table 2.2: Recommended LC3 sampling rates [10].

Finally, one also has to consider the overall audio quality in relation to the sampling rate. Nick Hunn, in his book *Introducing Bluetooth LE Audio* [10], provides an intuitive explanation on how the sampling rate should be chosen, based on the particular application as well as independent test results published by the Bluetooth SIG. This explanation is illustrated in Table 2.2.

2.5.2 Low Complexity Communications Codec Plus

The Low Complexity Communications Codec Plus (LC3plus) is the sibling of LC3 and provides additional functionality. It is not only targeted towards extreme low delays and high quality audio transmission, but also towards transmission robustness. LC3plus implements a high performance PLC algorithm as well as forward error correction to sustain a high level of perceived audio quality on connections suffering from packet loss [6].

According to the work of Schnell et al. [22], LC3plus is able to sustain a fair to good audio quality[3] even at a packet loss rate of up to 6%.

2.6 Zephyr RTOS

Zephyr, an open-source project hosted by the Linux Foundation, aims to create the "best-in-class small, scalable, real-time operating system (RTOS) optimized for resource-constrained devices, across multiple architectures"[21].

One of the main goals of Zephyr is to include "silicon vendors, OEMs, ODMs, ISVs, and OSVs"[21] directly in the development process, to shorten the time to market period and, therefore, allow for faster and more efficient technology evolution. It is based on a "small-footprint kernel"[21], optimized for resource-constrained embedded systems and supports, simple sensor based embedded systems as well as sophisticated IoT applications [21].

Additionally, particularly applicable to this use-case, Zephyr also supports Bluetooth Low Energy, providing an open-source, Bluetooth SIG qualified BLE controller for Nordic Semiconductor devices [31].

2.6.1 Bluetooth Low Energy Stack Architecture

In general, the Zephyr BLE protocol stack consists of three main layers: host, controller, and radio hardware [31].

[3] Based on ITU-T P.800.2, a standard for subjective quality assessment [18].

As illustrated in Figure 2.17, the host layer sits in between the controller and the actual application, and consists of multiple network and transport protocols. These protocols allow applications to communicate with peer devices in an interoperable manner [31].

Moving one layer down in the Zephyr Bluetooth stack, one can find the BLE controller. This layer, together with the radio hardware that is responsible for the 2.4 GHz digital baseband functionality, implements the actual link layer protocol and, therefore, enables over-the-air communication [31]. Specifically, it schedules the packet transmission, reception and handles all link-layer control procedures [31].

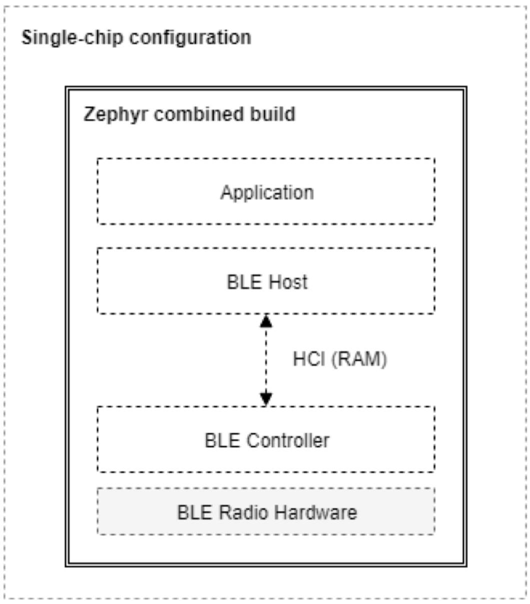

Figure 2.17: Zephyr single-chip BLE architecture [31].

As indicated by the double arrow in Figure 2.17, the host also needs a way to communicate with the controller and vice versa. This functionality is implemented in Zephyr with the usage of the Host Controller Interface (HCI). The HCI can be implemented over different transport media such as "UART, SPI, or USB"[31] and ensures vendor-independent interoperability.

Up to this point, we have only considered the single-chip configuration, in which one system-on-chip (SoC) implements both host and controller. However, Zephyr also allows for dual-chip configuration, where the host and controller can run on separate dedicated hardware. This allows for energy and performance optimized applications [31].

2.7 Hardware

Hardware of Nordic Semiconductor has been chosen to perform multiple comparable evaluations in this thesis. This section provides a detailed insight in the Nordic Semiconductor development hardware used throughout this work.

Generally, all Nordic nRF5 series devices can also use proprietary Bluetooth controller implementations. The most prominent one is the so called SoftDevice, provided by Nordic Semiconductor, as an alternative to the Zephyr open-source controller. This SoftDevice provides Nordic-certified Bluetooth functionality at the cost of a closed-source implementation.

2.7.1 Nordic Semiconductor nRF52840 DK

The nRF52840 DK is a single board development kit manufactured by Nordic Semiconductors. It contains a nRF52840 SoC that, amongst other features, provides a 2.4 GHz radio. This radio supports Bluetooth 5.3, IEEE 802.15.4[4] as well as ANT[5]. Additionally, the SoC contains a 32-bit Arm Cortex-M4 processor, clocked at 64 MHz and 1 MB flash as well as 256 KB RAM memory. Furthermore, it is fully supported in ZephyrOS, and additionally, is used in the D-Cube testbed. Figure 2.18 illustrates the nRF52840 DK.

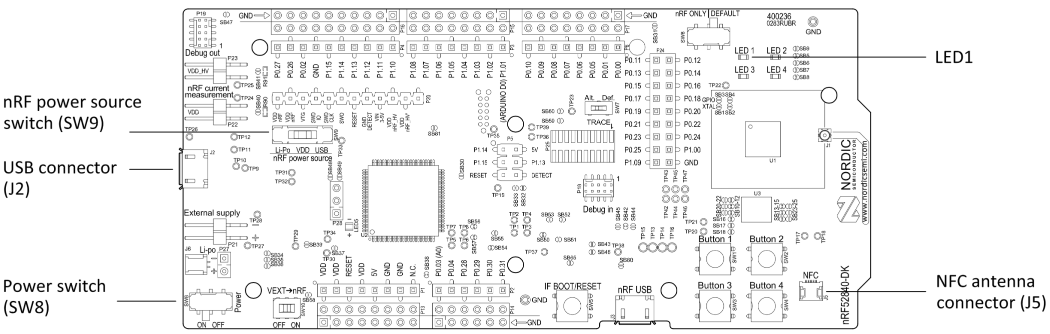

Figure 2.18: Illustration of a Nordic Semiconductor nRF52840 development kit [19].

[4]A technical standard for low-rate wireless personal area networks.
[5]A protocol for ultra-low power wireless data transmission [7].

3 Related Work

This chapter provides an overview of some related work to this thesis. Section 3.1 introduces approaches for audio and voice transmission over BLE. Section 3.2 illustrates previous work on isochronous channels, whereas Section 3.3 lists mechanisms for dynamic parameter adaptation in BLE-based systems.

3.1 Audio over BLE

Since audio transmission over BLE is not specified in the Bluetooth Core Specification, no standard procedure has been defined. This section introduces two approaches to bypass this limitation. Both propose the implementation of a custom service and the usage of GATT notifications, given their asynchronous nature and minimal packet overhead.

BlueVoice. Gentili et al. proposed an implementation with two separate BLE characteristics, an audio characteristic and a sync characteristic [8]. Via these two characteristics, audio data and synchronization data can be transferred periodically at different connection intervals. For example, audio data could be transmitted at a connection interval of 10 ms while synchronization data only gets transmitted at a connection interval of 160 ms [8]. In combination with sophisticated audio processing, they were able to achieve an average throughput of 64.3 kbps at an average latency of 15.80 ms.

Texas Instruments - Voice over BLE. Based on the same basic principle as BlueVoice, the Texas Instruments Voice over BLE mechanism also proposes the usage of a custom service with two separate BLE characteristics [11]. In contrast to BlueVoice, these two characteristics are described as the start characteristic and the audio characteristic. The start characteristic is used to transmit a start and stop command, indicating the first and last packet of the stream [11]. The audio characteristic is used for the actual data transmission and is dedicated to audio packets only [11].

Limitations. Although both mechanisms are able to transmit audio data over BLE, neither of them is able to guarantee isochronous data transmission. Therefore, simultaneous transmission to multiple devices will result in different rendering points. Furthermore, in contrast to isochronous streams, neither approach is able to restrict nor fix the actual latency of the transmission.

3.2 Isochronous Channels

Wireless isochronous data transmission is becoming increasingly important – not only in the realm of audio transmission, but also in industrial applications. This can be, for example, the need for isochronous control of multiple machine components, or the data extraction with high temporal requirements. In 2013, Trsek et al. already investigated such mechanism for IEEE 802.11 [28]. As the need for more power efficient systems increased, the interest in isochronous data transmission for Bluetooth Low Energy started to grow. Finally, with the introduction of Bluetooth 5.2, isochronous data transmission over BLE became standardized, and is referred to as isochronous channels. This feature can fulfill temporal requirements and operates on a fixed amount of retransmissions to ensure packet delivery. Although these retransmissions are beneficial and can increase the link quality, they may lead to unnecessary packet transmission in good quality links. One possible solution, to avoid such unnecessary transmissions, is the determination of the optimal retransmission number based on a stochastic method and the adaptation of the broadcast mechanism proposed by Jaeho Lee [13]. This solution can be more energy efficient under certain conditions but requires a high reception probability. Nevertheless, research in this area is still at its infancy and further research is necessary.

3.3 Parameter Adaptation

Multiple dynamic parameter adaptation mechanisms have emerged over time. In general, they can be separated into mechanisms for the adaptation of the connection interval [14][20][26], the data rate [3], or the transmission power [15][20][25]. Even though they are adapting different parameters, all mechanisms have one thing in common, the need for connection information on the sending device. This information can either be retrieved via the connection itself or with a dedicated feedback connection.

One particular adaptation mechanism dedicated to parameter adaptation over Bluetooth Low Energy is AdaptaBLE, a data rate and transmission power adaptation mechanism, proposed by Park et al. [20]. It illustrates the possible improvements in terms of energy consumption while retaining a reliable connectivity compared to static parameters. This mechanism varies the transmission power as well as physical mode of a BLE connection, based on the link quality behavior over a certain amount of time [20]. AdaptaBLE proved to outperform the baseline scheme but consumed more energy than the static selection of the optimal parameters.

However, the approaches listed in this section are insufficient for this use-case due to eventual fast-changing environmental conditions and tight link quality constraints.

4 Experimental Campaign

This chapter presents a series of experiments determining the influence of key parameters (illustrated in Section 4.3) on the performance of broadcast isochronous streams. The goal of these experiments is to characterize the behavior of these parameters and provide a solid foundation for the design and implementation of the adaptation algorithm that will be illustrated in Chapter 5.

Specifically, Section 4.1 describes the experimental setup, whereas Section 4.2 illustrates the actual software used and the implemented isochronous synchronization mechanism. Finally, Section 4.3 introduces all tested parameters as well as their limitations and Sections 4.4 and 4.5 present the results and our conclusions.

4.1 Experimental Setup

To be able to determine how BIS parameters and transmission power influence broadcast isochronous streams, two different experimental setups were used.

4.1.1 D-Cube

D-Cube, a public benchmark infrastructure for accurate determination of dependability metrics [23], was used to ensure repeatable and accurate results throughout parts of the experimental campaign. D-Cube allows, in the case of broadcast isochronous streams, to specify one sender node and concurrently benchmark multiple receiver nodes in terms of end-to-end latency and power consumption. Additionally, it allows for comparison between good quality links and intermediate quality links, which can provide additional information needed for edge-case behavior determination.

An example of one D-Cube experiment is illustrated in Figure 4.1, in which the blue node indicates the ISO broadcaster and all other nodes are configured as ISO receivers. Green nodes indicate all nodes that were able to receive the BIS, while red nodes indicate a packet reception rate of 0%.

All experiments were conducted ten times over a time period of 120 seconds each.

Figure 4.1: D-Cube topology.

4.1.2 Field Test

For parameter determination over distance, two different experimental setups were established.

Figure 4.2: Outdoor setup; test track marked in blue [4].

One experiment was executed outdoors on a flat, straight dirt track which is illustrated in Figure 4.2, while the other experiment was conducted indoors in a narrow, open hallway. In both experiments, the Nordic nRF52840 DK development boards were mounted on tripods, at a height of one meter, at the center of the track/hallway and were facing each other.

All measurements in this experiment were conducted three times over a time period of 30 seconds each.

4.2 Test Application

In general, the applications used to carry out all experiments in this chapter are based on the *isochronous broadcaster* and *isochronous receiver* samples provided by Zephyr, and are carried out on the 2M physical mode, given that Zephyr only supports the 2M physical mode, in relation to broadcast isochronous streams, at this point in time.

To ensure interoperability with future versions of Zephyr, the time synchronization over broadcast isochronous streams has been implemented on the application layer itself. The ISO receiver captures the current timestamp in the ISO receive callback, computes the time difference between the SDU synchronization reference (contained in the ISO packet), and the current timestamp as well as starts a timer with this time difference to ensure synchronization. It is important that the current timestamp is captured in the same way as the SDU synchronization reference timestamp. This can either be achieved with a free-running timer with a correct offset or by capturing the actual controller timestamp used to calculate the SDU synchronization reference. In the case of the nRF52840 DK, which was used for the experimental campaign, this can be achieved by capturing the value of RTC0.

4.2.1 D-Cube

To utilize all the functionality provided by D-Cube, without significant overhead of serial data capturing, an 8-bit style encoding on 8 GPIO pins has been implemented. These GPIO pins represent the ID of the current packet. Due to the output triggering at the moment of the ISO broadcast and the reception of the packet, accurate latency determination is ensured. This information can be used to calculate the packet reception rate and throughput.

Additionally, it is important to highlight that the receiver power consumption is subject to hardware variability caused by the nodes themselves.

This phenomenon can be seen in Figure 4.3, in which a separate test over ten nodes and ten D-Cube runs with similar parameters has been conducted.

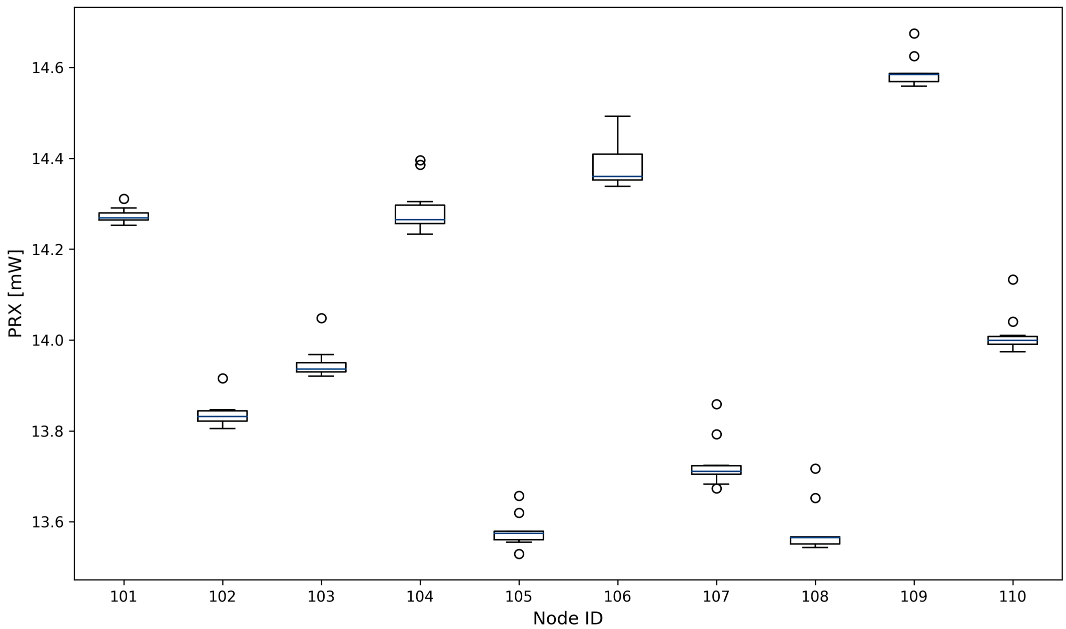

Figure 4.3: Receiver power consumption variability in D-Cube nodes.

4.2.2 Field Test

In contrast to the D-Cube based experiments, the field test experiments require a live packet reception rate calculation on the receiver side. This calculation is achieved via the implementation of a windowed moving average filter with a window size of 250 samples.

4.3 Explored Parameters

In an isochronous broadcast application, one needs to select multiple parameters before the initial establishment of the stream. Typically, the three main parameters in terms of Broadcast Isochronous Streams are the SDU size, retransmission number, and the SDU Interval. Additionally, one of the key parameters in terms of range, is the transmission power, which, therefore, also needs to be considered in the experimental campaign.

4.3.1 SDU Size

This parameter refers to the actual size of an SDU in octets [2]. In the current version of Zephyr, this value can be between 23 and 251 bytes, given the correct corresponding MTU setting.

4.3.2 Retransmission Number

This parameter refers to the "number of times every PDU should be retransmitted, irrespective of which isochronous events the retransmissions occur in" [2] and can be selected in the range from 0 to 30.

It is important to note that this parameter is a "recommendation to the controller which the controller may ignore" [2]. However, in the specific version of Zephyr used throughout the thesis, the retransmission number set by the user equals the retransmission number of the controller, as long as the SDU interval and the maximum transport latency is large enough to fit all retransmissions.

4.3.3 SDU Interval

This parameter refers to the "time interval of the periodic SDUs" [2], can be set to a value between 255 us and 1048575 us, and specifies the time between two consecutive BIG events. This correlation is illustrated in Figure 2.8.

4.3.4 Transmission Power

The transmission power refers to the output power of the radio in dBm, and can be set between -40 dBm and +8 dBm with a granularity of 4 dBm, in the case of the Nordic nRF52840. This parameter is subject to the actual SoC as well as to country-specific limitations. Furthermore, the transmission power in all experiments refers to the total energy consumption of a single nRF52840 DK with the J-Link debugger detached.

4.3.5 Metrics of Interest

It is important to highlight that this thesis focuses on the reliability, robustness and resilience of a broadcast isochronous stream while retaining throughput and minimizing the receiver and transmitter power consumption. To be able to determine these metrics, the following measurement approaches, based on the application layer, were used:

- **Packet Reception Rate (PRR):** Number of received packets divided by the total number of sent packets.
- **Throughput:** Number of received packets multiplied by the packet size over a period of one second.
- **Receiver and Transmitter power consumption:** Total energy consumption provided by D-Cube.
- **Latency:** End-to-end latency between a sent packet and the time of reception.

4.3.6 Default Parameters

The following parameters form the basis for all experiments in Chapter 4 and, unless otherwise mentioned, remain fixed while varying the parameter under study:

- SDU Size: 250 byte
- SDU Interval: 20 ms
- Maximum Transport Latency: 20 ms
- Retransmission Number: 2
- Transmission Power: +8 dBm

4.4 Results D-Cube

4.4.1 Impact of SDU Size

This experiment illustrates the impact of a varying SDU size under the default parameters denoted in Section 4.3.6.

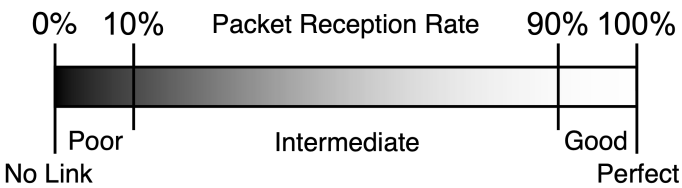

Figure 4.4: Terminology to describe the link quality in terms of PRR [27].

Figure 4.5 illustrates nodes 103 and 212 of the D-Cube infrastructure. Details on the node locations are illustrated in Figure 4.1. In this experiment, node 103 corresponds to a good quality link and node 212 corresponds to an intermediate quality link. The categorization of intermediate quality links and good quality links is illustrated in Figure 4.4.

Interpreting the results, one can denote that both links show similar behavior in terms of the packet reception rate (PRR), throughput and receiver power consumption (PTX), considering the variability as well as in terms of latency.

Additionally, the following findings can be extracted:

Packet Reception Rate (PRR):
The packet reception rate does not show a significant difference over a multitude of different SDU sizes.

Throughput:
The throughput correlates to the SDU size and packet reception rate. The higher the SDU size, the higher the throughput.

Transmitter Power Consumption (PTX):
The transmitter power consumption correlates to the SDU size. The higher the SDU size, the longer it takes to send the packet, thus the higher the transmitter power consumption.

Receiver Power Consumption (PRX):
The receiver power consumption correlates to the SDU size. The higher the SDU size, the longer it takes to receive the packet, thus the higher the receiver power consumption.

Latency:
This plot shows the correctness of the time synchronization mechanism (isochronous channel) implemented in the experimental setup.

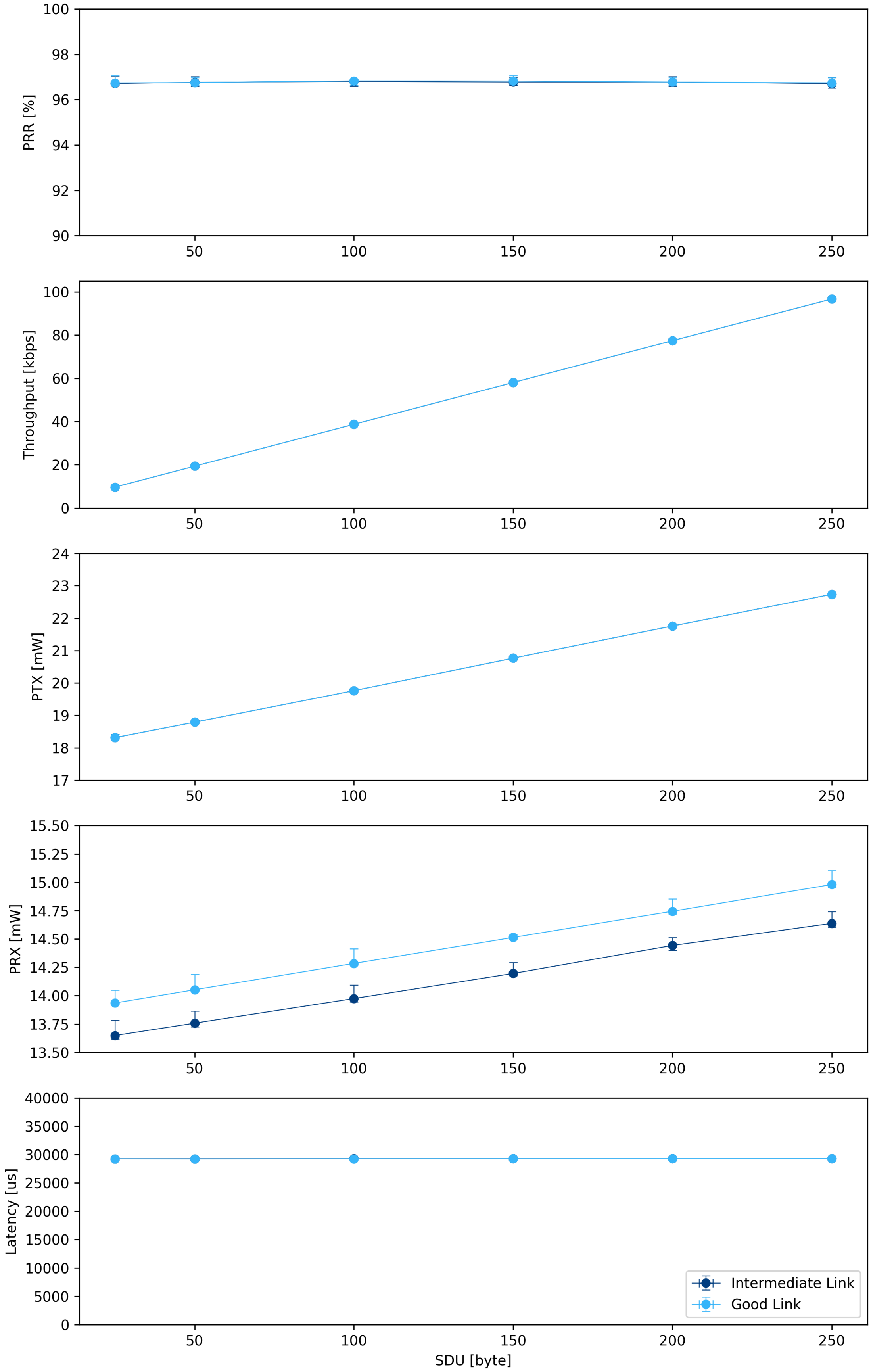

Figure 4.5: Impact of varying SDU size on broadcast isochronous streams. A higher SDU size results in a higher throughput as well as power consumption while the latency remains constant.

4.4.2 Impact of the Retransmission Number

This experiment illustrates the impact of a varying retransmission number.

Interpreting the results of Figure 4.6, which represent nodes 103 and 213 of the D-Cube infrastructure, one can extract the following findings:

Packet Reception Rate (PRR):
The packet reception rate increases with the number of retransmissions in an intermediate quality link while the packet reception rate in a good quality link remains constant.

Throughput:
The throughput correlates to the SDU size and packet reception rate. Therefore, it increases at the same rate as the packet reception rate with the same premise in terms of link quality.

Transmitter Power Consumption (PTX):
The transmitter power consumption, in this particular case, is the result of the SDU size as well as the number of retransmissions including the time period needed in each retransmission. The higher the number of retransmissions, the longer the radio will be active, thus the higher the transmitter power consumption.

Receiver Power Consumption (PRX):
The receiver power consumption is in relation to the packet reception rate as well as the link quality. Ideally, the receiver node can switch off its radio as soon as the first valid packet has arrived. However, if the link quality is not sufficient, the packet reception might need more retransmissions, thus the radio needs to stay activated for a longer period of time, leading to an increased receiver power consumption. Again, it is important to denote the jitter in the receiver power consumption measurements, as mentioned before.

Latency:
This plot shows the relation between the retransmission number and the latency. Given that isochronous channels exchange the number of retransmissions at the time of the establishment, the actual latency is determined by the initial value.

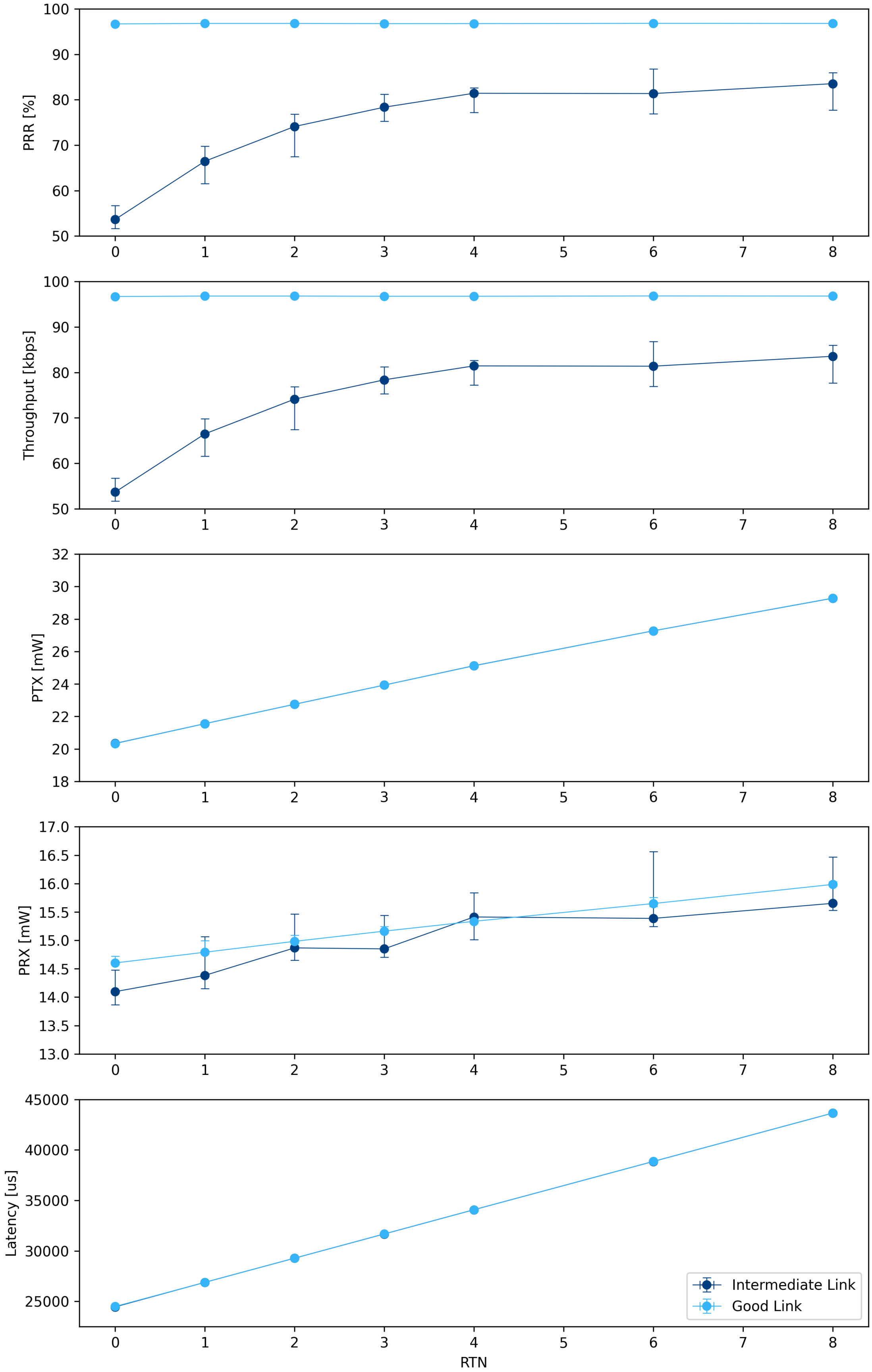

Figure 4.6: Impact of a varying retransmission number. A higher retransmission number can lead to a significant increase of PRR in intermediate quality links at the cost of overall latency.

4.4.3 Impact of SDU Interval

This experiment illustrates the impact of a varying SDU interval.

It is important to highlight that the maximum transport latency in this experiment is set to 10 ms to be able to provide test results that surpass the typical range of an audio application.

Interpreting the results of Figure 4.7, which represent nodes 103 and 212 of the D-Cube infrastructure, one can determine the following outcome:

Packet Reception Rate (PRR):
The packet reception rate decreases with the SDU interval due to limitations in the current version of the Zephyr Controller. Furthermore, one can denote that the packet reception rate of an intermediate quality link is lower than the packet reception rate of a good quality link.

Throughput:
The throughput correlates to the SDU size, packet reception rate, and SDU Interval. Therefore, it decreases with a higher SDU Interval.

Transmitter Power Consumption (PTX):
The transmitter power consumption also correlates to the SDU size and consequently decreases with an increase of the SDU Interval.

Receiver Power Consumption (PRX):
The receiver power consumption correlates to the SDU size and, therefore, decreases with an increase of the SDU Interval.

Latency:
The latency correlates to the SDU interval. A higher SDU interval means that packets are sent less often, therefore, increasing the latency.

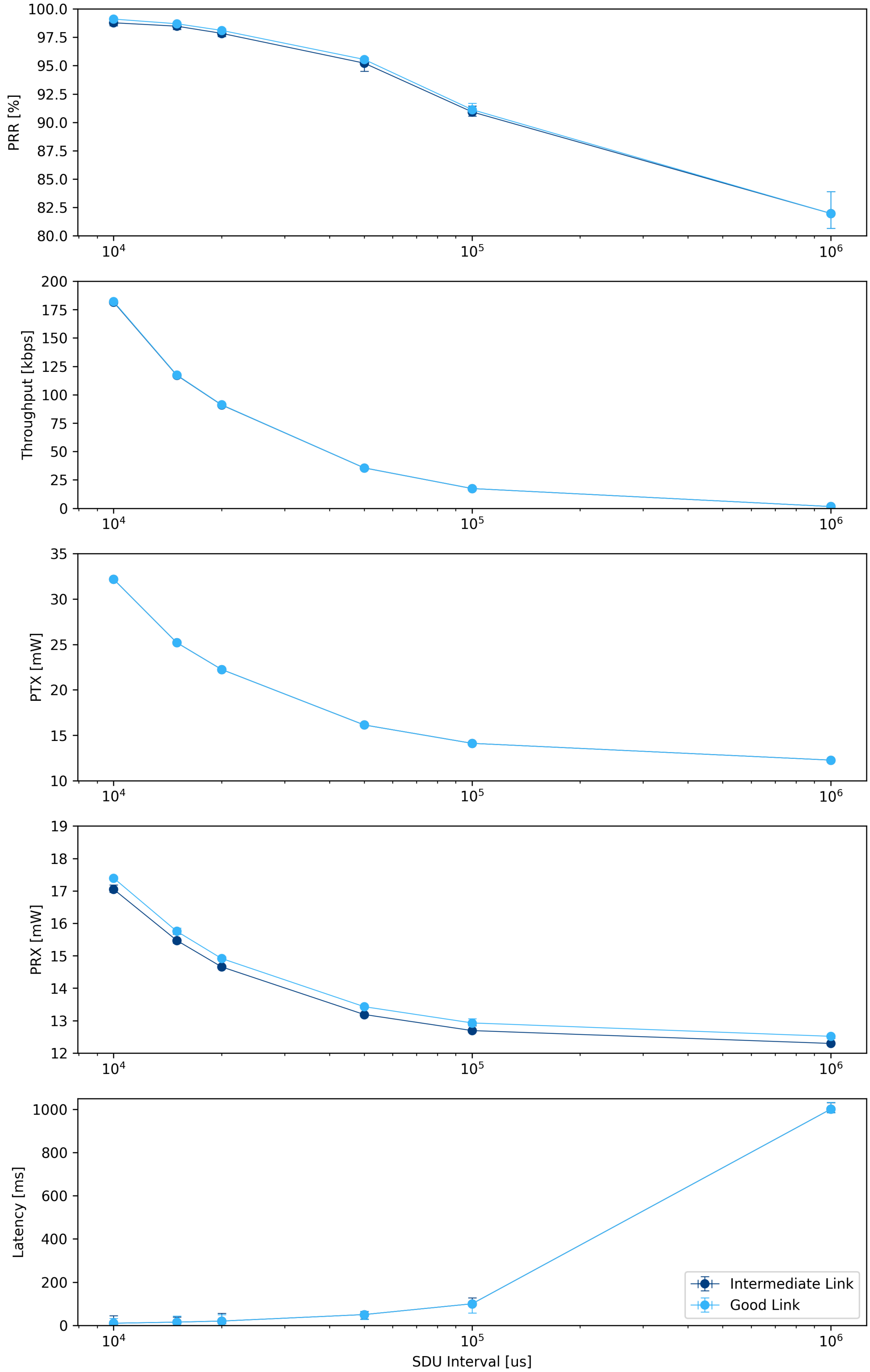

Figure 4.7: Impact of a varying SDU interval. A higher SDU interval correlates to a lower throughput and power consumption but increases the overall latency.

4.4.4 Impact of Transmission Power

This experiment illustrates the impact of a varying transmission power.

Interpreting the results of Figure 4.8, which represent nodes 103 and 212 of the D-Cube infrastructure, one can extract the following findings:

Packet Reception Rate (PRR):
The packet reception rate increases with the transmission power in intermediate quality links while it remains constant in good quality links.

Throughput:
The throughput correlates to the SDU size and packet reception rate. As a result, it increases with a higher SDU Interval.

Transmitter Power Consumption (PTX):
The transmitter power consumption correlates to the SDU size and transmission power. Therefore, it increases with an increase of the transmission power.

Receiver Power Consumption (PRX):
The receiver power consumption correlates to the packet reception rate and link quality. In conclusion, it decreases with an increase of the transmission power assuming that receiver nodes can switch off their radio as soon as the first packet has been received.

Latency:
Under the correct implementation of isochronous channels, the latency is fixed.

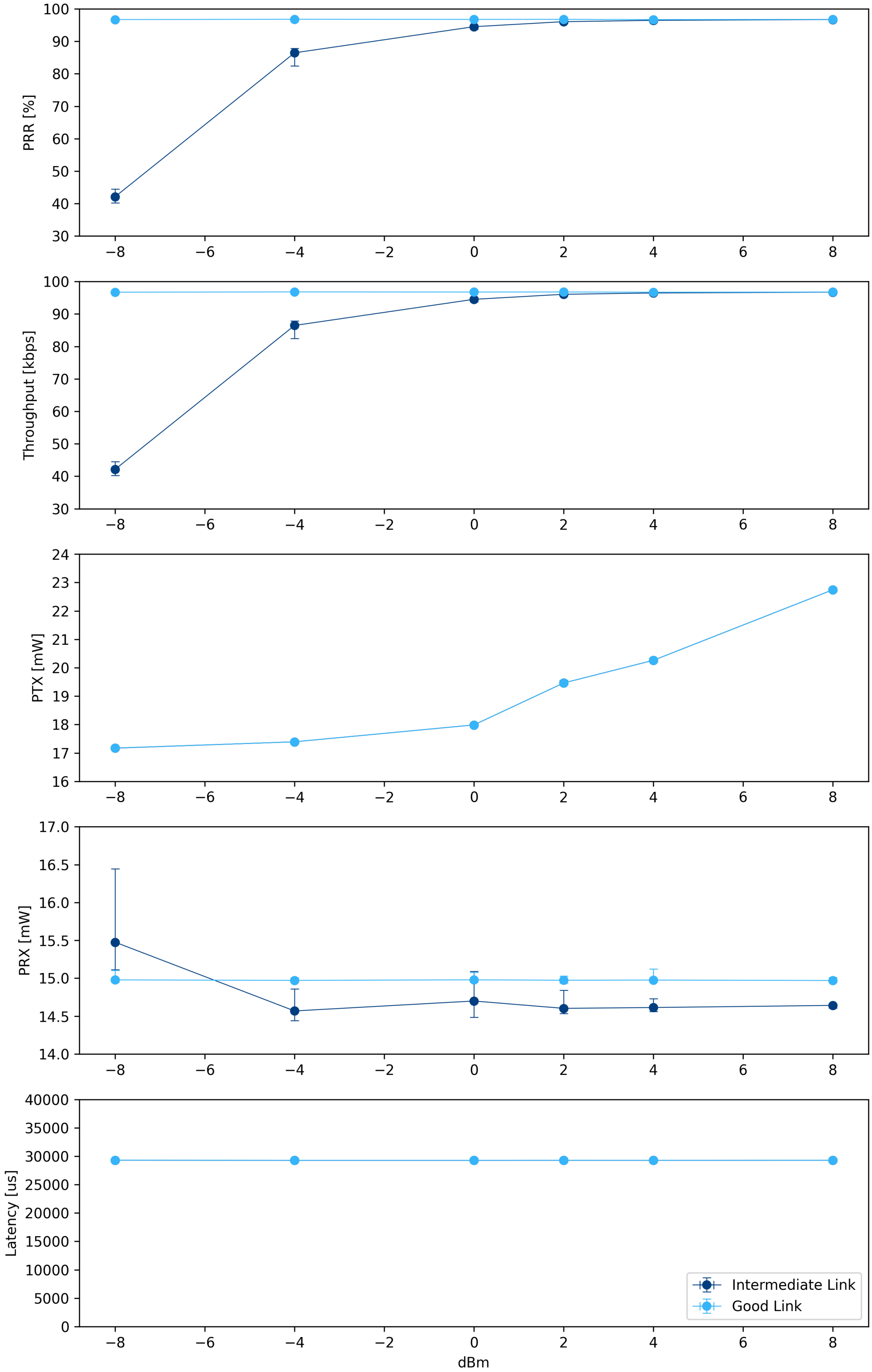

Figure 4.8: Impact of a varying transmission power. A higher transmission power leads to a significant increase of PRR in intermediate quality links at the cost of transmitter power consumption.

4.5 Results Field Test

4.5.1 Impact of Retransmission Number

These experiments illustrate the impact of a varying retransmission number over multiple distances in an outdoor and indoor environment.

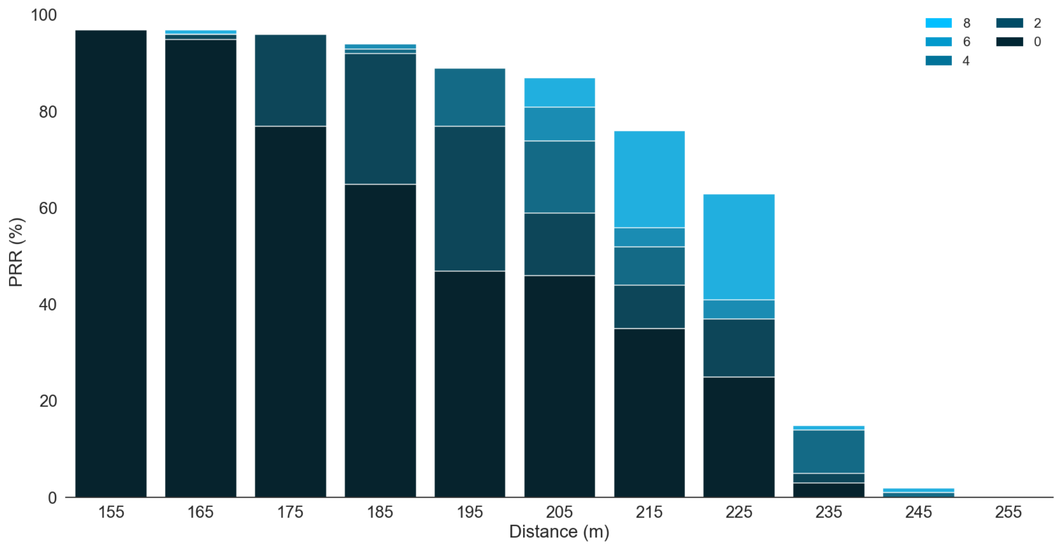

Figure 4.9: Experimental campaign field test result - outdoor retransmission number.

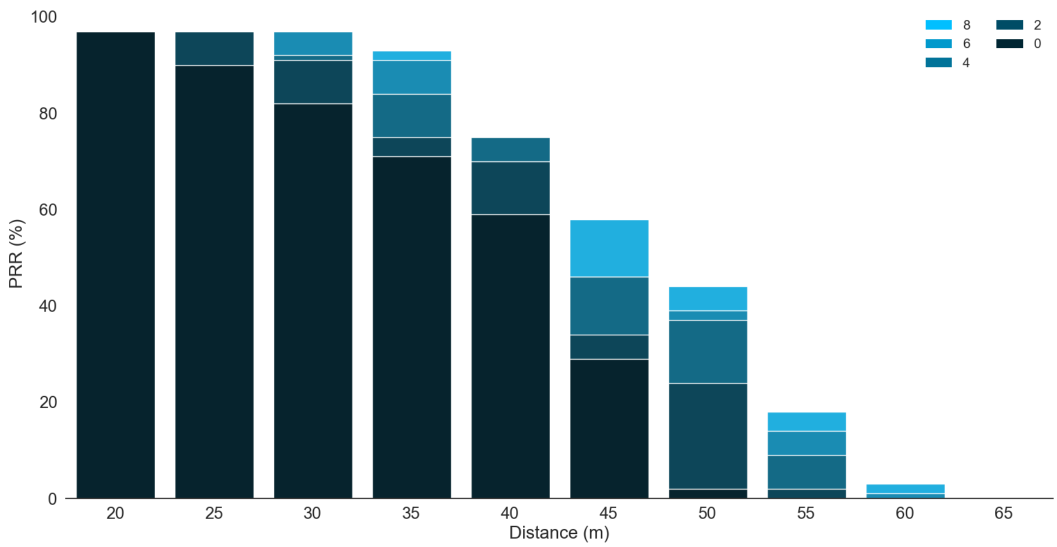

Figure 4.10: Experimental campaign field test result - indoor retransmission number.

It is important to highlight that these tests were conducted at a fixed transmission power setting of 0 dBm.

Interpreting the results of Figure 4.9 as well as Figure 4.10, one can determine that in both types of environment, a higher retransmission number correlates with a higher packet reception ratio over distance. Especially in outdoor environments, a small increase in the retransmission number can lead to a substantial increase in terms of the packet reception ratio, in distances closer to the ISO broadcaster and vice versa in indoor environments. For example, an increase from a retransmission number of 0 to 2 in an outdoor environment is able to extend the usable range, assuming the 6% acceptable packet loss of LC3plus, by more than 10%.

4.5.2 Impact of Transmission Power

These experiments illustrate the impact of a varying transmission power over multiple distances in an outdoor and indoor environment with a fixed retransmission number of two.

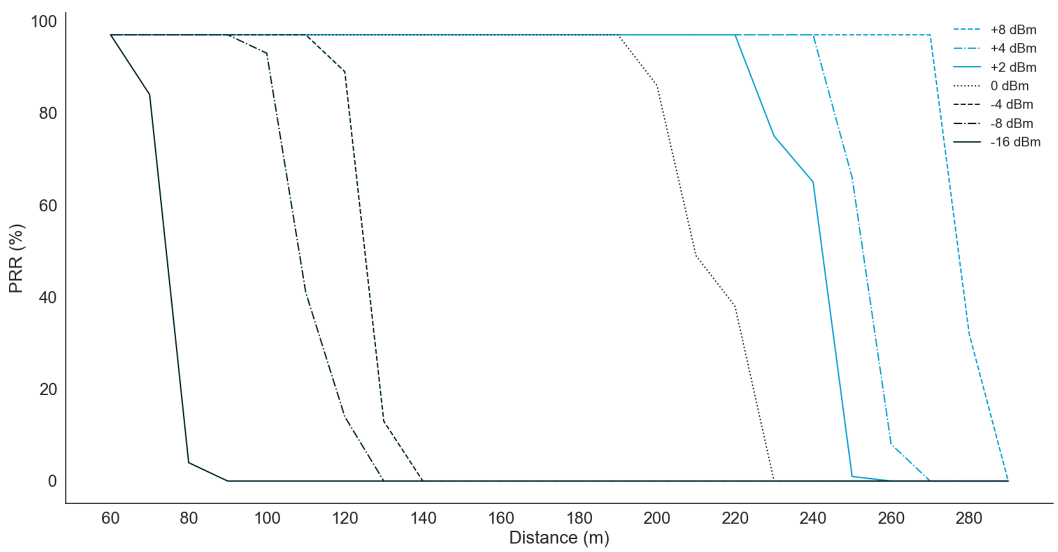

Figure 4.11: Experimental campaign field test result - outdoor transmission power.

Interpreting the results of Figure 4.11 as well as Figure 4.12, it can be seen that in both types of environment, a higher transmission power is related to a higher packet reception rate over distance. Especially in an outdoor environment, a higher transmission power can yield impressive distances while sustaining a good link quality. Moreover, every transmission power increase is able to bring the PRR to 100%, considering the 6% acceptable packet loss of LC3plus, which makes the adjustment of the transmission power more potent than the adjustment of the retransmission number.

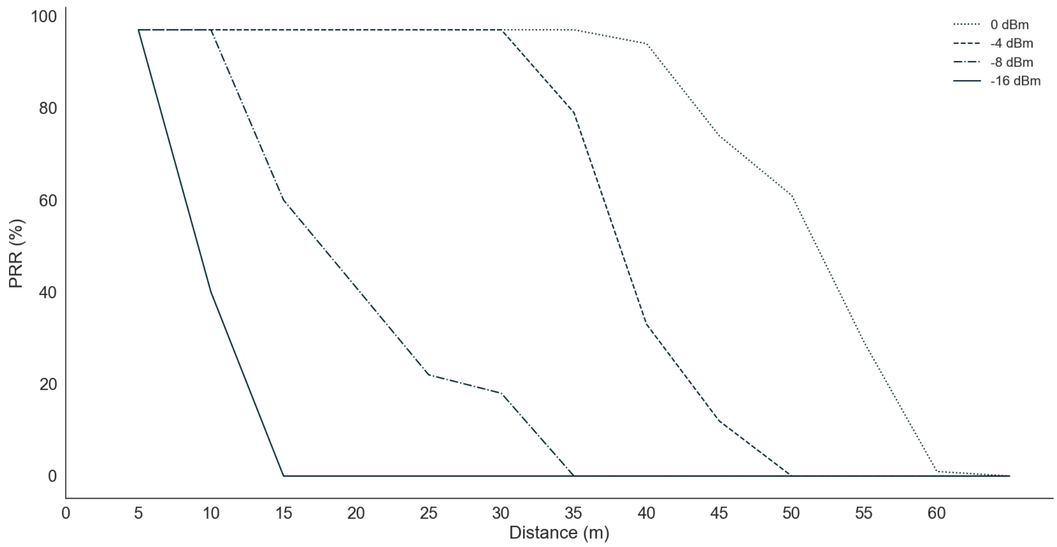

Figure 4.12: Experimental campaign field test result - indoor transmission power.

4.6 Conclusions

Table 4.1 provides a concise overview over all conducted experiments. One can conclude that the SDU size has a linear impact on throughput and power consumption. The retransmission number can significantly increase the PRR in intermediate quality links at the cost of power consumption and overall latency. Furthermore, it can improve the PRR in distances close to the broadcasting device in outdoor environments, while in indoor environments, the retransmission number has linear impact on the PRR. A higher SDU interval decreases the throughput as well as the power consumption while increasing the latency. Additionally, the transmission power can substantially increase the PRR in indoor as well as outdoor environments, at the cost of a rising transmitter power consumption.

Combining the gathered knowledge of Section 4.4 with the findings of Section 4.5, one can determine that the two most influential parameters in terms of broadcast isochronous streams, for this particular use case, are the retransmission number and transmission power, especially given the linear relationship between SDU size and power consumption as well as the fixed SDU interval of a typical audio streaming application.

Explored Parameter	**PRR**	**Throughput**	**Power Consumption**	**Latency**
SDU Size	-	↑	↑	-
Retransmission Number	↑↑	↑↑	↑	↑
SDU Interval	-	↓↓	↓↓	↑↑
Transmission Power	↑↑	↑↑	↑↑	-

Table 4.1: Explored parameter trade-off on intermediate quality links. Green representing a positive impact, red indicating a negative one. Single arrows correspond to a linear trend while double arrows represent a non-linear trend.

To be able to select the optimal parameters for the dynamic adaptation algorithm in Chapter 5, a comparison between the power consumption, PRR, and the metrics of interest has to be carried out. To this extent, one can consider each distance measurement of the experiments in Section 4.5 as an independent link.

Categorizing these links in the same way as in Section 4.4.1, one can derive a multitude of intermediate quality links in regards to the retransmission number as well as the transmission power.

Combined with the data of the corresponding intermediate quality link of the D-Cube experiments, a correlation between the metrics of interest, the transmitter power consumption, measured in the D-Cube experiments, and the increase in PRR can be constructed.

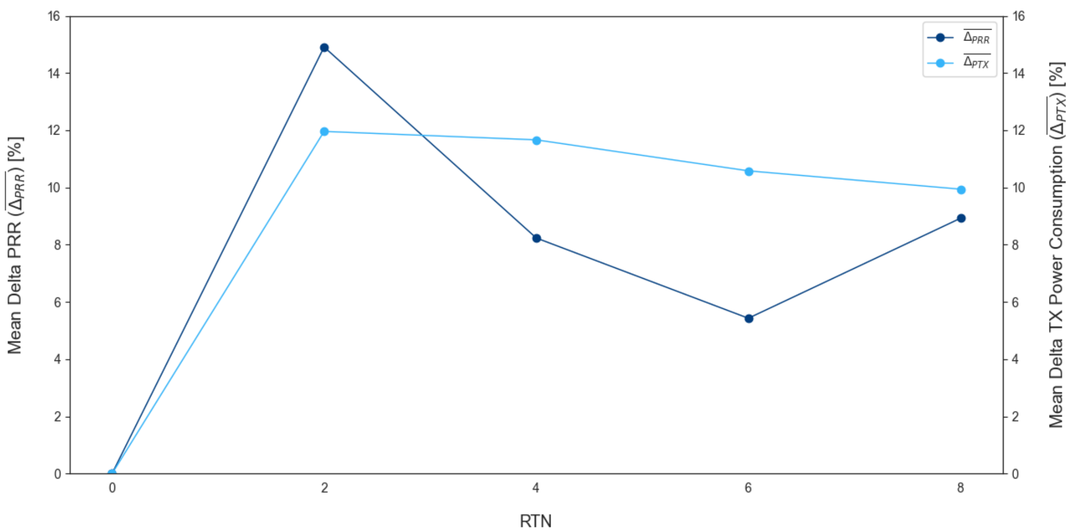

Figure 4.13: Average difference between consecutive retransmission number settings.

Figure 4.13 illustrates such a correlation in terms of the retransmission number. For example, an increase from a retransmission number of 0 to 2 will, on average, increase the PRR by over 14% but also increases the power consumption by more than 11%.

The same logic can be applied in regards to the transmission power, as illustrated in Figure 4.14. For example, an increase from 0 dBm to 2 dBm is expected to yield a PRR increase of over 30 percent at the cost of over 8 percent increase of the transmitter power consumption.

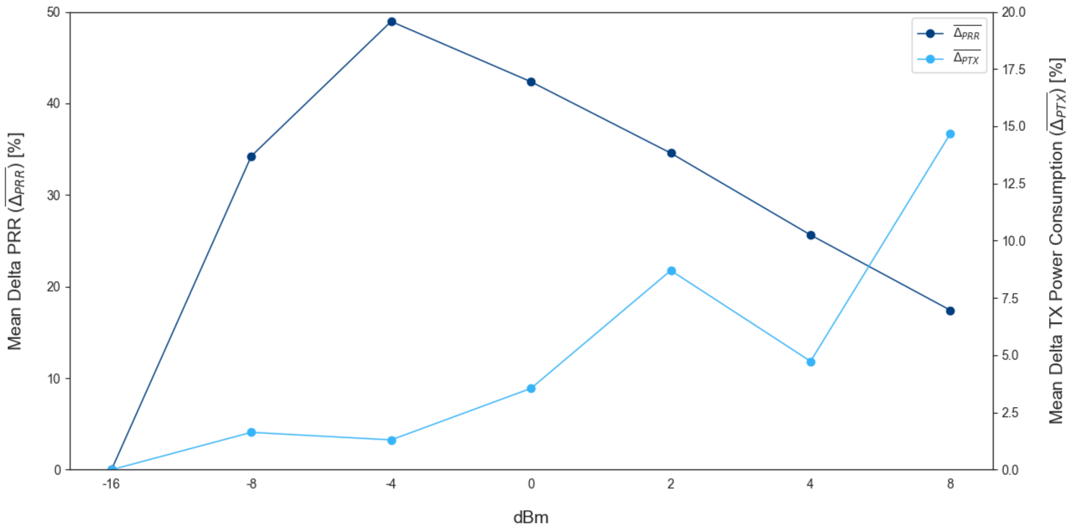

Figure 4.14: Average difference between consecutive transmission power settings.

Furthermore, interpreting Figure 4.13 and 4.14, one can see that the transmission power has a higher impact on the PRR than the retransmission number while consuming less energy at settings below +8 dBm. This impact can also be seen in Figure 4.15.

To further enhance the understanding of retransmission number and transmission power influence, one needs to derive a common baseline. This can be achieved by calculating the power consumption at the most power efficient setting for both metrics. This baseline can then be used to calculate the power consumption per percent of PRR increase for all possible metric combinations.

As illustrated in Figure 4.16, high retransmission numbers in low transmission power settings yield little PRR increase while consuming a lot of power. Therefore, it is advisable to first increase the transmission power and then the retransmission number.

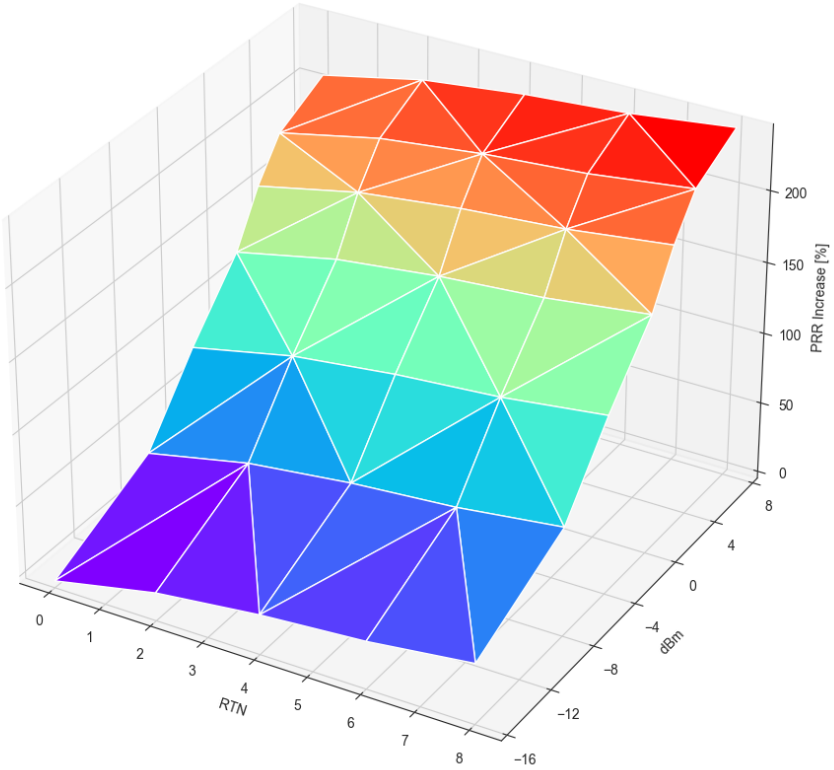

Figure 4.15: Impact of retransmission number and transmission power on the PRR.

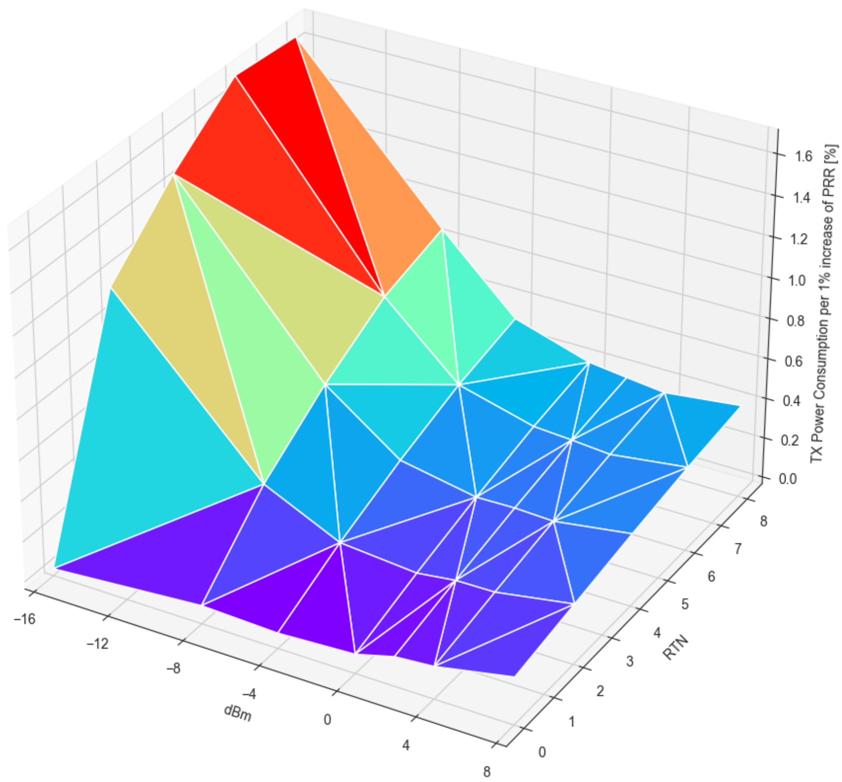

Figure 4.16: Transmitter power consumption per one percent increase of PRR.

5 Adapting Isochronous Parameters

Given the gained knowledge from the previous chapter, one can determine the need for a dynamic parameter adaptation algorithm. This chapter presents such an adaptation algorithm, which selects the optimal isochronous parameters in regards to link quality and throughput while reducing the power consumption as much as possible.

Figure 5.1 illustrates the basic building blocks while Section 5.1 provides insight into the design decisions and Section 5.2 describes the algorithm in further detail.

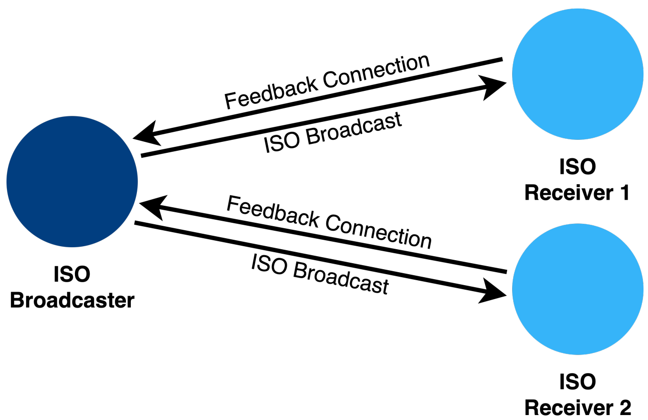

Figure 5.1: Illustration of the isochronous parameter adaptation algorithm architecture.

5.1 Design Rationale

Due to the use-case as a motorcycle intercom system, and the possible LC3plus codec implementation, the adaptation mechanism must be able to sustain a link quality of 94%.

Given the asymmetric nature of broadcast isochronous streams, the broadcasting device has no knowledge about the link quality. To be able to dynamically adapt the two most important isochronous parameters, specified in Section 4.6, a feedback connection is necessary. This connection is used to periodically transmit the PRR, calculated on the ISO receiver, to the ISO broadcaster. It is important to highlight that the algorithm used throughout this thesis is assuming one ISO receiver. In the case of multiple ISO receiver nodes, as illustrated in Figure 5.1, every node must establish a feedback connection and triggering must be based on the worst link, to ensure a correct operation of the dynamic parameter adaptation algorithm.

Most parts of the adaptation algorithm are implemented on the ISO broadcaster to be able to use the RSSI measurement of the feedback connection. This measurement is continuously monitored and in the case of a negative trend, the adaption algorithm adapts its parameters. This mechanism is referred to as proactive triggering. Similarly, if the ISO broadcaster receives a PRR, indicating the loss of multiple packets, the algorithm adapts its parameters accordingly (reactive triggering).

5.2 Adaptation Algorithm

5.2.1 Overview

Algorithm 1 describes a dynamic parameter adaptation algorithm, which uses a combination of a proactive and reactive trigger to be able to determine the next optimal move. This move can either be to increase the link quality, or to save power by selecting an adjacent value from the lookup table, described in Section 5.2.4.

In general, the adaptation algorithm is composed of 4 cases. In the first two cases, a higher lookup table setting is chosen based on a proactive or reactive trigger. These triggers are described in Sections 5.2.2 and 5.2.3. The third case describes an acceptable link quality between two threshold values and the fourth case describes the power optimization mechanism based on the proactive trigger.

It is important to highlight that the PRR is calculated on the ISO receiver and transmitted to the ISO broadcaster via the feedback connection. This design decision offloads the processing overhead caused by the PRR calculation, which can be beneficial for a use-case with multiple simultaneous receiver nodes. Ideally, the calculated PRR value is transmitted to the ISO broadcaster at every connection event. So, the ISO broadcaster can use this value as a heartbeat signal to determine if a receiver device is still in reception range and adjustment is needed or, if the device has terminated the reception of the broadcast isochronous stream.

Variables $status_{PRR}$ and $status_{RSSI}$ of Algorithm 1 describe global variables with values between 0 and M_{PRR} and M_{RSSI}, respectively. M_{PRR}

and M_{RSSI} are used as tuning parameters[1] to adjust the overall reaction speed. This reaction speed is also referred to as *stickyness*, describing the margin for which the algorithm does not adjust the corresponding parameters if the status variable is below a certain number. A low value corresponds to a fast reaction time and a high value to a slow reaction time. These values are application specific and in relation to the connection interval of the feedback connection.

Similarly, θ_L and θ_H are tuning parameters used as threshold values that define an acceptable range of lost packets. If the PRR is between θ_L and θ_H, no actions are taken by the adaptation algorithm.

Algorithm 1: Dynamic Parameter Adaptation Algorithm

Variable: $status_{PRR}$ ← *PRR counter variable*;
Variable: $status_{RSSI}$ ← *RSSI counter variable*;
Input: PRR ← *packet reception rate*;
Input: M_{PRR} ← *PRR margin ("stickyness")*;
Input: M_{RSSI} ← *RSSI margin ("stickyness")*;
Input: θ_L ← *lower PRR threshold*;
Input: θ_H ← *upper PRR threshold*;
Input: θ_{RSSI} ← *RSSI threshold*;

if *isRSSITrendDeclining* ***or*** *(lastNIsoPacketsMissed* ***and*** $status_{RSSI} > M_{RSSI}$*)* **then**
    selectHigherLUTSetting;
    $status_{RSSI} \leftarrow 0$;
else if *PRR* $< \theta_L$ **then**
    selectHigherLUTSetting;
    $status_{PRR} \leftarrow 0$;
else if *PRR* $< \theta_H$ **then**
    do nothing;
else
    if $status_{PRR} < M_{PRR}$ **then**
        $status_{PRR}{+}{+}$;
    else if *rssiAfterDownshift* $< \theta_{RSSI}$ **then**
        selectLowerLUTSetting;
        $status_{PRR} \leftarrow 0$;
$status_{RSSI}{+}{+}$;

[1] Parameters that need to be experimentally determined for the specific application.

Furthermore, θ_{RSSI} is used in combination with the result of Algorithm 2, described in Section 5.2.2, as a threshold for switching to more power saving parameters. θ_{RSSI} is dependent on the radio hardware itself, as well as environmental factors, and should, therefore, be considered a tuning parameter.

5.2.2 Proactive Trigger

To be able to detect a degradation of the link quality before packet loss occurs, or a possible wasteful behavior in terms of power consumption, two additional algorithms are needed.

Algorithm 2 describes one of the two conditions that could trigger the selection of a higher lookup table setting. This algorithm inserts every RSSI reading, measured at the ISO broadcaster via the feedback connection, into a fixed size ring buffer. If all values contained in the buffer are in descending order and the absolute difference between the highest and lowest buffer value is above a certain threshold, the RSSI trend is considered declining.

Algorithm 2: isRSSITrendDeclining

Input: $RSSI \leftarrow$ *RSSI value*;
Input: $\theta_{RT} \leftarrow$ *RSSI trend threshold*;
Variable: $R_{RSSI} \leftarrow$ *ring buffer of size* N_{RSSI}, *pre-filled*;

put(R_{RSSI}, $RSSI$);
for $i \leftarrow 0$ **to** $len(R_{RSSI}) - 1$ **do**
    $r_{curr} \leftarrow peek(R_{RSSI}, i)$;
    $r_{next} \leftarrow peek(R_{RSSI}, i + 1)$;
    if $r_{curr} < r_{next}$ **then**
        return false;
$\Delta_r \leftarrow max(R_{RSSI}) - min(R_{RSSI})$;
if $\Delta_r > \theta_{RT}$ **then**
    return true;
return false;

Furthermore, it is important to highlight that the *peek* function in Algorithm 2 refers to a retrieval of the value inside the buffer at the given index without removing it. Additionally, the RSSI value handed to Algorithm 2 must be filtered to provide resilience against outliers. This filtering is achieved by applying a moving average filter.

The second algorithm in the realm of proactive triggering is Algorithm 3. This algorithm uses the difference between the current transmission power setting and the next lower transmission power setting in the lookup table to calculate the possible future RSSI value. Additionally, this algorithm implements a *MAX* function to prevent an out of bounds access of the lookup table.

Algorithm 3: rssiAfterDownshift

Input: $RSSI \leftarrow$ *RSSI value*;
Input: $i_{LUT} \leftarrow$ *current LUT setting index*;
Variable: $LUT \leftarrow$ *ISO setting lookup table*;

$L_{current} \leftarrow LUT[i_{LUT}]$;
$L_{lower} \leftarrow LUT[MAX(0, i_{LUT} - 1)]$;
return $RSSI + L_{lower}.txp - L_{current}.txp$;

5.2.3 Reactive Trigger

If packet loss has already occurred, reactive triggering is needed to prevent further degradation of the link quality.

Algorithm 4 is one algorithm to fulfill this requirement. This algorithm indicates the loss of the last N packets by iterating over a ring buffer and checking if all last N packets have been lost. This algorithm is implemented on the ISO receiver and every positive result of this algorithm is transmitted to the ISO broadcaster via the feedback connection. Negative outcomes are ignored to ensure maximum power efficiency.

Algorithm 4: lastNIsoPacketsMissed

Input: $N \leftarrow$ *window margin*;
Variable: $R_{PRR} \leftarrow$ *PRR ring buffer, pre-filled with values from 0-1*;

for $i \leftarrow len(R_{PRR}) - N$ **to** $len(R_{PRR})$ **do**
  if $peek(R_{PRR}, i)$!= 0 **then**
    return false;
return true;

Additionally, case 2 of Algorithm 1 can be considered a reactive trigger. If the PRR drops below a certain threshold (θ_L), an immediate action is required to prevent further packet loss.

5.2.4 Parameter Selection

Building on the findings of Section 4.6, one can use all derived, baseline adjusted[2], intermediate quality links to determine the optimal isochronous parameter combinations in order to maximize the PRR and minimize the power consumption. These combinations are a crucial part of this adaptation algorithm and can be used to choose the next adaptation step, to either save power or increase the link quality.

These parameters combinations are derived by sorting the baseline adjusted, intermediate quality links by their power consumption and excluding links, which produce a lower average PRR increase, at a higher power consumption, compared to the last setting. The result of this procedure can be found in Table 5.1 and is referred to as the isochronous parameter lookup table (LUT).

Interpreting Table 5.1, one can determine the expected average PRR and transmitter power consumption increase in percent for all feasible TXP and RTN combinations. These average increases are the relative difference of the current setting compared to the baseline of -16 dBm and a retransmission number of 0.

Index	TXP	RTN	$\overline{\Delta_{PRR}}$ [%]	$\overline{\Delta_{PTX}}$ [%]
1	-16	0	0.0	0.0
2	-8	0	34.24	1.91
3	-4	0	83.18	3.43
4	0	0	125.53	7.58
5	2	0	160.10	17.74
6	4	0	185.72	23.27
7	4	2	200.63	40.07
8	4	4	208.87	56.45
9	8	2	218.04	57.22
10	8	4	226.28	73.60
11	8	8	240.63	102.43

Table 5.1: Isochronous parameter lookup table.

The adaptation algorithm can use this LUT to determine the next step. For example, if the current setting is -4 dBm and 0 retransmissions, the algorithm could determine that a switch to -8 dBm and 0 retransmissions will, on

[2]Links must be adjusted to a common transmission power of -16 dBm.

average, decrease the PRR by 48.94%, but save 1.52% power relative to the baseline setting.

In more general terms, a switch from -16 dBm and 0 retransmissions to 0 dBm and 0 retransmissions will, on average, yield a PRR increase of over 100%. It is important to highlight that the PRR can be at most 100%, despite a higher theoretical number.

To make use of this lookup table in Algorithm 1, two additional algorithms are needed, Algorithm 5 and Algorithm 6. In both algorithms, the current lookup table index is provided as the input, and a valid LUT result is returned. These algorithms can use the lookup table in a sequential fashion due to a PRR increase of at least 6% per setting, which correlates to the maximum acceptable packet loss of LC3plus.

Algorithm 5: selectHigherLUTSetting

Input: i_{LUT} ← *current LUT setting index*;
Variable: LUT ← *ISO setting lookup table*;

return $LUT[MIN(i_{LUT} + 1, len(LUT) - 1)]$;

Algorithm 6: selectLowerLUTSetting

Input: i_{LUT} ← *current LUT setting index*;
Variable: LUT ← *ISO setting lookup table*;

return $LUT[MAX(0, i_{LUT} - 1)]$;

5.3 Limitations

Due to the lack of standardization, the dynamic adaptation of the retransmission number during runtime of a broadcast isochronous stream is not defined. This thesis relies on a custom adaptation of the open-source Zephyr BLE controller, which establishes the BIS with the highest retransmission number inside the lookup table and stops the transmission of broadcast packets if a smaller retransmission number is selected.

6 Evaluation

This chapter presents a series of experiments to evaluate the performance of the dynamic parameter adaptation algorithm, described in Chapter 5, against two naïve parameter adaptation algorithms. Section 6.1 describes the experimental setup and Section 6.2 presents the results as well as conclusions.

6.1 Experimental Setup

6.1.1 Hardware Setup

All experiments in this chapter were conducted on Nordic nRF52840 DK boards. The devices were spaced 2 meters apart on a flat surface with direct line of sight. The device acting as the ISO broadcaster, was modified (PCB antenna removed) and connected to a programmable attenuator (Mini-Circuits RCDAT-8000-90). To be able to accurately determine the current consumption, a Nordic Power Profiler Kit II was used.

6.1.2 Software Setup

Applications

The applications used for the evaluation are based on the *isochronous broadcaster* and *isochronous receiver* samples provided by Zephyr.

To be able to illustrate the possible performance improvements of parameter adaptation algorithms, a baseline application, using static parameters is required. This application will be referred to as *static maximum* and implements the default *isochronous broadcaster* and *isochronous receiver* samples without a feedback connection. Furthermore, the static parameters will be set to a maximum, which translates to a transmission power of +8 dBm and a retransmission number of 8. This setting was chosen to support the same communication range as the parameter adaptation algorithms and to ensure comparability.

As described in Chapter 5, a feedback connection is a prerequisite for all parameter adaptation algorithms. This feedback connection informs the ISO broadcaster about the link quality and is implemented as a BLE connection, in which the ISO receiver acts as a peripheral device and the ISO broadcaster acts as a central device.

Furthermore, as stated in Section 5.3, the open-source Zephyr BLE controller was adapted to allow a dynamic retransmission number adaptation on an existing BIS.

Similar to the experiments conducted in Section 4, the evaluation is carried out on the 2M physical mode, given that Zephyr only supports the 2M PHY, in relation to broadcast isochronous streams, at this point in time.

Algorithms

In general, the evaluation is conducted with three different parameter adaptation algorithms. One dynamic algorithm, described in Section 5.2, and two naïve algorithms. These naïve algorithms are referred to as aggressive and conservative.

In contrast to the dynamic parameter adaptation algorithm, which uses an optimized lookup table, these algorithms adapt the transmission power and retransmission number on the basis of a simplified lookup table, illustrated in Table 6.1. This table is built on the findings of Section 4.6, which concluded that it is advisable to first increase the transmission power and then the retransmission number.

Index	TXP	RTN
1	-16	0
2	-12	0
3	-8	0
4	-4	0
5	0	0
6	0	1
7	2	1
8	2	2
9	3	2
10	3	3

Index	TXP	RTN
11	4	3
12	4	4
13	5	4
14	5	5
15	6	5
16	6	6
17	7	6
18	7	7
19	8	7
20	8	8

Table 6.1: Naïve isochronous parameter lookup table.

Algorithm 7 illustrates the implementation of both naïve algorithms, in which the input variable *Aggressive* allows the selection between an aggressive and conservative behavior. The aggressive algorithm selects the highest lookup table setting as soon as sufficient packet loss is detected, whereas the conservative algorithm increases the LUT setting sequentially.

Algorithm 7: Naïve Parameter Adaptation Algorithm

```
Variable: status_PRR ← PRR counter variable;
Input: PRR ← packet reception rate;
Input: M_PRR ← PRR margin ("stickyness");
Input: θ_L ← lower PRR threshold;
Input: θ_H ← upper PRR threshold;
Input: Aggressive ← toggles the usage of an aggressive approach;

if PRR < θ_L then
    if Aggressive then
        selectHighestLUTSetting;
    else
        selectHigherLUTSetting;
    status_PRR ← 0;
else if PRR < θ_H then
    do nothing;
else
    if status_PRR < M_PRR then
        status_PRR++;
    else
        selectLowerLUTSetting;
        status_PRR ← 0;
```

6.1.3 Emulating a Varying Link Quality

To be able to evaluate the performance of the adaptation algorithms, a simulation analog to a real world motorcycle ride has been created. This simulation assumes a maximum acceleration of 13.9 m/s^2, which translates to a motorcycle accelerating from 0 to 100 kph in 2 seconds.

The simulation is executed over a time period of 180 seconds with random acceleration and deceleration sequences and uses a maximum attenuation of 24 dB, which correlates to the results of Section 4.5. To ensure repeatable results throughout the evaluation, a programmable attenuator, described in Section 6.1.1, was used.

A graphical representation of a random simulation can, for example, be seen in the top plot of Figure 6.2.

6.1.4 Metrics of Interest

It is important to highlight that audio codecs rely on a high link quality to be able to provide the optimal audio quality. Therefore, in the case of LC3plus, a minimum link quality of 94% should be sustained while minimizing the receiver and transmitter current consumption. To be able to determine these metrics, the following measurement approaches were used:

- **Packet Reception Rate (PRR):** Ratio between the number of received and lost packets, calculated on the ISO receiver implementing a moving average filter with a window size of 100 samples.
- **Throughput:** Linear correlation to the PRR. Can be calculated by multiplying the PRR with the maximum theoretical throughput per second.
- **Receiver and Transmitter current consumption:** Average current consumption measured with a Nordic Power Profiler Kit II.

6.1.5 Tuning Parameters

To evaluate the adaptation algorithm, we use the following tuning parameters:

Tuning Parameter	Dynamic Algorithm	Naïve Algorithms
M_{PRR}	10	20
M_{RSSI}	15	-
θ_L	97	98
θ_H	100	100
θ_{RSSI}	80	-

Table 6.2: Adaptation algorithm tuning parameters.

6.1.6 Default Parameters

The following parameters form the basis for all evaluation experiments and remain fixed throughout the evaluation:

- SDU Size: 50 byte
- SDU Interval: 20 ms
- Maximum Transport Latency: 20 ms

6.2 Evaluation Results

The first evaluation was conducted with the goal to illustrate the performance of all adaptation algorithms as well as a static setting in terms of the link quality (PRR).

Therefore, a randomly generated simulation sequence, based on the definitions of Section 6.1.2, was executed 5 times and a cumulative distribution function (CDF) was computed.

As illustrated in Figure 6.1, applications using static parameters, set to the highest possible values, are able to achieve the highest link quality under the proposed evaluation setup. The dynamic parameter adaptation algorithm, introduced in Chapter 5, is able to fulfill the 94% PRR requirement while both naïve algorithms fail to sustain the necessary link quality. It is important to highlight that an aggressive behavior typically achieves the best performance but due to the lack of proactive triggering, the dynamic algorithm can achieve a better result in terms of PRR.

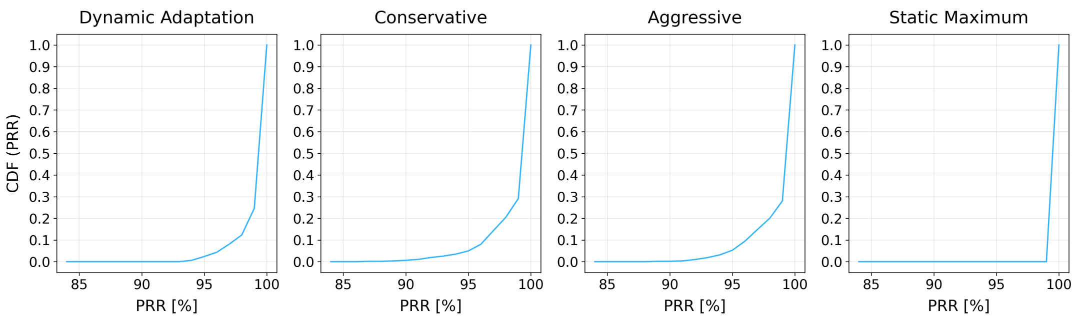

Figure 6.1: Impact of varying attenuation on parameter adaptation algorithms. Neither an aggressive nor conservative adaptation algorithm is able to sustain the required link quality.

To illustrate the general behavior of each adaptation algorithm, three additional plots were generated. Figures 6.2, 6.3, and 6.4 represent a single simulation run with a detailed insight on the selected transmission power and retransmission number.

Figure 6.2 shows the behavior of a typical aggressive adaptation algorithm. Such algorithm reacts to packet loss by the immediate selection of the highest LUT setting. After this setting has been applied, an elevated PRR is detected and the algorithm adjusts the LUT setting to the next lower setting. This procedure will continue until packet loss will be detected again. Therefore, a staircase curve can be observed in both, transmission power and retransmission number. Additionally, such algorithm is not able to sustain the required link quality. If packet loss has already occurred, even a fast adaptation is not sufficient in certain acceleration scenarios, such as between seconds 15 and 20.

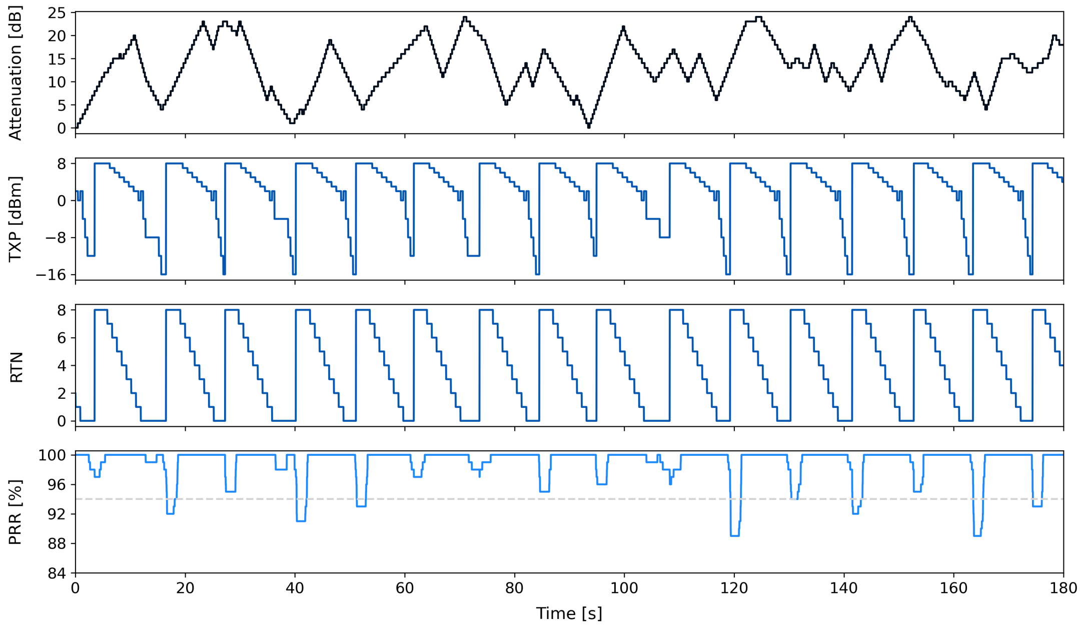

Figure 6.2: Impact of varying attenuation on an aggressive parameter adaptation algorithm.

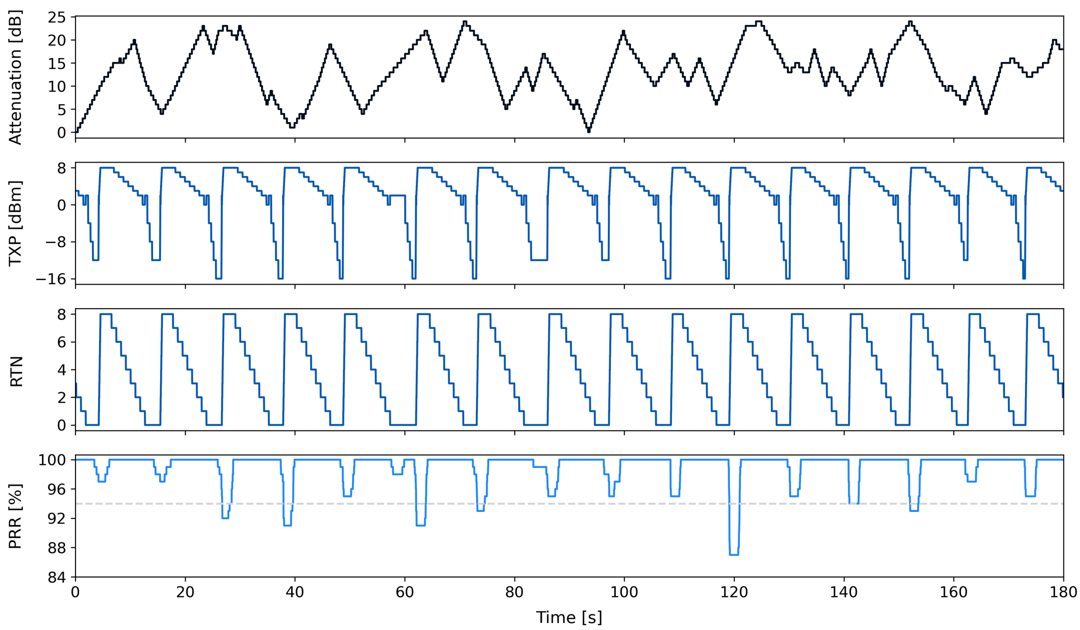

Figure 6.3: Impact of varying attenuation on a conservative parameter adaptation algorithm.

Similarly, a conservative adaptation algorithm also reacts to packet loss with the selection of a different LUT setting. In contrast to an aggressive algorithm, a conservative algorithm selects the next higher LUT setting instead of the highest LUT setting. Figure 6.3 illustrates the behavior of such conservative parameter adaptation algorithm on the simulation sequence.

Furthermore, the same simulation has been executed with the dynamic parameter adaptation algorithm. Figure 6.4 shows the behavior of such algorithm, which selects the best possible LUT setting depending on a proactive and reactive trigger. It is important to highlight that this algorithm is able to sustain the required link quality regardless of the acceleration or deceleration sequence, in the realm of the simulation.

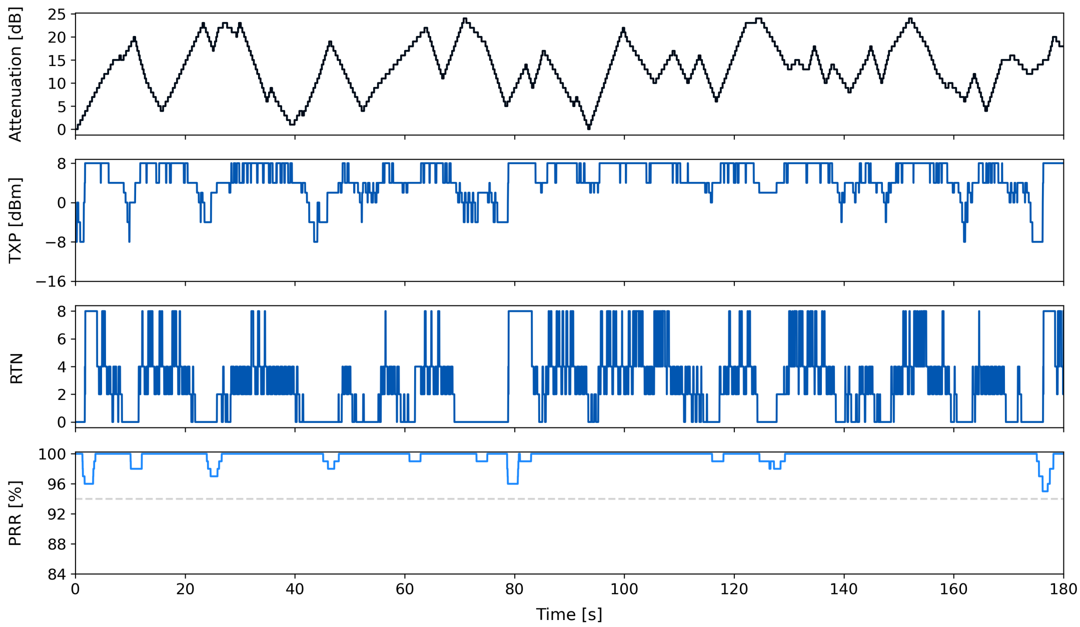

Figure 6.4: Impact of varying attenuation on a dynamic parameter adaptation algorithm.

Additionally, Figure 6.5 illustrates a comparison of all three algorithms over a subsection of the simulation sequence. This figure particularly highlights the difference between the reaction time of the aggressive and conservative algorithm. A conservative behavior increases the lookup table setting in a sequential way whereas an aggressive behavior simply selects the highest LUT setting as soon as the adaptation mechanism is triggered. Moreover, the behavior of the dynamic algorithm can be observed, which selects the optimal lookup table setting depending on a proactive and reactive trigger. It is important to highlight that the *static maximum* application always achieves a PRR of 100% in this simulation setup and, therefore, is not included in this comparison plot.

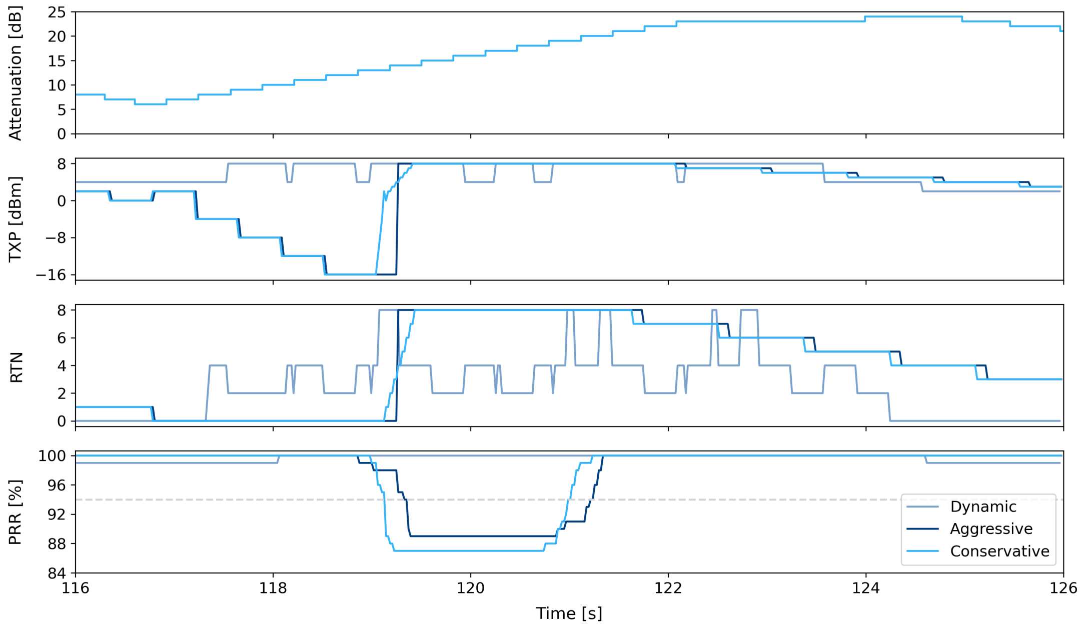

Figure 6.5: Combined excerpt of Figures 6.2, 6.3 and 6.4. Neither an aggressive nor conservative increase of the transmission power and retransmission number is able to sustain a link quality of 94%.

Finally, the second part of the evaluation aims to show the differences between the current consumption of all algorithms as well as the static maximum.

To achieve this goal, each algorithm was executed 5 times with the same simulation sequence over the same time period as in the previous evaluation experiments.

Figure 6.6 shows that the dynamic parameter adaptation algorithm has the lowest average transmitter current consumption, followed by the aggressive and conservative algorithms. The static setting had the highest transmitter current consumption while minimizing the receiver current consumption. All three adaptation algorithms had an increased receiver current consumption due to the overhead of the feedback connection.

Given this particular use case, where devices, on average, are expected to transmit and receive equal amounts of time, the savings in terms of transmitter current consumption of up to 49% outweigh the approximate 30% increase of receiver current consumption.

Therefore, the dynamic parameter adaptation algorithm proves to be a viable option for motorcycle intercom systems, retaining a high link quality and throughput while providing a significant reduction of the current consumption.

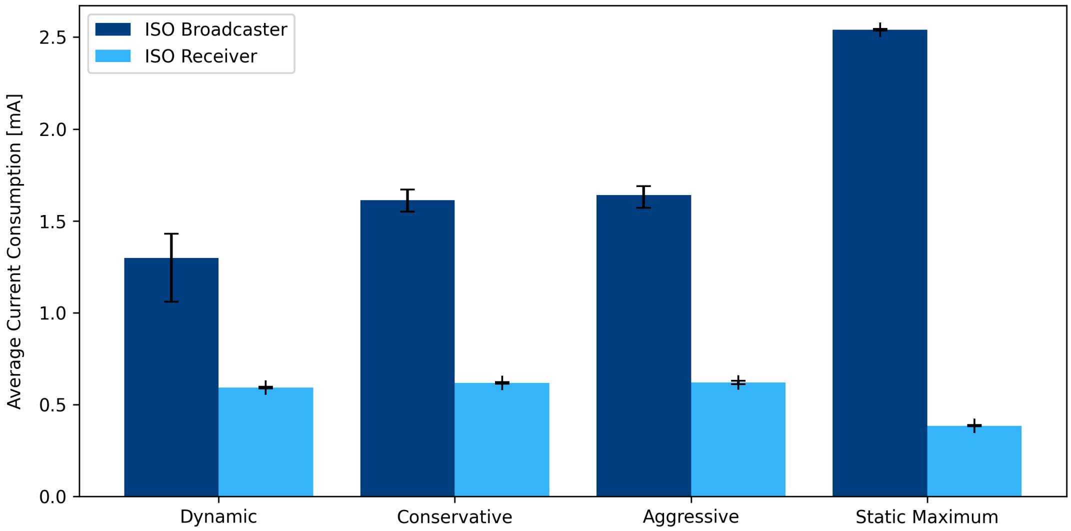

Figure 6.6: Impact of different parameter adaptation algorithms on current consumption. The proposed dynamic parameter adaptation algorithm outperforms other approaches in terms of transmitter current consumption. A static parameter setting is able to achieve the lowest receiver current consumption.

7 Conclusions & Future Work

7.1 Conclusions

Although intercom systems have become an indispensable tool for motorcycle riders, the lack of reliability as well as the unsatisfactory audio quality leaves room for improvement. Bluetooth Low Energy, especially the advances in Bluetooth 5.2, could be the solution motorcycle riders have been waiting for.

Building on top of this idea, this thesis investigated the influence of multiple isochronous parameters on broadcast isochronous streams. Especially the impact of the retransmission number and the transmission power has been studied in detail. Multiple D-Cube based experiments as well as field tests concluded that it is advisable to first increase the transmission power and then the retransmission number in order to sustain a certain throughput and link quality while reducing the power consumption as much as possible.

Using this knowledge, a dynamic parameter adaptation algorithm was created, which dynamically selects the optimal retransmission number as well as transmission power. This algorithm ensures that the communication between two devices is always at an optimal level in regards to link quality and throughput while reducing the current consumption.

Finally, an experimental evaluation was conducted to highlight the improvements introduced by a dynamic parameter adaptation algorithm against two naïve adaptation algorithms as well as a static setting. This evaluation showed that a dynamic algorithm can outperform both naïve algorithms in terms of link quality as well as transmitter current consumption. Therefore, it is a viable option for power-constrained devices such as motorcycle intercom systems.

7.2 Future Work

Dynamic parameter adaptation algorithm. Given the use case of a motorcycle intercom system, a real world test with audio data on a moving vehicle is required to validate the effectiveness of the proposed dynamic parameter adaptation algorithm.

Scalability. Although the dynamic parameter adaptation algorithm outperformed the naïve algorithms, further research is required to support multiple receiver nodes. Especially the selection process in regards to the worst link and the feasibility of a parameter adaptation based on one particular link has to be investigated.

Tradeoff between responsiveness and current consumption. Selection of a lower connection interval for the feedback connection will reduce the current consumption even further but reduces the responsiveness of the proactive trigger. Further experiments are required to determine the optimal ratio between connection interval and responsiveness.

Performance under interference. Given that this thesis does not account nor test for interference, further research is required to enhance our understanding in this regard.

Multiple BLE controller versions. Since the open-source BLE controller provided by Zephyr is under constant development, different controller versions have been used over the course of the thesis. This irregularity was caused by the implementation of multiple features, necessary for the adaptation algorithms. Therefore, it is advisable to rerun all D-Cube experiments with a new BLE controller implementation.

Bibliography

[1] Mohammad Afaneh. *Intro to Bluetooth Low Energy*. Novel Bits, 2018.

[2] *Bluetooth Core Specification*. v5.3. Bluetooth Special Interest Group. July 2021.

[3] Yifeng Cao et al. "Mrs. Z: Improving ZigBee throughput via multi-rate transmission." *Proceedings of the IEEE* 25^{th} *International Conference on Network Protocols (ICNP)*. Oct. 2017, pp. 1–10.

[4] OpenStreetMap contributors. *Map retrieved from www.openstreetmap.org*. 2022.

[5] Andreas Denner. *Communication Systems, connecting at last with (almost) all of them.* June 2022. URL: https://motorcycles.news/en/communication-systems-connecting-at-last-with-almost-all-of-them/.

[6] *Digital Enhanced Cordless Telecommunications (DECT); Low Complexity Communication Codec plus (LC3plus)*. ETSI TS 103 634 V1.2.1. ETSI. Oct. 2020.

[7] Garmin Canada Inc. *ANT / ANT+ DEFINED*. https://www.thisisant.com/developer/ant-plus/ant-antplus-defined. 2022.

[8] Maurizio Gentili, Roberto Sannino, and Matteo Petracca. "BlueVoice: Voice Communications over Bluetooth Low Energy in the Internet of Things scenario." *Computer Communications*, vol. 89-90 (Apr. 2016).

[9] Henry Anfang. *Bluetooth PHY – How it Works and How to Leverage it*. https://punchthrough.com/crash-course-in-2m-bluetooth-low-energy-phy. 2019.

[10] Nick Hunn. *Introducing Bluetooth LE Audio*. Nick Hunn, 2022. ISBN: 979-8-72-723725-0.

[11] Texas Instruments. *Voice over BLE*. 2016. URL: https://software-dl.ti.com/lprf/sdg-latest/html/voice/ble_voice.html.

[12] *Introduction to Bluetooth Classic*. Apr. 2019. URL: https://www.argenox.com/library/bluetooth-classic/introduction-to-bluetooth-classic/.

[13] Jaeho Lee. "Broadcast Audio Transmission for Bluetooth LE on an Interfered ISM Band." *IEEE Internet of Things Journal*, vol. 6, issue 4 (Oct. 2018).

[14] Taeseop Lee et al. "CABLE: Connection Interval Adaptation for BLE in Dynamic Wireless Environments." *Proceedings of the* 14^{th} *Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. June 2017.

[15] Shan Lin et al. “ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks.” *ACM Transactions on Sensor Networks*, vol. 12 (Jan. 2015).

[16] *Low Complexity Communication Codec*. v1.0. Bluetooth Special Interest Group. Sept. 2020.

[17] Bowen Marshall. “Advances in technology offer promise of an expanding role for telecoils.” *The Hearing Journal*, vol. 55, issue 9 (2002), pp. 40–41.

[18] *Mean opinion score interpretation and reporting*. ITU-T P.800.2. International Telecommunication Union. July 2016.

[19] *nRF52840 DK User Guide*. v1.4.1. Nordic Semiconductor. Apr. 2020.

[20] Eunjeong Park, Myung-Sup Lee, and Saewoong Bahk. “AdaptaBLE: Data Rate and Transmission Power Adaptation for Bluetooth Low Energy.” *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*. Dec. 2019, pp. 1–6.

[21] Zephyr Project. *About the Zephyr Project*. June 2022. URL: `https://zephyrproject.org/learn-about/`.

[22] Markus Schnell et al. “lc3 and lc3plus: the new audio transmission standards for wireless communication.” *Journal of the audio engineering society* (May 2021).

[23] Markus Schuß et al. “A Competition to Push the Dependability of Low-Power Wireless Protocols to the Edge.” *Proceedings of the 14^{th} International Conference on Embedded Wireless Systems and Networks (EWSN)*. Feb. 2017, pp. 54–65.

[24] Nordic Semiconductors. *nRF5340 Audio*. 2022. URL: `https://developer.nordicsemi.com/nRF_Connect_SDK/doc/latest/nrf/applications/nrf5340_audio/README.html`.

[25] Dongjin Son, Bhaskar Krishnamachari, and John Heidemann. “Experimental Study of the Effects of Tx Power Control and Blacklisting in Wireless Sensor Networks.” *Proceedings of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*. Nov. 2004, pp. 289–298.

[26] Michael Spörk, Carlo Alberto Boano, and Kay Römer. “Improving the Timeliness of Bluetooth Low Energy in Noisy RF Environments.” *Proceedings of the 16^{th} International Conference on Embedded Wireless Systems and Networks*. 2019, pp. 23–34.

[27] Kannan Srinivasan et al. “The β-factor: Measuring wireless link burstiness.” *Proceedings of the 6^{th} International Conference on Embedded Networked Sensor Systems (SenSys)*. Jan. 2008, pp. 29–42.

[28] Henning Trsek et al. “Towards an Isochronous Wireless Communication System for Industrial Automation.” *Proceedings of the 18th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Sept. 2013.

[29] International Telecommunication Union. *Method for the subjective assessment of intermediate quality level of audio systems*. 2015.

[30] Neil Zeghidour et al. *SoundStream: An End-to-End Neural Audio Codec*. 2021. arXiv: `2107.03312 [cs.SD]`.

[31] *Zephyr Project Documentation*. Release 3.1.0-0. Zephyr Project. June 2022.