Master's Thesis

Computer Science

Reliable Data Streams in the Presence of External Interference

by

Florian Meyer

ORCID iD: 0000-0002-0901-9408

February 2018

Supervised by

Florian Kauer
Institute of Telematics, Hamburg University of Technology

First Examiner	Prof. Dr. Volker Turau Institute of Telematics Hamburg University of Technology
Second Examiner	Prof. Dr. Timm-Giel Institute of Communication Networks Hamburg University of Technology

Please cite by using the DOI: 10.15480/882.1667

Institute of Telematics
Hamburg University of Technology
TUHH

Declaration by Candidate

I, FLORIAN MEYER (student of Computer Science at Hamburg University of Technology), hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been acknowledged.

Hamburg, February 22nd, 2018

Florian Meyer

Abstract

Wireless mesh networks, as a tool for the wireless communication in large industrial applications, gained more and more popularity over the last few years. Thereby, these applications introduce new demands on the wireless solutions regarding their reliability and predictability in the presence of external interference, especially in the controlling and monitoring of safety critical systems. Additionally, many applications require a loss-free data transmission, in the sense that every packet sent at the application layer will eventually arrive at the receiver. This can mainly be achieved by retransmitting packets until they are successfully delivered and lowering the throughput of the application, so that no packet loss occurs due to queue drops.

In the course of this thesis, two popular protocols for the application in wireless mesh networks and industrial applications are examined regarding their performance in the presence of external interference, namely Bluetooth Low Energy and IEEE 802.15.4 DSME. For these, a theoretical model is developed, which allows the calculation of the maximum transmission rate of an application in a multi-hop environment, so that a loss-free transmission is guaranteed. Furthermore, the model is evaluated with respect to different metrics and parameters and a comparison of the performance of the two protocols in the presence of external interference is provided. As a second contribution, a series of hardware experiments with and without artificially generated interference is conducted to assess the influence of interference under realistic conditions. The results are compared to the model and show good conformance. Finally, two mechanisms on top of the Bluetooth Low Energy application layer are introduced that significantly increase its reliability and performance in the presence of external interference.

Table of Contents

Chapter 1

Introduction

With the advances in the research of wireless communication technology and, consequently, the evolution of wireless networks, those have become an ever-increasing part of everyday life. Thereby, their use extends far beyond traditional wireless networks like IEEE 802.11 Wi-Fi or cell phone networks, which are used for the rapid transmission of large amounts of data. The advent of the Internet of Things (IoT) in normal households, for example in home automation or even in body area networks (BANs) with fitness trackers or sensors for medical purposes, shifts the demands on these technologies towards more flexibility, connectivity, security and energy efficiency. Bluetooth Low Energy (BLE) is a communication technology that fulfills many of these demands out of the box. Therefore, it has become a driving force in the development of the IoT with its application in many consumer level products [HOR+17].

But not only the consumer market benefits from the advances in the wireless communication technology. Under the term Industry 4.0, a lot of effort is currently done to automate processes in industrial applications, which requires the interconnection and exchange of data between different parts of industrial plants. Thereby, traditional cable-bound networks are often replaced by wireless solutions, with the advantage of a significantly reduced setup cost and complexity [CM11]. On the other hand, the wireless medium is prone to external interference and therefore reduces the reliability and predictability of the transmission. This effect is yet reinforced by traditional channel access methods like CSMA/CA, which guarantee a high reliability in small networks, but fail in dense industrial networks with hundreds of nodes. Therefore, more advanced coordination mechanisms are needed, which can be found in popular wireless sensor network (WSN) protocols like Bluetooth or WirelessHART. Additionally, the IEEE 802.15.4e standard offers MAC layer protocols for this purpose, which utilize different time-division multiple access (TDMA) and frequency-division multiple access (FDMA) schemes for channel access. One notable candidate is the IEEE 802.15.4 Deterministic and Synchronous Multi-Channel Extension (DSME), which utilizes a TDMA and FDMA scheme that divides time and frequency in different slots. Those offer exclusive channel access to

Figure 1.1.: Exemplary wireless sensor network for the controlling and monitoring of a power plant.[1]

specific nodes and therefore enable the reliable communication in large industrial applications.

Many WSN protocols, including DSME, promise a guaranteed packet delivery for most network configurations, which, however, greatly varies depending on the network's size and topology and the amount of external interference. In the effort to replace wired connections, reliability is an essential requirement. This can even go to the extreme case where an industrial application cannot tolerate any packet loss at all, which is usually the case for safety critical applications like chemical plants or power plants, as exemplarily shown in Figure 1.1. Packet loss on a single wireless link can in principle be prevented by resending the lost packets until they finally arrive. Thereby, a successful transmission is recognized by receiving an associated acknowledgement. However, this introduces new problems by lowering the possible throughput on that link. This can cause problems if the other nodes in the network do not adapt to the lowered throughput and keep sending data at the same rate. In this case, the receive and transmit buffer could overflow and again cause packet loss.

Therefore, it essential to know how much data can be sent in a network without causing any packet loss, when considering external interference. This thesis provides attempts to answer this question using two popular protocols that already support many reliability mechanisms, namely Bluetooth Low Energy and DSME.

[1] Original image from www.flickr.com/photos/20119750@N00/5070115067 (licensed under CC BY 2.0) - Author: Shubert Ciencia

1.1. Goals and Requirements

The goal of this thesis is the evaluation of Bluetooth Low Energy and IEEE 802.15.4 DSME in the presence of external interference. These protocols are common solutions for the reliable data transmission in wireless mesh networks, with BLE focusing more on consumer products and IEEE 802.15.4 DSME concentrating on industrial applications. Bluetooth Low Energy, however, is also becoming increasingly popular in industrial environments. Therefore, the objectives of this work can be further subdivided as follows:

- Development of a theoretical model for the maximum amount of packets an application can send, using Bluetooth Low Energy, so that no packets are lost at all.
- Conducting of hardware experiments to assess the performance of Bluetooth Low Energy with regard to a loss-free data transmission in the presence of external interference.
- Comparison of Bluetooth Low Energy and IEEE 802.15.4 DSME, regarding the performance of their interference mitigation techniques in the presence of external interference.

In the course of this work loss-free data transmission is defined as: Every packet that is handed from the application layer to the lower layers of the stack is eventually delivered successfully. This is mainly done, by resending packets until they are successfully delivered and lowering the throughput of the application until no more drops occur due to finite buffers. Of course, in a real application, other influences might lead to data loss, such as nodes that run out of energy or losses due to hard- or software failures. These effects are hard to predict and not considered in this thesis.

1.2. Outline

Following on from this, Chapter 2 gives a general overview about the state of the art of interference mitigation in wireless sensor networks. Therefore, different sources of interference are first classified, before mechanisms for mitigation are briefly discussed. Additionally, the prevalent communication protocols in wireless sensor networks are described and a more detailed insight into the Bluetooth Low Energy and IEEE 802.15.4 DSME protocols is provided as a preparation for the analytical model, which is described in Chapter 3. The last chapter covers the experimental evaluation of the two protocols. Thereby, a focus is put on the performance of both protocols with respect to an increasing amount of external interference. Furthermore, some techniques to mitigate the influence of external interference in Bluetooth Low Energy are implemented and evaluated. The thesis is finally concluded in Chapter 5.

Chapter 2

State of the Art

Especially in industrial applications like process automation and monitoring, communication over long distances is often required to reach remote devices. Therefore, wireless mesh networks are a common choice of technology, since they do not require any additional infrastructure and enable the transmission of messages over multiple hops. Additionally, they usually provide self-configuration and a high network capacity, simplifying the deployment and maintenance of large networks. However, one problem that quickly arises is the influence of external interference on critical links. This can also occur in the form of self-interference, due to a high density of the network, or mutual interference between different protocols, because many of them utilize the same unlicensed 2.4 GHz frequency band. For this reason, wireless mesh protocols have to comprise mechanisms to mitigate the influence of external interference, while still maintaining a low power consumption. Although similar results can be achieved with many protocol, e.g. the IEEE 802.11 Wi-Fi standard, dedicated protocols like Bluetooth and IEEE 802.15.4 significantly reduce the setup-cost and energy consumption of these networks. Additionally, they usually achieve notable reliability increases by providing custom physical and medium access control layers that are adapted to the demands of the particular wireless mesh networks.

The following sections describe existing approaches to common challenges in wireless mesh networks with respect to the influence of external interference. Therefore, Section 2.1 first classifies the different types of interference in wireless mesh networks, while the following section describes the concepts of commonly used mechanisms to mitigate their influence on the transmission. After that, Section 2.3 to 2.7 provide an overview of popular protocols for industrial applications and wireless sensor networks. These include Bluetooth in two flavors and IEEE 802.15.4 based protocols. At last, an overview of research that is related to this work is given.

2.1. Types of Wireless Interference

Wireless interference refers to the "fading or disturbance of received radio signals caused by unwanted signals from other sources"[1]. Thereby, it can be distinguished between internal and external interference. Internal interference is caused inside a network, for example when multiple nodes simultaneously try to access the same physical channel. External interference, on the other hand, can have various reasons. For example it can be caused by normal household items like microwaves or other unshielded electrical equipment [SA05]. Another source of external interference are jamming transmitters that are not part of the network. In [PYP06], this kind of interference is further separated into *frequency-static interference* and *frequency-dynamic interference*. Frequency-static interference occurs on a fixed frequency range and is considerably longer than a packet duration, which makes it easy to mitigate or avoid. This is not the case for frequency dynamic interference, since it is caused by a network external transmitter that periodically switches frequencies and only sends small amounts of data.

The amount of interference on a wireless link is usually specified as a *signal-to-interference-plus-noise ratio* (SINR) or simply as a *signal-to-noise ration* (SNR), if the interference can be classified as noise. Additionally, metrics like the *packet reception rate* (PRR), *bit error rate* (BER) or deviation of a measured throughput from the nominal throughput are used as performance indicators that are directly correlated to the interference on a wireless link [SDTL06, PAP+05].

2.2. Reliability-Improvement Methods

Over the last few years, a lot of research has been conducted in the field of interference mitigation of WSNs. While many of the resulting techniques are quite specific and depend on the used protocol and modulation scheme, also some general concepts can be emphasized that are applied in a range of popular protocols. Thereby, almost all WSN protocols comprise mechanisms for deterministic and coordinated channel access so that the influence of self interference can be mitigated. On the other hand, the mutual interference between different WSN protocols still imposes problems. Standard protocols for channel access like CSMA/CA often fail in these situations, because of the hidden node problem, which occurs more frequently in dense networks. Additionally, different protocols often employ different modulation schemes and different power levels for transmission, making the interaction between them unpredictable and hard to model. Solutions could be *cognitive radios*, which permanently monitor the wireless channels and automatically switch to an unused frequency for transmission [TMLU08]. Additionally, many other coexistence protocols have been proposed, which can be classified into collaborative protocols, where different protocols work together to mitigate

[1] en.oxforddictionaries.com/definition/interference

the influence of mutual interference, and non-collaborative protocols [CR03]. A promising non-collaborative mechanism is *adaptive frequency hopping* (AFH), where frequencies are periodically switched while the quality of the according channels is monitored. Channels that are unsuitable for transmission are sorted out in the process and are not considered for transmission anymore. At last, the reliability in those networks can always be increased by introducing redundancy, for example by sending data on multiple frequencies using multiple transceivers or by using multipath routing protocol like AOMDV [AKM+14].

2.3. Bluetooth

Bluetooth, as defined in the IEEE 802.15.1 standard and often referred to as *Bluetooth Classic* nowadays, was originally designed as a short-range, cost-efficient cable replacement for portable personal devices like computer peripherals, cellular phones, digital cameras and many more [LSS07]. It underlies the constant development by the Bluetooth Special Interest Group (SIG), which releases it as an open industrial technology standard. Version 4.0 introduced significant changes to the Bluetooth technology with the extension by a new flavor called *Bluetooth Low Energy* (BLE). BLE is supposed to prepare the Bluetooth technology for the novel demands of the Internet of Things by narrowing the functionality of the Bluetooth stack and drastically reducing the data rate and energy consumption. However, Bluetooth Classic is still used for the short-range transmission of large amounts of data, which BLE cannot handle, and in applications without high energy constraints. For a better demarcation between Bluetooth Classic and Bluetooth Low Energy, both protocols are depicted in more detail in the next two sections.

2.3.1. Bluetooth Classic

As already mentioned, Bluetooth Classic (v3.0) was originally designed as a short-range wireless technology to replace cables in personal area networks (PANs). Therefore, it offers a high data rate and native support for popular applications like data and audio streaming [PW10]. Bluetooth operates in the 2.4 GHz frequency band and allows ad hoc connectivity in the form of so-called piconets. Piconets are wireless personal area networks (WPANs), which consist of a single device acting as a master and up to seven devices acting as slaves, forming a star topology. Additionally, slaves can be put in the *parked* state with a distinct address space, where they are not actively part of the communication in the piconet, but can be added by the master on demand at any time. At last, a device is also allowed to take part in multiple piconets, joining them to a so-called scatternet, to relay data between them [Bis01]. An exemplary scatternet is illustrated in Figure 2.1. For communication, Bluetooth utilizes a fast frequency-hopping mechanism, with about 1600 hops per second to diminish the influence

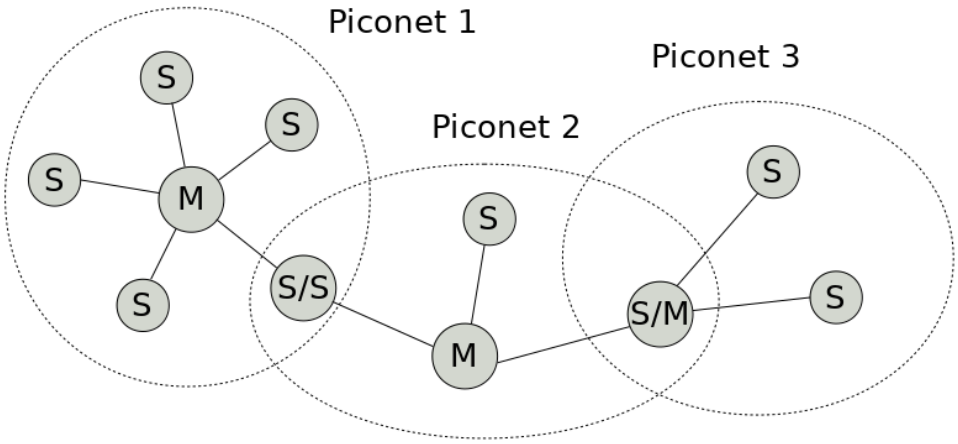


Figure 2.1.: Example of a Bluetooth scatternet consisting of several piconets (M=master, S=slave, S/S=slave-slave bridge, S/M=slave-master bridge).

of external interference. The frequency hopping channel is thereby determined by the address of the piconet's master, enabling the spacial and temporal coexistence of piconets. If desired, the frequency hopping algorithm can operate in an adaptive mode which can reduce the set of 79 available channels to a minimum of 20 channels, by sorting out channels with a bad quality [Tor14]. Because of the popularity of Bluetooth Classic and its broad field of application, it has been subject to a lot of research over the last few years. Thereby, various different areas of Bluetooth Classic are covered, including the evaluation of application scenarios as well as its reliability in the presence of external interference and its coexistence with other protocols [SM00, JPLK01].

2.3.2. Bluetooth Low Energy

Due to the recent developments in the Internet of Things and the consequent new requirements for energy-efficient transmission protocols, the original Bluetooth standard has been extended by Bluetooth Low Energy, also known as Bluetooth Smart, which is supposed to achieve a significantly lower energy consumption and cost per piece, while increasing the communication range. The extension was introduced to cover the field of sensor and controlling applications on low-resource platforms like wearables and small sensor devices in industrial applications. Thereby, BLE works on a limited function set of Bluetooth Classic, but also introduces some new features. Since Bluetooth Low Energy in version 4.1 is used throughout this thesis, a more detailed description of the Bluetooth Low Energy stack and its timing according to the standard [ble13] can be found in Section 2.3.2.2 and following. Additionally, the next section gives a brief overview of the different BLE versions, for a better demarcation between them.

2.3.2.1. Version Overview

Bluetooth Low Energy was first introduced to the standard in 2010 with the release of the version 4.0 specification. It significantly improves the connectivity and range of Bluetooth and introduces the Low Energy protocol, which allows the operation on resource-restricted devices. However, it lacks many features from the previous Bluetooth versions, like meshing capabilities, automatic evaluation of the channels' quality for adaptive frequency hopping and many security features. These were traded for a more lightweight stack. The v4.0 standard also provides the specification for the physical layer of the Bluetooth stack. After that, BLE v4.1 introduced multiple link-layer roles, effectively allowing the construction of mesh networks due to the fact that devices can act as masters and slaves at the same time. Thereby, v4.1 is only a software update to the previous version and also improves the bulk data exchange rate and coexistence with Mobile Wireless protocols like 4G. Many features of the new version are not mandatory but can be chosen by the manufacturers as needed, so that many stack implementation do not comprise certain features at all. At last, BLE v4.2 introduces a significantly higher data rate with about 2.6 times the transmission rate of prior versions. In addition to that, new security features and profiles for home automation are added.

Finally, one should say that the Bluetooth standard has lately been released in version 5. The new version is characterized by a higher communication range and data rate in comparison to Bluetooth Low Energy, theoretically rendering Bluetooth Classic obsolete [DMSL+17]. There are, however, not many devices supporting Bluetooth 5 yet, so that Bluetooth Low Energy is still the prevalent technology. For more information about different versions refer to website of the Bluetooth Special Interest Group[2].

2.3.2.2. The Stack

The Bluetooth Low Energy stack is a multilayer architecture, as shown in Figure 2.2, with a strong separation between the host layer and the controller layer. As one can easily see, the controller layer consists of the Physical Layer (PHY), the Link Layer (LL) and one side of the Host Controller Interface (HCI) which transports commands and events between the two layers and creates an abstract interface to allow an easy interchangeability of the host. The HCI is usually implemented through serial transport protocols like UART or SPI or callbacks and function calls in the case of an embedded wireless MCU. The host layer, on the other hand, comprises the other side of the Host Controller Interface (HCI), the Logical Link Control and Adaption Protocol (L2CAP), the Attribute Protocol (ATT), the Security Manager (SM), the Generic Attribute Profile (GATT) and the Generic Access Profile (GAP). Thereby, L2CAP's main task is to provide logical channels to multiplex the different upper

[2] www.bluetooth.com

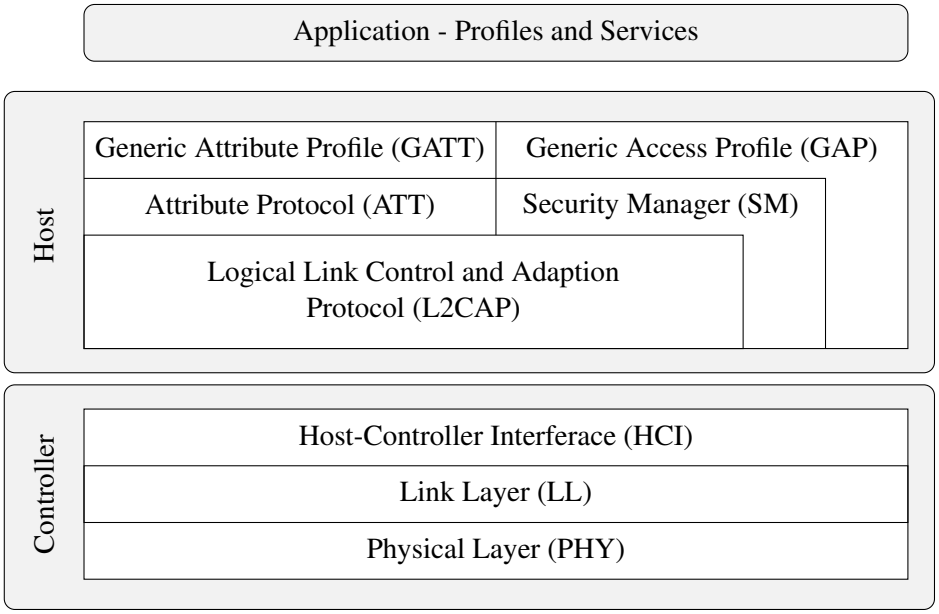


Figure 2.2.: Structure of the Bluetooth Low Energy stack.

level protocols over the same physical link such as the ATT protocol which defines so-called attributes - data resources that can be manipulated in a request/reply manner through the GATT protocol. Additionally, the latter also allows to send one-way messages, so-called notifications, to subscribed communication partners when the value of an attribute changes. At last, the GAP manages the connection process and defines four different roles for how the devices can interact with one another. In this work only two of the four roles are used, namely the central, which is usually a device without any resource constraints that acts as the master of a connection, and the peripheral, which is usually resource restricted and acts as the slave that gathers sensor data and sends them to the central. Usually, Bluetooth chip manufacturers only distribute the respective Controller layer in form of a precompiled binary file, which is only accessible through the functions and events defined in the HCI.

2.3.2.3. Physical Layer

On the physical layer, Bluetooth Low Energy comprises a total number of 40 channels, with a bandwidth of 1 MHz each, that are mapped to the frequencies from 2400 MHz to 2483.5 MHz as shown in Figure 2.4. For modulation the *Gaussian Frequency Shift Keying* (GFSK) with a modulation index of 0.5 is used that significantly reduces the power consumption and increases the range in comparison to classic Bluetooth. This results in a data rate of 1 Mbps.

The main task of the physical layer, however, is the transmission of link layer packets, as depicted in Figure 2.3. Those have a maximum size of 47 bytes and carry *protocol data units*

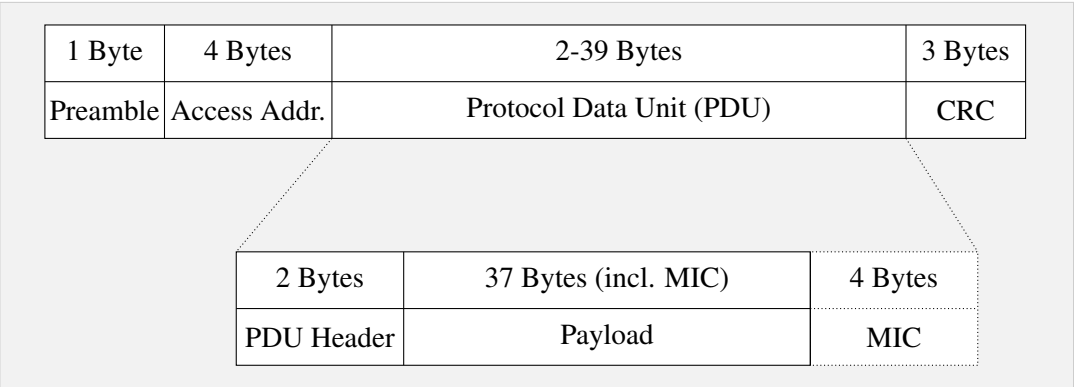


Figure 2.3.: Structure of a Bluetooth Low Energy data packet.

(PDUs), which are either data packets or advertising packets for the connection establishment. The *preamble* is used for synchronization, while the *access address* makes the packet relatable to a specific connection. The last two fields are the *message integrity check* (MIC), which is used for encryption of the packet, and the *cyclic redundancy check* (CRC) for error detection.

2.3.2.4. Mesh Networks

In Bluetooth Low Energy, a peripheral can only maintain a single connection at a time, while centrals are able to manage multiple connections at once using a simple TDMA scheme. The time slots in which the central exchanges data with a particular peripheral are called connection events and have a fixed length and frequency which is negotiated at the establishment of the connection. The parameters of a connection event are not dynamically managed by the Bluetooth Low Energy stack but remain static so that the maximum number of nodes in the network should be known beforehand to avoid the costly reconfiguration of all connections by the application. Since version 4.1, the Bluetooth standard allows BLE devices to act in multiple link layer roles at the same time. This enables a device to act as a master for one one communication partner and a slave for another, effectively enabling the construction of mesh network, similar to the piconets from the original Bluetooth standard. Such a piconet is depicted in Figure 2.1. As one can see, these mesh networks are a combinations of piconets, in which a single master manages multiple slaves in a star topology. The piconets are then connected by nodes that fulfill two roles at the same time and in this way create a bridge between two piconets. One should, however, notice that there is no native support for the synchronization of two masters in different piconets. For example the masters of piconet one and piconet two could both send to the slave/slave node in the middle at the same time, resulting in a collision of the packets. In the extreme case, a piconet only contains a master that acts as a bridge between two piconet, allowing the construction of arbitrary mesh networks.

2.3.2.5. Frequency Hopping

Since Bluetooth Low Energy operates on the 2.4 GHz frequency band, the same as the omnipresent Wi-Fi and the IEEE 802.15.4 protocols, it consolidates some mechanisms to ensure a faultless operation in the presence of external interference. Therefore, the broadcast channels 37 to 39, also used for the discovery of devices and the establishment of connections, are placed at frequencies that are not interfering with the popular Wi-Fi channels 1, 6 and 11 as shown in Figure 2.4. Additionally, BLE provides a total number of nine channels that are not overlapping with the Wi-Fi frequencies and uses a frequency hopping mechanism that makes nodes change their channel right before every connection event. Thereby, the next channel ch_{next} is chosen by the communication partners by using the following formula

$$ch_{next} = (ch_{curr} + ch_{inc}) \; mod \; 37, \tag{2.1}$$

where ch_{curr} is the current channel and ch_{inc} is a random integer between 5 and 16 that is chosen by the central at establishment of a connection. At last, the set of data channels, specified in the *channel map*, can be reduced, so that only a subset of the 37 channels is used for transmission. This way, the application can assess the channels and discard the ones that are subject to the influence of external interference. In this case, the channel hopping mechanism still iterates over all 37 channels. If a channel is selected that is not used, according to the current channel map, the channel will simply be mapped to a used channel by creating a remapping table that contains N_{used} usable channels in ascending order. Then, a $remappingIndex$ is calculated as

$$remappingIndex = ch_{curr} \; mod \; N_{used}, \tag{2.2}$$

which is used to select a usable channel from the remapping table.

2.4. IEEE 802.15.4

The IEEE 802.15.4 standard defines a communication protocol for *Low-Rate Wireless Personal Area Networks* (LR-WPAN) with a focus on low-cost, low-energy applications [iee16]. Therefore, it is especially suitable for application areas in process automation and wireless sensor networks. The standard separates the devices of a PAN into two distinct groups, namely *full function devices* (FFD) and *reduced function devices* (RFD). As the name implies, these devices respectively support the full functionality or a reduced functionality of the standard. Thereby, RFDs can only communicate with FFDs, while FFDs can communicate with all device types. A PAN consists of at least one FFD which acts as the PAN coordinator and manages the essential tasks of the network like device association, addressing and routing.

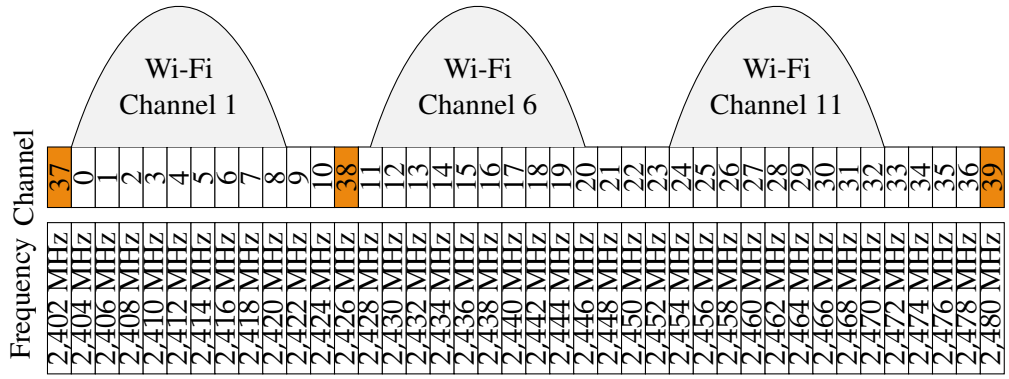


Figure 2.4.: Frequencies and corresponding channels of Bluetooth Low Energy.

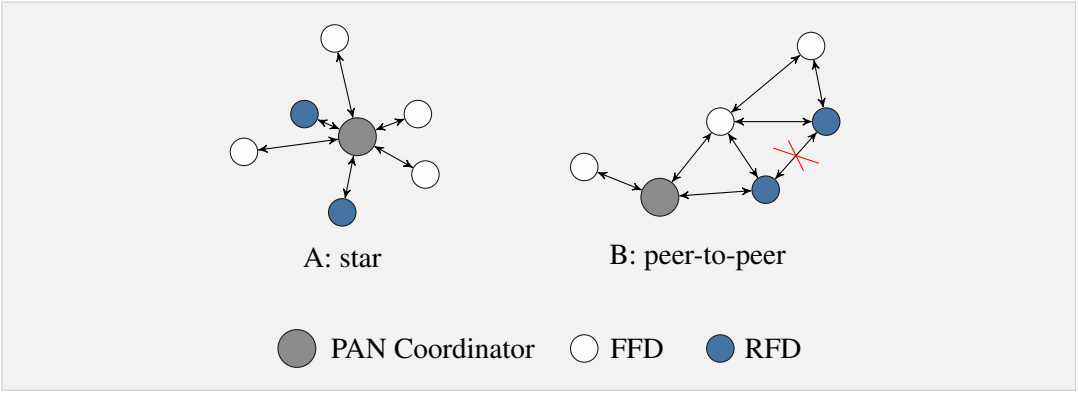


Figure 2.5.: Two network topologies as defined in the IEEE 802.15.4 standard. The red cross signifies that a connection between two RFDs is not allowed.

Furthermore, it sets up the network to one of the topologies depicted in Figure 2.5. The first one is a star topology with the PAN coordinator as the center node and all regular devices distributed around it. Thereby, all communication has to flow through the PAN coordinator and only a single hop is supported, so that the range of the network is naturally limited by the communication range of the coordinator. To counteract this problem, also a peer-to-peer topology is provided. Here, the devices do not have to communicate via the PAN coordinator but can directly exchange messages. Additionally, other FFDs can act as coordinators that manage a subset of the devices in the network. Multiple of these peer-to-peer networks can be joined in a *cluster tree*, where a single FFD acts as a bridge between the single clusters. This enables the transmission of messages over multiple nodes, which is, however, subject to higher level protocols like ZigBee or WirelessHART, since the IEEE 802.15.4 standard only defines the physical and medium access control (MAC) layers.

2.4.1. Physical Layer

The standard defines various physical layer modes, some of which operate on region-restricted frequencies. In the unlicensed 2.4 GHz frequency band, however, it provides 16 non-overlapping channels with a 5 MHz spacing using *O-QPSK* for modulation. Therefore, a theoretical data rate of 250 Kb/s can be achieved. Since the 2.4 GHz band is utilized by many popular protocols, this number of channels enables different mechanisms, e.g. frequency hopping, to mitigate the influence of external interference. The main task of the physical layer is the transmission and reception of *physical protocol data units* (PPDUs), which carry the packets of upper layer protocols.

2.4.2. Medium Access Control Layer

The MAC layer provides logical channels on top of the physical layer channels and offers two different methods to access them, a beacon enabled mode and a non-beacon enabled mode. The former utilizes a superframe structure, as shown in Figure 2.6, and uses beacons for synchronization. Those are periodically sent by the PAN coordinator. Thereby, the time between two consecutive beacons is called beacon interval and contains an active period, where time is divided into 16 equally sized slots, and an optional inactive period, where no communication takes place and devices enter low-power states to preserve energy. Additionally, this time allows other network coordinators to send their own beacons. While the first slot is reserved for the network beacon, the other slots can be further divided into a *Contention Access Period* (CAP) and a *Contention Free Period* (CFP). During the CAP, devices can communicate using a CSMA/CA algorithm, while the CFP offers up to seven preassigned *Guaranteed Time Slots* (GTS). Access to these is exclusive to one specific device. In the non-beaconed mode, on the other hand, no superframe structure is used so that devices always have to stay active. Channel access is accomplished with an unslotted CSMA/CA algorithm and energy saving is subject to the higher level protocols.

2.5. IEEE 802.15.4e

With the increasing utilization of wireless sensor networks in industrial applications, the original IEEE 802.15.4 standard quickly reached its limits with respect to the new demands. Especially, the scalability and determinism were a problem caused by the limited availability and flexibility of the GTSs [DGBA16]. Additionally, the protocol lacks the support for channel hopping so that the communication is prone to external interference. Because of this, IEEE 802.15.4e extends the existing standard by several MAC layer behavior modes [iee16], each for its own specific application domain:

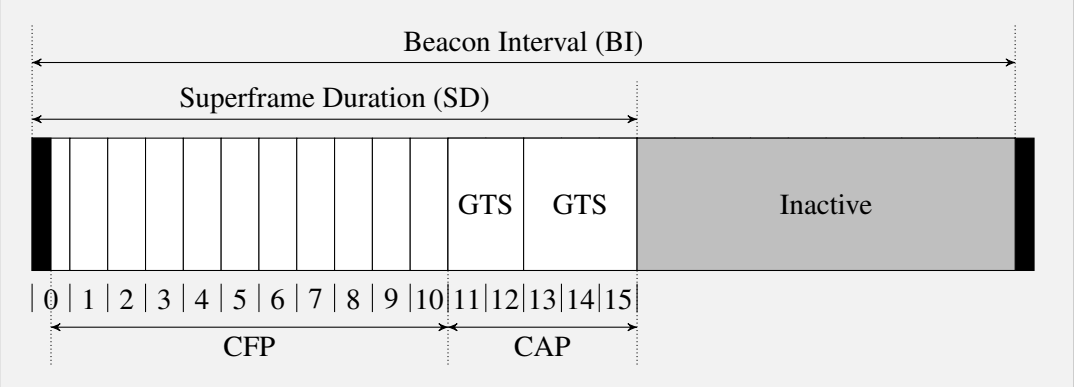


Figure 2.6.: Superframe structure according to the IEEE 802.15.4 standard.

- *Radio Frequency Identification Blink (BLINK):* For applications like location, tacking and item and people identification. Messages can be sent without prior association with the network.
- *Asynchronous Multi-Channel Adaptation (AMCA):* For applications with a large deployment requirements like process automation or infrastructure monitoring. Only usable in non beacon-enabled PANs.
- *Deterministic and Synchronous Multi-channel Extension (DSME):* For industrial and commercial applications with strict reliability and latency requirements.
- *Low Latency Deterministic Networks (LLDN):* For applications with minimal latency requirements, but with restriction to a star topology. The PAN coordinator may use multiple transceivers on different channels to increase the overall number of nodes in the network.
- *Time Slotted Channel Hopping (TSCH):* For process automation applications with the goal to minimize the influence of external interference.

From this list MAC modes, DSME and TSCH seem to be the most relevant ones for the application in wireless mesh networks in industrial environments, since they do not only provide a high reliability, but also a low and deterministic delay. Therefore, they further described in the next two sections.

2.5.1. Time Slotted Channel Hopping

Time Slotted Channel Hopping (TSCH) combines the concept of time slotted channel access with multiple logical channels and therefore provides an excellent predictability with bound delays and a guaranteed bandwidth per node. Additionally, it increases the network capacity

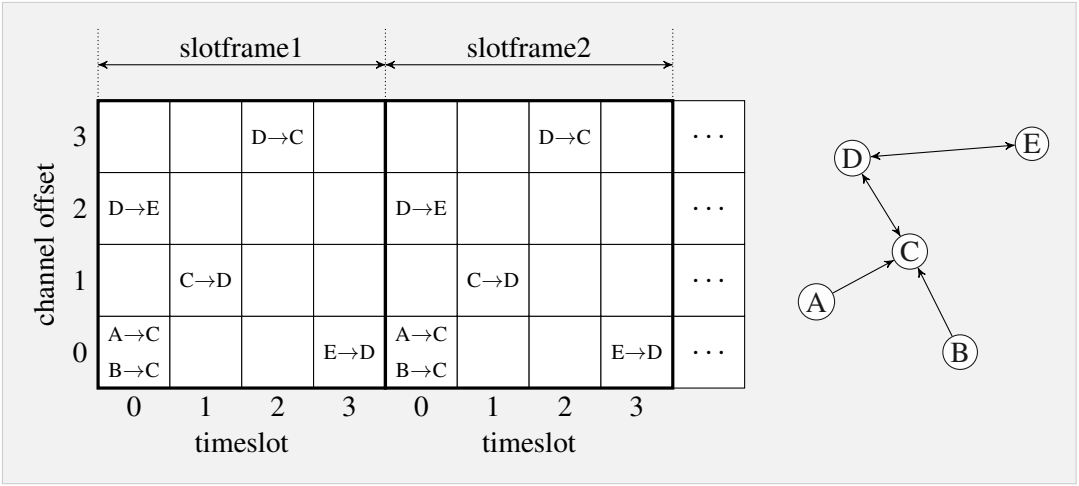


Figure 2.7.: Structure of an IEEE 802.15.4 TSCH slotframe with slot occupation for an exemplary network.

due to the multi-channel approach and mitigates the influence of external interference by utilizing a channel hopping algorithm. These properties make the protocol suitable for the application in process automation and monitoring. In TSCH synchronization is not achieved through beacons, but all nodes synchronize on so-called *slotframes*, which are shown in Figure 2.7 for an exemplary network. A slotframe consists of multiple time slots with a fixed length, so that they can hold a maximum length data packet and the related acknowledgement. Multiple slotsframes can be used on different channels at the same time. A link between two nodes is represented by the pairwise assignment of a directed communication to distinct time slots with an arbitrary channel offset, as also shown in Figure 2.7. Thereby, *dedicated links* provide exclusive access to the slot, while *shared links* use the CSMA/CA algorithm for channel access. Shared links are usually used for discovery and routing messages with multiple sender and receivers. At last it should be said that TSCH is topology independent and therefore a common choice for mesh network. The algorithm for the link schedule, however, is not in the scope of the standard and left for higher level protocols. Finding a feasible and efficient link schedule is not a trivial task that even gains in complexity in highly dynamic networks. In static networks, however, TSCH has been proven to work well with a static link schedule [FEO+].

2.5.2. Deterministic and Synchronous Multi-channel Extension

The Deterministic and Synchronous Multi-channel Extension (DSME) supplements the IEEE 802.15.4 standard since 2012 as one of the newly introduced MAC layer protocols that are supposed to increase the reliability and energy-efficiency in wireless mesh networks. Therefore, it expands the standard by a distributed slot assignment mechanism which works without the

participation of a PAN coordinator. Additionally, it considers multiple channels to enable the spatially and temporally overlapping transmission of data. The most significant changes, including the altered superframe structure, the distributed GTS allocation and frequency hopping are briefly summarized in the next few sections.

2.5.2.1. Superframe structure

The DSME MAC protocol utilizes a superframe structure that builds upon that of the original IEEE 802.15.4 standard. As shown in Figure 2.8, a DSME superframe always starts with an enhanced beacon which contains essential information about the network, like the duration of a superframe and indications for time synchronization. Beacons are only sent by network coordinators which exclusively reserve the beacon slots to avoid collisions. The next phase is the *contention access period* (CAP) where participants of the network can exchange messages in a contention based manner, e.g. by using CSMA/CA. This is mainly used for management packets, since the *contention free period* (CFP) offers exclusive access to *guaranteed time slots* (GTS). In comparison to the original IEEE 802.15.4 standard, the GTSs are not only spread in time but also over different frequencies. This enables the temporally and spatially overlapping transmission of data on different channels. Additionally, in DSME multiple superframes can be joined to a *multi-superframe* to increases the number of allocatable GTSs. This is especially useful for low data rate applications. As one can easily see, the size of a superframe did not change in comparison to the original standard and still contains 16 distinct time slots, whereof 7 are used for the CFP and 9 are used for the CAP and the beacon. DSME, however, allows the operation in a *CAP reduction* mode where the CAP of every superframe in a multisuperframe but the first is omitted and replaced by GTSs, resulting in a total number of 15 distinct time slots for the GTSs. Thereby, the length of a superframe, multisuperframe and beacon interval are given by the *superframe order* (SO), *multisuperframe order* (MO) and *beacon order* (BO) in potencies of two so that

$$0 \leq SO \leq MO \leq BO \leq 14. \tag{2.3}$$

This allows the calculation of the superframes per multisuperframe as 2^{MO-SO} and the number of multisuperframes per beacon interval as 2^{BO-SO}.

2.5.2.2. GTS Management

In contrast to the default IEEE 802.15.4 MAC layer, DSME manages GTSs in a distributed and more efficient manner, without the involvement of the PAN coordinator. Thereby, slots have to be allocated with a communication partner before they can be used for transmission. For the allocation a modified three-way handshake is used, as depicted in Figure 2.9, where

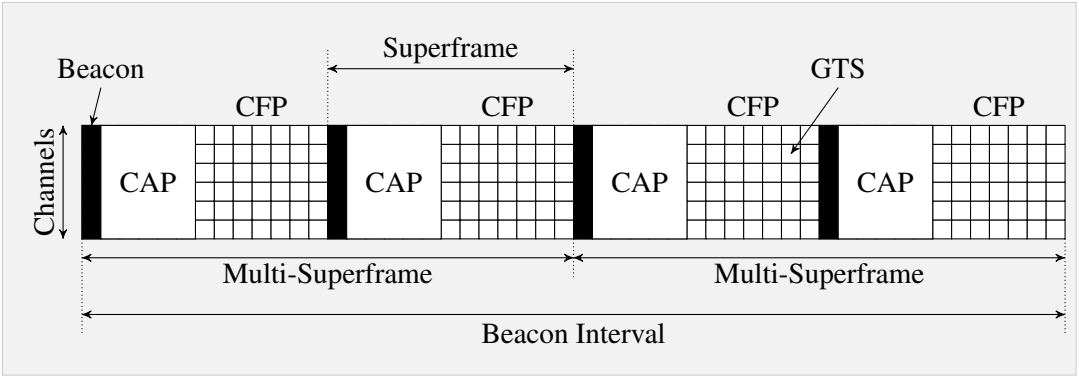


Figure 2.8.: Structure of an IEEE 802.15.4 DSME multi-superframe

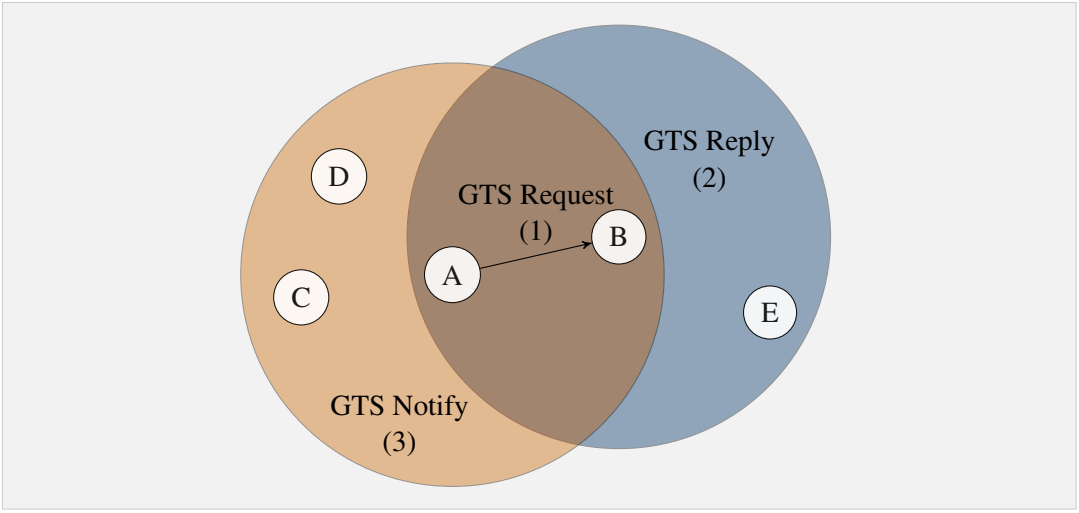


Figure 2.9.: Procedure of a GTS allocation handshake in DSME

the last two messages are sent via broadcasts. The slot allocation starts with a *GTS request* (1) from a device A to its communication partner B, containing a preferred slot, channel and superframe, and a *slot allocation bitmap* (SAB) with all GTSs that are not used by either A or its neighbours. B can then select the preferred GTS if it is free or another free GTS from the SAB that fits both A and B. After that, B broadcasts a *GTS reply* (2) with the updated SAB, so that all devices in the neighbourhood can mark the selected slot as occupied. At last, A also broadcasts a *GTS notify* (3) to inform the devices in its neighbourhood about the newly allocated slot. For both broadcasts, the neighbours of A and B respectively have to check if the newly allocated slot is conflicting with their local SAB. If this is the case, they send a *GTS duplicate allocation notification* to roll back the slot allocation procedure. The GTS deallocation works in the same way as the allocation, using the modified three-way handshake.

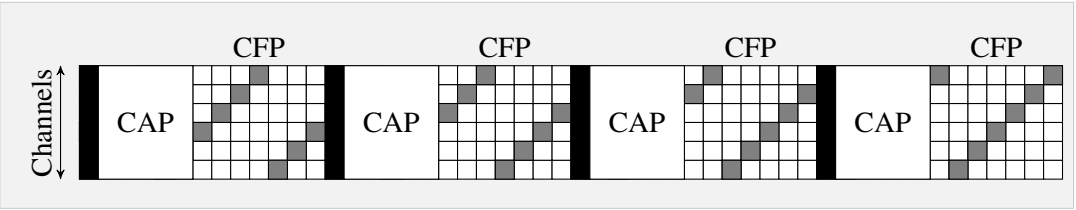


Figure 2.10.: Channel hopping sequence for IEEE 802.15.4 DSME with channelOffset=3 and hoppingSequence={1,2,3,4,5,6}.

2.5.2.3. Channel Hopping

With the extension by multiple channels, also the support for channel hopping was introduced to the IEEE 802.15.4 standard as an optional feature. DSME supports channel hopping in a slotted mode, where a GTS is not bound to a single channel but hops over a predefined list of frequency channels, the so-called *hopping sequence*, to receive data. This procedure is exemplarily depicted in Figure 2.10 for a single devices with a channel offset of three and {1,2,3,4,5,6} as the hopping sequence. A transmitting device has to switch to the channel used by the receiving device to send data. Thereby, the next channel C at a given GTS i in superframe j can be calculated as

$$C(i) = hoppingSequenceList[(j * l + i + channelOffset + PANCoordinatiorBSN) \ \% \ hoppingSequenceLength]. \quad (2.4)$$

Here, l is the number of GTSs in j and equal to 7 if CAP reduction is disabled or j is zero, and 15 otherwise. $PANCoordinatorBSN$ is an enhanced beacon sequence number of the PAN coordinator, $hoppingSequenceList$ is the list of all channels in the hopping sequence and $hoppingSequenceLength$ is the number of channels in the hopping sequence. Finally, $channelOffset$ is the channel offset of the receiving device and unique to all devices in the neighbourhood. Two devices in channel hopping mode can only communicate if they support the same set of channels and agree on a common hopping sequence, which can be predefined or communicated through enhanced beacon frames. Additionally, a default hopping sequence is provided for the hopping sequence id zero. It is determined by shuffling the available channels in a pseudorandom manner using a linear feedback shift register.

2.6. ZigBee

Since 2002, the ZigBee Alliance develops ZigBee as a high-level communication protocol based upon the physical and MAC layer of the IEEE 802.15.4 standard. It is intended for low power, low data rate applications like home automation or small sensor networks. Thereby, it

provides short range, self-organized mesh networking using the ad-hoc on-demand distance vector (AODV) algorithm for routing, so that every device is responsible for route discovery itself and there is no need for a central management device. Additionally, all devices in a ZigBee network share a common channel, usually the one with the least interference at startup, and there is no frequency hopping mechanism. This means that the network is prone to any kind of external interference. ZigBee can run in a non-beaconed mode and a beaconed mode, where devices are synchronized and the underlying IEEE 802.15.4 superframe is divided into 16 time slots. Up to seven of these slots can be allocated for specific devices to increase reliability and predictability, since access to the slots is normally contention based using CSMA/CA [LSH08]. Due to the low robustness of ZigBee, it is rarely used in industrial applications nowadays.

2.7. WirelessHART

WirelessHart is an industrial standard developed by the HART Communications Foundation (HCF) and is based upon the physical layer of the IEEE 802.15.4 standard. It is the wireless counterpart to the *Highway Addressable Remote Transducer* (HART) protocol, which is commonly used for industrial automation. In comparison to the IEEE 802.15.4 standard, however, it redefines the data-link, network, transport, and application layers to accomplish some essential requirements on the protocol. This includes the simplicity of deployment and usage, flexibility, scalability, reliability, security and the ability for self organisation and self healing. Therefore, it utilizes a mesh structure with a single global *Network Manager*, which takes care of the network's formation, resource scheduling, path configuration and many more tasks related to the wireless mesh network. As one can easily imagine, the complexity for the Network Manager increases drastically for large mesh network. All devices are synchronized using the same global clock and use a TDMA scheme with pre-scheduled, fixed-length slots for communication. Since WirelessHART also operates in the 2.4GHz frequency band, it employs a Frequency Hopping Spread Spectrum (FHSS) mechanism with 16 frequencies to diminish interference with other popular wireless protocols [LSH08]. WirelessHART has its origins in the unsuitability of ZigBee for industrial applications and especially increases the reliability and energy efficiency. Therefore, it is considered one of the most popular wireless protocols for industrial process automation applications [PC09].

2.8. Related Work

Since the integration of Bluetooth Low Energy in the official standard in 2010, it has been subjective to extensive research, covering almost every aspect of it. Thereby, a large part of

the published papers deals with its applicability in different scenarios, for example in body area networks for medical purposes [YXL12, OK10], indoor positioning applications [FH14] or inter-vehicular communication [LTT15]. Another large part of the research is dedicated to the increased energy-efficiency [SHNN12] and the simplified device discovery process [CPH+14, Mik14a] of Bluetooth Low Energy in comparison to Bluetooth Classic, since those were the major changes that enabled its application in wireless sensor networks. More relevant for this thesis, is probably the vast research about BLE mesh networks. While many of the meshing protocols are not relevant for this thesis [GMIS16, Mik14b], since they were proposed for the Bluetooth standard v4.0 which did not support multiple link layer roles, some papers provide useful approaches for the design of those networks. In [KLJ15] a meshing protocol is described, which served as a basis for the official mesh implementation in the Bluetooth standard v5. Additionally, [PLB16] describes a synchronization method for the connection of multiple masters in a BLE mesh network that is implemented in a similar way in the BlueNRG-MS stack [blu16]. On the other hand, the performance of Bluetooth Low Energy in the presence of external interference is not researched as well as many of the other areas. In [TFM+15] a method to improve the reliability of data transmissions is described, but only for connectionless transmissions, which are not considered in this thesis due to their limited performance and flexibility. Furthermore, the round-trip delay of Bluetooth Low Energy is modeled and evaluated in [RGL17] for different bit error rates. Thereby, the obtained results provide a good reference for the average packet delay, obtained in the hardware experiments in this thesis. The most relevant work, however, is [GDP11], which describes the development of a theoretical model for the calculation of the throughput with respect to different bit error rates. The transmission probabilities from the paper serve as the basis for the theoretical model of this thesis.

To the authors knowledge, the only works that directly deal with BLE's channel hopping algorithm are [AKR14] and [AKR15], which model the selection probabilities of the individual channels in the channel hopping algorithm. However, the results suggest that the difference between the probabilities is extremely small so that they can be neglected in the majority of this thesis.

At last, one should mention work that examines the coexistence of Bluetooth Low Energy with other protocols. For example [NDV15] investigates the coexistence with the IEEE 802.15.4 protocol and [SSF+14] considers many popular protocols like Zigbee and Wi-Fi.

Chapter 3

Analytical Model

This chapter describes the development of an analytical model for the estimation of the maximum throughput of Bluetooth Low Energy and DSME in the presence of external interference. The calculation of the transmission probabilities is based on [GDP11] and is applied in an absorbing Markov chain for each protocol. Since the structure of the two models is very similar, it is paradigmatically described for Bluetooth Low Energy in section 3.1 and only the differences for DSME are briefly covered in section 3.2. Additionally, an evaluation of the model, regarding the most important parameters, is provided in the end of this chapter.

3.1. Bluetooth Low Energy

In the course of the next sections, the incremental development of a theoretical model for Bluetooth Low Energy in the presence of external interference is described. Hence, Section 3.1.1 describes a way to calculate the theoretical limit of packets that can be sent during a single connection event, assuming a unidirectional error free transmission. Afterwards, an error model for a single link is developed that allows to calculate the probability of transmitting a total number of n successful packets during a single connection event given the bit error rate of that link. Section 3.1.2 extends this model to multiple channels with distinct bit error rates and also considers the frequency hopping mechanism of Bluetooth Low Energy. Here, the maximum transmission rate λ of an application is calculated so that no packets are lost. This calculation constitutes the main result of this work. At last, Section 3.1.6 applies the model to a multi-hop network.

To provide a better overview of the model, its general structure is depicted in Figure 3.1. Thereby, the single incremental steps, as mentioned above, are illustrated by the different numbers. At first, the theoretical limit of packet per connection interval for a given connection event length has to be calculated (1), which is then used by the channel model. The channel model represents the retransmission behavior of Bluetooth Low Energy in a Markov chain, which also considers the BER of different channels and different packet sizes (2). The result is

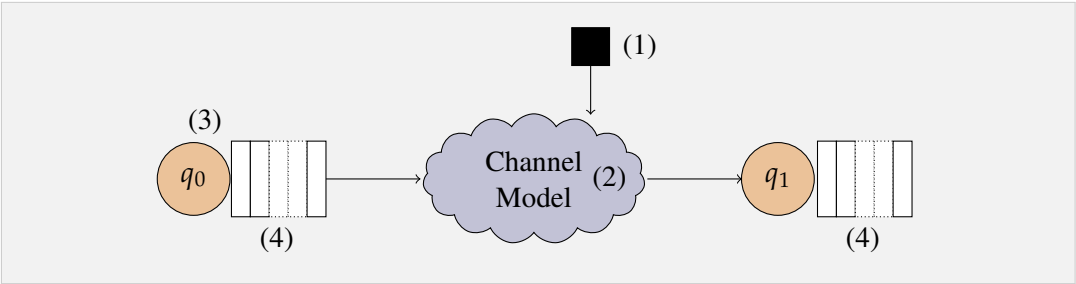


Figure 3.1.: Visual representation of the developed model for reliable data transmissions.

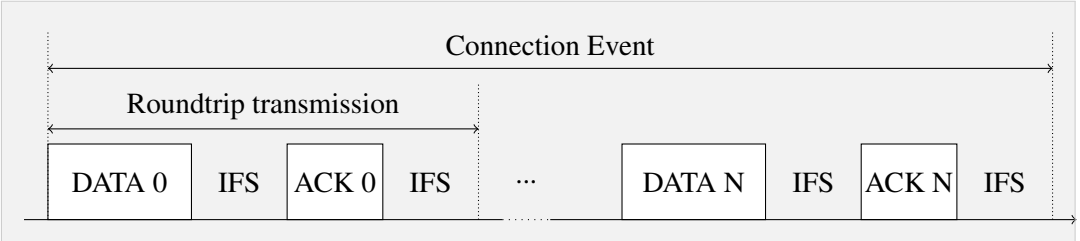


Figure 3.2.: Timing of a Bluetooth Low Energy connection event.

the actual number of packet per channel, considering the influence of external interference, and can be used as the service rate for the queueing model (3). Finally, multiple queuing models can be considered to analyze whole networks under the influence of external interference (4).

3.1.1. Packets per Connection Event

As already mentioned in Section 2.3.2, Bluetooth Low Energy only allows the point-to-point transmission of data in so-called connection events. Thereby, both participants of a connection alternately get the chance to send data, until there remains no data to send or the connection event ends. To ensure the reliability of the connection, every packet has to be acknowledged using a distinct sequence number, carried in the header of regular data packets, as shown in Figure 2.3. If one of the devices has no more data to send, it simply transmits an empty data packet as an acknowledgement. As a consequence, a successful one-way transmission consists of a single packet with application data, an empty data packet for the acknowledgement and two interframe spaces - one after every sent packet as shown in Figure 3.2. In accordance with the notation of [GDP11], such an transmission process is called round trip transmission and the required time is called round trip duration.

The maximum number of round trip transmissions N_{max} per connection interval is limited by the connection interval CI and the length of the connection event T_{CE}, but since the number of transmitted packets is supposed to be maximized, the length of the connection event is assumed to be equal to the connection interval for now. One connection event directly follows

the previous one after the duration of an IFS. Therefore, the maximum number of round trip transmissions N_{max} per connection interval can, assuming an error free connection, be calculated as

$$N_{max} = \left\lfloor \frac{CI}{T_{Data} + T_{ACK} + 2 * T_{IFS}} \right\rfloor, \tag{3.1}$$

with T_{Data} and T_{ACK} being the time it takes to transmit a data packet or ACK packet respectively. For Bluetooth Low Energy these durations can be calculated using 3.2 and 3.3. T_{IFS} is the duration of an interframe space, used to separate two consecutive packets on the channel and equal to 150 μs.

$$T_{Data} = \frac{S_{Data}}{R_{BLE}} \tag{3.2}$$

$$T_{ACK} = \frac{S_{ACK}}{R_{BLE}} \tag{3.3}$$

R_{BLE} is the bit rate of the Bluetooth Low Energy physical layer and equal to 1Mbps. And S_{Data} and S_{ACK} is the packet size of a data packet and an empty data packet respectively. Looking at Figure 2.3 one can easily see that the size of an empty ACK packet is ten bytes while the maximum size of a packet is 47 bytes. Using these values for 3.2 and 3.3, one gets the following transmission times for an ACK packet and a data packet with maximum payload:

$$T_{Data} = \frac{S_{Data}}{R_{BLE}} = \frac{47bytes}{1Mbps} = 376\mu s, \tag{3.4}$$

$$T_{ACK} = \frac{S_{ACK}}{R_{BLE}} = \frac{10bytes}{1Mbps} = 80\mu s. \tag{3.5}$$

These durations can then be used in 3.1 to calculate the maximum number of round trips per connection event. In 3.6, the calculation for a minimum connection interval of 7.5 ms is exemplarily shown.

$$N_{max} = \left\lfloor \frac{CI}{T_{Data} + T_{ACK} + 2 * T_{IFS}} \right\rfloor = \left\lfloor \frac{7.5ms}{0.376ms + 0.08ms + 2 * 0.15ms} \right\rfloor = 9 \tag{3.6}$$

At last one should notice that these values are theoretical. In most implementations of the Bluetooth Low Energy stack, e.g. Android or iOS, the maximum number of packets per connection interval is often limited, because of processing or propagations delays. For the hardware boards used in the BLE experiments, however, the calculation seems to yield almost exact result, as verified with a software provided by STMicroelectronics [1].

[1] http://www.st.com/en/embedded-software/stsw-bnrg001.html

3.1.2. Reliability Model

Until this point, the theoretical model does not take the influence of external interference into account. Therefore, it is extended by following [GDP11] for the calculation of the retransmission probabilities and then developing an own solution using an absorbing Markov chain in the next section. For the sake of simplicity, this section only considers a uniform bit error rate for all channels, which can, however, be easily extended by distinct bit error rates per channel, as shown in Section 3.1.4.

In general, a connection event is kept open until there is no more data to send or the next connection event starts. It can, however, be closed prematurely due to bit errors in the data or acknowledgement packets to save energy, when there is too much interference on the channel to transmit data. This reduces the overall number of packets per connection event and results in a lower throughput. For the model, four different cases can be distinguished, which influence the number of packets in a connection event:

1. A round trip is successful. That means, there is no bit error in the data or acknowledgement packet.
2. A bit error in a single data or acknowledgement packet causes a retransmission of the data packet.
3. A bit error in two consecutive data or acknowledgement packets closes the connection event immediately.
4. A bit error in the *Access Address* field of a packet header closes the connection event immediately.

The probability of all four cases depends on the bit error rate (BER) of the current channel, whereby bit errors are assumed to be uncorrelated. It allows to calculate the probability of a successful transmission of N bits, which is denoted as Γ_N. The complementary probability Ω_N, is the probability that a transmission of N bits leads to at least one bit error. The two probabilities can be calculated as

$$\Gamma_N = (1 - BER)^N, \tag{3.7}$$

$$\Omega_N = 1 - (1 - BER)^N. \tag{3.8}$$

Case 1: The probability of a successful round trip Γ_{RT} is the probability that there is no bit error in the data packet and that there is no bit error in the acknowledgement packet. This is illustrated by the following equation:

$$\Gamma_{RT} = \Gamma_{DATA}\Gamma_{ACK}, \tag{3.9}$$

where Γ_{DATA} and Γ_{ACK} are the probabilities that a data or an acknowledgement packet is transmitted successfully and can be written as shown in 3.7. L_{DATA} and L_{ACK} are the lengths in bit of a data packet and an acknowledgement packet respectively so that

$$\Gamma_{DATA} = (1 - BER)^{L_{DATA}}, \tag{3.10}$$

$$\Gamma_{ACK} = (1 - BER)^{L_{ACK}}. \tag{3.11}$$

Case 2: The probability Ω_{RT} that there is a bit error during the round trip transmission is the probability of either the data packet, the acknowledgement packet or both packets having a bit error that is not in the *AccessAddress* field and can be written as

$$\Omega_{RT} = \Omega_{DATA}\Gamma_{ACK} + \Gamma_{DATA}\Omega_{ACK} + \Omega_{DATA}\Omega_{ACK}. \tag{3.12}$$

Ω_{DATA} and Ω_{ACK} are the probabilities that there is a bit error in the respective packet that is not affecting the *AccessAddress* field

$$\Omega_{DATA} = (1 - BER)^{L_{AA}}(1 - (1 - BER)^{L_{DATA} - L_{AA}}, \tag{3.13}$$

$$\Omega_{ACK} = (1 - BER)^{L_{AA}}(1 - (1 - BER)^{L_{ACK} - L_{AA}}, \tag{3.14}$$

with L_{AA} being the length of the *AccessAddress* field in the BLE packet header.

Case 3: To save energy, a connection event is closed prematurely when one of the participating devices receives two consecutive packets with bit errors. The probability for this event is denoted Ω^i_{BE} and depends on the current number of round trip transmissions i. It can be calculated as

$$\Omega^i_{BE} = \begin{cases} 0, & for\ 0 \leq i \leq 1 \\ (\Omega_{DATA}\Gamma_{ACK})^2 + (\Gamma_{DATA}\Omega_{ACK})^2 + (\Omega_{DATA}\Omega_{ACK})^2 & \\ +2(\Omega_{DATA}\Gamma_{ACK} + \Omega_{DATA}\Omega_{ACK}) & \\ +2(\Gamma_{DATA}\Omega_{ACK} + \Omega_{DATA}\Omega_{ACK}), & otherwise \end{cases}. \tag{3.15}$$

The Equation is the result of the probabilities that the Markov chain, depicted in Figure 3.3, ends up in one of the three bit error states $s_{i,1}$, $s_{i,2}$ or $s_{i,3}$ on the first round trip transmission and then goes to the error state $s_{i,4}$ in the second round trip transmission.

Case 4: At last, there is the case that a connection event is closed, when the *AccessAddress* field of a BLE packet does not match the expected value. This can either be caused by bit

errors or by another device transmitting a packet during the same connection event, on the same channel. This does not mean that a collision has to occur. However, since this model only considers the influence of external interference and not the impact of falsely sent packets, the probability Ω_{AA} of a bit error in the *AccessAddress* field during a round trip transmission can be written as

$$\Omega_{AA} = 2(\Gamma_{AA}\Omega_{EAA}) + \Omega_{EAA}^2, \tag{3.16}$$

with Ω_{EAA} being the probability that at least one bit in the *AccessAddress* field is corrupted and Γ_{AA} being the probability that the *AccessAddress* is transmitted without any bit errors.

$$\Gamma_{AA} = (1 - BER)^{L_{AA}}, \tag{3.17}$$

$$\Omega_{EAA} = 1 - (1 - BER)^{L_{AA}}. \tag{3.18}$$

As shown in Equation 3.16, it is not necessary to distinguish between a bit error in the *AccessAdress* of an acknowledgement packet and a data packet, since the size is the same in both headers.

3.1.3. The Markov Chain

Figure 3.3 shows the absorbing, time discrete Markov chain used to determine the number of round trip transmissions per connection event. It can be described by the state space $S = \{s_{0,0}, ..., s_{i,j}, ..., s_{N_{max},0} \mid 0 \leq i \leq N_{max} - 1, 0 \leq j \leq 4\}$ and the transition probability matrix

$$P = [p_{i,j}]_{|S|\times|S|} \quad s.t. \quad \sum_{j \in S} p_{i,j} = 1, \tag{3.19}$$

where $p_{i,j}$ denotes the probability of moving to state j when being in state i. The Markov chain contains N_{max} absorbing states $A \subset S$ for which

$$p_{i,i} = 1, \quad p_{i,j} = 0 \; for \; i \neq j. \tag{3.20}$$

In general the Markov chain describes the transmission behavior of Bluetooth Low Energy and consists of $N_{max} - 1$ equal trees, as shown in Figure 3.3, with the ith tree symbolizing i successful round trip transmissions. Thereby, the state $s_{i,0}$ symbolizes i successfully transmitted packets. The state $s_{i,4}$ signifies that the connection event has been closed after i successful round trips and the state $s_{N_{max},0}$ shows that all N_{max} round trips were successful. The states $s_{i,1}$, $s_{i,2}$ and $s_{i,3}$ can be translated to a bit error in the data packet, acknowledgement packet and both packets after i successful round trip transmissions respectively.

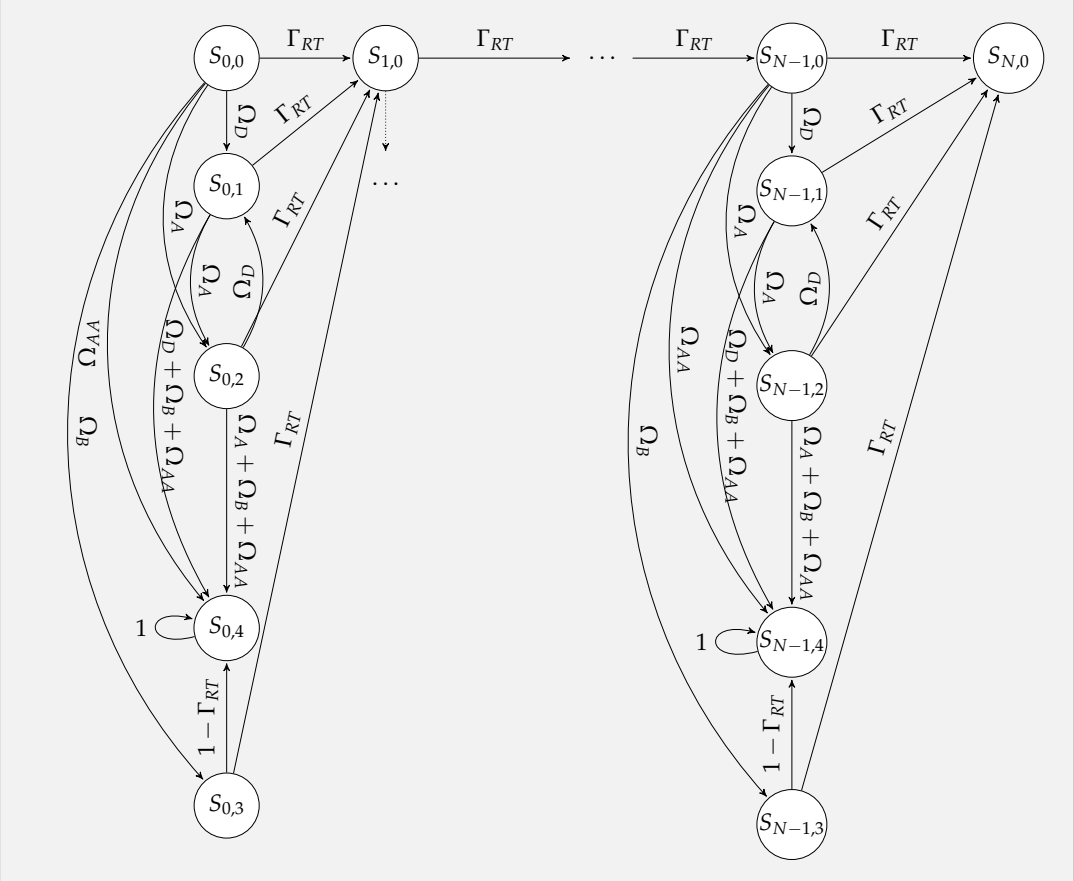


Figure 3.3.: Markov chain for the number of roundtrips per connection interval with $\Omega_D = \Omega_{DATA}\Gamma_{ACK}$ and $\Omega_A = \Gamma_{DATA}\Omega_{ACK}$ and $\Omega_B = \Omega_{DATA}\Omega_{ACK}$.

The transition probabilities between states in the *ith* tree are given by

$$
\begin{aligned}
p_{s_{i,j},s_{i+1,0}} &= \Gamma_{RT}, && 0 \leq j \leq 3 && (case\ 0)\\
p_{s_{i,j},s_{i,1}} &= \Omega_{DATA}\Gamma_{ACK} && j = 0,2 && (case\ 1.1)\\
p_{s_{i,j},s_{i,2}} &= \Gamma_{DATA}\Omega_{ACK} && j = 0,1 && (case\ 1.2)\\
p_{s_{i,0},s_{i,3}} &= \Omega_{DATA}\Omega_{ACK} && && (case\ 1.3)\\
p_{s_{i,0},s_{i,4}} &= \Omega_{AA} && && (case\ 4)\\
p_{s_{i,1},s_{i,4}} &= \Omega_{DATA}\Omega_{ACK} + \Omega_{DATA}\Gamma_{ACK} + \Omega_{AA} && && (case\ 3+4)\\
p_{s_{i,2},s_{i,4}} &= \Omega_{DATA}\Omega_{ACK} + \Gamma_{DATA}\Omega_{ACK} + \Omega_{AA} && && (case\ 3+4)\\
p_{s_{i,3},s_{i,4}} &= 1 - \Gamma_{RT} && && (case\ 3+4)
\end{aligned}
\tag{3.21}
$$

For the further calculations, the transmission probability matrix needs to be represented in the canonical form

$$P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}. \tag{3.22}$$

For an absorbing Markov chain with r absorbing states and t transient states I is a $r \times r$ identity matrix, 0 is a $r \times t$ zero matrix, R is a nonzero $t \times r$ matrix and Q is a $t \times t$ matrix. Thereby, Q contains the probabilities to transition from one transit state to another and Q the probability to transition from a transit state an absorbing state. The resulting matrix is shown in B.1 in the appendix. To calculate the probability to be in a certain state after n round trip transmissions, one can use $v^{(n)} = v * P^n$, where v is a row vector of length $\mid S \mid$, containing the starting states of the Markov chain. Since a connection event always starts with zero round trip transmissions, it is assumed that

$$v = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}. \tag{3.23}$$

At last, the probability Ψ_i for i successfully transmitted packets during a connection event, is the combination of all probabilities in the ith tree of the Markov chain. As a consequence, the average number of packet per connection event is given by

$$\mu = \sum_{i=0}^{N_{max}} \Psi_i \cdot i. \tag{3.24}$$

3.1.4. Distinct Bit Error Rates

In comparison to the last section, not only a single channel is considered, but a set of N distinct channels $C = \{c_0, \dots, c_i, \dots, c_{N-1}\}$. Additionally, each channel c_i is assigned a bit error rate BER_i, so that a Markov chain, as shown in the last section, can be created for every channel. The input to the individual Markov chains are the bit error rate BER_i of the specific channel c_i and the connection interval and connection event length, which stay constant over all channels. The output is the average number of packets per connection interval μ_i on channel c_i. To find out, how many packets packets can be transmitted over multiple channels with distinct bit error rates, the Bluetooth Low Energy channel hopping algorithm has to be considered or more precisely, the worst case channel schedule in regard to the bit error rate BER_i. A worst case schedule is a repetitive sequence of channels $D = (d_i \mid d_i \in C,\ BER(d_i) \leq BER(d_{i+1}))$ of size $|C|$, so that the channels with a high bit error rate μ_i are grouped together and create the bottleneck of the system. The hopping sequence is basically, a succession of Markov chains for every channel, ordered by their resulting throughput. Thereby, the selection probability of

all channels in C is assumed to be equal. In [AKR15], it has been shown that this is not the case, but since every channel appears exactly once in D, the different selection probabilities for the channels cancel out in the end. At last, it should be said that in BLE only a single channel is used per connection event , so that an unsuccessfully transmitted packet can only be transmitted on two different channels if it is in front of the queue and the current connection event ends before the packet can be retransmitted. In this case, it is simply considered the first packet in the queue in the Markov chain on the next channel.

3.1.5. Impact of Transmission Queues

A queueing system can be defined by an average packet arrival rate λ, an average service rate μ and the size of the queue K. Thereby, the service rate is the rate of packets that can be processed by a single server and therefore leave the queue. In this case, the service rate is given by μ_i as the output of the Markov chain on channel c_i and is time-varying but periodic, due to the repetition of the hopping sequence. In a setup without transmission queues, the maximum transmission rate is limited by the channel with the smallest μ_i. With the introduction of queues, on the other hand, more packets can be transmitted by temporarily buffering packets. Therefore, this section gives an approximation of a BLE transmission queue with a model similar to D/D/1/K.

Since the throughput is supposed to be maximized without loosing any packets and considering the worst case channel schedule D from last section, an arrival rate λ has to be found, so that

$$\arg\max_{\lambda \in \mathbb{R}_0^+} n_t + \lambda \leq K \qquad for\ \forall t \in \mathbb{R}_0^+, \tag{3.25}$$

where n_t is the number of packets in the queue at time t. In other words: An arrival rate has to be found, so that the transmit buffer never blocks an incoming packet.

3.1.5.1. D/D/1/K Model

In this queuing model, the arrival rate and the service rate are assumed to be deterministic or constant. Even though, this does not perfectly reflect the conditions of a real world application, it can be used to approximate the maximum worst case arrival rate. Thereby, it is assumed that departures (D) from the queue are strictly separated from the arrivals (A) and happen instantaneously, as shown for an exemplary worst case hopping sequence in Figure 3.4. Additionally, it is assumed that the arrival of packets happens before the departure of packets.

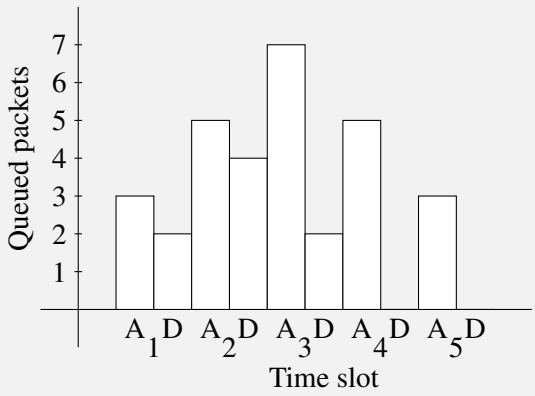


Figure 3.4.: Exemplary variation of the queue size over time for a worst case channel schedule with $d_1 = 1,\ d_2 = 1,\ d_3 = 5,\ d_4 = 6,\ d_5 = 7$ and K=7 and $\lambda = 3$.

The size of the queue n_t at time t, assuming an empty queue at the start of the connection event, can be iteratively calculated as

$$\begin{aligned} n_0 &= (\lambda - \mu_0)^+, \\ n_1 &= (n_0 + \lambda - \mu_1)^+ = (2\lambda - \mu_0 - \mu_1)^+, \\ n_t &= (n_{t-1} + \lambda - \mu_t)^+ = (t\lambda - \sum_{k=0}^{t} \mu_k)^+, \end{aligned} \tag{3.26}$$

where $z^+ = max(z,0)$, because the number of packets in the queue cannot be negative. Thereby, it should be noticed that the queue's maximum size is completely neglected until this point, which is why the maximum number of packets in the queue has to be restricted to the maximum size of the queue. The maximum number of packets is actually reached right after the arrivals or before the point of departure, as shown in Figure 3.4, so that the maximum size of the queue at time t can be expressed as $n_{max,t} = (t+1)\lambda - \sum_{i=0}^{t-1} \mu_i$. Since the maximum arrival rate λ is supposed to be found, so that no packets are lost, λ has to be maximized with respect to the following constraints:

$$n_{max,t} = (t+1)\lambda - \sum_{k=0}^{t-1} \mu_k^- \leq K \Leftrightarrow \lambda \leq \frac{K + \sum_{k=0}^{t-1} \mu_k^-}{t+1}, \qquad 0 \leq t \leq N \tag{3.27}$$

$$\lambda \leq \frac{\sum_{k=0}^{N} \mu_k^-}{N+1}. \qquad t = N+1 \tag{3.28}$$

Here, $z^- = min(z,K)$, since there cannot be more packets removed from the queue than the queue is long. The first constraint restricts the maximum number of packets in the queue at time t to be less than the queue's length, while the second constraint restricts the queue size to

be zero at the end of the hopping sequence, since K is omitted. The reason for this is that in the next execution of the channel hopping mechanism the same sequence of channels is used which would result in an accumulation of the packets in the queue if it is not empty in the end.

3.1.6. Extension to Mesh Networks

For the extension to mesh networks, a series of nodes $Q = \{q_i, i \in \mathbb{R}^+\}$ with transmission rates $\lambda_i = \lambda(q_i)$ has to be considered. Thereby, the individual nodes q_i build a path from the source of a transmission to the sink. The goal is to find the maximum value for λ_0, since q_0, as the first node on the path to the destination, is the transmitter of the packet itself. Therefore, it applies $\lambda_0 \leq \min_i \lambda_i$, or in other words: the transmission rate of the transmitter is limited by the smallest transmission rate on the path to the destination, since packets are handed from one node to the next and only the node with the smallest transmission rate limits the overall maximum transmission rate.

Besides that one also has to consider the number of neighbours for all nodes q_i on the path to the destination. Assuming that all nodes get same priority and therefore the same time to transmit their packets, the time of a connection event is limited by the number of neighbouring nodes. Thereby, a connection to n nodes, results in a CE length of CI/n. This effectively limits the throughput of all nodes in the network and has to be taken into account, before calculating the λ_i of node q_i.

3.2. IEEE 802.11.4 DSME

For the DSME model, the calculation of the transmission probabilities is a lot simpler, due to the usage of GTSs. Thereby, only the communication in the CFP is considered so that the number of distinct, usable GTSs during a multi-superframe is the limiting factor for the maximum throughput and can be expressed as

$$\begin{aligned} N_{GTS} &= (N_{SPM} - 1) \cdot N_S + 7 \\ &= (2^{MO-SO} - 1) \cdot Slots + 7 \end{aligned} \tag{3.29}$$

where N_{SPM} is the number of superframes per multi-superframe and N_S is the number of slots per superframe and 15 if CAP reduction is enabled and 7 otherwise. The first superframe in a multi-superframe always contains 7 slots. The round trip notation from the BLE model is reused in this section, as well as the calculation of a successful round trip probability as shown in Equation 3.9. On the other hand, the probability of a bit error during a round trip transmission changes slightly since the $AddressAccess$ field does not exist in DSME. Therefore, the bit

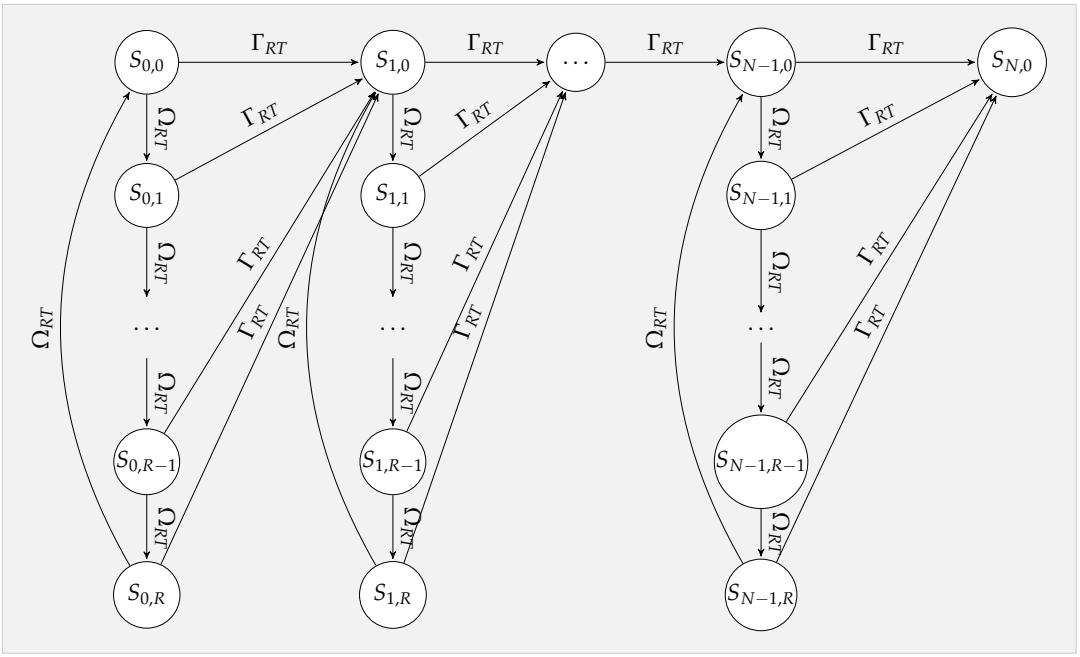


Figure 3.5.: The Markov chain for the retransmission behavior of IEEE 802.15.4 DSME

error probability is still given by $\Omega_{RT} = \Omega_{DATA}\Gamma_{ACK} + \Gamma_{DATA}\Omega_{ACK} + \Omega_{DATA}\Omega_{ACK}$, but with

$$\Omega_{DATA} = 1 - (1 - BER)^{L_{DATA}}, \tag{3.30}$$

$$\Omega_{ACK} = 1 - (1 - BER)^{L_{ACK}}. \tag{3.31}$$

As a results $\Omega_{RT} = 1 - \Gamma_{RT}$. The according Markov chain can be found in Figure 3.5 and describes the retransmission behavior of DSME, which, unlike BLE, does not retransmit packets indefinitely but drops a packet after R failed retransmissions. Thereby, the states $S_{i,0}$ represent the successful transmission of i packets, while the ith tree with the states $S_{i,1}$ to $S_{i,R}$ represent the number of retransmissions after i packet have been delivered successfully. Additionally, the expected number of times to visit state $S_{i,R}$ gives the expected number of packet drops after i successfully delivered packets and can be described by the fundamental matrix

$$N = (I - Q)^{-1}. \tag{3.32}$$

Apart from that, the DSME model works in the same way as the BLE model and also considers the worst case channel schedule of the frequency hopping algorithm. The total base

duration of a superframe $baseSD$ is fixed and can be calculated as

$$\begin{aligned} baseSD &= symbolsPerSlot * N_S * timePerSymbol, \\ &= 60 * N_S * 16\mu s, \end{aligned} \tag{3.33}$$

so that the number of packet per second using DSME can be calculated as $T = \frac{N_{GTS}}{baseSD * 2^{MO}}$. In accordance to the BLE model, the number of available GTSs at a node q_i has to be split on all nodes in the neighbourhood. Since no node is supposed to be prioritizes, it is assumed that all n neighbours of q_i are assigned N_{GTS}/n GTSs.

3.3. Evaluation

Before verifying the accuracy of the theoretical model by using hardware experiments, a short evaluation of it is given with regard to the most important metrics. Thereby, different influences like the bit error rate of the channels, the size of the packets and the size of the transmission queue are examined. Therefore, Section 3.3.3 evaluates the probability to send i packets during a connection event or a multi-superframe respectively and shows the related throughput. After that, the influence of the packets' size is examined and the influence of the transmission queue's size on the maximum throughput is evaluated.

3.3.1. Different Configurations

Different configurations can be used for both protocols, which have a significant influence on their performance. Thereby, the maximum throughput is limited by the length of a connection event in BLE and the superframe order in DSME, as illustrated in Figure 3.6. In addition to that, the superframe order also determines the maximum packet size of DSME that can be sent during a single GTS. As shown in the figure, a higher connection event length and a lower superframe order leads to a higher throughput using BLE and DSME respectively. This is due to the reason that in BLE more packets can be sent during a single connection interval and in DSME more GTSs are available for data transmission. Furthermore, it can be seen that BLE reaches a higher throughput than DSME, although it only utilizes a bandwidth of 1 MHz per channel, in comparison to 2 MHz per channel in DSME. The reason for this is that DSME makes use of O-QPSK with the *Direct Sequence Spread Spectrum* (DSSS) modulation technique, which introduces an overhead for every sent information bit by spreading it over a larger bandwidth. The result is a reduced data rate of 250 kb/s in comparison to 1 Mb/s for BLE, which does not use DSSS and utilizes GFSK for modulation. On the other hand, DSME becomes a lot more robust with a minimal reception sensitivity of -100 dBm [iee16] in contrast to -70 dBm for BLE [ble13]. This also results in a significantly wider transmission

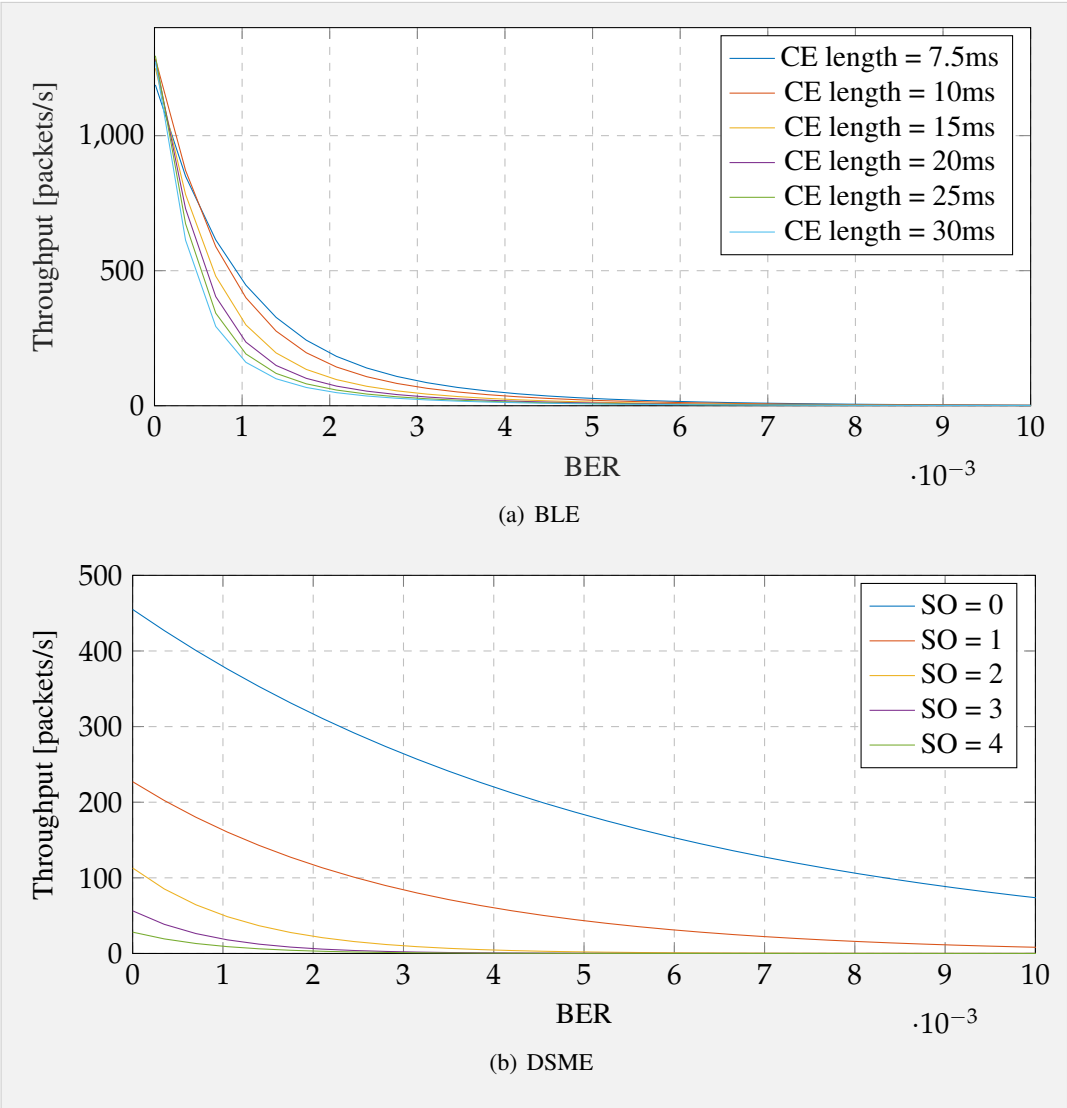
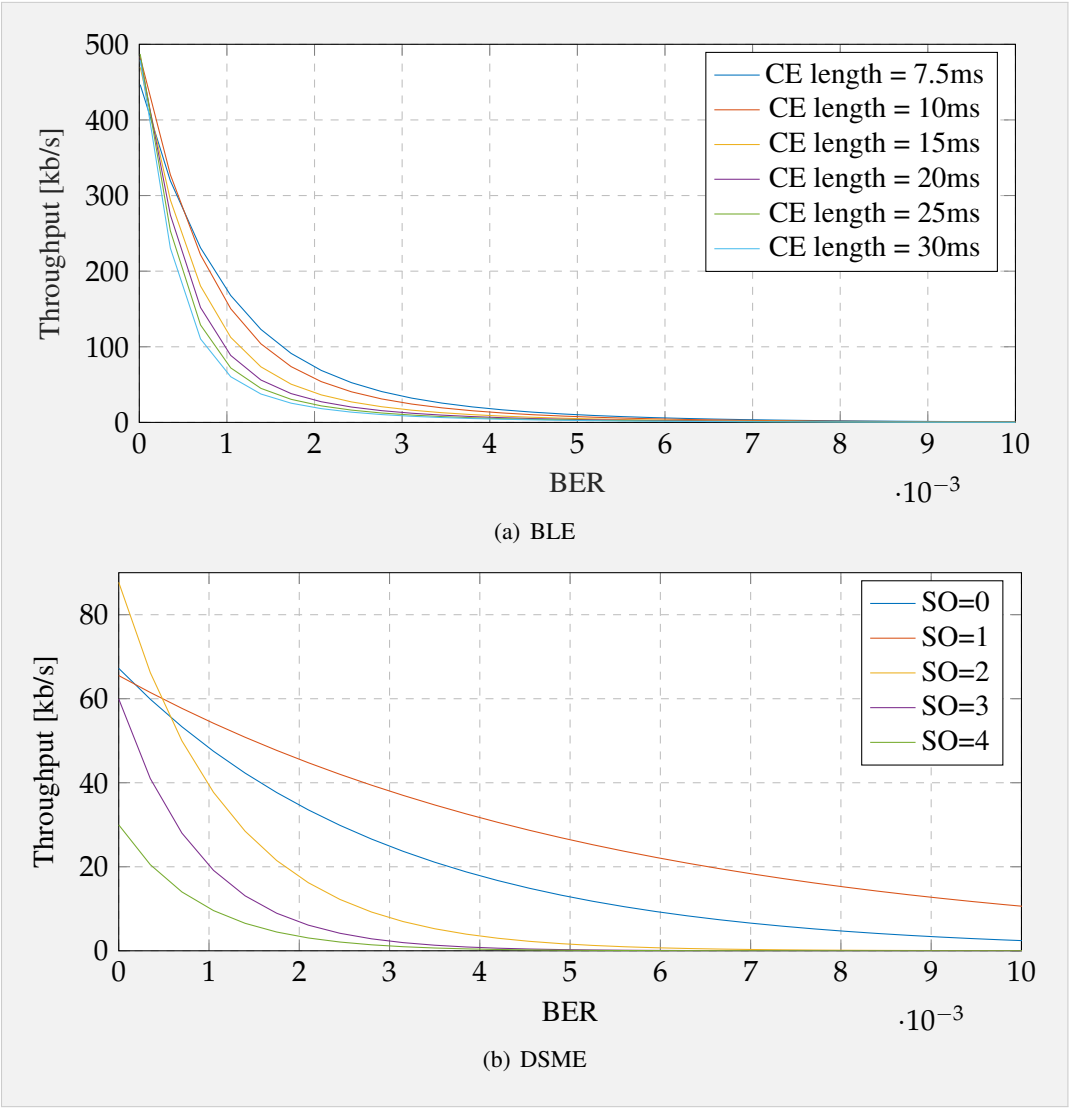


Figure 3.6.: Theoretical limit of packets per second for DSME and BLE for different configurations of the superframe order and connection event length respectively.

range of DSME than of BLE. Therefore, in the same environment, DSME will face a lower BER than BLE.

Figure 3.7 shows the theoretical maximum physical layer throughput of both protocols for different configurations. For BLE, it depicts the exact same picture as before, since the packets per second are only multiplied by the packet size. For DSME, where the maximum size of a packet is also limited by the superframe order, a more mixed image emerges. As one can see, the highest throughput can be achieved with $SO = 2$, but in this case a GTS cannot hold a full size physical layer packet of 133 bytes. Therefore, a superframe order of $SO = 3$ is used for the following evaluation, because even though it offers less throughput in terms of packets per



(a) BLE

(b) DSME

Figure 3.7.: Theoretical throughput limit for DSME and BLE for different configurations of the superframe order and connection event length respectively.

second, it can carry the full packet size. One should just keep in mind that if an application only requires the transmission of small amounts of data, a smaller superframe order can be chosen to increase the maximum throughput. At last, Figure 3.8 presents the throughput in packets per seconds for various bit error rates and an increasing connection interval for BLE. As one can easily see, a higher connection interval and therefore a higher connection event length does not automatically results in a higher throughput in terms of packets per second. Already for a bit error rate of about $3.6 \cdot 10^{-3}$, the throughput decreases with an increasing connection interval, because the connection event is more likely to close prematurely, resulting in fewer packets per second.

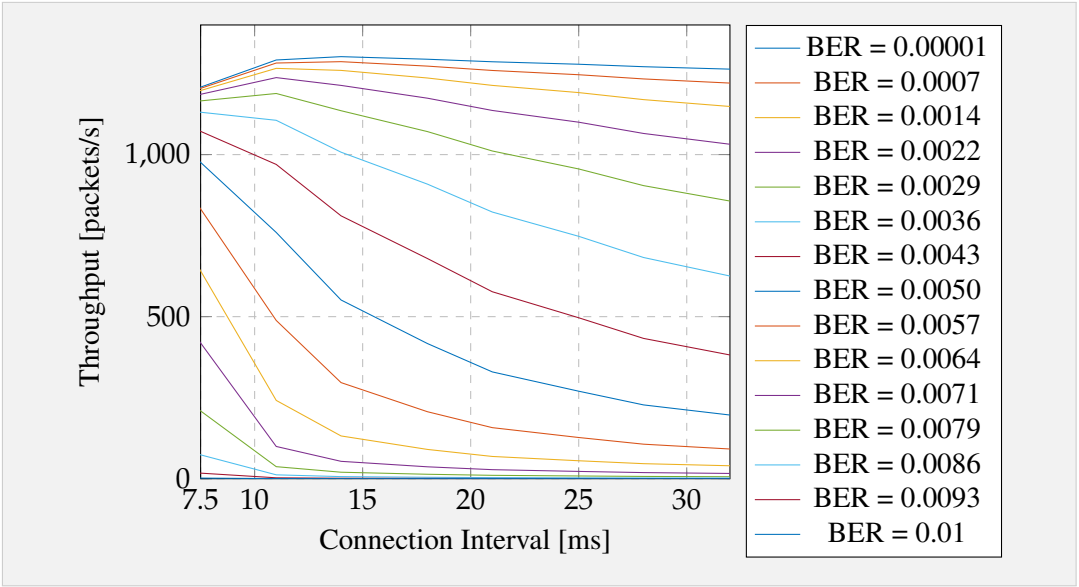


Figure 3.8.: Packets per second for an increasing connection interval and connection event length for different bit error rates for 10^{-5} to 10^{-2}.

3.3.2. Probability Distribution

Figure 3.9 shows the distribution of the probability Ψ_i to send i packets per connection event or multi-superframe respectively. Thereby, the connection event length and multi-superframe length was chosen in a way that a theoretical maximum of 14 packets can be transmitted with both protocols. In Figure 3.10, one can easily see how Ψ_i changes with an increasing BER for BLE. It is obvious, that for a low BER of 10^{-5} the probability to send 14 packets is almost one, while for a high BER of about 10^{-2} the probability to send no packets at all is almost one. This behavior is the same for DSME, as depicted in Figure 3.9 and 3.11. However, for a bit error rate between about 10^{-3} and $4 \cdot 10^{-3}$, also the probability to send one to 13 packets is higher. Surprisingly, the probability for five to ten packets is lower in BLE than for the rest, which is not the case in DSME, where Ψ_i is equally high for every i at least once, depending on the different bit error rates. This is due to the reason that the packet transmission in DSME does not depend on the probability of prior packets, so that a packet is either transmitted or not, with the same probability for every packet. In BLE, on the other hand, it has to be considered that the connection event can be closed prematurely, resulting in the shown probability distribution.

With regard to the probability to successfully transmit i packets, Figure 3.12 shows the resulting average number of packets per connection event or multi-superframe for both protocols. As one would expect, the average number of packets it is the highest for a low bit error rate. Additionally, it can be seen that DSME performs slightly better than BLE, with the result that for a fixed number of 14 packets DSME can transmit more packets at a higher bit

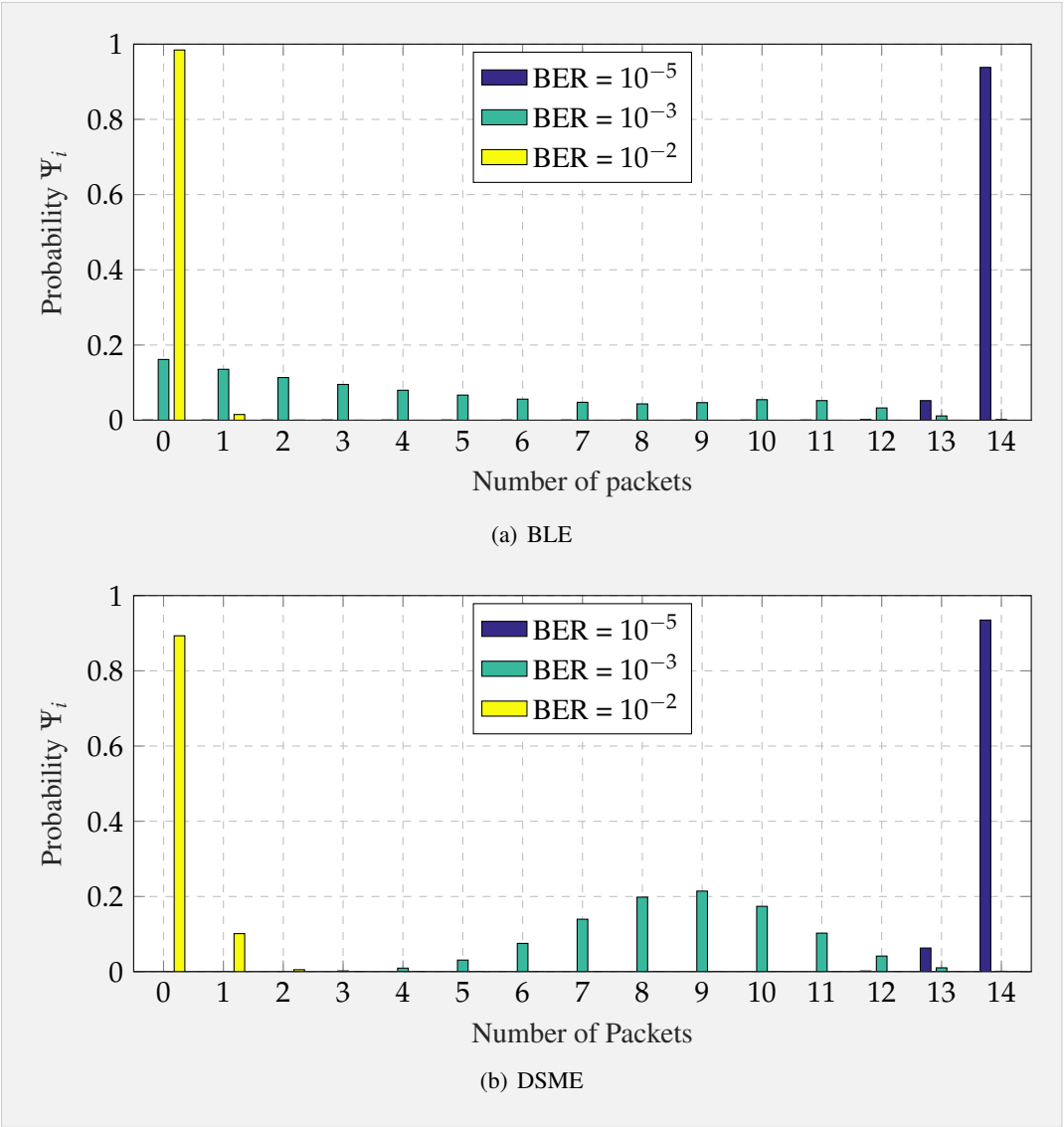


Figure 3.9.: Probability Ψ_i to send i packets during a connection event in for different bit error rates.

error rate. As already mentioned, the reason for this is that the connection events in BLE are more likely to end before all 14 packets have been sent, due to bit errors in two consecutive packets or a bit error in the *AccessAddress* field. This, however, is only the case for a fixed number of packets, as shown in the next section.

3.3.3. Packet size

This section evaluates the influence of different physical layer packet sizes on the performance of BLE and DSME for an increasing BER. Therefore, the sizes of the acknowledgement

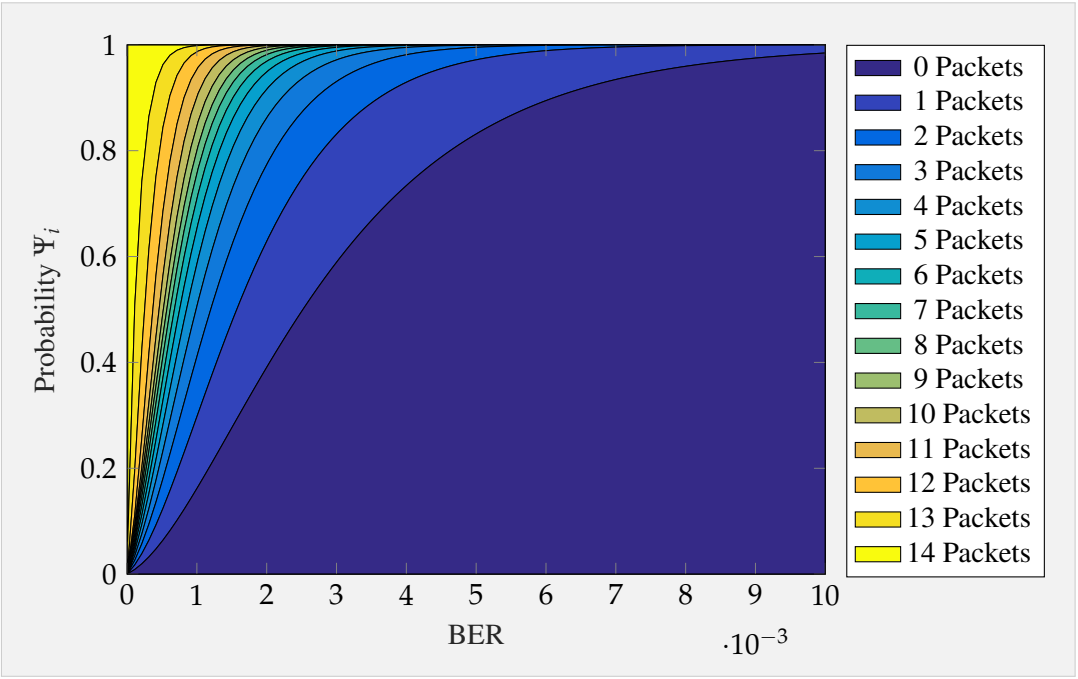


Figure 3.10.: Probability Ψ_i to send i packets during a connection event in BLE for a bit error rate from 10^{-5} to 10^{-2}.

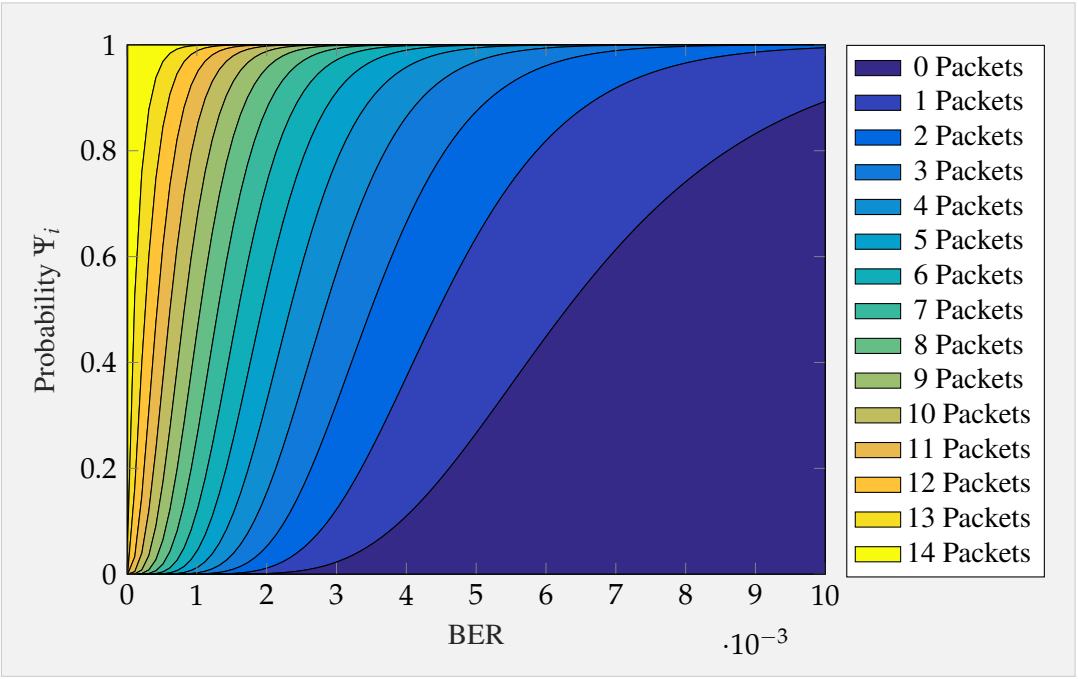


Figure 3.11.: Probability Ψ_i to send i packets during a connection event in DSME for a bit error rate from 10^{-5} to 10^{-2}.

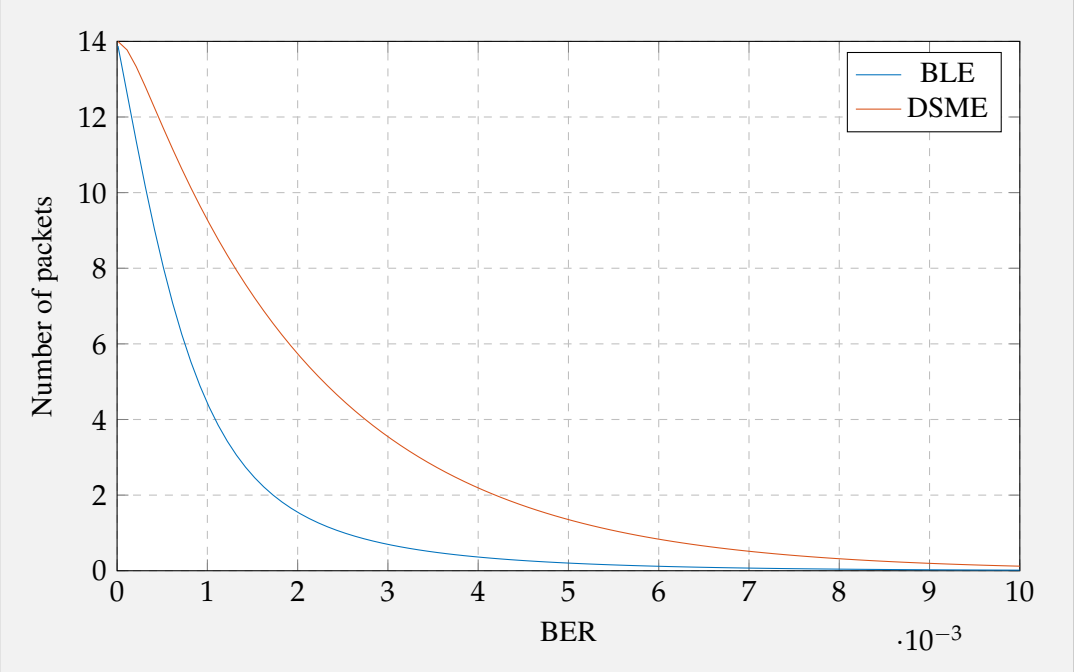
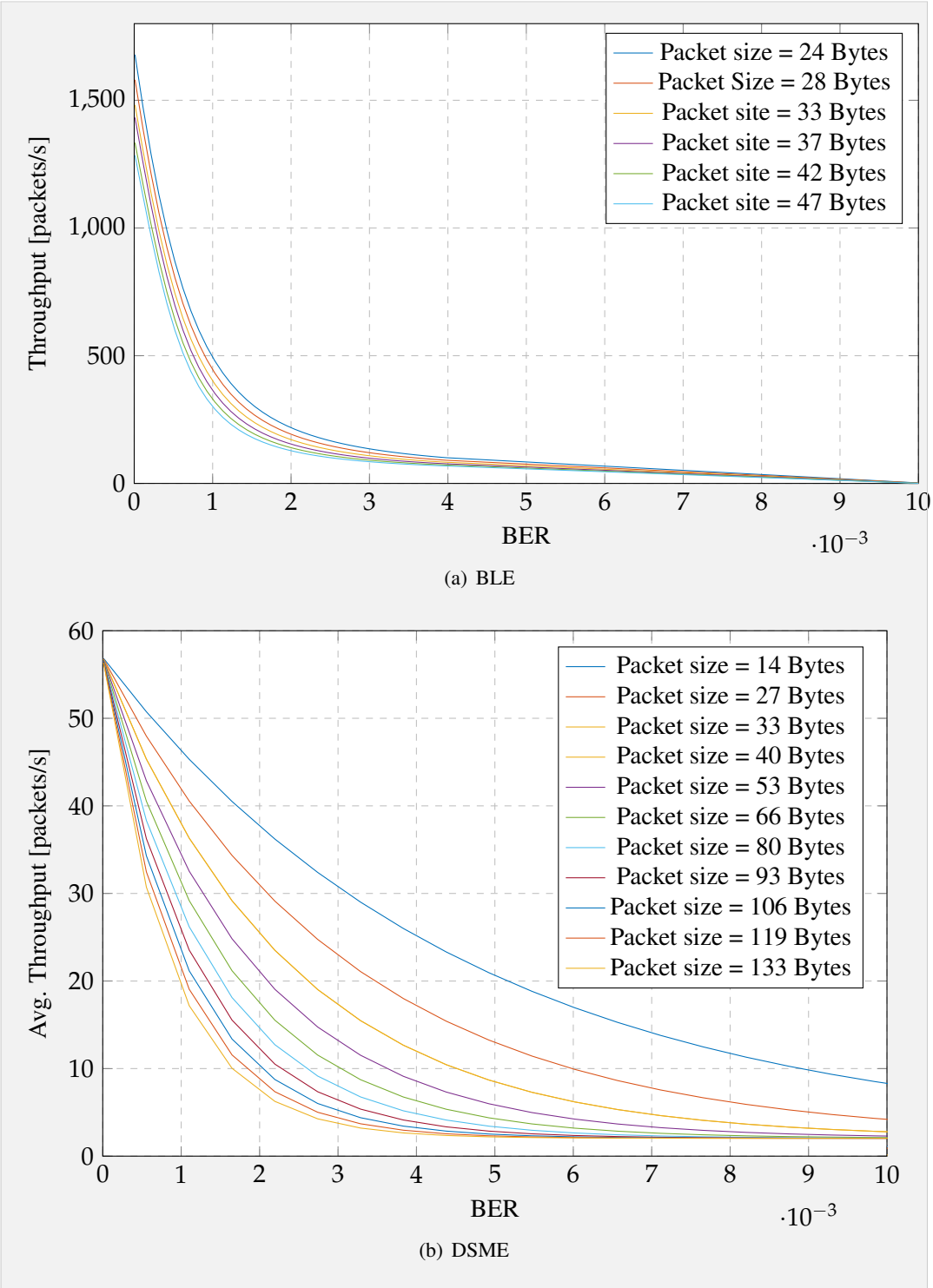


Figure 3.12.: Comparison of the average number of transmitted packets for BLE and DSME with a theoretical limit of 14 packets and a bit error rate from 10^{-5} to 10^{-2}.

packets are fixed to the smallest possible size and different data packet sizes are considered, as shown in Figure 3.13. As one can easily see, the physical layer throughput of BLE is significantly higher than for DSME in terms of packets per second, simply because DSME is limited by the number of available GTS slots, while in a connection event, the packets can be sent one after another. Apart from that, both protocols show the same behavior, with a higher packet size leading to a lower throughput, because the bigger packets naturally contain more bits so that the probability of an error in the packet is higher. In contrast to DSME, the throughput of BLE for a very low bit error rate of 10^{-5} is not the same for all packet sizes. That is, because the packet size influences the number of packets per connection interval and therefore also the number of packets per second.

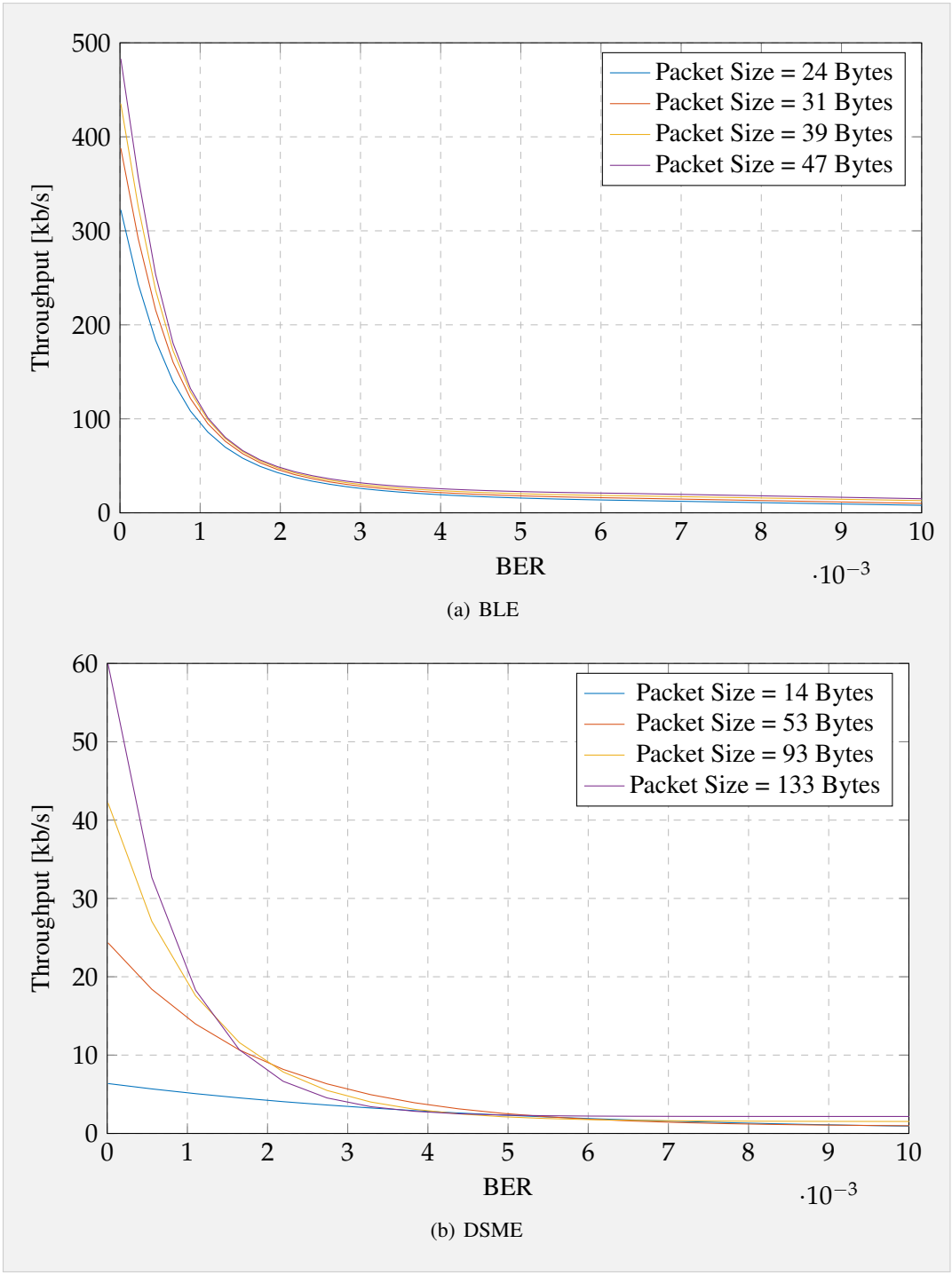
The physical layer throughput in kb/s is depicted in Figure 3.14. For BLE, the resulting throughput shows the same behavior as already discussed - the higher the packet size, the lower the throughput. For DSME on the other hand, which supports significantly higher packet sizes in comparison to BLE, the results are more interesting. Figure 3.14 shows that the maximum achievable throughput is not only dependent on the BER but also on the packet size. For a BER between $2 \cdot 10^{-3}$ and $5 \cdot 10^{-3}$, the throughput for a packet size of 53 Bytes is higher than for a packet size if 133 Bytes. Afterwards, however, it sinks below the throughput of the higher packet size again. The reason for this is that for a certain bit error rate, the probability that a small packet arrives is higher than the probability that a big packet arrives, as shown in



(a) BLE

(b) DSME

Figure 3.13.: Throughput in packets/s for an increasing bit error rate from 10^{-5} to 10^{-2} and different physical layer packet sizes.



(a) BLE

(b) DSME

Figure 3.14.: Physical layer throughput for different physical layer packet sizes.

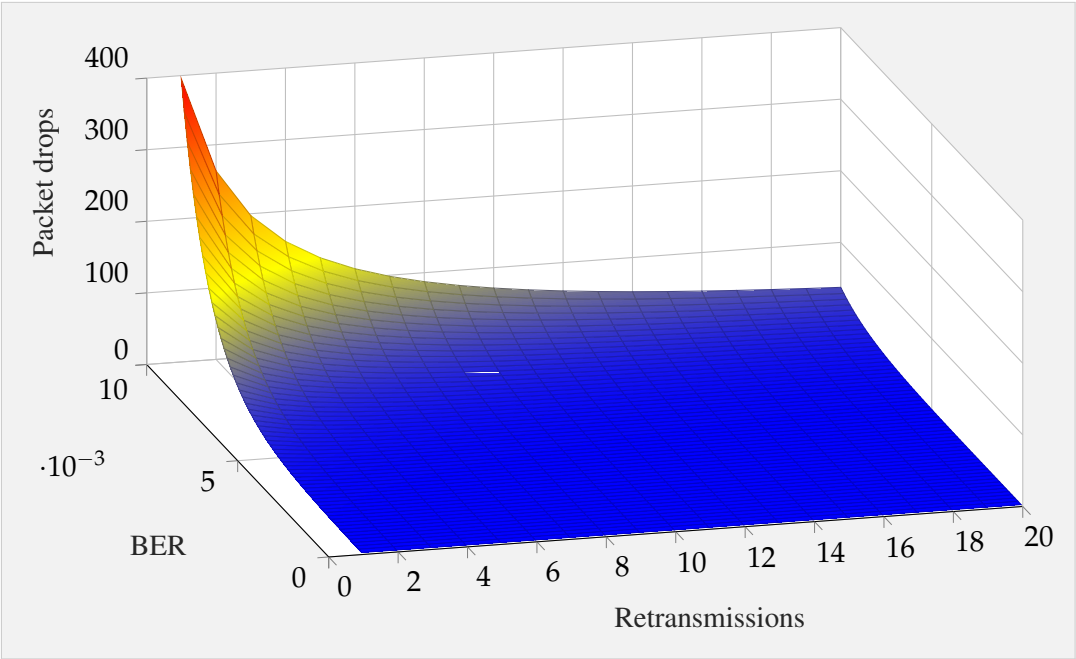


Figure 3.15.: Number of drops before a packet is successfully delivered in DSME for an increasing BER and increasing number of retransmissions.

the last section. In the end, however, this effect flattens, so that big packets that occasionally arrive successfully still provide a higher throughput. Therefore, the model can be used to find an optimal packet size for a given BER.

3.3.4. Dropped Packets

As discussed in Section 3.2, the Markov chain for DSME can also be used to estimate the expected number of times a packet is dropped before successfully arriving at the receiver for a given number of retransmissions. Therefore, it is assumed that the application immediately resends a packet, as soon as it is dropped. To model this behavior, a Markov chain with two main states is used, namely zero packets arrived and one packet arrived. In addition to that, the Markov chain contains a number of states for the retransmissions. The results are depicted in Figure 3.15 and give an indication of how high the maximum number of retransmissions has to be set for a certain bit error rate, so that no packets are dropped. It is obvious that the highest number of drops, before a successful transmission, occurs for a high bit error rate and a low amount of retransmissions.

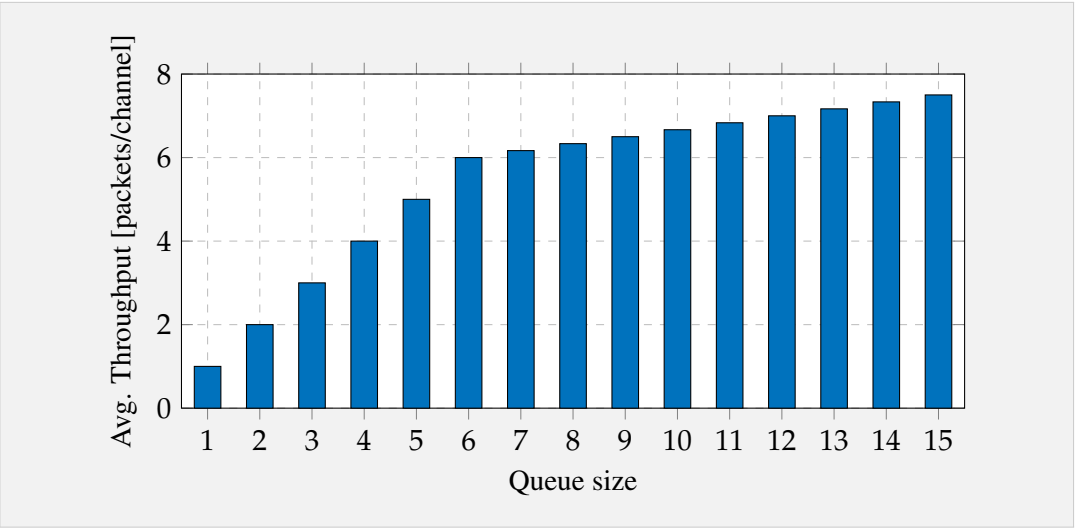


Figure 3.16.: Average throughput in packets per channel for an increasing queue size and a worst case hopping sequence with $\mu_i = \{6, 6, 6, 6, 6, 10, 10, 10, 10, 10\}$.

3.3.5. Queue Size

At last, the impact of different sizes of the transmissions queue on the maximum achievable throughput is evaluated for an exemplary hopping sequence consisting of ten channels with the service rates $\mu_i = \{6, 6, 6, 6, 6, 10, 10, 10, 10, 10\}$, symbolizing a sequence of five bad and five good channels. Figure 3.16 shows the maximum arrival rate λ to the queue, so that no packets are lost due to queue drops. One can easily see that the size of the queue has a significant impact on maximum λ. Thereby, the maximum arrival rate scales linearly with a queue size of up to six packets, because until this point, it is the only limiting factor. The lowest service rate is six packets on the first five channels, so that all packet that arrive to the queue can be immediately removed and the queue size is effectively zero after every transmission on a specific channel. After that, the bad channels, with a service rate of six, start to limit the average throughput, so that packets have to be queued after the transmission on a single channel, since the arrival rate is higher than six packets. Additionally, the figure illustrates that a further increase in the queue size results in a higher throughput, because more packets can be buffered.

Another noticeable effect is that the maximum number of packets per channel still rises when the buffer is larger than the largest μ_i, which is ten in this example. That is because the only constraint at the end of the queue is that the queue must be empty, but the throughput can theoretically still be increased until ten packets are in the queue in the last step of the hopping sequence.

Chapter 4

Experiments

In the course of this thesis, a couple of hardware experiments have been conducted to evaluate the performance of Bluetooth Low Energy and IEEE 802.15.4 DSME in the presence of external interference. Thereby, the experiments provide a reference point for the accuracy of the theoretical model from Chapter 3 and introduce different improvements on the BLE application layer. The following sections first describe the experimental design, followed by a brief BLE parameter study to achieve the best possible results. After that, results are presented for the different experiments and compared with the results of the theoretical model. At last, two additions on the applications layer of BLE are presented in the form of a flow control and an adaptive frequency hopping algorithm to increase the performance and predictability of BLE in the presence of external interference.

4.1. Experimental Design

The hardware experiments were conducted in different rooms of the Institute of Telematics at TUHH as depicted in Figure 4.1. For the Bluetooth experiments, measurements were performed using STM32F401RE evaluation boards with X-Nucleo-IDB05A1 extension boards, which feature the BlueNRG-MS network processor, conforming to BLE in the v4.1 specification. The BLE module offers support for multiple GAP roles, so that devices can be configured as master and slave at the same time and mesh networks can be constructed. For the DSME evaluation, the BLE devices were replaced by custom made boards with an ATmega256RFR2 chip. Additionally, openDSME was used, an open source implementation of the DSME protocol, developed at the Institute of Telematics [KKLT17]. A superframe order of three and a multisuperframe order of 5 is chosen for the hardware experiments to enable the transmission of a physical layer packet with a size of 133 bytes. To control the level or external interference during the experiments, additional boards were used to artificially disturb certain frequencies during the experiments.

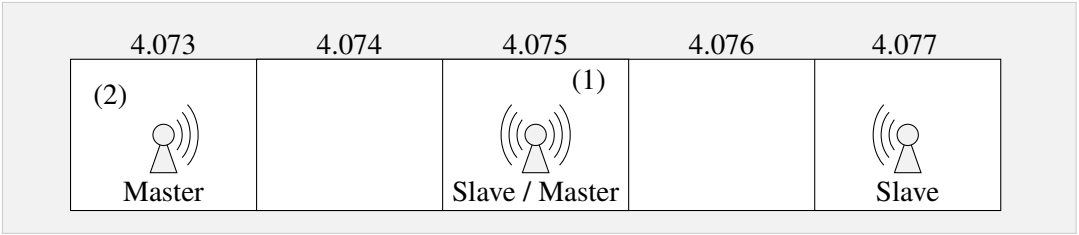


Figure 4.1.: Experimental setup for the hardware experiments.

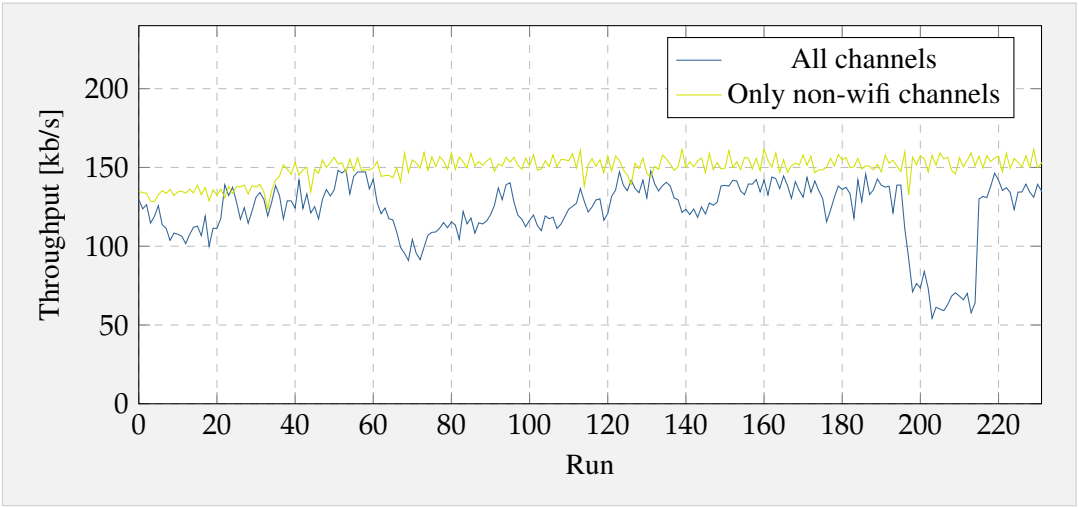


Figure 4.2.: The influence of Wi-Fi interference on Bluetooth Low Energy throughput.

For both protocols, the set of used channels had to be reduced, so that for Bluetooth Low Energy only nine out of the total 37 available data channels were used and for DSME four out of 16 available channels. This has mainly two reasons: On the one hand, an artificially created interference on one of the 37 or 16 data channels, respectively, does not yield any significant differences in the results, since the interference is simply not big enough. On the other hand, most of the frequencies of the two protocols interfere with the Wi-Fi frequencies. There was no way to turn of the Wi-Fi signal during the experiments, because the routers are inaccessible and supply the whole corridor. For that reason only channels that are not interfering, as shown in Figure 2.4 for Bluetooth Low Energy, were chosen for the reduced channel set. Figure 4.2 exemplarily shows the influence of Wi-Fi interference on the throughput of Bluetooth Low Energy. Therefore the throughput was once measured with the reduced, not interfering channel set and once with all 37 channels. As one can easily see, the throughput with all channels is heavily and unpredictably fluctuating so that the influence of artificial interference can not be distinguished from the Wi-Fi interference. The reduced set of channels is given by {9,10,21,22,23,33,34,35,36} for BLE and {15,20,25,26} for DSME.

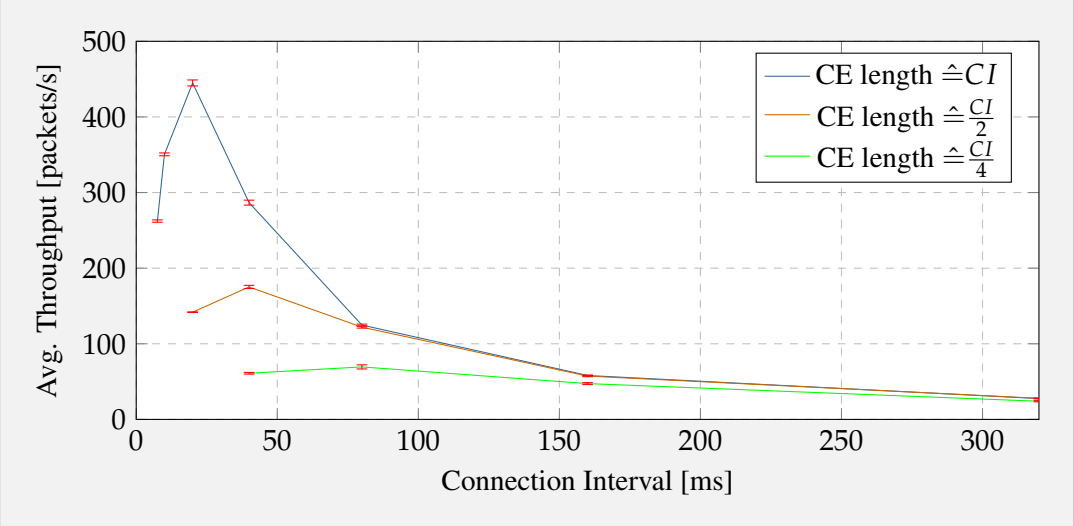


Figure 4.3.: Bluetooth Low Energy throughput for a varying connection interval and connection event length.

4.2. Bluetooth Low Energy Parameter Study

The parametrisation of Bluetooth Low Energy is a well researched area in the literature [GOP12, LCM12, AKBBR16], especially with regard to its advertising and connection process. Most of these studies, however, only provide theoretical results and lack the verification on hardware platforms. Therefore, this section analyses the influence of the connection interval and connection event length on the performance of BLE using the analytical model from Chapter 3 and hardware experiments.

For the evaluation in this section, two BLE devices were placed in adjacent rooms, as illustrated by position (1) and (2) in Figure 4.1. The device at position (2) was configured as the source of the transmission, which constantly sent data packets to the device at position (1) in a way that no packets are lost due to queue drops. To ensure that the choice of the optimal parameters for BLE is not affected by the influence of external interference, no jamming devices were used in this setup.

At first, one might assume that a longer connection event is directly related to a higher throughput, because more packets can be sent in a single event. However, this is not necessarily the case, as one can easily see in Figure 4.3. Here, e.g. for a CE length $\hat{=}$ CI, the throughput first rises until a connection interval of 20 ms and then rapidly falls with an increasing connection interval. The reason for this is that a high bit error rate often leads to preliminary termination of the connection event and therefore fewer packets sent. The next connection event starts again after the beacon interval but in the time between the termination of the connection event and the start of the next connection interval, no packets can be sent. Therefore,

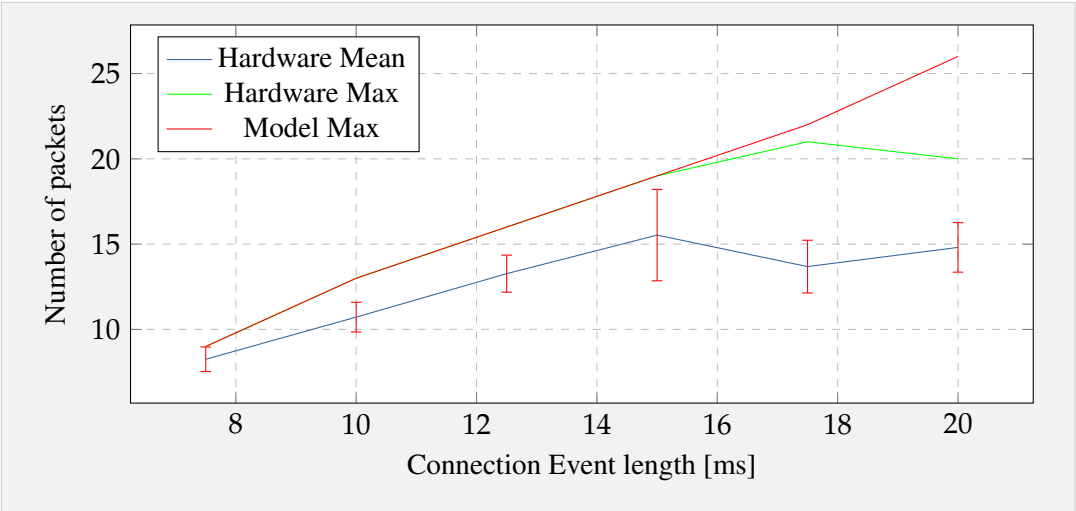


Figure 4.4.: Transmitted packets per connection event for an increasing connection event length.

the throughput decreases exponentially for a medium to high bit error rate, as also illustrated by the results of the model in Figure 3.8.

Additionally, Figure 3.2 shows that reducing the connection event length also leads to a significantly reduced throughput. For example for a connection interval of 20 ms and a reduction of the CE length from CI to $\frac{CI}{2}$, the throughput drops by about 300 packets/s. On the other hand, the difference disappears for high connection intervals and connection event lengths, as one can exemplary see at a connection interval of about 160 ms. Here, the connection event is closed prematurely again so that the BER limits the overall throughput. In Bluetooth Low Energy, it is not possible to choose the connection event length completely free, but the connection interval is split between all connected devices. Thereby, every device usually gets the same connection event length so that for a connection interval of 160 ms it would not matter if only one device is connected with a connection event length equal to the connection interval or 4 devices share the connection interval - the throughput of every single device stays the same.

At last, it can be seen that a connection event length of 20 ms yields the highest throughput for all configurations on the hardware platform. Therefore, this CE length is chosen for all further experiments to maximize the throughput and the connection interval is kept as low as possible so that all devices in the experiment can be connected.

4.3. Packets per Connection Interval

Section 3.1.1 provides a way to calculate the theoretical limit of packets per connection event. Since this is the foundation of the theoretical model, it is verified on the hardware platform.

This is especially important, because, as already mentioned, the number of transmittable packets is often lower than the model suggest due to processing delays. The results for the model and the hardware experiments are shown in Figure 4.4.

For the hardware experiment, two BLE boards were placed right next to each other to mitigate the influence of external interference. Additionally, no jamming device was used to artificially disturb certain frequencies. It is easy to see that the model fits the hardware results up to a connection interval of 15 ms. After this, the maximum number of packets becomes almost constant on the hardware, which probably happens due to the influence of external interference. Another indicator for this is that also the mean number mean number of packets per connection event stagnates after about 15 ms. Therefore, the model seems to describe the maximum number of packets per connection interval perfectly in ideal conditions.

4.4. Global Interference

To assess the influence of external interference on BLE and DSME, both protocols were evaluated on the hardware platforms with an experimental setup as depicted in Figure 4.1. Therefore, a number of three devices was used, whereof the rightmost device in the figure is configured as the source of the transmission, which constantly sends data packets to the device at position (2) in a way that no packet losses occur. The only task of the middle node in this setup is to forward packets. For the evaluation of DSME, the sink of the transmission had to be placed at a larger distance of about 20 meters from the middle node on the corridor in front of the test rooms. This is necessary to ensure that the source and the sink of the transmission do not establish a direct connection. It was not possible to place the sink for the Bluetooth experiments at the same position, due to its limited range in comparison to DSME. Therefore, one should keep in mind that the results for the DSME experiments do not only suffer from artificially created interference, but also strengthened from other effects like unwanted external interference or fading, affecting the comparison of the two protocols. Apart from that, the idea of this experiment is to created a global influence of interference, affecting all links in the network, by placing the source of the interference next to the center node. This allows the evaluation of the performance of the frequency hopping algorithm of both protocols. The throughput, average packet delay and packet reception ratio were measured, for an increasing amount of interfered channels.

To verify that both links are subject to the same amount of interference, the average transmission time of a single packet and the average throughput were measured for both links individually using BLE. The results are presented in Figure 4.5, which also gives an indication about the performance of each link. As one can easily see, the single hop throughput decreases almost linearly with an increasing number of interfered channels, while the average packet

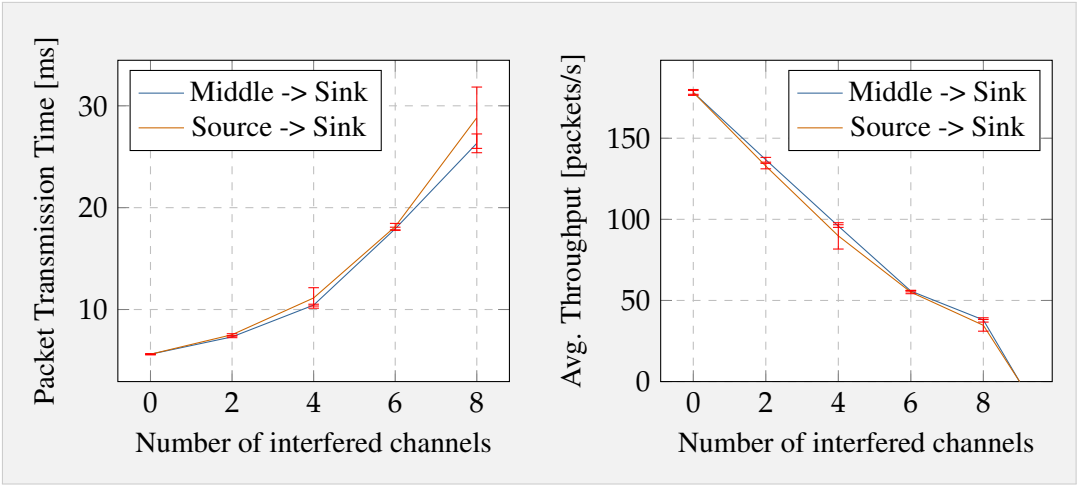


Figure 4.5.: Packet transmission time and average throughput in packets per second of the two links for an increasing number of interfered channels in BLE.

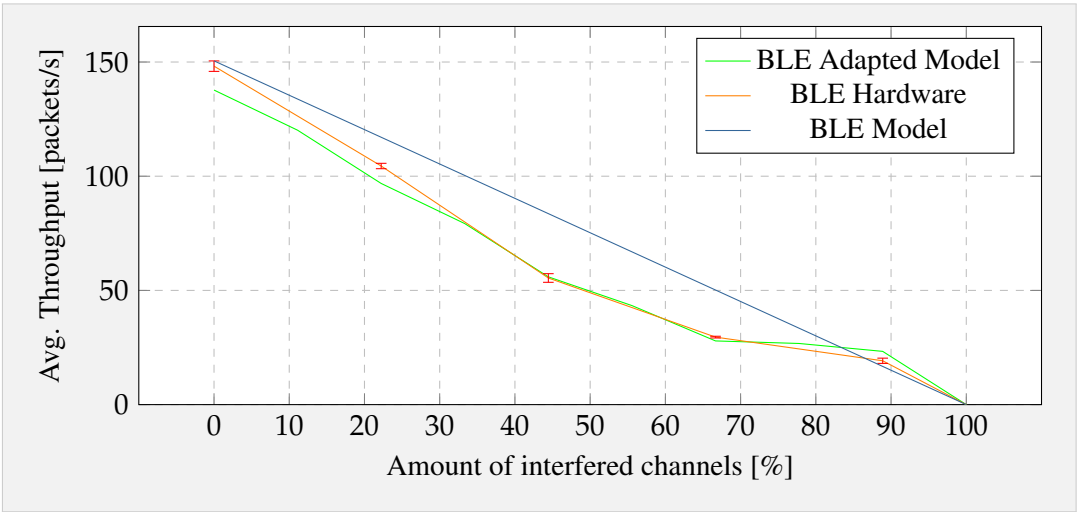


Figure 4.6.: Comparison of the measured packets per second for BLE with the theoretical model.

transmission time seems to increase exponentially. Thereby, the small spike for a number of eight interfered channels is due to the different bit error rates of the links. The obtained results from the model show the same characteristics as the hardware measurements, supporting an average bit error rate of about $1.32 \cdot 10^{-3}$ that was measured for the hardware experiments using BLE.

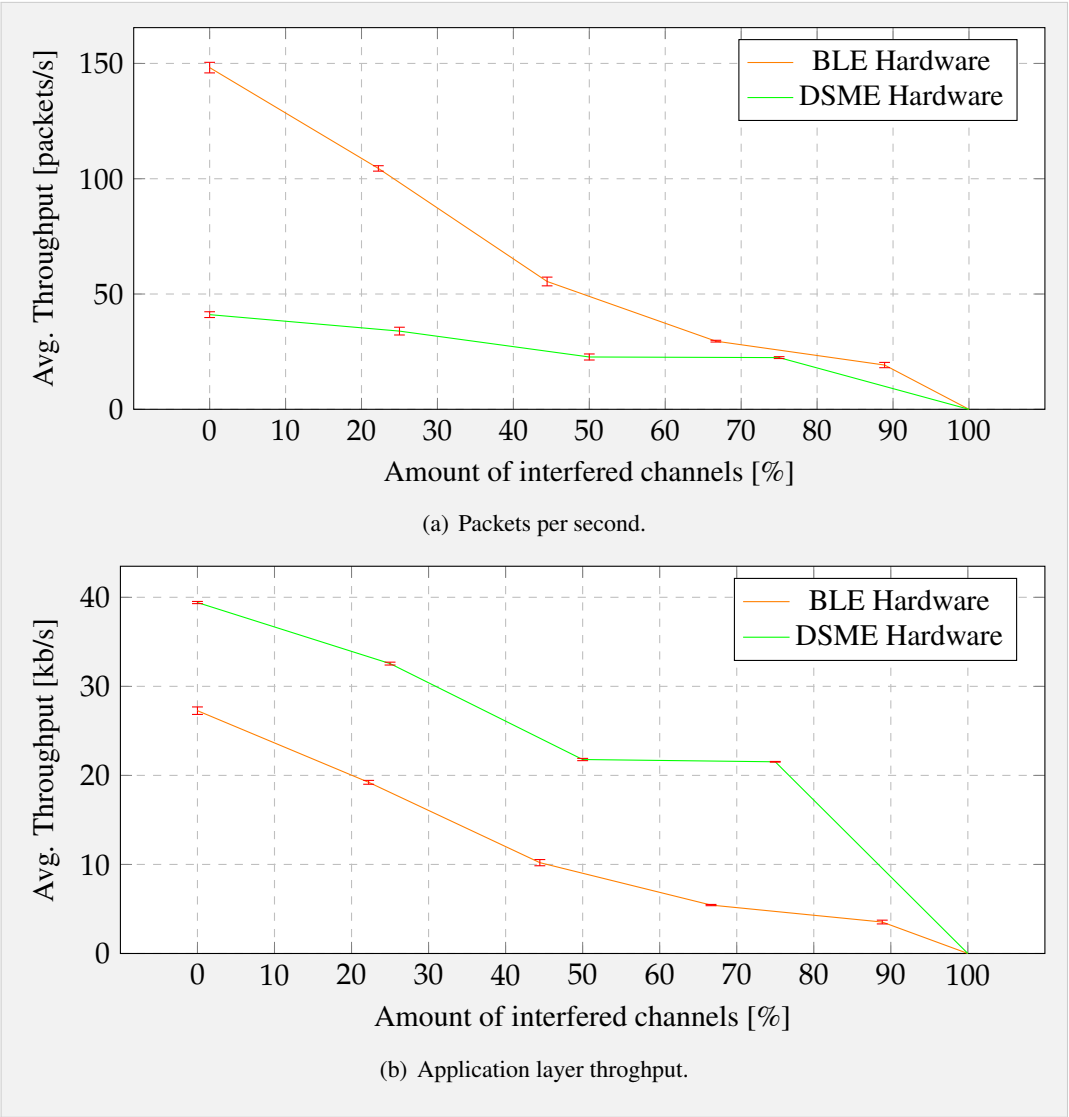
Figure 4.6 shows the measurements of the average throughput for BLE and compares them with the results of the model. As one can easily see, the results of the regular model reflect the trend of the measurements, but do not exactly match them. This is due to the reason that, for sake of simplicity, all non interfered channels are assigned the same BER, resulting in a linear

CH9	CH10	CH21	CH22	CH23
$1.259 \cdot 10^{-3}$	10^{-3}	$1.259 \cdot 10^{-3}$	10^{-3}	10^{-3}
CH33	**CH34**	**CH35**	**CH36**	
$1.585 \cdot 10^{-3}$	$1.380 \cdot 10^{3}$	$5.012 \cdot 10^{-3}$	$3.162 \cdot 10^{-3}$	

Table 4.1.: Measured bit error rates for the used Bluetooth Low Energy channels.

decrease of the throughput. Since the channels are subject to different levels of interference, that are not created on purpose for this work, one has to take different bit error rates into account. The measurement of the BER on hardware, however, is tedious, so that it has only been exemplarily done for BLE. Therefore, Table 4.1 shows the different BERs of the used BLE channels and the resulting throughput of the adapted model is shown in Figure 4.6. For the experiment the number of interfered channels was gradually increased in steps of two, so that the channels were reduced in the order {9,10}, {21,22}, {33,34} and {35,36}. As a result, channel 23 remains the only channel without interference. One can easily see, that the accuracy of the model is drastically increased by taking a distinct bit error rate for every single channel into account. The drawback, however, is that the bit error rate of every single channel has to be known beforehand. Nevertheless, one can say that the hardware measurements validate the accuracy of the theoretical model for BLE.

Despite the small inaccuracies, Figure 4.7(a) provides a good comparison between the performance of BLE and DSME on the hardware. As illustrated, the average number of packets per second is about three times higher in BLE than in DSME, which is also supported by the model. Furthermore, both protocols scale linearly with the number of of interfered channels, although it has to be said that for BLE a total number of nine channels was available, while for DSME only four channels were tested, as already described in Section 4.1. Thereby, it can be seen that the throughput for DSME is sinking slower with the number of interfered channels than BLE. The reason for this is that in DSME only a single packet is lost if an interfered channel is selected by the channel hopping algorithm. In BLE, on the other hand, all packets that could be sent in a connection event are lost and the device has to wait until the next connection event to resend them. However, the physical layer packet size and therefore the transmittable amount of user data varies greatly between the protocols, with 23 bytes of user data for BLE and up to 127 bytes for DSME, Figure 4.7(b) shows the maximum application layer throughput of both protocols. It can be seen that even though a lot less packets were transmitted in DSME for a maximum packet size, it provides a significantly higher application layer throughput with about 10 kb/s more.



(a) Packets per second.

(b) Application layer throghput.

Figure 4.7.: Comparison of the measured throughput for DSME and BLE on the hardware platform.

Delay

During the experiments, also the delay of BLE and DSME has been measured for a global source of interference over a single hop. Thereby, the delay is especially important in industrial applications, because small delays are essential to control remote applications or retrieve critical data from sensors before it becomes invalid. Therefore, Figure 4.8 shows the average delay of BLE and DSME for an increasing amount of interfered channels.

As one can easily see, the delay for BLE starts with a value of about seven milliseconds and slowly increases for until a number of six interfered channels, where is rapidly starts to rise.

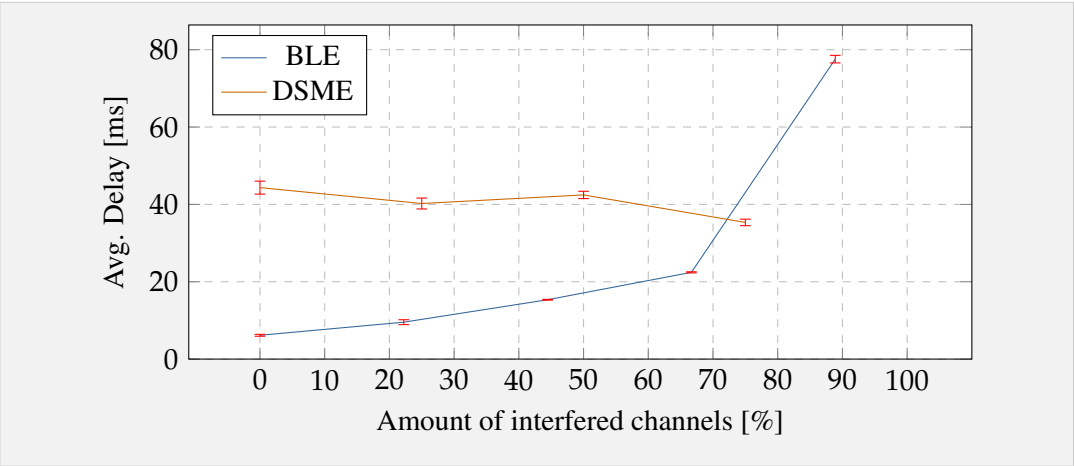
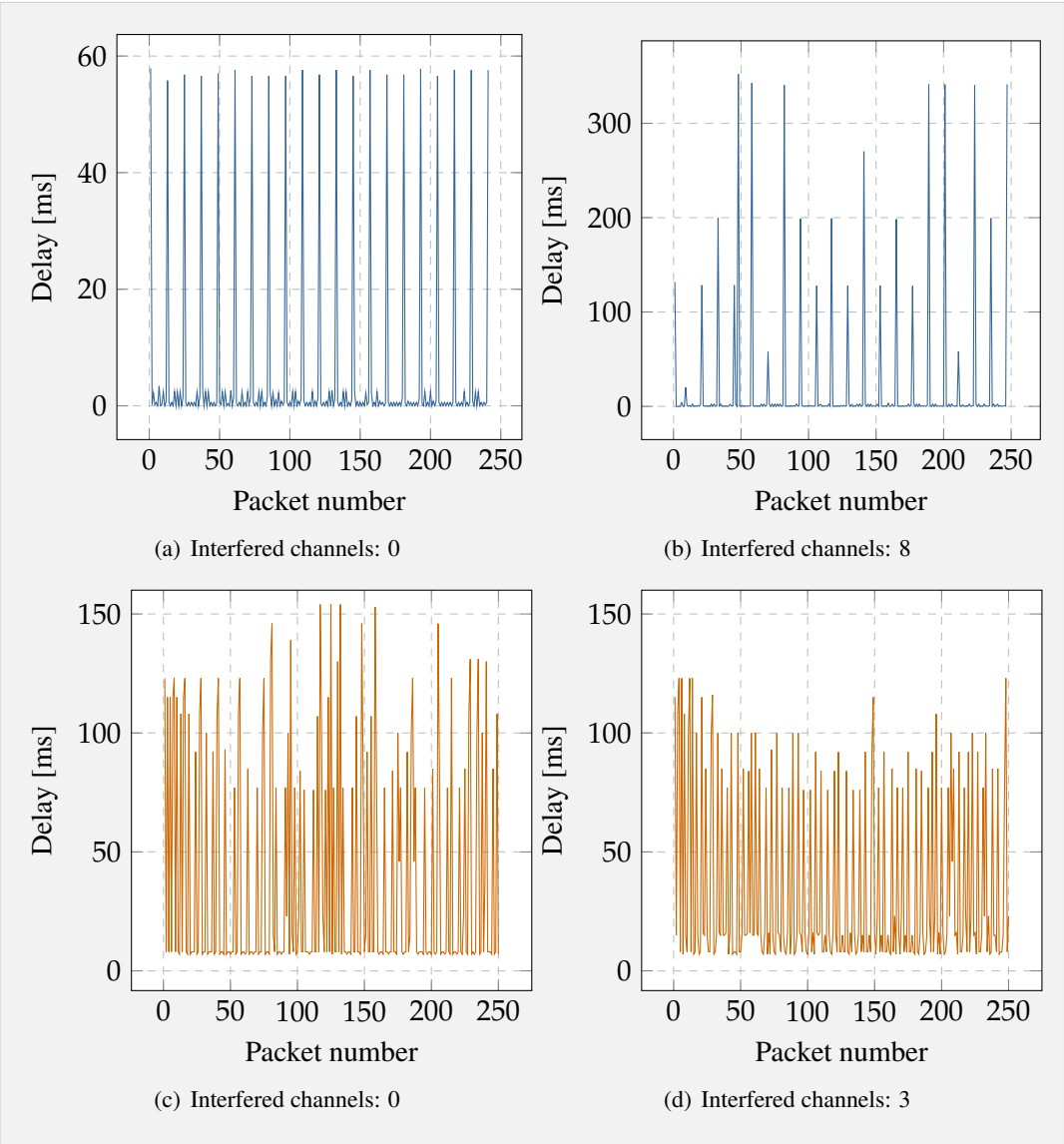


Figure 4.8.: Comparison of the average delay of BLE and DSME.

This is due to the fact that for a number of eight interfered channels, only a single channels is left for transmission, while for example for six interfered channels, three are still left. Since those three channels probably do not follow one after another in the hopping sequence, the time until the next non interfered channel is significantly reduced in comparison to the other case, where the whole hopping sequence has to be executed until a packet can be sent again. It is clearly visible that DSME follows a completely different behavior. Here, the average delay stay almost constant around a value of 40 milliseconds. This is due to the reason that for the experiment the maximum available number of 28 GTSs was allocated so that a packet can be sent in each of the slots. The time until the next slot on an uninterfered channel is therefore rather short in comparison to BLE.

In addition to the average delay, also the variation of the delay between two consecutive packets over time was measured for a number of zero and eight interfered channels for BLE and zero and three interfered channels for DSME. The results for the first 250 packets are presented in Figure 4.9 and show how the data transmission in Bluetooth Low Energy works. For the experiment a connection interval of 40 milliseconds and a connection event length of 20 milliseconds was chosen. In Section 4.3, it has been shown that the average number of packets per connection interval in such a configuration is about 15 packets. This is also illustrated in Figure 4.9 by the periodic spikes, which occur about every 15 packets. At these, the connection event is prematurely terminated due to the corruption of a data packet or repetitive packet loss and the next packet only arrives at the start of the next connection event. As long as no error occurs, however, BLE guarantees low packet inter arrival times of only a few milliseconds and even in the case of the premature termination of a connection event the delay stays below 60 milliseconds, which is often enough for the application in industrial applications. For a number of eight interfered channels, however, as shown in Figure 4.9(b),



(a) Interfered channels: 0

(b) Interfered channels: 8

(c) Interfered channels: 0

(d) Interfered channels: 3

Figure 4.9.: Variation of the delay over time for Bluetooth Low Energy (a)(b) and DSME (c)(d).

the delay spikes are much higher. As one can see, there are still about 15 packets that are successfully delivered with a low delay per connection interval, but once the connection event is terminated, the next packet only arrives after a huge delay since the other channels are interfered and do not allow the transmission of packets after a huge delay, since the other channels are interfered and do not allow the transmission of packets. However, the median delay is the same for both with a value of 600 microseconds. Therefore, one can say that BLE tries to mitigate the influence of external interference by sending as many packets as possible, when a good channel is found.

For DSME, on the other hand, the delay is fluctuating a lot more, as depicted in Figure 4.9(c) and 4.9(d). A reason for this could be the increased packet size of DSME in comparison with BLE and the therefore increased vulnerability to the influence of external interference. Additionally, it is obvious that long packets naturally need more time for transmission. During the measured period, the absolute delay stays below 150 milliseconds for both measurements and it is easy to see that the absolute values are not significantly increasing for a higher number of interfered channels, as it was the case for BLE. This is due to the fact that during the experiments the totally available amount of 28 GTSs was allocated, so that a packet can be sent in every time slot. As a result, delays are kept short in comparison to BLE, where it has to be waited until the next connection event to retransmit the packets. In fact, only the lower bound of the delay seems to be shifted up for DSME, as it can be seen at the bottom of the two orange graphs. This is reflected by the median of the delay of over the observed period, which is 8 milliseconds for zero interfered channels and 15 milliseconds for three interfered channels.

4.5. Bidirectional Traffic

A lot of traffic in industrial applications consists of unidirectional data transmissions. However, at some point the bidirectional transmission of data is required, may it be for control messages or the communication with monitoring devices. Therefore, Figure 4.10 shows the throughput over two hops for a bidirectional transmission of data. The experimental setup is chosen, as presented in the last section and the two outer devices exchanges packets with the maximum size of 47 bytes for Bluetooth Low Energy. As a result of the increased packet size, the maximum number of packets for the used connection interval of 20 milliseconds decreases from 26 to 19.

An important information is the fact that not every packet needs to be acknowledged for both sides. Instead, the acknowledgements are carried in the header of regular data packets, as already mentioned, so that effectively only packets with the full size are exchanged. The figure shows that an increase of the packet size results in a significant lower throughput of about 10 packets/s for eight interfered channels up to 68 packets/s for no interfered channels at all. However, the reason for this is not only the reduced number of packets that can be send per connection interval, but also the increased vulnerability of the packets to the influence of external interference due to their increased size. To demonstrate this, also the throughput for only one byte of user data per packet is shown. For this result, the number of packets per connection event was artificially limited by setting it to the same number as for the 47 byte packets. Therefore, the only difference between those configurations is the packet size and as shown, the average number of packets per second is much higher for a smaller packet size.

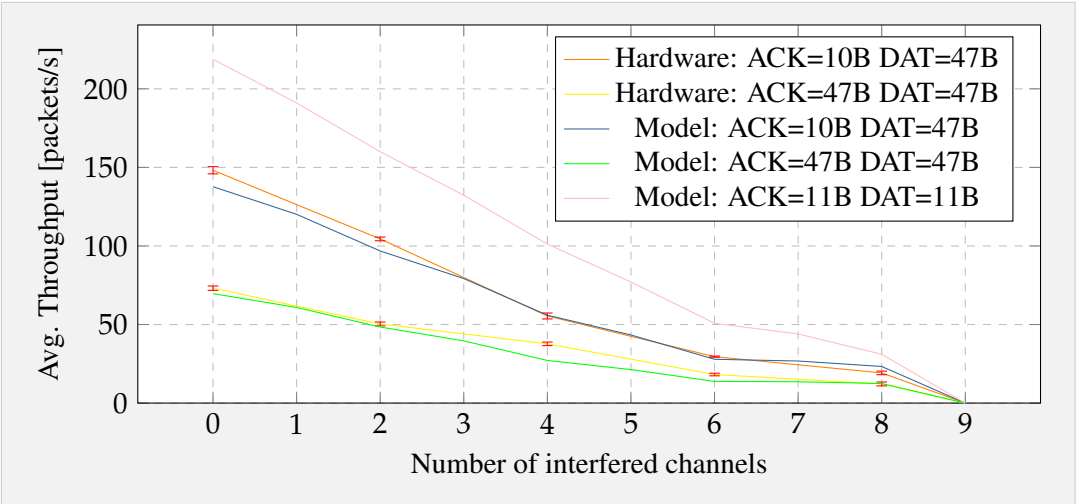


Figure 4.10.: Bluetooth Low Energy throughput over two hops for bidirectional and unidirectional traffic.

As a result, the packet size should be chosen as small as possible for networks with a high amount of interference and even the segmentation of large data packets into multiple smaller ones might drastically increase the performance despite the increased packet overhead.

4.6. Asymmetric Interference

Until this point of the evaluation, one of the main sources of packet loss in wireless networks has been completely neglected, namely the overflow of internal packet buffers. Thereby, the reception queue usually does not cause any problems, since dropped packets are simply not acknowledged and therefore immediately resent by the transmitter. Eventually, the queue will have enough space for the packet and it is successfully delivered. For transmission queues, on the other hand, dropped packets are usually lost, because otherwise the application would have to wait until there is space available again in the queue. This is particularly a problem in scenarios, where two links are subject to different amounts of external interference. In a scenario where a node A wants to send data to a node C via a node B, B just forwards messages from A to C. But since the throughput on the second link is much lower, the transmission queue will eventually overflow. The BlueNRG-MS stack handles this situation by buffering a small amount of packets and indicating packet drops back to the application.

To see, how many packets are lost due to queue drops and how the queue size affects such a scenario, an additional queue was implemented in front of the transmit buffer of the BlueNRG-MS stack, which mimics its behavior. Thereby, packets that are queued for transmission will not get lost, because they are resent until they successfully arrive. As a consequence the

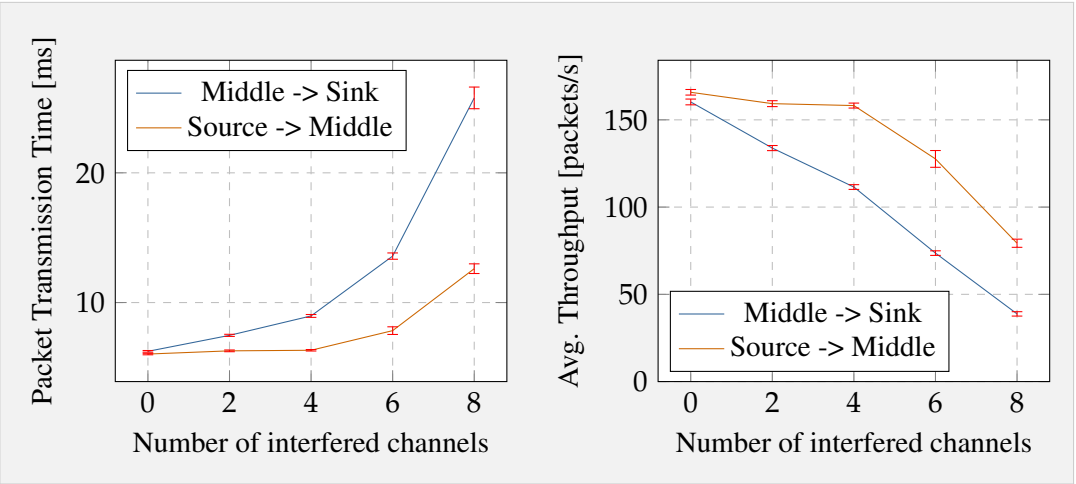


Figure 4.11.: Packet transmission time and average throughput of the two links for an increasing number of interfered channels in BLE.

application has to wait for a short time, until it is assured that the packet has been sent to the lower layer of the BlueNRG-MS stack and will therefore be transmitted successfully.

The experimental setup was chosen as depicted in Figure 4.1 and the throughput and packet loss were measured. Therefore, three boards were used, whereof the rightmost device, as shown in the figure, was configured as the source of the transmission and the leftmost device was configured as the sink. The only task of the middle device is to forward packets from the source to the sink. Furthermore, the main idea of this experiment is to created an asymmetric distribution of interference on the two links so that the link from the middle node to the sink forms the bottleneck of the system. Therefore, the boards that create the artificial interference were placed next to the sink of the connection, so that the link from the sink to the middle node is not affected by the interference. The number of interfered channels was gradually increased from zero to eight for the second link to create a discrepancy in the throughput. In order to verify this and give a reference point for the achievable throughput on both links, Figure 4.11 shows the transmission time of a single packet and the average throughput. One can easily see that the throughput of both links diverges as the number of interfered channels is increased.

The results are presented in Figure 4.12 and 4.13, whereby the former shows the number of dropped packets for an increasing number of interfered channels and different queue sizes and the latter shows the according throughput over two hops. Surprisingly, the number of dropped packets stays about constant for an increasing amount of interfered channels. This is due to the fact that the application possibly has to wait when pushing packets from the queue to the BlueNRG-Ms stack. In this time no new packets can be added to the queue, so that the number of dropped packets depends also on the throughput of the second link and stays constant. As one can see in Figure 4.13, the resulting throughput heavily varies with the

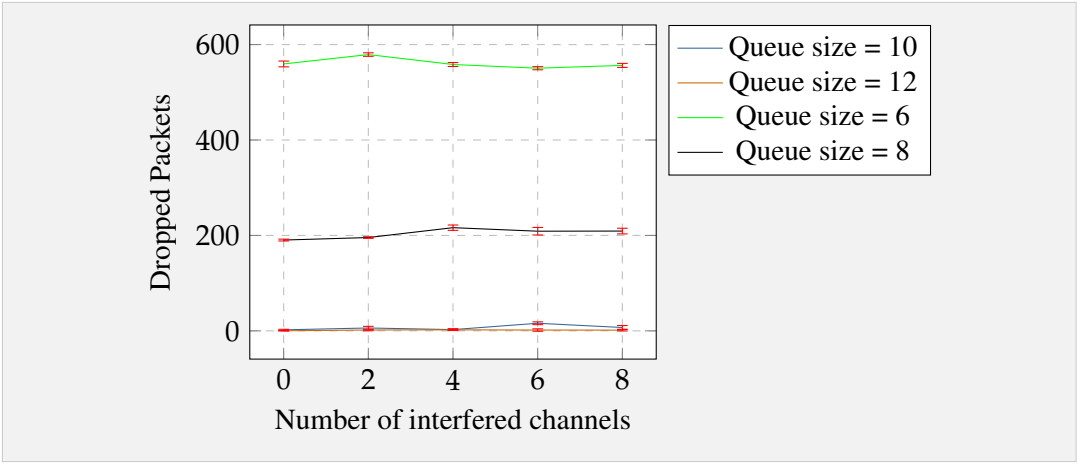


Figure 4.12.: Throughput from the middle node to the sink for an increasing amount of interference

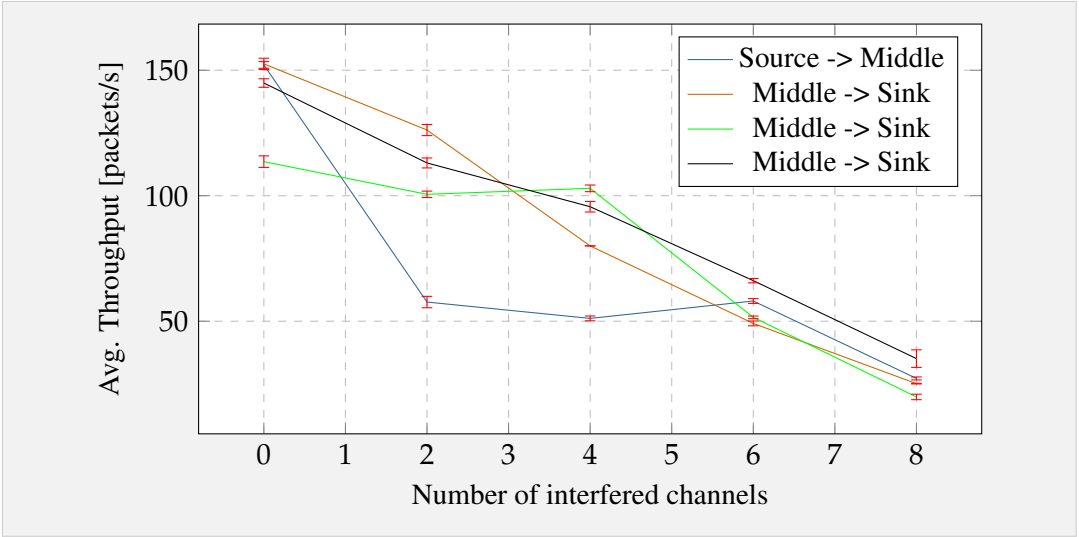


Figure 4.13.: Dropped packets at the queue for different queue sizes

number of interfered channels, but in general decreases as expected.

4.7. Application Layer Extensions

In order to further improve the reliability and predictability of Bluetooth Low Energy under the influence of heavy interference, two mechanisms were implemented and tested on its application layer. Section 4.7.1 first covers a credit-based flow control for BLE, followed by the description of an adaptive frequency hopping algorithm in Section 4.7.2. These kinds of algorithms are usually implemented on the MAC layer. However, most BLE implementations

only offer access to the higher layers, while the controller is commonly provided as a pre-compiled binary. Therefore, the implementation on the application layer is chosen to support a wide range of devices.

4.7.1. Credit-Based Flow Control

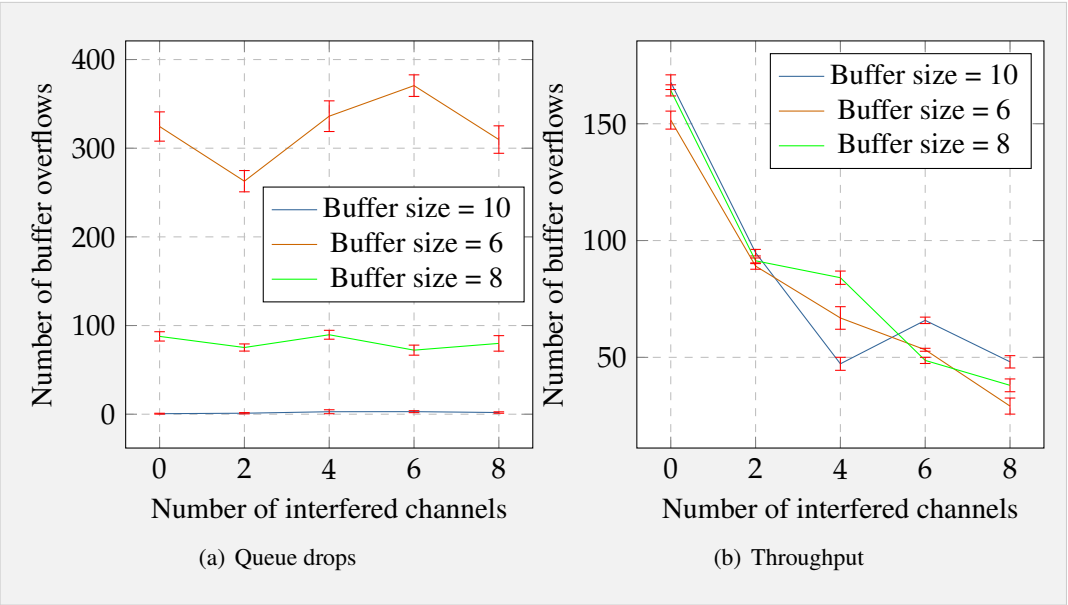
As shown in Section 4.6, asymmetric interference naturally leads to high packet loss if the source is not adapting to lower data rates on the path to the destination. A commonly used technique to counter this type of packet loss is *flow control*. There are several types of flow control but for this work a simplified version of the credit-based flow control from [KM95] is chosen. A similar approach is already presented in [Ale], where a credit-based flow control is designed on the L2CAP protocol of the BLE stack.

As the name implies, the core idea of credit-based flow control is that a node can issue credits to its neighbours based on its available buffer space. If a node A wants to transmit data to a node B, it has to consume its assigned credits. Thereby, a node cannot transmit any data if the assigned credits are depleted. The receiver periodically issues new credits when more buffer space becomes available again [KM95].

In this work, the algorithm was implemented on the application layer by using a GATT service with a single byte characteristic for the credit assignment. One credit is set to correspond to one BLE data packet, so that a buffer size of up to 256 packets is supported. On establishment of a connection, the participating nodes subscribe to the credit characteristic of the other node to receive updates. A node can write to its local characteristic to issue credits to all subscribed nodes. However, since there is only one credit characteristic, all neighbours receive the same number of credits. As a consequence, a node has to calculate its local credit value as the available buffer space divided by the number of established connections. This algorithm can be easily extended to distinct credit values per connection by adding more characteristics to the credits service, which is, however, not required in this work. At last every node simply checks if there are any credits left before sending a data packet and holds back the packet transmission if applicable.

The algorithm was tested with the same experimental setup as used for the experiments in Section 4.6, yielding the results shown in Figure 4.14. Thereby, Figure 4.14(a) show the number of buffer overflows for an increasing number of interfered channels. It is easy to see that the number of overflows is about halved in comparison to the results from section 4.6 when using the described algorithm. This, however, is not the intended level of reliability and the credit-based flow control algorithm should be able to work without any packets losses.

The problem seems to arise on the hardware platform, due to the two component architecture of the BLE stack. At the moment a device receives an update of the available credits, there can still be packets buffered for transmission in the controller layer, in this case in the BLE



(a) Queue drops

(b) Throughput

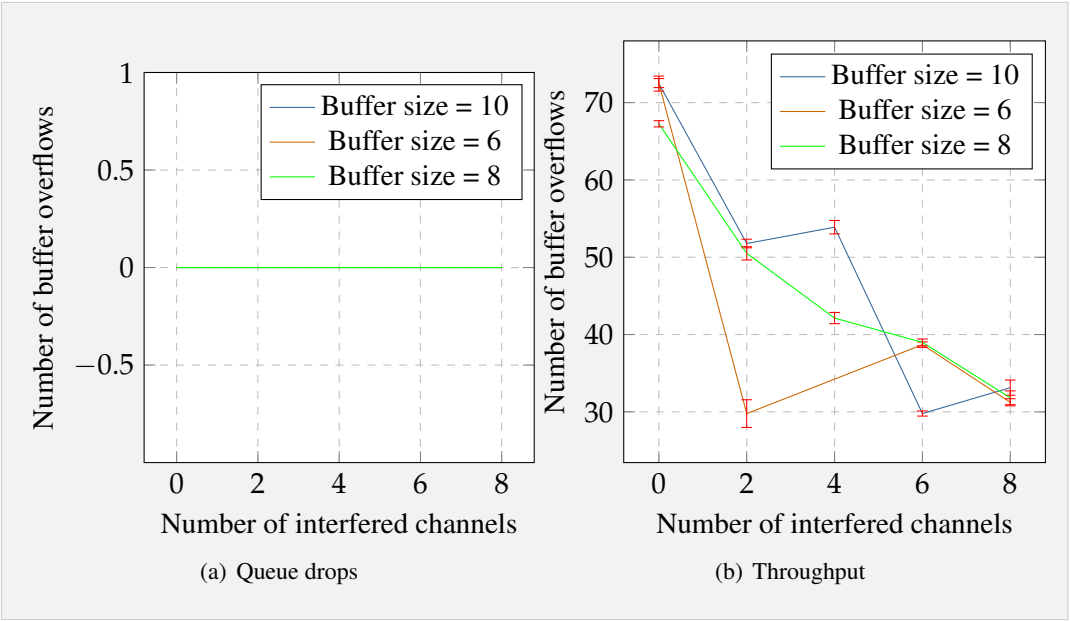
Figure 4.14.: Number of dropped packets at the queue and average throughput for different queue sizes for BLE.

expansion board. If the device now uses all its credits for transmission, buffer overflows may occur, because the already buffered packets are not taken into account. Therefore, the receiving device has to consider these additional packets when handing out credits. This is done by reducing the number of packets per device by the amount of packets that can be buffered in the lower layer queues. Another solution, which however introduces a higher management effort because the receiver has to keep track of the packets that it still has to receive from every connected device. Additionally, the credits would have to be updated individually for every connected device.

Figure 4.15 shows the results for the updated solution. As it can be seen in Figure 4.15(a), the number of lost packets due to buffer overflows is zero for any buffer size. The throughput, on the other hand, decreases to about half at the same time.

4.7.2. Adaptive Frequency Hopping

As already discussed in Section 2.2, adaptive frequency hopping (AFH) can be seen as an extension of regular frequency hopping. It monitors the quality of all channels, while periodically switching the transmission frequency. In this process, channels with a bad quality are sorted out and not used for transmission anymore to increase the overall performance and reliability of the transmission. In this section the implementation of a simplistic AFH algorithm on the BLE application layer is described and results of hardware experiments are presented.



(a) Queue drops (b) Throughput

Figure 4.15.: Number of dropped packets at the queue and average throughput for different queue sizes using the updated algorithm for BLE.

Usually adaptive frequency hopping algorithms employ metrics like the bit error rate (BER), packet reception ratio (PRR), signal-to-noise ratio (SNR) or received signal strength indicator (RSSI) for assessment of a channel's quality. This, however, yields problems in the implementation on the BLE application layer. The first problem is that metrics like the BER or PRR can be measured if the receiver knows exactly how many packets were sent, but the number of retransmissions is usually not available on the application layer. Additionally, the SNR and RSSI have to be measured at the physical layer, creating the same problem. At last, the most important information, namely the current transmit channel, is also not available which highly restricts the freedom of the implementation.

Therefore, the AFH algorithm is built under the assumption of a traffic scenario with a constant transmission rate, for example a sensor that sends its data towards a controlling application every few milliseconds. The desired throughput can be easily estimated beforehand and is used as a threshold in the AFH algorithm. Thereby, the designed algorithm constantly monitors the throughput on the receiving link and compares it with the predefined threshold value. If the threshold is violated, the algorithm starts to examine the quality of specific channels. This implementation has the advantage that changes only have to be applied to the receiving application and the transmitter can send data as usual, benefiting especially resource and power restricted sensor nodes.

Since the current channel number is not available on the application layer, the algorithm cannot directly sort a channel out if its quality is bad, but has to check which of the channels in

Time	Used Channels	Time	Used Channels
t4	{10,9,23,22,21,36,35,34,33}	t11	{23,22,21,36,35,34,33}
t5	{10,9,23,22,21,36,35,34,33}	t12	{23,22,21,35,34,33}
t6	{10,9,23,22,21,36,35,34,33}	t13	{23,22,21,35,34,33}
t7	{9,23,22,21,36,35,34,33}	t14	{23,22,21,35,33}
t8	{23,22,21,36,35,34,33}	t15	{23,22,21,35}
t9	{23,22,21,36,35,34,33}	t16	{23,22,21,35}
t10	{23,22,21,36,35,34,33}	t17	{23,22,21,35}

Table 4.2.: Alteration of the channel map over time with {9,10,33,34} being the set of interfered channels.

the current channel map reduces the overall throughput. Therefore, it diminishes the channel map by a single channel at a time and measures the throughput over the duration of multiple connection events - with at least one connection event for every channel in the channel map. If the measured throughput is getting lower, the removal of the channel was beneficial and the tested channel is removed from the channel map for further transmissions. On the other hand, if the measured throughput is higher, the tested channel is written back to the channel map and used for further transmissions. Additionally, a tolerance can be provided so that only channels, which contribute significantly to a lower throughput, are sorted out and small variations are neglected. In this way the channel map is reduced one by one until the measured throughput is higher than the original threshold. If only one channel is left in the channel map, it is reset to its original state and the AFH algorithm fails. This is due to the reason that BLE, according to the standard, requires at least two channels to work. The AFH algorithm, however, can then start a new run on the reset channel map.

The AFH algorithm was evaluated on the hardware platform by measuring the throughput and the alteration of the channel map between two BLE boards. The threshold was set to 140 packets/s and one throughput measurement encompassed 2500 packets to increase the accuracy of the measurement. After a short delay, an artificial interference was created on up to eight channels. The results are depicted in Figure 4.16, which shows the variation of the measured throughput for different numbers of interfered channels and start and end time of the AFH algorithm using the red dashed lines. As one can easily see, the algorithm manages to reduce the set of used channels in a way that the throughput rises above the threshold again, a short time after it was violated. This is even the case for six interfered channels, as shown on the bottom right of Figure 4.16, where the throughput was reduced from about 160 packets/s to about 20 packets/s due to the external interference. Table 4.2, exemplarily shows the alteration of the channel map for a number of four interfered channels (CH 9, 10, 33, 34). The table corresponds to the throughput in Figure 4.16 (3) and gives a deeper insight

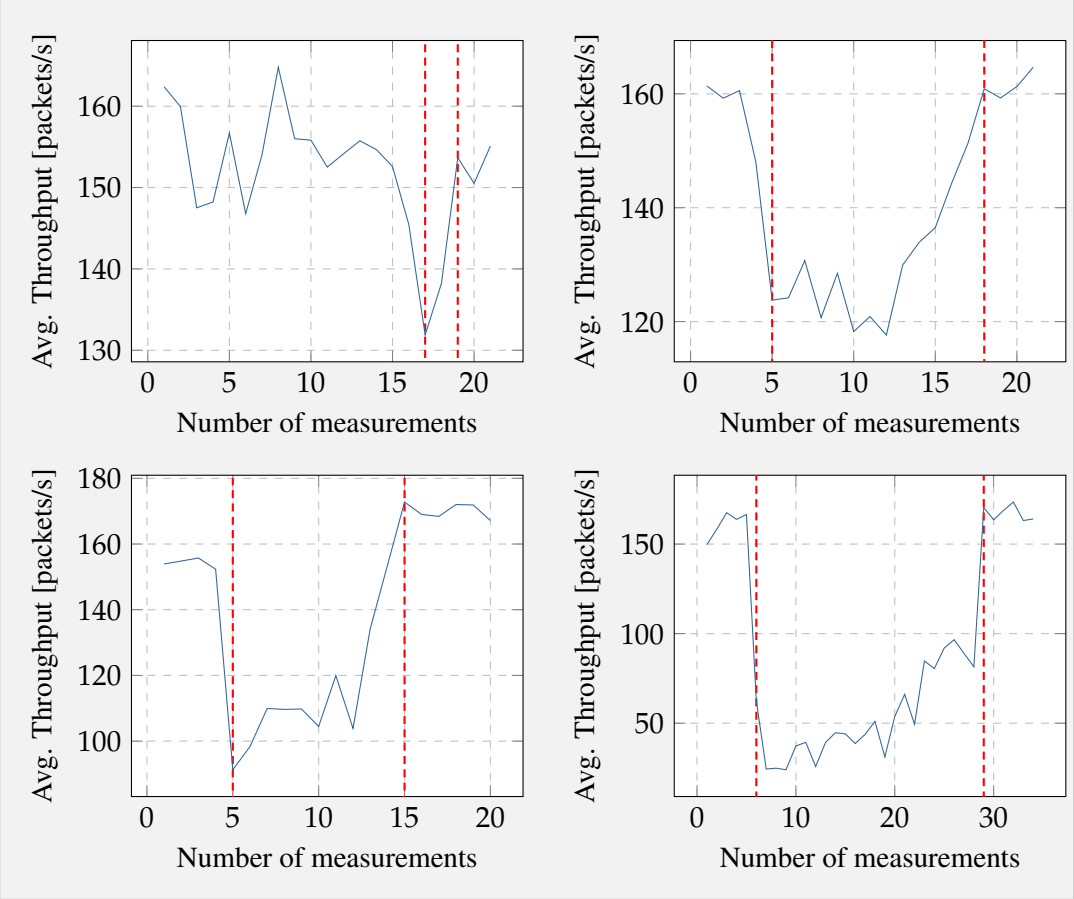


Figure 4.16.: Variation of the BLE throughput over time using the AFH algorithm for different numbers of interfered channels (top left 0, top right 2, bottom left 4, bottom right 6).

on how the algorithm works. For example, one can see that at measurement t11 channel 36 was removed, because of the spike in the throughput graph. At t12, on the other hand, the throughput is getting worse than in the best seen configuration so far, so that channel 35 stays in the channel map for further transmissions. In the resulting channel map, all interfered channel were found and removed in addition to channel 36, which was apparently subject to disturbance by another source.

At last, one should say that even if the application is not using a constant transmission rate, this algorithm could be used. In this case, a *test mode* has to be entered when the network is started. The nodes then evaluate the channels using the described mechanism, whereby one node periodically sends test packets. Additionally, it should be said that a central automatically updates the channel map for all its slave connections at once. This reduces the flexibility because the channel map cannot be chosen individually for every link but drastically increases the simplicity of the implementation.

Chapter 5

Conclusion

In the course of this thesis, a theoretical model to evaluate the performance of Bluetooth Low Energy and IEEE 802.15.4 DSME in the presence of external interference has been developed. The model allows the calculation of a device's maximum transmission rate λ in a wireless environment, while considering the interference of the wireless channels, so that no packets are lost. Therefore, the results contribute significantly to the predictability and optimal usage of the protocols in industrial applications with dense wireless mesh networks. The results of the theoretical model have been evaluated and both protocols were compared, with the result that Bluetooth Low Energy provides a higher amount of packets per second, which is often more useful in industrial applications, since sensors frequently have to send small amounts of data. On the other hand, IEEE 802.15.4 provides a higher throughput in the presence of high amounts of interference, due to a higher packet size, so that it has to be concluded that the choice of the protocol is completely depended on the particular application. Additionally, it has to be said that the maximum throughput of DSME could be much higher for a different configuration of the superframe order, if the transmission of a maximum physical layer packet is not required. The main argument for DSME, however, is its extended range in comparison to Bluetooth Low Energy, which allows the efficient construction of mesh networks over large industrial plants. The only, outstanding advantage of Bluetooth Low Energy over IEEE 802.15.4 DSME is its broad availability in a variety of devices, like smartphones and other wearables. This implies a good support of the protocol and a large pool of resources in case of problems, which is often a key factor in the industry. Therefore, it has to be concluded that Bluetooth Low Energy is the better choice for applications with limited range, but DSME is superior for large scale mesh networks over larger distances and with a higher amount of external interference. The model was verified by a series of hardware experiments which proved its correctness and accuracy in realistic wireless environments. Additionally, they provided some further results like the delay of IEEE 802.15.4 DSME and Bluetooth Low Energy in a multihop environment for an increasing amount of interference on the channels. At last, some improvements were applied to the Bluetooth Low Energy application layer to

further improve the reliability of the connection. This includes a token-based flow control, which mitigates the packet loss due to queue drops at the transmission queue. The second was the test of an adaptive frequency hopping algorithm, which was able to significantly improve the throughput in the test setup by sorting out channels, which underlie influence of external interference.

Future Work

Due to the limited availability of the hardware platforms, the experiments could only be conducted in an environment with a maximum number of two hops and up to four interfered channels. Therefore, no verification of the model's accuracy in mesh network, where also a lot of different factor like self interference play a role, could be given. As a consequence, the test with more boards would significantly increase the reliability of the model and at the same time yield some interesting results of the protocols in dense environments with large amounts of interference. Also the model itself can be extended to include other traffic scenarios such as Poisson distributed traffic or traffic bursts. Finally, these results could be provided by simulation, which, however, would require the implementation of a Bluetooth Low Energy stack in one of the popular, freely available network simulators like OMNeT++[1] or ns-3[2]. To the authors knowledge, there is currently no Bluetooth Low Energy simulator with support for mesh networks, so that this would be a significant improvement. Such a simulation would be useful to support and verify the model and the hardware results of the thesis to design more robust and reliable wireless networks in industrial applications.

[1] https://www.omnetpp.org/
[2] https://www.nsnam.org/

Bibliography

[AKBBR16] Mohamad Omar Al Kalaa, Walid Balid, Naim Bitar, and Hazem H Refai. Evaluating bluetooth low energy in realistic wireless environments. In *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*, pages 1–6. IEEE, 2016.

[AKM+14] N Azmi, LM Kamarudin, M Mahmuddin, A Zakaria, AYM Shakaff, S Khatun, K Kamarudin, and MN Morshed. Interference issues and mitigation method in WSN 2.4 GHz ISM band: A survey. In *Electronic Design (ICED), 2014 2nd International Conference on*, pages 403–408. IEEE, 2014.

[AKR14] Mohamad Omar Al Kalaa and Hazem H Refai. Bluetooth standard v4.1: Simulating the Bluetooth low energy data channel selection algorithm. In *Globecom Workshops (GC Wkshps), 2014*, pages 729–733. IEEE, 2014.

[AKR15] Mohamad Omar Al Kalaa and Hazem H Refai. Selection probability of data channels in Bluetooth Low Energy. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, pages 148–152. IEEE, 2015.

[Ale] Georgel Bodgan Alexandru. Bluetooth le credit-based flow control for l2cap connection-oriented channels.

[Bis01] Chatschik Bisdikian. An overview of the Bluetooth wireless technology. *IEEE Communications magazine*, 39(12):86–94, 2001.

[ble13] Specification of the Bluetooth System, Covered Core Packageversion: 4.1, 2013.

[blu16] BlueNRG,BlueNRG-MS stacks programming guidelines. Technical report, 2016.

[CM11] John COLPO and Diederik MOLS. No strings attached. *Hydrocarbon engineering*, 16(11), 2011.

[CPH+14] Keuchul Cho, Woojin Park, Moonki Hong, Gisu Park, Wooseong Cho, Jihoon Seo, and Kijun Han. Analysis of latency performance of Bluetooth low energy (BLE) networks. *Sensors*, 15(1):59–78, 2014.

[CR03] Carla-Fabiana Chiasserini and Ramesh R Rao. Coexistence mechanisms for interference mitigation in the 2.4-GHz ISM band. *IEEE Transactions on Wireless Communications*, 2(5):964–975, 2003.

[DGBA16] Domenico De Guglielmo, Simone Brienza, and Giuseppe Anastasi. IEEE 802.15.4e: A survey. *Computer Communications*, 88:1–24, 2016.

[DMSL+17] Piergiuseppe Di Marco, Per Skillermark, Anna Larmo, Pontus Arvidson, and Roman Chirikov. Performance Evaluation of the Data Transfer Modes in Bluetooth 5. *IEEE Communications Standards Magazine*, 1(2):92–97, 2017.

[FEO+] Xenofon Fafoutis, Atis Elsts, George Oikonomou, Robert Piechocki, and Ian Craddock. Adaptive Static Scheduling in IEEE 802.15. 4 TSCH Networks.

[FH14] Ramsey Faragher and Robert Harle. An analysis of the accuracy of bluetooth low energy for indoor positioning applications. In *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation*, pages 201–210, 2014.

[GDP11] Carles Gomez, Ilker Demirkol, and Josep Paradells. Modeling the maximum throughput of Bluetooth low energy in an error-prone link. *IEEE Communications Letters*, 15(11):1187–1189, 2011.

[GMIS16] Asmir Gogic, Aljo Mujcic, Sandra Ibric, and Nermin Suljanovic. Performance Analysis of Bluetooth Low Energy Mesh Routing Algorithm in Case of Disaster Prediction. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(6):929–935, 2016.

[GOP12] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9):11734–11753, 2012.

[HOR+17] Diego Hortelano, Teresa Olivares, M Carmen Ruiz, Celia Garrido-Hidalgo, and Vicente López. From sensor networks to internet of things. Bluetooth low energy, a standard for this evolution. *Sensors*, 17(2):372, 2017.

[iee16] 802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks, 2016.

[JPLK01] Kyunghun Jang, Jonghun Park, Tae-Jin Lee, and YongSuk Kim. Reliable delivery of broadcast packet in Bluetooth. In *Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd*, volume 2, pages 1119–1123. IEEE, 2001.

[KKLT17] Florian Kauer, Maximilian Köstler, Tobias Lübkert, and Volker Turau. OpenDSME-A portable framework for reliable wireless sensor and actuator networks. In *Networked Systems (NetSys), 2017 International Conference on*, pages 1–2. IEEE, 2017.

[KLJ15] Hyun-Soo Kim, Jungyub Lee, and Ju Wook Jang. Blemesh: A wireless mesh network protocol for bluetooth low energy devices. In *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, pages 558–563. IEEE, 2015.

[KM95] NT Kung and Robert Morris. Credit-based flow control for ATM networks. *IEEE network*, 9(2):40–48, 1995.

[LCM12] Jia Liu, Canfeng Chen, and Yan Ma. Modeling and performance analysis of device discovery in bluetooth low energy networks. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 1538–1543. IEEE, 2012.

[LSH08] Tomas Lennvall, Stefan Svensson, and Fredrik Hekland. A comparison of WirelessHART and ZigBee for industrial applications. In *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*, pages 85–88. IEEE, 2008.

[LSS07] Jin-Shyan Lee, Yu-Wei Su, and Chung-Chou Shen. A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. In *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pages 46–51. Ieee, 2007.

[LTT15] Jiun-Ren Lin, Timothy Talty, and Ozan K Tonguz. On the potential of bluetooth low energy technology for vehicular applications. *IEEE Communications Magazine*, 53(1):267–275, 2015.

[Mik14a] Konstantin Mikhaylov. Accelerated connection establishment (ACE) mechanism for Bluetooth Low Energy. In *Personal, Indoor, and Mobile Radio Communication (PIMRC), 2014 IEEE 25th Annual International Symposium on*, pages 1264–1268. IEEE, 2014.

[Mik14b] Konstantin Mikhaylov. Simulation of network-level performance for Bluetooth Low Energy. In *Personal, Indoor, and Mobile Radio Communication (PIMRC), 2014 IEEE 25th Annual International Symposium on*, pages 1259–1263. IEEE, 2014.

[NDV15] PrithviRaj Narendra, Simon Duquennoy, and Thiemo Voigt. BLE and IEEE 802.15. 4 in the IoT: Evaluation and Interoperability Considerations. In *International Internet of Things Summit*, pages 427–438. Springer, 2015.

[OK10] Alf Helge Omre and Steven Keeping. Bluetooth low energy: wireless connectivity for medical monitoring. *Journal of diabetes science and technology*, 4(2):457–463, 2010.

[PAP+05] Jitendra Padhye, Sharad Agarwal, Venkata N Padmanabhan, Lili Qiu, Ananth Rao, and Brian Zill. Estimation of link interference in static multi-hop wireless networks. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 28–28. USENIX Association, 2005.

[PC09] Stig Petersen and Simon Carlsen. Performance evaluation of WirelessHART for factory automation. In *Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on*, pages 1–9. IEEE, 2009.

[PLB16] Gaetano Patti, Luca Leonardi, and Lucia Lo Bello. A Bluetooth low energy real-time protocol for industrial wireless mesh networks. In *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*, pages 4627–4632. IEEE, 2016.

[PW10] Maulin Patel and Jianfeng Wang. Applications, challenges, and prospective in emerging body area networking technologies. *IEEE Wireless communications*, 17(1), 2010.

[PYP06] Petar Popovski, Hiroyuki Yomo, and Ramjee Prasad. Dynamic adaptive frequency hopping for mutually interfering wireless personal area networks. *IEEE Transactions on Mobile Computing*, 5(8):991–1003, 2006.

[RGL17] Raúl Rondón, Mikael Gidlund, and Krister Landernäs. Evaluating Bluetooth Low Energy Suitability for Time-Critical Industrial IoT Applications. *International Journal of Wireless Information Networks*, pages 1–13, 2017.

[SA05] Marvin K Simon and Mohamed-Slim Alouini. *Digital communication over fading channels*, volume 95. John Wiley & Sons, 2005.

[SDTL06] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 419–420. ACM, 2006.

[SHNN12] Matti Siekkinen, Markus Hiienkari, Jukka K Nurminen, and Johanna Nieminen. How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4. In *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*, pages 232–237. IEEE, 2012.

[SM00] Slim Souissi and Eric F Meihofer. Performance evaluation of a Bluetooth network in the presence of adjacent and co-channel interference. In *Emerging Technologies Symposium: Broadband, Wireless Internet Access, 2000 IEEE*, pages 6–pp. IEEE, 2000.

[SSF+14] Sérgio Silva, Salviano Soares, Telmo Fernandes, António Valente, and António Moreira. Coexistence and interference tests on a Bluetooth Low Energy front-end. In *Science and Information Conference (SAI), 2014*, pages 1014–1018. IEEE, 2014.

[TFM+15] Evgeny Tsimbalo, Xenofon Fafoutis, Evangelos Mellios, Mo Haghighi, Bo Tan, Geoffrey Hilton, Robert Piechocki, and Ian Craddock. Mitigating packet loss in connectionless Bluetooth low energy. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 291–296. IEEE, 2015.

[TMLU08] Tanim M Taher, Matthew J Misurac, Joseph L LoCicero, and Donald R Ucci. Microwave oven signal interference mitigation for Wi-Fi communication systems. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 67–68. IEEE, 2008.

[Tor14] Karl Torvmark. Three flavors of Bluetooth®: Which one to choose? *Texas Instruments. Lit. no.: SWRY007*, 2014.

[YXL12] Bin Yu, Lisheng Xu, and Yongxu Li. Bluetooth low energy (BLE) based mobile electrocardiogram monitoring system. In *Information and Automation (ICIA), 2012 International Conference on*, pages 763–767. IEEE, 2012.

Appendix A

Content of the DVD

The contents of the attached DVD are described in Table A.1.

Filename	Description
thesis.pdf	This document
/latex/	LaTeX sources of this document
/model/	Matlab sources for the theoretical model
/ble/	Source code of the Bluetooth Low Energy experiments for Chapter 4
/dsme/	Source code of the IEEE 802.15.4 DSME experiments for Chapter 4

Table A.1.: Contents of the DVD.

Appendix B

Transition matrix for the BLE Markov chain

$$
(Q|R) = \begin{pmatrix}
 & s_{0,0} & s_{1,0} & \cdots & s_{N,0} & s_{0,1} & s_{0,2} & s_{0,3} & \cdots \\
s_{0,0} & 0 & \Gamma_{RT} & \cdots & 0 & \Omega_{DATA}\Gamma_{ACK} & \Gamma_{DATA}\Omega_{ACK} & \Omega_{DATA}\Omega_{ACK} & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\
s_{N-1,0} & 0 & 0 & \cdots & \Gamma_{RT} & 0 & 0 & 0 & \cdots \\
s_{N,0} & 0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots \\
s_{0,1} & 0 & \Gamma_{RT} & \cdots & 0 & 0 & \Gamma_{DATA}\Omega_{ACK} & 0 & \cdots \\
s_{0,2} & 0 & \Gamma_{RT} & \cdots & 0 & \Omega_{DATA}\Gamma_{ACK} & 0 & 0 & \cdots \\
s_{0,3} & 0 & \Gamma_{RT} & \cdots & 0 & 0 & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\
s_{N-1,1} & 0 & 0 & \cdots & \Gamma_{RT} & 0 & 0 & 0 & \cdots \\
s_{N-1,2} & 0 & 0 & \cdots & \Gamma_{RT} & 0 & 0 & 0 & \cdots \\
s_{N-1,3} & 0 & 0 & \cdots & \Gamma_{RT} & 0 & 0 & 0 & \cdots
\end{pmatrix}
$$

(B.1)

$$
\begin{pmatrix}
s_{N-1,1} & s_{N-1,2} & s_{N-1,3} & s_{0,4} & \cdots & s_{N-1,4} \\
0 & 0 & 0 & \Omega_{AA} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\Omega_{DATA}\Gamma_{ACK} & \Gamma_{DATA}\Omega_{ACK} & \Omega_{DATA}\Omega_{ACK} & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \Omega_{DATA}\Gamma_{ACK} + \Omega_{DATA}\Omega_{ACK} + \Omega_{AA} & \cdots & 0 \\
0 & 0 & 0 & \Gamma_{DATA}\Omega_{ACK} + \Omega_{DATA}\Omega_{ACK} + \Omega_{AA} & \cdots & 0 \\
0 & 0 & 0 & 1 - \Gamma_{RT} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \Gamma_{DATA}\Omega_{ACK} & 0 & 0 & \cdots & \Omega_{DATA}\Gamma_{ACK} + \Omega_{DATA}\Omega_{ACK} + \Omega_{AA} \\
\Omega_{DATA}\Gamma_{ACK} & 0 & 0 & 0 & \cdots & \Gamma_{DATA}\Omega_{ACK} + \Omega_{DATA}\Omega_{ACK} + \Omega_{AA} \\
0 & 0 & 0 & 0 & \cdots & 1 - \Gamma_{RT}
\end{pmatrix}
$$