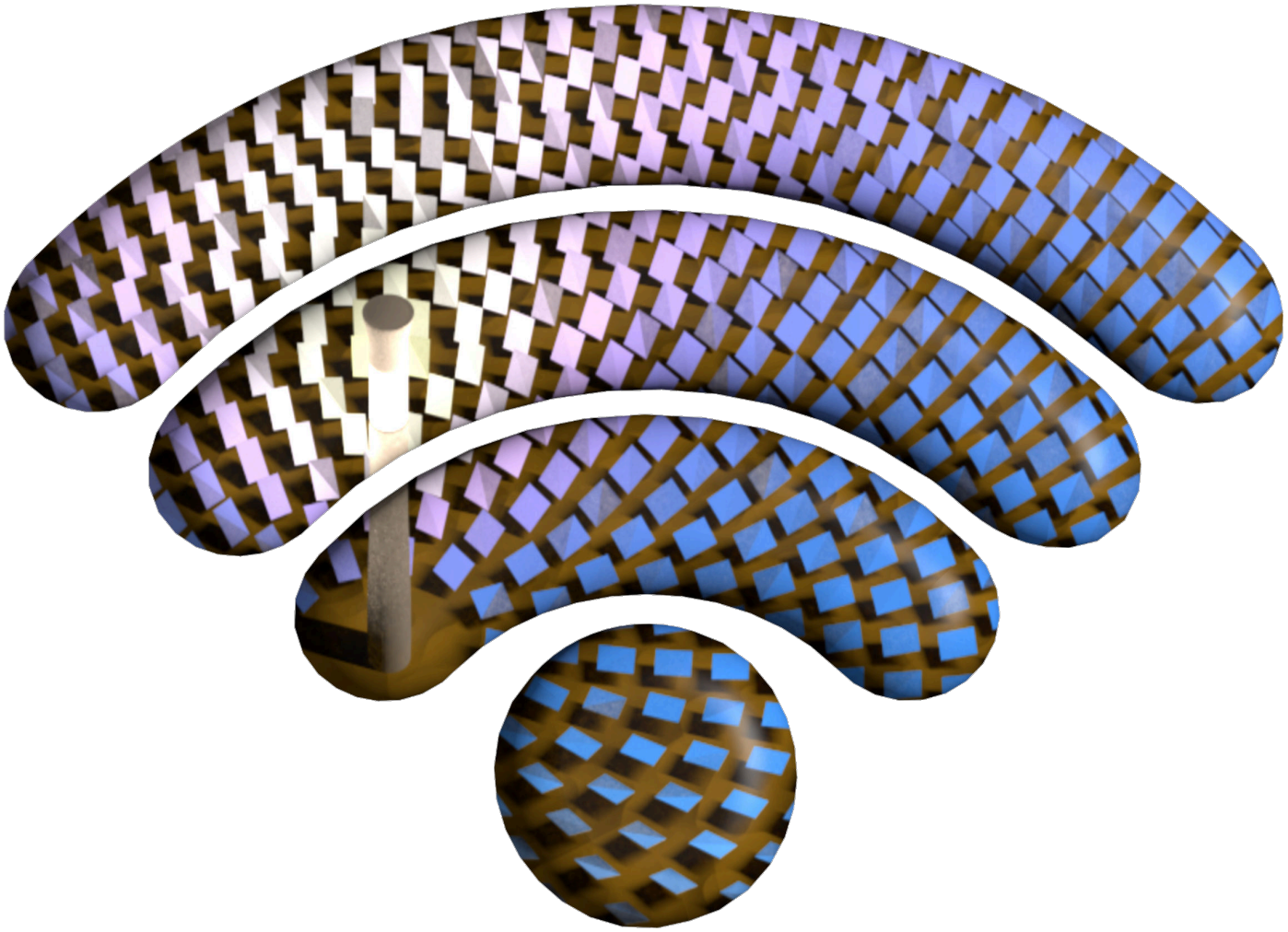
Scalable Wireless Multi-Hop Networks for Industrial Applications

Florian Kauer



DOI 10.15480/882.2259

Scalable Wireless Multi-Hop Networks for Industrial Applications

Vom Promotionsausschuss der
Technischen Universität Hamburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von

Florian Kauer

aus

Hamburg, Deutschland

2019

Date of Oral Examination	May 6th, 2019
Chair of Examination Board	Prof. Dr.-Ing. Gerhard Bauch Institute of Communications Hamburg University of Technology
First Examiner	Prof. Dr. rer. nat. Volker Turau Institute of Telematics Hamburg University of Technology
Second Examiner	Prof. Dr.-Ing. Udo Zölzer Department of Signal Processing and Communication Helmut Schmidt University
Digital Object Identifier	https://doi.org/10.15480/882.2259
ORCID-iD	https://orcid.org/0000-0002-8712-2072

Acknowledgment

I would like to express my deep gratitude to the numerous people who have accompanied and supported me for the last years. Foremost, this includes my supervisor Prof. Dr. Volker Turau, who has always provided me the perfect balance of support and freedom for realizing this thesis.

Valuable knowledge about the requirements of real-world applications was provided by the project partners of the AutoR research project at TRINAMIC, the DLR as well as the staff at the solar tower power plant in Jülich. I am very thankful for this input and the opportunity to be part of such a large team. I would also like to thank the numerous students that I supervised over the last years for providing valuable input and different ways of thinking.

I wish to thank my colleagues at the Institute of Telematics for many fruitful discussions, including the fun controversies during lunchtime. In particular, this includes Andreas Weigel, Maximilian Köstler and Florian Meyer with whom I shared the office. I owe them, as well as Tobias Lübkert, Sandrina Köstler, Axel Neuser and Stefan Unterschütz, my deepest respect and gratitude for their remarkable effort during the development and debugging of the software implementations for openDSME or CometOS or both.

I am very grateful for my supportive and encouraging family and friends. Especially, for my parents Martina Meier and Matthias Meyne, who have sparked my interest in informatics and electronics, respectively. A special thank you goes out to my wife, Friederike Kauer, for all her love and for many constructive mathematical discussions.

Last, but not least, I would like to thank my son Emil. His upcoming birth was a huge motivation burst during the final phase of writing.

Abstract

Wireless technology promises flexible and cost-efficient machine-to-machine communication. However, high packet loss can emerge from simultaneous transmissions of many devices, undermining the reliability required for industrial applications. This thesis analyzes and develops techniques for time-slotted multi-hop communication with focus on IEEE 802.15.4 DSME. In a holistic approach, simulations, formal and analytical analyses and testbed experiments are combined, concluding the utility of the proposed methods. For future usage in research and real-world deployments, openDSME is developed as open-source implementation of DSME and application-specific insights are provided.

Zusammenfassung

Funktechnik verspricht flexible und kosteneffiziente Maschinenkommunikation. Allerdings können gleichzeitige Übertragungen von vielen Geräten zu Paketverlust führen und die für industrielle Anwendungen notwendige Zuverlässigkeit untergraben. Die Dissertation untersucht und entwickelt Techniken für vermaschte IEEE 802.15.4 Netzwerke mit Fokus auf dem Zeitschlitzverfahren DSME. Die Nützlichkeit wird in einem ganzheitlichen Ansatz mit Simulationen, formalen und analytischen Analysen und realen Netzen gezeigt. Für Nutzungen in Forschung und Anwendungen wurde die frei verfügbare DSME Implementierung openDSME entwickelt und es werden anwendungsspezifische Einblicke gegeben.

Table of Contents

Chapter 1

Introduction

Wireless communication is a core technology for realizing the vision of the Industrial Internet of Things. It promises to interconnect autonomous components with large flexibility and low costs to exchange information for process control but also about machine conditions and performance. This enables applications such as predictive maintenance and condition monitoring for efficient use of resources or mass customization with accurate monitoring of every workpiece and repeated reconfiguration of the machines. Using wireless technology is especially advantageous when retrofitting an existing plant with new sensors and actuators, but also for new plants where cabling is a complex and cost-intensive part of the deployment.

As running example for this thesis, a solar tower power plant as depicted in Fig. 1.1 is introduced, where thousands of steerable mirrors, so-called heliostats, reflect sunlight to a central receiver. In existing plants, every motor has a direct wired connection to a single field control unit and is actuated with a high frequency. In the concept proposed in the research project AutoR [PRM14], every mirror is a stand-alone unit that can autonomously track the sun. Since this allows for less frequent communication, mainly for power control and state-of-health messages, it is possible to use wireless connections to the mirrors, largely reducing the deployment costs for cabling. Additional mirrors can easily be added without modifications to a wired field bus. The particular challenge in this application is the huge

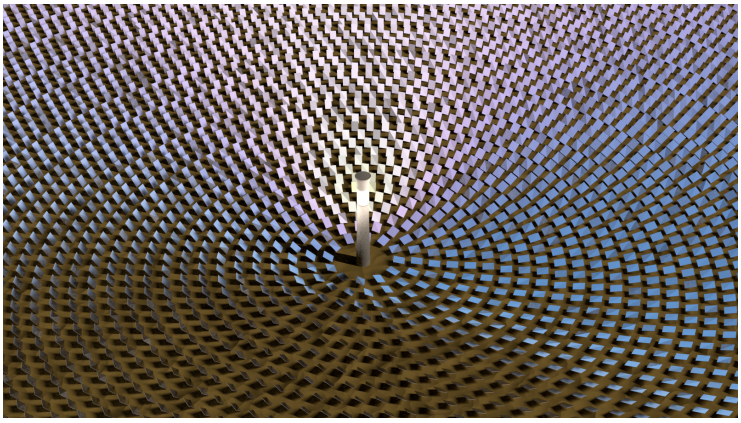


Figure 1.1: A solar tower power plant.

number of devices. This calls for particular consideration to allow for a coverage of a large area while still maintaining a high throughput and reliability. These requirements are very characteristic of many industrial applications, so a broad application of the techniques developed in this context is expected.

A prosperous technology for the realization of wireless networks for such applications is the standard IEEE 802.15.4 due to its low energy consumption and the availability of cost-effective hardware. It allows to build mesh networks that can efficiently forward messages over multiple hops to achieve a large coverage. There is a multitude of standards that use the physical layer of IEEE 802.15.4, including ZigBee, most prominently used for home automation, and the competing standards WirelessHART and ISA100.11a for industrial automation. Furthermore, 6LoWPAN is a set of IETF standardizations for seamless integration of IEEE 802.15.4 into IPv6 networks and finally the Internet.

An inherent property of wireless technologies is the need to coordinate the access to the shared electromagnetic spectrum. In order to mitigate the interference of multiple transmissions, several medium access techniques are being developed, ranging from simple listen-before-talk schemes to complex distributed algorithms for prereservation in the time and frequency domain. The latter includes two extensions to the IEEE 802.15.4 standard, namely Time Slotted Channel Hopping (TSCH) and the Deterministic and Synchronous Multi-channel Extension (DSME). Especially in large networks under heavy load, these approaches are expected to perform much better than the commonly used Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). DSME is of particular interest for large-scale and efficient wireless multi-hop networks since it specifies a slot allocation handshake for distributed construction of a schedule that avoids collisions of transmissions by neighboring transceivers.

In order to enable the application of these techniques for a broad usage in industrial contexts, several building blocks are missing. Most importantly, techniques for slot scheduling are required that can adapt to heterogeneous traffic demands and improvements of the slot allocation handshake are needed to maintain consistency. Also, before the development of openDSME as presented in this thesis, no implementation of DSME was publicly available for research and usage on physically deployed wireless devices. Finally, the analysis of the performance is of particular interest to estimate if a given application can be realized with sufficient performance. While in many consumer applications, best-effort communication is sufficient, unmet expectations can thwart the whole investment in an industrial application.

The performance evaluation of techniques for wireless networks is possible in multiple ways. Building physically deployed testbeds provides good insights into the performance of an actual application. However, it is very cost- and time-intensive to build and maintain such deployments. Therefore, event-based simulators are developed that try to replicate the reality as closely as possible, while allowing reproducible experiments without the need of specialized hardware. The disadvantage is a complex and extensive implementation and large execution times. An alternative is the analysis of mathematical models. An analytical model in the sense of this thesis is a system of equations that can be solved by numerical means. While such a model usually only describes a small subset of the behavior of a real system and is therefore only an approximation to the real behavior, the calculations are often faster than a simulation by magnitudes and influences from external sources are

avoided. The results can also be verified and reproduced quite easily. Last, but not least, the comparison of a simulator and an analytical model helps to find inaccuracies and bugs in both approaches and gives new insights into the underlying principles. Finally, formal methods are useful for verifying properties of a protocol such as correctness and consistency.

1.1 Contributions

This thesis combines the different approaches to take a holistic view on the scalability of wireless multi-hop networks. They also form the basis for proposing improvements and missing building blocks for effective industrial deployments. In this context, the following contributions are made:

- The slot allocation handshake of DSME is analyzed by formal methods under consideration of link and node failures. A modification for the handshake is proposed to improve the consistency of the schedule.
- Three traffic-aware scheduling algorithms for distributed generation of stable slot schedules on the basis of token-passing are presented as well as a technique for decentralized and dynamic construction of traffic-aware slot schedules based on the slot allocation handshake of DSME.
- The open-source implementation openDSME for executing DSME in simulators and on wireless hardware is presented[1]. It was initiated by the author and developed with support by a team of students at the Institute of Telematics of the Hamburg University of Technology.
- By using openDSME and the OMNeT++ simulator, a simulative evaluation of DSME and the proposed distributed scheduling mechanism is conducted and its performance is compared to CSMA/CA.
- Analytical models for fast and verifiable assessment of large-scale wireless multi-hop networks are developed[2]. They incorporate the specific properties of CSMA/CA, TSCH and DSME to give insights into the characteristics of these techniques and demonstrate the importance of traffic-aware schedules.
- Also, openDSME is evaluated in the FIT/IoT-LAB testbed for demonstrating the applicability to a physically deployed network. In addition, a technique for generating multi-hop topologies in dense wireless testbeds is proposed[3].
- Several aspects relevant for system integration in industrial applications are presented. This includes a method for over-the-air software updates and a dual-radio approach for reliable signalization of emergency conditions.

[1] Implementation available at http://www.opendsme.org
DOI 10.5281/zenodo.3147550

[2] Implementation available at https://github.com/koalo/AnalyticalMultiHop
DOI 10.5281/zenodo.3148187

[3] Implementation available at https://github.com/koalo/iotlab_topologies
DOI 10.5281/zenodo.3148096

1.2 Dissertation Structure

Chapter 2 presents requirements and foundations of wireless communication in industrial contexts. Common wireless technologies are presented and compared. The IEEE 802.15.4 standard is identified as an adequate candidate for scalable industrial networks and thus Chapter 3 presents important building blocks and existing solutions for achieving maximum reliability with IEEE 802.15.4. For this, a time-slotted medium access is promising and therefore a medium access layer on the basis of DSME is constructed in Chapter 4, based on the formal analysis of the slot allocation handshake and a decentralized traffic-aware and prediction-based slot scheduling. This results in the open-source implementation openDSME presented in Sect. 4.4.

In order to get a comprehensive insight into the performance of existing and proposed techniques for medium access, extensive analyses are conducted. In Chapter 5 this is done by means of the OMNeT++ simulator. Chapter 6 develops analytical models for CSMA and TDMA and proposes slot schedules aiming for maximum throughput. In Chapter 7, the FIT/IoT-LAB is used for analyzing the performance of openDSME in a physically deployed network. Some challenges and solutions that occur when building actual industrial applications are presented in Chapter 8 and Chapter 9 finally concludes the thesis.

Chapter 2

Wireless Networks for Industrial Applications

The use of wireless instead of wired communication technologies promises a high flexibility and a low cost for the deployment. It does, however, come with several challenges that have to be addressed to properly fulfill the stringent requirements in industrial applications.

One issue is the lower determinism of wireless connections. While building long transmission lines with very high data rates is a challenging task, too, the parameters of a given cable connection can be predetermined and measured comparatively easy and the performance of a once built cable connection rarely varies as much as the one of a wireless link does. For wireless connections, on the other hand, link qualities can only be approximated roughly since the environment has a huge impact on the transmission. Furthermore, these conditions can change rapidly and by a huge amount. Reasons for this include the addition and removal of damping or reflecting material, for example doors or human bodies, in the transmission path but also outside of a straight line between sender and receiver due to multi-path propagation where the direct path is often not the one that leads to the highest signal energy at the receiver. This already shows that the consideration of the peculiarities of wireless communication are very relevant for the success of a deployment.

In this chapter, the requirements for industrial communication are presented together with an overview of existing techniques to fulfill these with wireless communication. The focus is on the tradeoffs that are necessary to select the adequate technology for a given application in the Industrial Internet of Things.

2.1 Requirements for Industrial Communication

As highlighted before, reliability and availability are critical requirements for the success of a deployment of a communication system in industrial applications. These, however, go hand in hand with many others of varying importance for a given application. The most relevant ones for the subsequent discussion are given in the following. Corresponding requirements for the use of wireless technology in industrial automation can for example be found in [NE124].

Reliability in general describes the ability of a system to fulfill its anticipated purpose over a given period of time. In the context of this thesis, it more specifically describes the probability that a given chunk of information is finally delivered at its destination.

Availability is very much related to reliability, but explicitly covers irregular events such as maintenance or occasional external disturbances. While a system might have a high reliability over hours or days, heavy use of other competing wireless devices, for example, might make it dysfunctional at some times.

Throughput is a measure for the possible amount of traffic that can be processed by the system. In particular, in the context of this thesis it is usually the number of packets that is successfully delivered per second. This requirements obviously depends a lot on the application.

Timeliness requirements can be very different depending on the application. While some applications can even be realized with latencies in the range of minutes or hours, for example in many predictive maintenance applications, other applications could struggle if messages are delayed by some milliseconds, for example in industrial control of fast processes.

Safety is essential in many industrial applications, where faulty, missing or delayed delivery of information can lead to expensive damages to the plant or even threaten the live and limb of people. It is highly dependent on reliability and availability, but deserves extra attention since otherwise even a single fault in years might lead to catastrophic consequences.

Security issues can threaten the integrity, availability and confidentiality of the system and also affect the safety of the plant. Wireless systems are especially vulnerable since traditional security measures such as walls and doors can only give partial protection against eavesdropping and intrusion. Autonomous devices are often under less supervision than a personal device such as a smartphone or laptop and maintaining software updates and security patches comes with additional effort (cf. Sect. 8.2). Thus, existing security measures for PCs are not sufficient for the IoT and new methods have to be developed [TPK17].

Coverage requirements can range from a radius of some millimeters where for example a slip ring is replaced by a wireless communication to many square kilometers in large solar plants or even larger if direct wireless communication between different facilities is required or for monitoring of large infrastructure assets such as pipelines.

Energy Consumption does not seem so much of an issue in an industrial plant where machines with several kilowatts of power are running 24/7, compared to wireless devices with at most a few watts. However, often the full potential of the flexibility that comes with the use of wireless technology can only be reached if the devices are powered by batteries or renewable energy sources such as solar panels. In these cases, minimizing the energy consumption is essential to increase the maintenance interval for replacing the batteries or to even enable the use of renewable energy sources.

Cooperativeness is the ability of a system to work in parallel with other systems competing for the same resources. In wireless systems this primarily means to have low

transmission power and a low bandwidth consumption. It is forbidden by law to exceed certain limits and exclusive spectrum assignments are usually not available for single industrial applications.

Costs should be kept as low as possible. If a wireless system would be too expensive, wired solutions are often more appropriate or even the application itself is not realizable at all since the expected revenue is too low.

Scalability is the ability of a system to support a high number of devices. The requirements can vary by a large degree. While often a point-to-point communication between two devices is sufficient, other applications such as a solar tower power plant call for hundreds of thousands of connected devices. As the main focus of this thesis, scalability incorporates several of the already mentioned requirements. While posing some challenges on its own, supporting a high number of devices usually comes with the demand for a large coverage and a high throughput. Also, deployment and maintenance get more complex, so a low energy consumption and low costs get even more important.

2.2 Foundations of Wireless Communication

In the following, a short overview over the foundations of wireless communication is given and concepts relevant for the thesis are introduced. Fig. 2.1 depicts the simplified transmission chain between two transceivers. The modulator generates a signal from the binary data which is amplified by the power amplifier (PA). The signal is then subject to various losses and gains. Besides losses between the transceivers and the antennas and the gain of the antennas, which highly depends on the mutual alignment of the transceivers, the main component is usually the path loss.

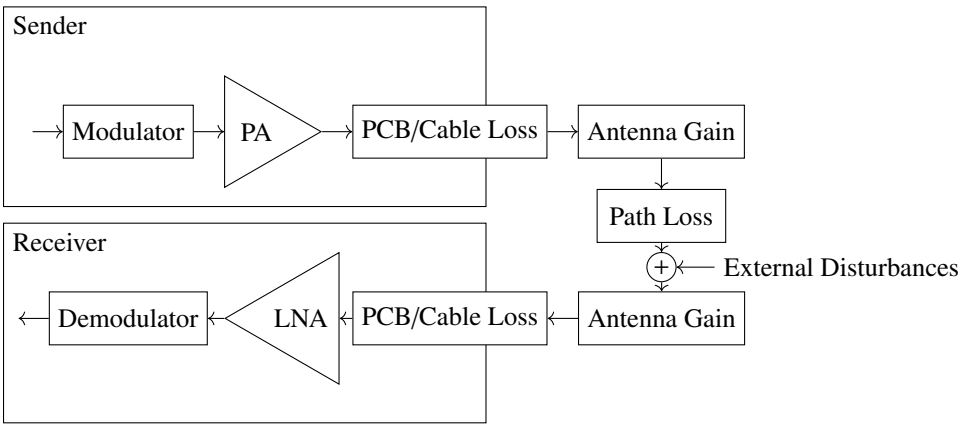


Figure 2.1: Simplified transmission chain between two transceivers.

In the most simple case, especially without any obstacles and without fading effects for example due to multi-path propagation, the path loss factor is given by the free-space path loss

$$FSPL = \left(\frac{4\pi r \cdot f}{c}\right)^2, \tag{2.1}$$

where r is the distance between sender and receiver, f the transmission frequency and c the speed of light. The loss increases quadratically with the distance and the frequency.

The signal is finally amplified by the low-noise amplifier (LNA) of the receiver and demodulated. The received signal power and the noise and interference power collected over the path determine the quality of the reception that is in particular the bit error ratio (BER). In contrast to voice communication where data is transmitted as a stream (even though it might be fragmented inbetween) and a certain amount of bit errors can be tolerated, this is not the case for most industrial applications where usually short data packets are transmitted (e.g. a certain sensor reading at a given point in time). In this case, a packet of L bits with at least one unrecoverable bit error is dropped and the crucial metric is the packet error ratio (PER) over a link calculated as

$$\text{PER} = 1 - (1 - \text{BER})^L. \tag{2.2}$$

The BER depends on the data coding and the modulation, but they can not reduce the BER indefinitely without affecting other system properties. In fact, there is a maximum achievable channel capacity C in bits per second, according to the Shannon-Hartley theorem

$$C = B \cdot \log_2\left(1 + \frac{S}{N}\right), \tag{2.3}$$

where B is the bandwidth in hertz, S is the received signal power in watts and N the noise and interference power in watts. Thus, a high channel capacity, required for a high throughput, calls for a large bandwidth, high signal power at the receiver and low noise and interference power.

However, a large bandwidth means to consume a larger portion of the frequency spectrum, decreasing the cooperativeness and it is only possible within certain ranges given by the regulations. Furthermore, for a constant spectral density of the noise, a larger bandwidth also implies that the receiver collects proportionally more noise and potentially more interfering signals, leading again to a lower channel capacity. While an increased bandwidth is a direct consequence of faster switching of symbols, a high signal to noise ratio only leads to a higher throughput if an adequate modulation and coding scheme is applied, so (2.3) is only an upper bound. This simplified example already shows that tradeoffs are inevitable as elaborated in more detail in Sect. 2.8.

2.2.1 Medium Access Control

Due to the inherently shared nature of wireless communication, regulatory, operational and technical measures have to be set up to mitigate mutual interferences. Thus, medium

access techniques are an essential part of wireless systems. While applications such as radio broadcasting can circumvent this by dedicated frequency assignments and in other applications such as amateur radio, human operators strive for minimizing interferences, autonomous sensors and actuators have to coordinate their transmissions automatically. The goal of the Medium Access Control (MAC) is to separate transmissions in time, space, coding or frequency to avoid disturbances.

Time separation can be achieved by two fundamental approaches to avoid simultaneous transmission. Either the transceivers agree on a common schedule beforehand and use it for future communication (Time Division Multiple Access, TDMA) or the channel is sensed for ongoing transmissions and transmissions are delayed until no other transmission is recognized (Carrier Sensing Multiple Access, CSMA).

Space separation can exploit the fact of different reception powers throughout the system, for example when using multiple base stations or multi-hop networks as presented in the next section. Directional antennas can increase the effect at the cost of flexibility.

Coding allows to differentiate signals from each other even if they would interfere otherwise (Code Division Multiple Access, CDMA).

Frequency is often used to separate different systems from each other, but using different frequency within the same network is also possible by an appropriate schedule (Frequency Division Multiple Access, FDMA).

Medium access is also an import feature in wired systems. However, techniques such as carrier sensing multiple access with collision detection (CSMA/CD) as used for example in Ethernet can not be applied since the signal power at the transmitter is usually much larger than at the receiver, so sniffing the channel to recognize collisions while transmitting at the same time is usually unfeasible in contrast to wired systems.[1] For the same reason, collision resolution as in CAN is not possible. Therefore, MACs that rely on carrier sensing (such as CSMA/CA as presented in Sect. 3.2) are prone to the hidden node problem.

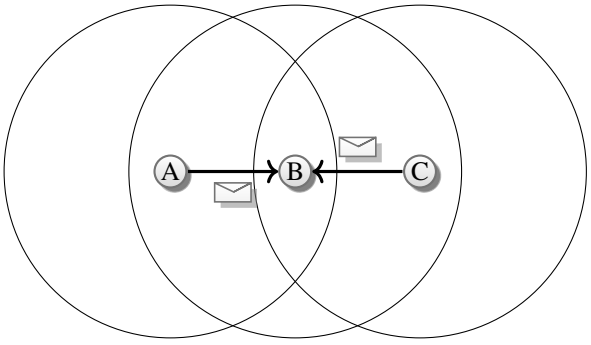


Figure 2.2: Hidden Node Problem.

[1] There are, however, approaches to mimic collision detection in wireless systems, denoted as collision notification [SCN12].

The hidden node problem denotes the situation where a disturbing transmission cannot be sensed by another transceiver as depicted in Fig. 2.2. Out of the wrong assumption that the channel is free, both transmitters might transmit at the same time, leading to interference and thus packet loss at the receiver. The influence on the performance is, for example, analyzed in [PMB09, ZL07]. The hidden node problem and increased incidence in very large wireless networks is the main motivation for the techniques analyzed and developed in this thesis.

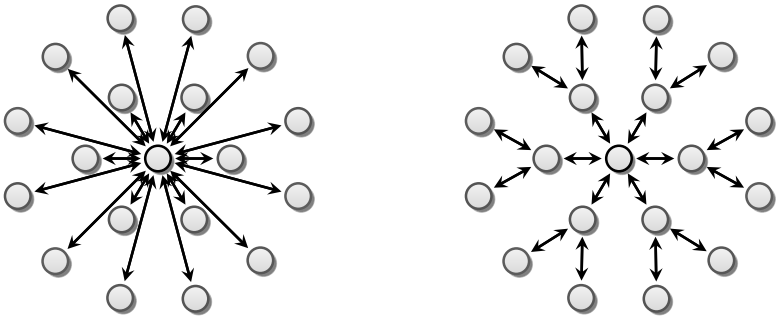
2.2.2 Multi-Hop Networks

One possibility to achieve a large coverage is to forward messages over multiple hops. This principle is depicted in Fig. 2.3. While in the single-hop case all transceivers communicate directly with the single basestation in the center, the basestation communicates with only 6 neighbors in the multi-hop case and messages by the outer nodes are forwarded via them.

Theoretical analysis [BS06] suggests that, depending on the distance, using multiple hops can be more efficient in terms of bandwidth and energy. Also [PMB10] confirms the energy-efficiency of multi-hop if the power consumption for transmitting is sufficiently larger than the one for reception. Though, the same paper notes that this does not imply that for a given wireless technology using more hops is always more energy-efficient, because the assumption of a high power consumption for transmission does not necessarily hold, especially for devices optimized for smaller ranges (such as ones for IEEE 802.15.4).

In practice, the primary reason for using multi-hop networks is that it allows extended coverage even under constrained conditions, for example, if the regulations disallow high transmission power. Multi-hop communication is especially interesting if the deployment of sensors (or actuators) is dense anyway, allowing the devices to act both as end device and forwarding device. The disadvantage is a high complexity that comes with the construction of large scale multi-hop networks, including the development of an adequate routing algorithm that is not necessary for single-hop transmission.

The expected gains of multi-hop communication together with the interesting theoretical and practical challenges have led to the flourishing research area of wireless sensor (and actuator) networks.



(a) Single-Hop (b) Multi-Hop

Figure 2.3: Illustration of wireless network topologies.

2.3 Low Power Wide Area Networks

The main feature of the Low Power Wide Area Network (LPWAN) approach is the ability to cover large areas without the need to forward messages within the wireless network. The term covers several technologies, including LoRa and Sigfox. They usually use license-exempt frequency bands lower than 1 GHz such as the band at 868 MHz in Europe, allowing for a high coverage. This, however, comes with the problem of the narrow width of these frequency bands. The LPWAN approach makes a virtue of necessity and reduces the bandwidth even further thus reducing the amount of noise collected by the receiver and by this improving the sensitivity. This allows to achieve a large range in the order of several kilometers, at least for good channel conditions such as mounting the base station on a tower and the end devices on the roof of a car as in [PMR15].

For LoRa, different transmission powers, carrier frequencies, spreading factors, bandwidths and coding rates can be selected to tailor the system properties to the application [BRV16]. For example, with a coding rate of $4/5$ and a bandwidth of 125 kHz, the selection of the spreading factor can result in a transmission duration for a 20 bytes packet between about 72 ms (corresponding to $\approx$ 2.2 kb/s) and about 1.8 s ($\approx$ 89 b/s) [FB18]. With higher spreading factors, the sensitivity can be improved to achieve a larger coverage but it comes at the cost of a very low data rate.

Sigfox drives the concept of a small bandwidth even further with 100 Hz subbands. It achieves 100 b/s with 12 bytes payload (plus 26 bytes overhead) in uplink and 500 b/s with 8 bytes in downlink [FB18]. It is a proprietary technology promoted and deployed by the Sigfox company on the basis of a operator model supporting end-to-end connectivity from IoT devices to cloud applications.

2.4 Cellular Networks

The ability of ubiquitous Internet access and telephony in most developed areas is foremost enabled by the use of cellular networks. Their unique feature compared to all other mentioned systems is the usage of a dedicated frequency allocation allowing for unobstructed channel access with high bandwidth. Since the required infrastructure in terms of mobile basestations is broadly deployed, it seems obvious to use the same technology for machine to machine communication.

In order to enable efficient use of cellular networks for the IoT, several improvements to the existing technology are required [MHM18]. For reduced power consumption, the methods presented in the following include a power saving mode that allows the devices to enter a sleep mode. While staying connected to the network they are not reachable for communication.

2.4.1 EC-GSM-IoT

EC-GSM-IoT is an extension to the GSM technology and does not require UMTS or LTE support of the basestation. This allows to deploy it to existing GSM basestations with a software update while allowing to multiplex EC-GSM-IoT and conventional GSM in a

single channel [FB18]. Thus, it covers a bandwidth of 200 kHz per channel in accordance to conventional GSM. For extended coverage (EC), the same packet is blindly retransmitted several times, resulting in a throughput down to 350 b/s for maximum coverage. The throughput can be increased at the cost of coverage, for example up to 240 kb/s by using less retransmissions and another modulation, i.e. 8PSK with EDGE [MHM18].

2.4.2 LTE-MTC

MTC (Machine Type Communication) is an extension to LTE and allows to reuse LTE basestations per software update [ASF17]. It provides for example low latencies by using a subframe of 1 ms. For low costs and long battery life, new device types are introduced, for example LTE Cat 0 with reduced data rate (1 Mb/s in up-and downlink). LTE Cat M1 is optimized for long battery life for example by optionally reduced transmission power (20 dBm instead of 23 dBm) [MHM18].

2.4.3 NB-IoT

In contrast to the other two, Narrowband-IoT (NB-IoT) defines a new physical layer, including different synchronization signals and different signal bandwidths [MHM18]. The new carrier defined by NB-IoT can be used standalone, in the guard interval of LTE or multiplexed with an LTE carrier. With single-tone modulation, 60 kb/s are possible in downlink and 20 kb/s in uplink. Furthermore, while conventional cellular technology uses scheduled transmission (FDMA, TDMA and CDMA), a random access channel is specified to support asynchronous traffic.

2.4.4 5G

5G is the term for the next mobile network generation with the introduction expected at about 2020 [LSW17, Ch. 10]. Besides massive increase in the provided data rate for new broadband applications (eMBB), it aims for supporting massive (mMTC) and critical (cMTC) machine type communication. 5G mMTC builds upon the above mentioned LTE-MTC and NB-IoT but proposes some improvements for reduced latency and better resource utilization.

2.5 IEEE 802.11

IEEE 802.11, also known as Wi-Fi after the Wi-Fi Alliance that promotes and certifies IEEE 802.11-compliant devices, is a technology primarily used for wireless communication of notebooks and smartphones with a local area network (WLAN). Since its first specification in 1997, a huge number of extensions were proposed and implemented to increase the achievable data rate or to tailor it to new application domains, such as IEEE 802.11p for vehicular environments. Thus, IEEE 802.11 comes with a high complexity, including a multitude of modulation schemes (BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM), channel codings (convolutional or LDPC with different coding rates) and channel bandwidths (from

20 MHz to 160 MHz) in the VHT PHY of IEEE 802.11-2016 [802.11-2016]. Furthermore, several spatial streams and beamforming are supported. With good channel conditions this can lead to very high throughputs (e.g. according to the standard up to 600 Mb/s for the HT PHY with 40 MHz bandwidth and four spatial streams), while for bad conditions the selection of more robust settings is possible to still maintain connectivity. For improving support for IoT applications, the IEEE 802.11ah extension [802.11ah-2016] is proposed that uses frequencies under 1 GHz for extended coverage. Still, a similar multitude of modulations and codings is available, resulting in a PHY rate from 150 kb/s to 78 Mb/s [MMG17].

IEEE 802.11 networks usually use CSMA/CA with RTS/CTS for medium access, but methods for scheduled transmissions, such as the Point Coordination Function and other proposals [PAB15] exist to provide more reliable medium access. Mesh networks are supported by IEEE 802.11s that specifies the routing protocol Hybrid Wireless Mesh Protocol (HWMP) on top of the IEEE 802.11 MAC. This allows for extended coverage, for example for the application in Advanced Metering Infrastructure [TA18].

The focus of IEEE 802.11 is on high data rate, but it nevertheless specifies methods for energy-efficiency, namely the optional Power Saving Mode (PSM). The access point regularly sends beacons, announcing pending packets. The wireless station regularly wakes up to receive beacons and potentially requesting pending data [HHR16].

2.6 Bluetooth

Bluetooth is a branch of wireless technologies promoted by the Bluetooth Special Interest Group. While originally targeted at peripherals such as headsets and keyboards, it has found broad application in many wireless short-range applications and is thus deployed in most smartphones [MMG17]. The first version was released in 1999 [Sch15].

Earlier versions of the standard were ratified as IEEE 802.15.1, but that standard is now inactive and newer versions are solely published by the Bluetooth SIG. A major step towards the IoT was Bluetooth 4.0, now superseded by Bluetooth 5 [Blue5]. It specifies multiple core configurations, including enhanced data rate (EDR) with up to 3 Mb/s and high speed (HS), actually an IEEE 802.11 PHY with Bluetooth adaption layer. The most relevant for the IoT, however, is the Bluetooth Low Energy (BLE) configuration. It supports data rates of 2 Mb/s, 1 Mb/s, 500 kb/s and 125 kb/s with GFSK modulation and 40 channels with 2 MHz channel spacing in the 2.4 GHz band.

BLE specifies connection-oriented and connection-less medium access. For a given connection over a link, one node is designated as master and one as slave. A master is responsible for scheduling all message transmissions towards and from connected slaves appropriately. For this, the master sends a packet (with or without payload) to the slave in regular time intervals (connection interval). This starts a connection event that optionally includes further back and forth transmission of packets between slave and master. For increased resistance to interference, the channel is changed for the next connection event. This is different to conventional Bluetooth where the frequency can even be changed during a single packet transmission (multi-slot packets). Outside of a connection, message transmission is possible in form of advertisements that can optionally include data payload.

These are usually sent in bursts on multiple channels and potential receivers hop over the channels to get the chance to receive the packet.

Using multi-hop networks to extend the coverage of Bluetooth networks is not a new topic [LLS03], but it has recently gained more momentum in the context of BLE, including the publication of an official Bluetooth Mesh Specification [BlueMesh]. In [DGD17], further approaches for BLE mesh are compared and [Mey18] develops an analytical model for connection-based multi-hop in BLE networks considering external interference.

2.7 IEEE 802.15.4

The IEEE 802.15.4 standard targets very low-cost and low-power communications [802.15.4-2015]. It was specifically designed to target application domains not properly addressed by the versions of Bluetooth and IEEE 802.11 at that time (initial version of IEEE 802.15.4 published 2003) and but looks back on a similarly changeful history by now [Sch15]. In the meantime IEEE 802.15.4 has incorporated several different PHY specifications (19 by now) [802.15.4-2015]. Popular ones include O-QPSK with Direct Sequence Spread Spectrum (DSSS) in the 2.4 GHz band with 250 kb/s bit rate and BPSK in the 868 MHz band with 100 kb/s. In the 2.4 GHz band, 16 channels with 5 MHz channel spacing can be used.

Even the first IEEE 802.15.4 version of 2003 was designed with flexibility and scalability in mind by providing mesh network capabilities, also known as peer-to-peer topology. For building mesh networks, CSMA/CA as specified by the standard is popular. Since it suffers from the hidden node problem, several approaches for medium access based on time slots (TDMA) and multiple channels (FDMA) were proposed for the use in demanding industrial applications that promise a higher resilience against collisions and packet loss compared to CSMA/CA. In 2008, the Time Synchronized Mesh Protocol (TSMP) [PD08] was proposed for IEEE 802.15.4 multi-hop networks. It was later standardized in the two competing standards WirelessHART [IEC10] and ISA100.11a [ISA11] that have gained a lot of attention for industrial automation [PL10]. In 2012, the amendment IEEE 802.15.4e [802.15.4e-2012] was published, including a standardization of TSMP as Time-Slotted Channel Hopping (TSCH). Furthermore, an alternative approach was included, the Deterministic and Synchronous Multi-Channel Extension (DSME) that is more closely related to the already existing slot structure for single-hop IEEE 802.15.4 networks. TSCH and DSME got merged with the IEEE 802.15.4 standard in the 2015 version [802.15.4-2015]. Since IEEE 802.15.4 is the technology used in this thesis, more details will be provided in Chapter 3.

2.7.1 ZigBee

The ZigBee specification [ZigBee12] defines a network (NWK), security (SEC) and application (APL) layer on top of IEEE 802.15.4. In ZigBee each application confirms to a application profile that specifies the formats and actions regarding the messages and provide a method for interoperable devices by multiple vendors. Also methods for device and service discovery are specified. Since it was the earliest complete stack specified for IEEE 802.15.4

(first draft in 2004) it has found broad usage for example for home automation. Thus, ZigBee is often wrongly used to refer to other IEEE 802.15.4-based networks.

2.7.2 6LoWPAN

The ongoing effort of the Internet Engineering Task Force (IETF) is to provide seamless communication in constrained wireless networks with Internet technology [TPK17]. One goal is to implement the so-called hour-glass model to avoid the combinatorial explosion while porting applications to other communication technologies.

IPv6 over Low power Wireless Personal Area Network (6LoWPAN) defines routing (details in Sect. 3.4), header compression and packet fragmentation for IEEE 802.15.4 networks. Obeying the hour-glass model, 6LoWPAN techniques are also ported to other technologies, for example IPv6 over BLE [NSI15]. In [GPB17] more details and an overview of the history of 6LoWPAN and related standardization efforts are provided.

2.7.3 WirelessHART

WirelessHART [IEC10] is based on the Highway Addressable Remote Transducer (HART) standard that has widespread use in the context of wired communication in industrial applications and is proposed by the FieldComm Group (called HART Communication Foundation before merging with the Fieldbus Foundation). Since WirelessHART shares most higher layer services and protocols with wired HART, it promises seamless integration with existing HART-based industrial networks and applications. The applicability to factory automation is for example analyzed in [PC09], also under consideration of external interferences. The medium access is based on time slots of 10 ms and employs channel hopping. The routes and the slot schedule are calculated and deployed by a central entity, the network manager, while no methods for calculating them are defined in the standard. The schedules are applied locally on every device.

For time synchronization, every device has at least one designated device as time source as neighbor. Since the time between slot start and start of the message transmission is predefined, a receiver is able to calculate the slot boundaries after it received a packet from a time source. This is possible as long as the drift is small enough. If the number of packets sent out per time is too low, keep-alive packets are sent by the time sources.

2.7.4 ISA100.11a

ISA100.11a [ISA11] is proposed by the International Society of Automation (ISA) and is also based on IEEE 802.15.4. In contrast to WirelessHART, it relies heavily on 6LoWPAN. The medium access is very similar to WirelessHART, but provides more options, such as the use of configurable slot lengths and slow hopping where the channel is constant over a longer duration, thus allowing for more relaxed time synchronization. The time synchronization is performed similar to WirelessHART by measuring the time a packet has arrived. It is also possible to synchronize a sender of a packet with the receiver by letting the receiver measure the arrival time and then sending the time difference back in the acknowledgment. According to [Nix12] that compares WirelessHART and ISA100.11a,

one major difference is the higher flexibility of ISA100.11a allowing it to better tailor it to a given application. The disadvantage is that this impedes interoperability between different devices.

2.8 Tradeoffs for Selecting a Wireless Technology

None of the technologies presented above is *the* one-size-fits-all solution for the IoT. They all have their advantages and disadvantages making trade-offs inevitable that have to be made on a per-application basis. Besides technological aspects, this also includes for example economical considerations and regulatory restrictions. Also, marketing models can influence the decision, for example the possibility to modify a provided stack [Mey18, Sect. 4.7][WT15]. Sjöström [Sjö17] extensively compares technical as well as non-technical aspects of wireless technologies for the IoT, including SigFox, LoRaWAN and cellular networks and discusses the advantages and disadvantages of the different approaches for network ownership, intellectual property and financing.

2.8.1 Range

The achievable range for single-hop communication is the result of the interaction of several factors as already covered in Sect. 2.2. These can be broadly summarized as transmission power, gains and losses towards the receiver and the ability of the receiver to decode a signal depending on the reception power while achieving a sufficiently low error rate, also denoted as sensitivity.

High transmission power leads to high power consumption during transmission. This might or might not have a negative influence on the total energy consumption as discussed in Sect. 2.8.4. Directional antennas for increasing the radiated power might also be feasible for some applications, especially point-to-point connections with high throughput, but they increase the effort for the deployment and make the system less flexible, especially if a device communicates with more than one other device or they are mobile[2]. A major reason for not using higher transmission powers, however, are regulations. For example, the maximum allowed effective radiated power between 868.0 MHz and 868.6 MHz is 25 mW in Germany [BNetzA18]. With regard to path loss, a main factor that distinguishes wireless technologies is the used frequency. Equation (2.1) treats sub-GHz technologies with favor. Special modulation techniques such as orthogonal frequency-division multiplexing (OFDM) allow for mitigation of fading effects caused by multipath propagation.

Achieving low noise at the receiver is technologically challenging, since the hardware design includes several tradeoffs in itself, such as costs, noise figure and power consumption [Lar14] making it difficult to compare the achievable sensitivity of different wireless technologies. Still, under comparable conditions, that is a transceiver chip integrating multiple wireless technologies, IEEE 802.15.4 can achieve a 6 dB better sensitivity than BLE [ZSY17], suggesting a distinctly larger coverage. This is confirmed by empirical measurements in a home IoT scenario resulting in a 5 dB difference of IEEE 802.15.4 and

[2] One possibility to support this is beamforming as in the IEEE 802.11ac standard.

BLE [FTZ16]. The theoretical range advantage of IEEE 802.15.4 compared to Bluetooth classic and IEEE 802.11b is also covered in [PRM06].

This advantage of IEEE 802.15.4 is mainly related to the lower bit rate in contrast to BLE. As mentioned in Sect. 2.2, in general a higher bandwidth leads to a higher noise power and thus for everything else being constant, the distance has to be smaller to get the same S/N. For this reason also, LPWANs are those of the above technologies with the lowest bit rate, but the largest range.

Though, the relevant bandwidth of the information transmission is not necessarily equivalent to the occupied bandwidth in the spectrum. This is in particular relevant for spread-spectrum techniques, where every information bit is transmitted as a high-frequent sequence of chips, or other similar coding techniques. They increase the occupied bandwidth for the same bit rate or reduce the bit rate for constant occupied bandwidth. The latter effect is the reason why spread-spectrum or other coding techniques are often seen as a way to increase the range, but this improvement can usually mainly attributed to the smaller bandwidth of the information transmission that reduces the noise power and thus increases the range for a given signal power. Still, the advantage of spread-spectrum is the increased resilience to narrowband disturbances and the enabling of code division multiple access, while other codes are for example useful to mitigate burst errors. Such coding is therefore applied in most of the above mentioned technologies and the spreading or coding factor is often even adjustable to trade bit rate and range. This is for example the case for LoRa and the LE Coded PHY of Bluetooth 5.

2.8.2 External Infrastructure

While most of the presented technologies can be deployed with independent infrastructure, some require the usage of external infrastructure and the consultation of a network provider. This is the case for Sigfox and for cellular networks, even though there are recent efforts to enable the deployment of dedicated cells (CBRS [Mun17]) or device-to-device communication [MHM18]. Using external infrastructure comes with several advantages and disadvantages [LSW17, Sect. 9.2].

First of all, using external infrastructure can largely simplify the deployment and maintenance, especially for city-scale or even larger application areas. Depending on the circumstances this might even outbalance the explicit regular payments that come with the usage of the external infrastructure, partly required for refinancing the utilization of the licensed spectrum. The advantage of a simplified deployment turns into the opposite if the intended application area is not yet covered by proper cellular connectivity. A solar tower power plant in the desert is an extreme case of this. Furthermore, for critical applications it can be crucial to have complete control over the system at any time, especially during longer down-times that are not frequent but definitely not unheard of for large communication networks [COP03]. In critical infrastructure assets, such as power plants, the usage of external infrastructure could even lead to complex circular dependencies in emergency situations.[3]

[3] During the 2003 power outage in Italy, one of the problems was the dependence of the SCADA system of the power grid on the availability of a telecom company that itself suffered from the outage [UCTE04, p. 47].

2.8.3 Scalability Issues

Some standards impose hard limits on the number of supported devices per network. For example, conventional Bluetooth allows only up to 7 connections per master. Such restrictions are rare, but several issues can constraint the performance of large-scale networks. Some are exemplarily presented in the following.

Very large LPWANs mainly suffer from the low bitrate and limited allowed duty cycle leading to a high number of collisions already for low traffic demands. The authors of [BRV16] conclude that only 120 nodes sending 20 Bytes every 16.7 min can be supported within 3.8 ha when using a single basestation and still only about 90% of the transmissions are successful when using typical parameters. In [AVT17] further limits of the higher layer complement LoRaWAN are identified, including the difficulty to transmit downlink traffic and the necessity to minimize the number of acknowledgments to avoid further reduction in throughput at the cost of reliability. One also has to consider the usage of the same band by others (using the same or another technology), so cooperativeness is especially important for LPWANs due to their large coverage.

A better medium access technique could improve the performance of LPWANs, but in typical deployments end devices have a much higher path loss to other end devices than to the base station, rendering simple listen-before-talk techniques ineffective. The LPWAN approach can, however, shine in smart city applications where only very few messages per day and device are required. Also according to [BRV16], a dense deployment of base stations would be beneficial, corresponding to the concept of cellular networks.

One issue with LTE networks is the scalability due to the expected high signaling overhead in M2M scenarios [JJM15]. Typical smartphone applications have high and bursty downlink demands while M2M traffic demands are often lower but more scattered. Thus, signaling overhead, for example for connection establishment, is very relevant especially if the devices disconnect regularly, in the case of mobility (e.g. asset tracking) or if their traffic patterns are synchronized (e.g. transmission of a data packet at the beginning of every second). At least, the existing deployments and broad dedicated frequency bands allow for large coverage in many areas.

This is different for BLE with its small range making the construction of mesh networks inevitable. This is, however, hampered by the medium access that is optimized for small star topologies. The main problem is overlapping of connection events leading to recurring collisions [PLB16]. To circumvent this, many approaches, first of all the official Bluetooth Mesh Specification [BlueMesh], propose the use of advertising packets instead of connections for data transfer. This, however, leads to a high probability of collisions [GVS18] and a significantly reduced throughput [MPT13]. Together with constraint low power operation for relays [BlueMesh, Sect. 2.2.5] and the use of managed flooding [BlueMesh, Sect. 2.2], it is questionable if BLE will allow the realization of scalable and efficient mesh networks in the future without significant changes of the core and mesh specification. Still, broad application of BLE in the IIoT can be expected with focus on smaller networks and especially in combination with smartphones. For other applications, IEEE 802.15.4 is favorable since is does not come with these impediments as outlined in Chapter 3.

2.8.4 Energy Consumption

The average energy consumption, and with it the achievable lifetime of a battery driven device, depends on the power consumption in every state, such as transmission, reception and sleep, as well as the time spend in the states. Thus, for example, a high power consumption in the transmission state can be compensated with a short transmission duration if it is possible to stay in a low-power sleep state most of the time.

In [HHR16], the energy consumption of IEEE 802.15.4 and IEEE 802.11 is compared for consideration in an energy harvesting application. In the hardware comparison given in the paper, the power required in the transmission and reception state is several times larger for IEEE 802.11. Still, due to the higher data rate of IEEE 802.11 the number of bits that can be transmitted per consumed Joule is is much larger for IEEE 802.11 than for IEEE 802.15.4. However, for exploiting this advantage efficient power saving techniques are required that are not broadly applied, yet, and even existing implementations are not free from problems that thwart the application in energy-constraint scenarios [HHR16].

The achievable lifetime of a battery-driven device with several wireless technologies is compared analytically under various traffic demands in [MMG17]. With the highest available data rate of 2 Mb/s and connection-oriented data transmission, the paper attests BLE the best lifetime of the considered wireless technologies over all traffic demands. Connection-less data transmission in advertisements is not considered, so no statement can be made about energy consumption in networks according to the BLE mesh specification.

For only a single packet per day, [MMG17, Fig. 16] shows the same lifetime for IEEE 802.15.4 and BLE, while LoRa is close. But LoRa gets significantly worse for increasing traffic demand. IEEE 802.11 PSM and IEEE 802.11ah have a comparatively short lifetime for low traffic, but can outperform IEEE 802.15.4 for short transmission intervals of about 100 s and below and at the same time large packets of about 1000 Bytes and above.

The paper also includes an interesting analysis of the impact of the bit rate depicted in [MMG17, Fig. 19]. Lowering the bit rate of BLE by means of the LE Coded PHY, the achievable lifetime drops significantly and for $S = 8$ it is below the one of IEEE 802.15.4, while on the other hand a non standard conforming IEEE 802.15.4 device with 2 Mb/s[4] would come close to the lifetime of BLE with 2 Mb/s.

2.9 Conclusion

Many technologies are being developed in the area of the Internet of Things and while all have their individual justifications and prime applications, currently IEEE 802.15.4 seems to provide the best compromise for scalable industrial wireless networks, in particular in the presented application of a solar tower power plant, making it worth for further consideration and improvements. But also the other technologies can and will profit from improvements to their specifications. This also leads to convergence of the technologies and higher diversity of the particular standards to aim for a maximum market share. An example is the extension of BLE with the LE Coded PHY and mesh support for higher coverage. This, on the other

[4] This is actually realizable with the ATmega256RFR2 [Atm14], a popular IEEE 802.15.4 SoC.

hand, thwarts several of the established advantages of BLE. A thorough evaluation and comparison of the existing technologies and their parameters for every given application is therefore inevitable.

Still, one might be tempted to develop a new wireless system from scratch to perfectly tailor it to a given application. In consideration of the high complexity required for achieving the performance of today's systems, this is not recommended. Though, with new achievements in the area of software defined radio (SDR) such as [CLK16] one can expect a future trend from dedicated physical layer specifications implemented by different ICs towards more flexible approaches that allow for extensive adaptation possibilities on standardized and open platforms.

Chapter 3

Scalable Multi-Hop Networks with IEEE 802.15.4

Building scalable and reliable wireless networks requires consideration of all layers. In this chapter basics of the IEEE 802.15.4 standard [802.15.4-2015] relevant for this thesis are presented, together with existing proposals for slot scheduling and routing. Prominent large-scale deployments include the Trio Testbed with 557 nodes in an area of 50000 m^2 [DHJ06] and CitySee [MMH12] with 1196 nodes that build a multi-hop topology with up to 20 hops. Such deployments present the applicability of IEEE 802.15.4 for large multi-hop networks, but also the weaknesses in terms of reliability. For example in [LHL13], 60% of the packets were received by the sink in a 330 nodes network where every node sends three packets per hour. Using time-slotted medium access promises much better reliability by reducing or even avoiding collisions, if it is coupled with scalable and efficient slot scheduling and routing.

3.1 Physical Layer

For the calculations in this thesis, only the O-QPSK PHY with DSSS in the 2.4 GHz is exemplarily considered. The presented methods, especially those for the MAC layer are, however, independent of the physical layer and can be easily transferred. Only the reduced number of channels and the reduced data rates in lower frequency band might provide some difficulties for the implementation. With Direct Sequence Spread Spectrum (DSSS), four bits are mapped to a pseudo random noise sequence of 32 chips that together form one symbol. These chips are then modulated with O-QPSK. The duration of one symbol is 16 μs, thus the transmission of one byte (two symbols) takes 32 μs, corresponding to a physical layer data rate of 31250 Bytes per second or 250 kb/s.

3.1.1 Frame Format

The frame structure of IEEE 802.15.4 is presented in Fig. 3.1. The four bytes (also denoted as octets in the standard) preamble together with the 1 byte start of frame delimiter (SFD) are used for synchronizing the receiver with the symbol and byte phase of the transmitter

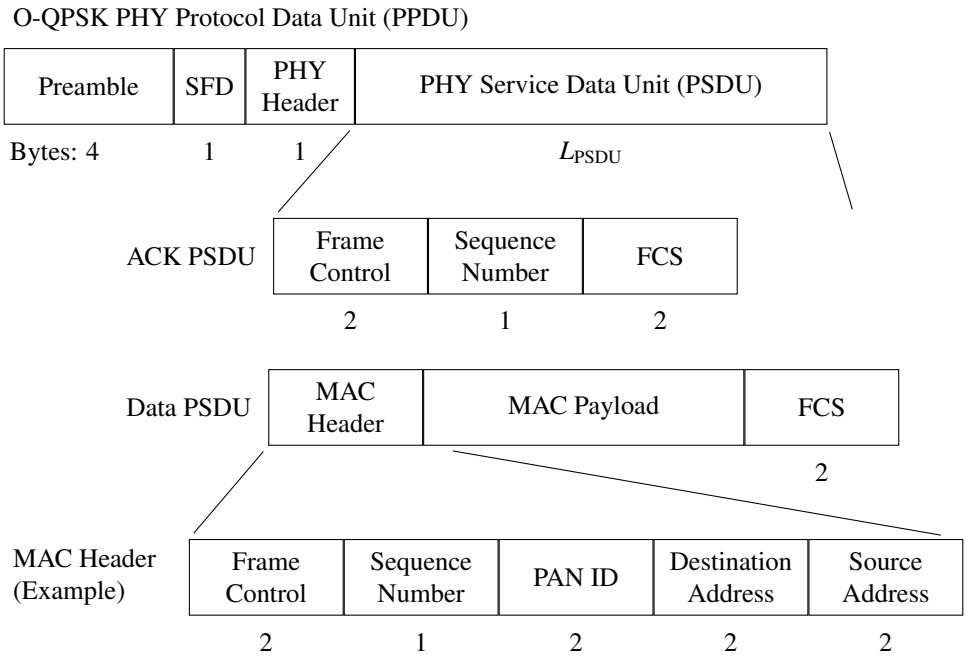


Figure 3.1: Frame Format.

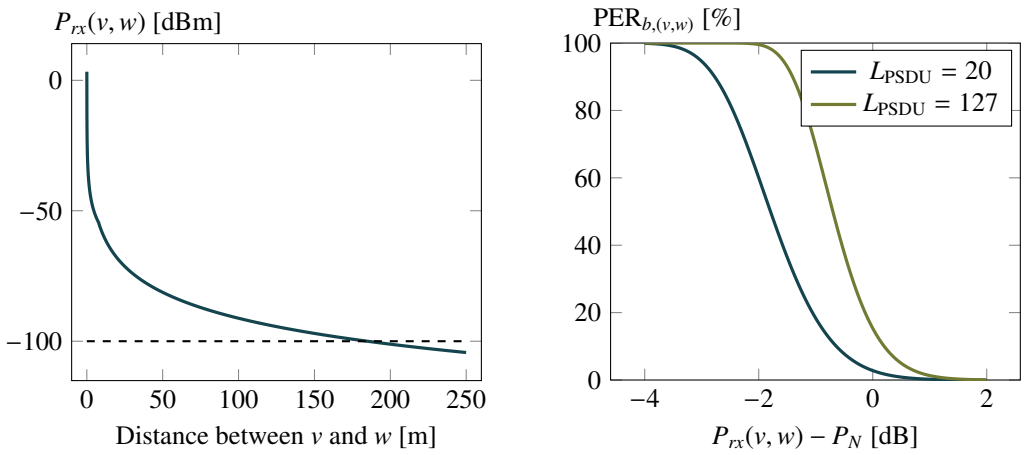
and to indicate the start of the frame. Besides one reserved bit, the PHY header consists of 7 bits indicating the length of the following PHY Service Data Unit (PSDU) in bytes, denoted as L_{PSDU} in this thesis. Thus, the maximum length of the PSDU is 127 bytes.

A data PSDU consists of the MAC header, payload and the frame check sequence (FCS), a 16-bit ITU-T CRC for recognizing transmission errors. The MAC header itself consists of the frame control field containing some flags to specify the type and composition of the packet and in particular the header. In Fig. 3.1, a commonly used header composition is shown, including a sequence number for recognizing duplicates, a network identifier (PAN ID) and 16 bit (short) addresses. Other possibilities include the specification of different PAN IDs for source and destination and 64 bit (extended) addresses. The PSDU of an acknowledgment only consists of the frame control field, sequence number to reference the transmitted data packet and an FCS.

3.1.2 Physical Layer Model

In order to analyze and simulate IEEE 802.15.4 networks in the course of this thesis, the following physical layer model is used. It is presented in the IEEE 802.15.4 standard in the version of 2003 [802.15.4-2003, Annex E] and is based on a breakpoint log-distance path loss model determined by measurements. The received signal power in dBm with distance $d_{v,w}$ between two nodes v and w and transmission power P_{tx} is given by

$$P_{rx}(v,w) = P_{tx} - \begin{cases} 58.5 + 33\log_{10}\left(d_{v,w}/8\,m\right) & d_{v,w} > 8\,m \\ 40.2 + 20\log_{10}\left(d_{v,w}\right) & d_{v,w} \leq 8\,m \end{cases}. \tag{3.1}$$



(a) Reception power for $P_{tx} = 3.5\,\mathrm{dBm}$ and the breakpoint log-distance path loss model.

(b) Packet error ratio for different packet lengths.

Figure 3.2: Exemplary plots for the IEEE 802.15.4 physical layer model.

Fig. 3.2a shows the reception power over the distance for a transmission power of $P_{tx} = 3.5\,\mathrm{dBm}$, that is the maximum transmission power of the ATmega256RFR2 [Atm14], an exemplary system on chip (SoC) integrating an IEEE 802.15.4 transceiver and a microcontroller (also see Sect. 8.1). The dashed line depicts the sensitivity of $-100\,\mathrm{dBm}$ of the ATmega256RFR2. That is the reception power that results in a packet error ratio of 1% for a packet with a PSDU of 20 octets. In real-world applications, obstacles will affect the path loss immensely. In [LHJ18] a path loss model is constructed on the basis of road-side tree and in [TJV08] an industrial indoor channel is characterized.

In order to calculate the probability of errors, the model presented in [802.15.4-2003, Annex E] also includes the computation of the bit error ratio dependent on the noise power P_N. For this, the signal-to-noise ratio

$$\mathrm{SNR}_{(v,w)} = 10^{\frac{1}{10}\cdot(P_{rx}(v,w)-P_N)} \tag{3.2}$$

is inserted into

$$\mathrm{BER}_{(v,w)} = \frac{8}{15}\frac{1}{16}\sum_{k=2}^{16} -1^k \binom{16}{k} e^{20\cdot\mathrm{SNR}_{(v,w)}\cdot\left(\frac{1}{k}-1\right)}. \tag{3.3}$$

For calculating the packet error ratio (PER) it is assumed for simplicity that bit errors in the preamble do not prevent the reception. Thus, together with SFD and PHY header, $L_{\mathrm{PSDU}} + 2$ octets have to be received without bit errors, so the PER can be calculated as

$$\mathrm{PER}_{L_{\mathrm{PSDU}},(v,w)} = 1 - \left(1 - \mathrm{BER}_{(v,w)}\right)^{8\cdot(L_{\mathrm{PSDU}}+2)}. \tag{3.4}$$

Fig. 3.2b depicts the packet error ratio over the margin of reception power and noise power. According to these calculations, the sensitivity of the ATmega256RFR2 corresponds to a P_N of -100.44 dBm. With interfering transmissions, the noise power P_N is replaced by the sum of the noise power and the signal powers of the simultaneously transmitted signals. The result of the calculation corresponding to (3.2) is denoted as signal-to-interference-plus-noise ratio (SINR).

3.2 Medium Access Control

As already covered in Sect. 2.2.1, the MAC is an integral component that largely influences the performance of a wireless system. The MACs specified by IEEE 802.15.4 relevant for this thesis are presented in the following. CSMA/CA is easy to use and does not require any management, but suffers from collisions due to the hidden node problem. DSME and TSCH provide a method for time-slotted communication and can use different frequency channels for parallel communication and also to bypass disturbed channels. In channel hopping mode, the used channel cycles over a predefined set of channels for every transmission. In channel adaption (only for DSME), the channel is constant for a given slot assignment for a link, but it is possible to change the assignment if a disturbed channel is noticed.

Several studies compare the different enhancements of the IEEE 802.15.4 standard, including [GBA16] and [JAG16]. The latter identifies a slight advantage of DSME compared to TSCH in terms of energy consumption. In [APM15] TSCH and DSME are compared under a constructed industrial scenario including multiple monitoring and control applications. The main metric is the end-to-end delay. For small network sizes DSME suffers from the fixed structure, but for large networks (50 nodes), the delay is better than for TSCH.

3.2.1 CSMA/CA

The basic procedure of CSMA/CA is depicted in Fig. 3.3. Before sending a packet, a backoff period in terms of backoff units of 320 μs is randomly selected from the interval $[0, 2^{m_0}]$ where m_0 (also denoted as *macMinBe*) is the initial backoff exponent. After this backoff period is over, the device senses the channel for an ongoing transmission (the so called Clear Channel Assessment (CCA)). If the channel is not sensed free, a new backoff period is selected. For every new backoff, the backoff exponent is increased up the maximum of m_b (*macMaxBe*) in the range 3 to 8. This is repeated until the message is transmitted or the maximum number of backoffs m is reached ($0 \leq m \leq 5$). Note that, according to the standard, m does not include the first unconditional backoff, so the actual number of backoffs is $m + 1$.

When the channel is sensed free, the transmission is started. If the transmission is successfully received by the receiver, an acknowledgment will be sent back unless none is requested as it is usually the case for broadcast messages. If the acknowledgment is not received by the original transmitter within a given time (*macAckWaitDuration*), the whole procedure is retried n (*macMaxFrameRetries*) times ($0 \leq n \leq 7$).

In contrast to IEEE 802.11, no RTS/CTS mechanism is provided that keeps the channel free from disturbing transmissions by a message handshake before the actual transmission.

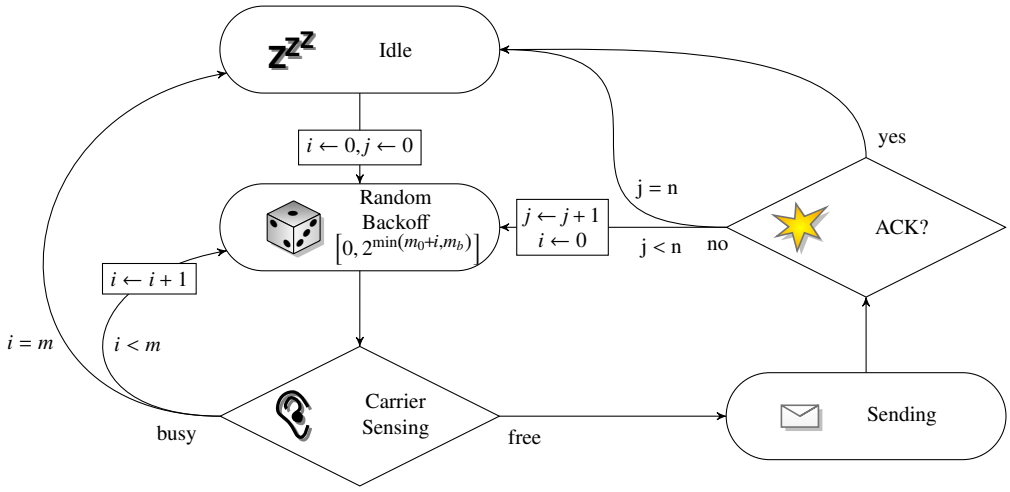


Figure 3.3: Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA).

This is sometimes seen as a weakness of IEEE 802.15.4 [PMB09], but it was shown [ZL07] that RTS/CTS is only really useful for high traffic load and considered too expensive for low traffic load. It is also important to note that the RTS/CTS messages are not protected against collisions themselves. If only short packets are sent anyway as typical for IoT applications, the probability of collisions can thus not be reduced. The reason for the effectiveness in IEEE 802.11 is the large difference in message size between the RTS/CTS messages and the payload packets and is therefore not transferable to IEEE 802.15.4.

3.2.2 TSCH and 6TiSCH

In contrast to the two TSMP-based standards WirelessHART and ISA100.11a that provide full stack solution, Time-Slotted Channel Hopping (TSCH) is an extension to the IEEE 802.15.4 standard that focuses on the slotted medium access only and relies on higher layers for configuration.

TSCH

Apart from that, TSCH is quite similar to the medium access in WirelessHART and ISA100.11a. It separates time into time slots of *macTsTimeslotLength* microseconds with the default of 10000 for the 2.4 GHz band corresponding to the 10 ms also used in WirelessHART. The slots are grouped into slotframes that repeat in time. It is also possible to define multiple slotframes with different length per network or even on a single device and transmissions are scheduled in the slotframe with the lowest *macSlotframeHandle* that currently has an active slot for a given device.

A device synchronizes its time with all neighbors configured as its time sources. The time synchronization is performed during the transmission or reception of a packet. The receiver enables its transmitter *macTsRxWait*/2 microseconds before the expected reception of a packet. The default of *macTsRxWait* is 2200. For perfect synchronization it is expected

that the frame starts after an interval of *macTsRxWait*/2 microseconds. For frame-based synchronization, the receiver adjusts its own clock according to the error between the expected and the measured begin of the frame. For acknowledgment-based synchronization, it records the difference and sends it back to the transmitter so that it can correct its clock accordingly. Also if no frame is transmitted within a configurable keep-alive period, a frame with no payload shall be sent to the time source for acknowledgment-based time synchronization.

Apart from these methods, TSCH mainly standardizes primitives for configuring the *macSlotframeTable* and *macLinkTable* by an upper layer. While the *macSlotframeTable* only consists of a list of *macSlotframeHandles* and their associated *macSlotframeSize*, a link in the *macLinkTable* consists of a larger set of parameters. This mainly includes the type of a link that can configured to be used for reception and/or transmission, shared with other devices, used for time synchronization, for high priority traffic or advertising. It also specifies, if applicable, the address of the neighbor device, the slotframe and position within the slotframe, as well as the offset for channel hopping.

6TiSCH

Configuring the *macSloframeTable* and *macLinkTable* is the responsibility of an upper layer. The Internet Engineering Task Force (IETF) has built the working group 6TiSCH to fill this gap and to connect TSCH to the 6LoWPAN stack. In [WPG15] the problems and goals of 6TiSCH are presented. Besides tasks such as network formation and security, it identifies slot scheduling as important building block. Several scheduling techniques were proposed in the context of 6TiSCH that will be presented along with other methods in Sect. 3.3.

In [Thu18], the architecture of 6TiSCH is presented. Its central component is the 6top sublayer and its 6P protocol for distributed scheduling [WVW18]. The latter defines two methods (2-step and 3-step 6P transaction) that can be used by two nodes in the network to agree on common time slots. In contrast to the slot allocation handshake of DSME as presented in Sect. 4.2.2, no precautions are taken to avoid slot collisions during the slot setup, though first research in this direction exists [ML16, FBA17].

In 6TiSCH the combination of a slot (time) offset and a channel offset is denoted as cell. In the following, the term slot is used for consistency with the other parts of the thesis.

3.2.3 DSME

The Deterministic and Synchronous Multi-Channel Extension (DSME) is based on the existing predefined, yet adaptable, superframe structure of the IEEE 802.15.4 standard. Note that TSCH defines its own slotframe structure. DSME relies solely on beacons for time synchronization, but extends the existing beacon mechanism to multi-hop networks. In fact, multi-hop networks were already possible with the existing beacon-enabled mode of IEEE 802.15.4, but this required proprietary methods for beacon scheduling as in GreenNet [VRV15] to avoid collisions. Furthermore, in GreenNet, the contention-free period is not used. Both features are enabled by defining a set of messages and distributed procedures for beacon and slot management and thus already includes many services provided by 6TiSCH for TSCH.

Superframe Structure

Fig. 3.4 shows an exemplary superframe structure. Every superframe consists of 16 slots (*aNumSuperframeSlots*) of duration

$$T_s = aBaseSlotDuration \cdot 2^{SO}, \tag{3.5}$$

where *aBaseSlotDuration* is 60 symbols of 16 μs each and SO is the *macSuperframeOrder* between 0 and 15, resulting in a superframe duration of

$$SD = aBaseSuperframeDuration \cdot 2^{SO}, \tag{3.6}$$

where *aBaseSuperframeDuration* = 960 symbols.

The first slot of a superframe is a beacon slot, followed by the contention access period (CAP) with 8 slots and a contention free period (CFP) with 7 time slots for allocating guaranteed time slots (GTS). During the CAP, contention based channel access is used. The beacon and the messages in the CAP are sent on a predefined, common channel.

A GTS is dedicated for communication between two nodes. The channel to be used can be configured for every GTS. Either as a fixed channel (channel-adaption mode) or to one out of a set of channel hopping sequences (channel-hopping mode), similar to TSCH. It is possible to assign the same time slot to several links if the links are spatially separated or a different channel or channel-hopping sequence is used. One GTS is dedicated to transmission for one device and to reception for the other. So the receiver will switch to reception mode shortly before the slot begins, the transmitter will transmit its frame within the slot and wait for an acknowledgment by the receiver. Afterwards, both devices can disable the transceiver for power saving.

Every GTS is repeated with an interval of one multi-superframe. A multi-superframe consists of 2^{MO-SO} superframes with the multi-superframe order MO. The duration of a multi-superframe is thus given by

$$T_m = T_s \cdot aNumSuperframeSlots \cdot 2^{MO-SO}. \tag{3.7}$$

For getting more GTSs per time, a CAP reduction mode is defined where only the first superframe of a multi-superframe has a CAP and all other superframes have an extended CFP instead. Finally, the beacon interval defines the repetition interval of a beacon. With the beacon order BO the number of multi-superframes in a beacon interval is 2^{BO-MO}. The structure can be optimized to the application within the ranges given in Table 3.1.

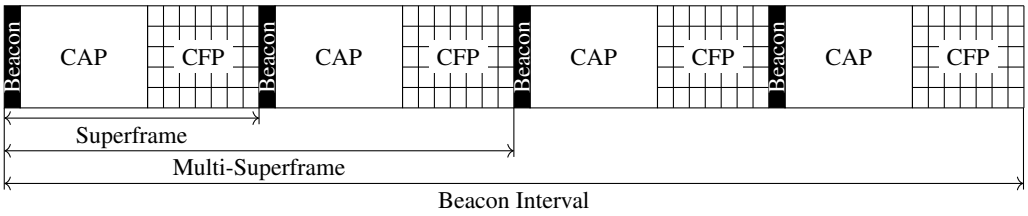


Figure 3.4: DSME Superframe structure for SO = 3, MO = 4 and BO = 5.

Parameter	Range	Default
macSuperframeOrder (SO)	$0-15$	15
macMultisuperframeOrder (MO)	$0-22$	15
aBaseSlotDuration	60 symbols	
aNumSuperframeSlots	16	
macResponseWaitTime (W_r)	$2-64$	32
macDsmeGTSExpirationTime (M_x)	$0-255$	7

Table 3.1: Important parameters of DSME.

In contrast to TSCH, where slotframes of arbitrary length can be constructed, the number of slots per multi-superframe is restricted to certain values given by

$$7 \cdot 2^{\mathrm{MO}-\mathrm{SO}} \tag{3.8}$$

or when using the CAP reduction mode

$$7 + 15 \cdot \left(2^{\mathrm{MO}-\mathrm{SO}} - 1\right) \tag{3.9}$$

GTS are available depending on the integer MO and SO.

As described by (3.5), the SO directly influences the slot length. To transmit full-length IEEE 802.15.4 packets with a PSDU of 127 Bytes and the acknowledgment within one slot, SO must be at least 3, resulting in a slot duration of $T_s = 7.68\,\mathrm{ms}$. If nothing else is specified, this is the value used throughout the thesis. If only shorter messages are transmitted, a smaller SO can increase the throughput in terms of packets per time. A larger SO can on the other hand be used to decrease the power consumption for networks with low traffic by turning off the transceiver after sending or receiving a packet. Another option to utilize long slots might be to send multiple packets per slot, but in principle, sending two packets per slot for a given superframe order s provides the same throughput as sending one packet each in two slots for superframe order $s-1$. The latter case requires more management traffic, but the flexibility is much higher, since in other parts of the network the two slots can be utilized by two different nodes. Thus, it is assumed throughout the thesis that only one packet is sent per slot. The additional benefit is that the receiver can turn off after receiving one packet.

The multi-superframe order MO is mainly responsible for setting the number of distinct slots. Since the multi-superframe is repeated in time, this number determines the granularity of the slot allocation. For example, with one superframe per multi-superframe, only 7 distinct slots can be assigned. No more than 7 nodes per neighborhood can allocate slots and they can only coarsely adapt the number of allocated slots to the traffic. With more slots, more nodes can be handled and the granularity is finer. On the other hand, MO should not be chosen too high for networks with a lot of traffic, since it increases the slot management overhead if a lot of slots have to be allocated for a single link.

As a good compromise it is convenient to choose SO and MO so that T_m is in the order of the average transmission interval. For example, with SO $= 3$ and the default as given

by the standard of MO = 15, a node can transmit a packet over one allocated GTS every $T_m \approx 8$ minutes. More slots are required to allow for the allocation of multiple slots for links with higher traffic, when heterogeneous transmission intervals should be supported.

Furthermore, the multi-superframe order MO determines the overall share of the CAP if CAP reduction is applied. With more superframes per multi-superframe and enabled CAP reduction, less time is available for management traffic, but more time is available for transmitting payload data via guaranteed time slots.

Time Synchronization

The time synchronization relies on beacons just like in IEEE 802.15.4 star networks. What is different is that multiple nodes can send beacons. This is necessary for multi-hop networks to distribute a common notion of time throughout the network. Before joining the network, a node scans for beacons and associates via the sender of a beacon. After that, it tracks the beacon of its time synchronization parent, that is not necessarily equal to the parent in a routing tree, to compensate its clock drift. It is important to not adapt the own clock to beacons of other devices, because this can easily lead to cyclic groups of nodes that drift away from the global notion of time.

A node can reserve a beacon slot to send an own beacon within its neighborhood by selecting a free beacon slot and sending a beacon allocation notification command as broadcast. A receiving node will either mark this beacon slot as occupied and will include this information in the enhanced beacons sent out by it or, if a conflict is detected, it sends a beacon collision notification to reject the selection of this beacon slot.

Since every beacon is repeated once within the beacon interval, a high BO will lead to a longer time until a node receives a beacon. Traditionally, according to the IEEE 802.15.4 standard, scanning has to be done on all available channels, so scanning will take very long to collect all beacons. Even after the association, a very large beacon interval can lead to problems if the clock drift of the nodes is too large. On the other hand, the BO determines the number of distinct beacon slots, so with a small BO there might not be enough beacon slots available for all nodes to be coordinator, even considering spatial reuse.

Slot Allocation Handshake

In DSME, a three-way slot allocation handshake is used for building up the schedule. Two arbitrary nodes in the network can perform this handshake to allocate or deallocate slots without the need of a central entity. A slot is always used in one direction only, but can be requested as a reception or transmission slot. The approach used by DSME has the unique feature of overhearing, not applied by the 6P transactions of 6TiSCH. The second and third message of the three-way handshake is transmitted as link-layer broadcast and can thus also be received by all neighbors of the respective nodes. This makes it possible to take this information about existing slot assignments into account during later slot allocations. This, in principle, leads to conflict-free schedules and can thus provide a maximum of reliability, while at the same time spatial reuse is still possible, maintaining the performance. The details of this procedure are analyzed in Sect. 4.2.

3.3 Existing Solutions for Slot Scheduling

This section presents existing research in the area of slot scheduling for multi-hop TDMA networks for the application to the standards presented before. This section is restricted to techniques directly applicable to one of the standards presented above, but much earlier research in this area exists, too. An example is TRAMA [ROG06] that uses random access slots for the exchange of two-hop neighborhood information and utilizes this information to calculate and announce slots to be used for packets currently in the queue. A unique node identifier together with the current time is used to establish priorities amongst the neighbors to avoid packet collisions. Also PEDAMACS [EV06] is a promising approach that uses a single high-powered node for time synchronization and slot scheduling for achieving high energy-efficiency of the other nodes in the network. In [TW07], schedules are built up by performing a distributed depth-first search in the network and the paper compares different types of schedules depending on the relation of the number of slots and nodes. The authors conclude that for data intensive networks it is advantageous to assign multiple exclusive slots to nodes to avoid collisions and for handling different traffic loads even if this leads to long slotframes.

3.3.1 Minimal 6TiSCH Configuration

The slot schedule proposed in the minimal 6TiSCH configuration [VPW17] (not to be confused with the Minimal Scheduling Function presented in Sect. 3.3.8) only consists of a single network-wide active slot that can be used by all devices as transmission and reception slot on a shared basis. Additionally, an arbitrary number of idle slots can be appended for energy-reduction and the parallel usage of other slotframe structures. It builds the baseline for other more complex scheduling functions for 6TiSCH and also specifies basic functionality such as the transmission of enhanced beacons to allow for joining the network, selection of time sources and security functionality.

3.3.2 Centralized Calculation of Schedules for Wireless Control

One application of WirelessHART is the usage in industrial control loops. In [FEI09] an approach is presented that generates a schedule including both control tasks and message transmissions in a WirelessHART network. It uses multiple channels for parallel transmission and multiple paths in order to provide redundant communication in case of packet loss. A heuristic off-line algorithm is applied that takes the different scan rates of multiple control loops into account while minimizing the delay. A simulation with TrueTime, based on Matlab/Simulink, shows the benefit of using redundant paths in case of message loss and the impact on the behavior of the controller.

Motivated by a deployment of a wireless network in a oil refinery, [PSB14] proposes the centralized calculation of a schedule based on the expected length of bursts of consecutive failed and successful transmissions. For this, these metrics are measured for every pair of nodes in the network and collected at a central entity. By iterating over all possible topologies generated by changes in the transmission power and the selection of links, the valid schedule with the smallest energy signature is selected. A heuristic is suggested to

reduce the computational complexity and a deployment in the oil refinery as well as two other testbeds show the improvements in terms of reliability, energy efficiency and reduction of interference with other networks achieved with the proposed technique compared to a handpicked topology.

3.3.3 Aloha- and Reservation-Based Scheduling

In [TWP11], a decentralized scheduling approach for TSCH is presented. It targets highly mobile networks and aims at full connectivity, that is the availability of a transmission and reception slot towards every neighbor. The paper proposes the aloha-based scheduling where nodes occasionally, with a chance of 10%, send advertisements in currently free slots as an offer for other nodes to establish a connection in this slot. These can then accept the offer and schedule data transmissions in this slot to the given node until the link is lost.

Several improvements to this technique are proposed, subsumed under the term reservation-based scheduling, including the usage of a dedicated advertisement slot, the dissemination of the two-hop neighborhood and the pairwise allocation of slots. Experiments are presented based on a simulation as well as a real-world application of drifters floating on a river. The results show that time slots can be scheduled for up to about 80% of the available links, but the ratio decreases significantly when increasing the density.

3.3.4 Centralized Scheduling for TSCH

A centralized calculation of a time slot schedule for TSCH is presented in [PAG13], called Traffic Aware Scheduling Algorithm (TASA). It is based on a combination of a matching and a vertex coloring algorithm and takes the traffic load of the individual links into account. While the method of collecting the information required to set up the schedule and the way for distributing the schedule in the network is out of the scope of the paper, some preliminary analysis of the signaling overhead identifies an average per-node signaling overhead of up to 300 Bytes.

The evaluation by means of a Python-based ad-hoc simulator reveals that the calculated schedule achieves a duty cycle very near ($> 97\%$) to the calculated optimum. Due to the reduced energy-consumption, this leads to a theoretical increase in node lifetime from 5 days to about 3 months. A short application of the analytical model for CSMA/CA presented also in [MPF12] demonstrates the high packet loss obtained by CSMA/CA and the need for collision-free slot schedules. Retransmission are, however, not considered in this analysis neglecting the fact that they are an important factor in CSMA/CA to mitigate the hidden node problem at least partially.

While TASA implicitly minimizes the energy consumption by only allocating the required number of slots for every node, [OGP17] explicitly targets minimum energy consumption by a detailed analysis of the energy-consumption and the formulation of an energy-efficiency maximization problem as nonlinear integer programming. For enabling efficient calculation, two heuristic schedulers are developed and evaluated.

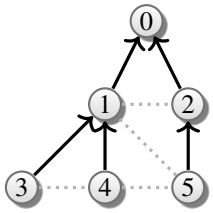
3.3.5 Hybrid Scheduling on Trees

Since the routing in many wireless sensor network applications is based on the construction of a routing tree, it can be useful to reuse this tree for schedule construction. In this thesis it is denoted as hybrid scheduling, in contrast to centralized and decentralized scheduling, because getting required information such as the depth in the tree requires communication over multiple hops even if it can be (partially) outsourced to the routing protocol such as RPL.

The authors of [MVV13] noticed the parallels between the problem of resource allocation in TSCH and wavelength switched optical networks (WSONs). They therefore propose to apply Generalized Multi-Protocol Label Switching (GMPLS) and the Resource-Reservation Protocol - Traffic Engineering (RSVP-TE) that are common in WSONs to slot scheduling in TSCH networks and explain how the end-to-end signaling can be done efficiently.

In order to fulfill the Quality of Service (QoS) constraints, they propose the Completely Fair Distributed Scheduler (CFDS). By interleaved assignment of time slots depending on the level in the routing tree, slots can be assigned without the need of a centralized off-line calculation of the schedule. Fig. 3.5 gives an example for a schedule generated by the basic mode, assuming slot 0 is reserved for control messages. As shown, nodes might transmit at the same time, potentially leading to collisions (node 3 and 5 in slot 1). However, the authors propose a method for selecting different channels to allow for parallel transmission. While this reduces the probability of collisions, they are not completely prevented by the design of the protocol. The paper therefore proposes to switch the channel if too many lost packets are noticed. Furthermore, an extended mode is proposed to better utilize the available time slots. Another proposal for a scheduling function that uses RSVP-TE is the SF1 draft of 6TiSCH [ALZ18], but it does not provide an equally sophisticated scheduling algorithm as CFDS.

In DeTAS [AVP15] a similar concept to the CFDS is used that also aligns transmission and reception slots in alternating positions and uses different channels for avoiding collisions. Instead of RSVP-TE messages, it uses MAC command frames for signalization. DeTAS is implemented in OpenWSN and evaluated in two testbeds with predefined tree routing. With DIS_TSCH [HWW17] nodes schedule their slots only based on the position in the routing



Slot / Node	0	1	2	3	4	5	6	7	8	9	10
0		Idle	RX	Idle	RX	Idle	RX	Idle	RX	Idle	RX
1		RX	TX	RX	Idle	Idle	TX	Idle	Idle	Idle	TX
2		RX	Idle	Idle	TX	Idle	Idle	Idle	TX	Idle	Idle
3		TX	Idle	Idle	Idle	Idle	Idle	Idle	Idle	Idle	Idle
4		Idle	Idle	TX	Idle	Idle	Idle	Idle	Idle	Idle	Idle
5		TX	Idle	Idle	Idle	Idle	Idle	Idle	Idle	Idle	Idle

Figure 3.5: Schedule generated by the basic mode of CFDS.

tree and no additional signaling is required. This approach is optimized for fully-balanced trees with constant degree.

Another promising distributed algorithm to generate a conflict-free schedule is Wave, presented in [SML16] and extensively analyzed in [SML15]. The schedule is constructed in multiple waves until all traffic demands are handled. It has many desirable properties such as being inherently conflict-free even with additional unused links, the support of a sink with multiple radio interfaces and the (limited) possibility to adapt to changes in the traffic demand without recalculation of the schedule.

3.3.6 Autonomous Slot Scheduling with Orchestra

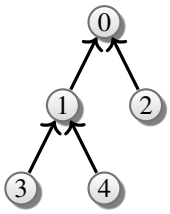
Orchestra [DNL15] is an approach for autonomous scheduling of time slots. Slots are assigned to nodes based on their IDs and thus, besides the neighbor discovery performed by RPL, no explicit messages are required for scheduling. There are four different types of slots:

Common Shared Slots (CS) corresponding to the single slot in the minimal 6TiSCH configuration (cf. 3.3.1).

Receiver-Based Shared Slots (RBS) as illustrated in Fig. 3.6 for a data-collection scenario where each node gets a reception slot based on (a hash of) its ID. Slot 0, again, is a common shared slot in the example. The nodes know about the position of the slot of their neighbors by their ID and will thus transmit in this slot. This transmission is contention-based since collisions are obvious in the figure, but the probability is reduced since fewer nodes try to transmit in a given slot. It is also possible to reduce the slotframe length so that several nodes share a reception slot.

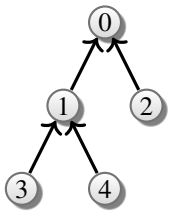
Sender-Based Shared Slots (SBS) where every node gets a transmission slot based on (a hash of) its ID. They are shared unless the slotframe is long enough to assign a dedicated slot to every node. In that case, they become Sender-Based Dedicated Slots.

Sender-Based Dedicated Slots (SBD) are unique for every node, thus no transmissions collide. It requires a slotframe that has as least as many slots as nodes in the network. This case is illustrated in Fig. 3.7. A node has to enable its receiver when a neighbor might transmit in a given slot.



Slot / Node	0	1	2	3	4	5
0		RX	Idle	Idle	Idle	Idle
1		TX	RX	Idle	Idle	Idle
2		TX	Idle	RX	Idle	Idle
3		Idle	TX	Idle	RX	Idle
4		Idle	TX	Idle	Idle	RX

Figure 3.6: Receiver-Based Shared Slots (RBS) with a Common Shared Slot at 0.



Slot / Node	0	1	2	3	4	5
0		TX	RX	RX	Idle	Idle
1		Idle	TX	Idle	RX	RX
2		Idle	Idle	TX	Idle	Idle
3		Idle	Idle	Idle	TX	Idle
4		Idle	Idle	Idle	Idle	TX

Figure 3.7: Sender-Based Dedicated Slots (SBD) with a Common Shared Slot at 0.

The paper also suggests to stack several slotframes with different lengths for different needs. For example, the use of a slotframe with a long period for beacons, a shorter one with common shared slots for RPL traffic and one with slots dedicated for application traffic. Several analytical considerations are made to calculate for example bounds for the duty cycle. An extensive evaluation is provided with testbeds of 98 and 25 nodes showing amongst others the higher packet delivery ratio achieved with Orchestra compared to an Always-On CSMA/CA and the Contiki MAC. A comparison with the offline scheduler proposed in [PSB14] demonstrates the advantage of a schedule that adapts to changes in the topology. Orchestra was the first non-trivial (cf. Sect. 3.3.1) scheduling function implemented for 6TiSCH in the Contiki operating system [DGV04].

3.3.7 Traffic-Aware Scheduling Based on 6top Transactions

For adapting to volatile traffic loads, [PWW16] proposes a technique for distributed slot scheduling based on the services provided by 6top. In particular, the number of required slots between two nodes is controlled, based on the amount of measured traffic. The 6top sublayer will then perform transactions between the nodes to adapt to the requirements. As soon as the estimated bandwidth exceeds a certain threshold, a configurable number of slots is overprovisioned. Deallocations will take place for a lower threshold. By this technique, excess management traffic is avoided if the estimated traffic is volatile.

A simulation-based evaluation is provided showing how the approach is able to adapt the number of slots to the estimated traffic. In the evaluation, the traffic load is estimated from the number of incoming packets per time smoothed by an auto-regressive filter. It is demonstrated that overprovisioning reduces the management traffic, but at the same time increases the probability of collisions.

In [DCV16] the 6top transactions are used to schedule slots based on the number of packets in the queue. A PID controller is developed for this task and a simulation-based evaluation shows how this approach can successfully handle bursty traffic. The number of packets in the queue is also used in [TP16] together with the estimated number of required transmissions per packet (ETX) to form a schedule. In order to support multiple parallel applications in a network this paper proposes to use isolated and convergent tracks and also the reduction of end-to-end latency by selecting transmission slots close to the reception slots for a given track.

A similar approach, the Low Latency Scheduling Function (LLSF), is proposed in [CWW16] for minimizing the end-to-end latency in scheduling functions that only specify the number but not the particular arrangement of slots. When a new transmission slot shall be allocated, the earliest available slot following the reception slot with the largest gap to the previous reception slot is selected. Correspondingly, when deallocating the transmission slot with the largest gap to the previous transmission slot is deallocated. In a simulation of a linear topology of 6 nodes, this approach is able to reduce the latency by a factor of about 3 to 6 depending on the slotframe length. DeBraS [ML16] explicitly targets dense networks where colliding slots are a major problem and like LLSF it can be seen as an addition to existing scheduling functions. With DeBraS, nodes regularly broadcast their own schedule so nodes will avoid scheduling slots known to be used in the neighborhood. The evaluation shows an improvement in terms of throughput and latency, but the need to broadcast the schedule increases the energy-consumption and in dense networks not all required information can be distributed timely.

3.3.8 6TiSCH Minimal Scheduling Function (MSF)

The 6TiSCH Minimal Scheduling Function (MSF) [CVV18] (not to be confused with the minimal 6TiSCH configuration presented in Sect. 3.3.1) is a recent proposal for a versatile scheduling function for 6TiSCH. MSF is based on the previous drafts SF0 [DGP17], SFX [DGP18] and ASF [DVW18]. While SF0 and SFX originate from the on-the-fly bandwidth reservation proposal (Sect. 3.3.7), ASF originates from Orchestra (Sect. 3.3.6).

The Minimal Scheduling Function builds upon the minimal 6TiSCH configuration and uses the shared slot for enhanced beacons and DODAG Information Objects (DIOs) for RPL. Second, every node maintains a unicast slot to each of its neighbors and it should use polling for ensuring that the neighbors are still within reach. Analogous to Orchestra RBS, the slot is computed from the hash of the node's address. This also means that this slot is not necessarily unique and thus is considered as shared slot. Third, additional slots are scheduled depending on the current traffic load. For this, every node regularly counts the number of unicast slots towards its neighbor as well as those that were used. If the ratio of used and available slots gets above/below a configurable threshold, a new slot will be requested/deallocated.

3.3.9 Discussion

In this section, popular existing approaches for slot scheduling are introduced. A more in-depth review is for example provided in [HGT17]. In the following, the presented approaches are compared and analyzed for their adequacy for very large wireless networks.

With the collection of global network knowledge at a single entity, centralized scheduling algorithms can generate very efficient and even, under the respective objectives, optimal schedules. The latter, however, usually requires high computational effort so heuristics are required. Further disadvantages are the high overhead for collection of information and dissemination of the schedule and the difficulty to react to changes of the conditions such as changing link conditions or changes in the traffic demand. Often, a complete recalculation of the schedule is required, multiplying the overhead. Therefore, they can

excel in small networks with strict requirements, such as high-rate control loops, but the named disadvantages lead to a bad scalability making them inappropriate for the applications considered in this thesis.

Hybrid scheduling algorithms improve the scalability since the overhead can be reduced. The global network topology is considered nevertheless and by this end-to-end QoS can be achieved by adequate alignment of a chain of slots. Still, a single broken link might still influence the schedule in large parts of the network leading to extensive recalculations, making them unsuitable for very large networks where volatility is inevitable.

Thus, we are left with scheduling algorithms that make their decisions only on the basis of local information. This may not lead to globally optimal schedules, but these approaches are especially useful for large or volatile networks. Orchestra falls in this category and since it requires no dedicated management traffic it is an appropriate schedule for many applications. However, the main disadvantage is its inability to handle different traffic loads efficiently since all nodes get the same number of slots. For example in a data collection scenario, the inner nodes have to handle the aggregated traffic of all nodes in their sub tree, so they would profit from using more slots than leafs. This negative effect is also known as funneling effect [WEC05] in networks with only one (or few) sinks. Thus, it is advised to take for example the estimated traffic over a link [PWW16] or the current queue level [DCV16] into account.

For the local agreement on slots, transactions such as the 6P transactions of 6TiSCH and the DSME slot allocation handshake are used. While the first does not avoid slot collisions, the second one does as outlined in Sect. 4.2. A recent achievement for distributed slot scheduling is the IETF draft of the MSF combining multiple promising approaches. However, it does not prevent slot collisions inherently, but tries to mitigate this by detecting and avoiding slots with a low packet delivery ratio. Furthermore, its possibility to adapt to the traffic is only given towards the parent and it is expected to come with a high overhead of management traffic for bursty or volatile traffic since it includes no smoothing of the measured traffic demand.

3.4 Multi-Hop Routing

In multi-hop networks, an algorithm is required for routing data packets from the source to the sink. A common traffic pattern in wireless multi-hop networks is sending to or from one or a few nodes that serve as gateways to other networks such as the Internet. Therefore, tree-based approaches or more generally ones that build upon a destination oriented directed acyclic graph (DODAG) are very common. During the effort of the Internet Engineering Task Force (IETF) to standardize the communication in such networks and make them compatible to the Internet, consolidated under the term IPv6 over Low power Wireless Personal Area Network (6LoWPAN), the Routing Protocol for Low power and Lossy Networks (RPL) was developed and is very popular today. It is a distance-vector routing protocol that builds up a DODAG by periodically sending out DODAG Information Objects (DIO) that, amongst others, includes the rank in the DODAG. This allows the nodes to select the best routing paths.

While RPL is useful for many applications, the need to build up and maintain the DODAG is especially cumbersome in very large networks. A different approach with much better scalability is the usage of geographic positions for routing decisions. This information can be determined by a GPS module or predefined in a static deployment. While this comes with an additional effort, the availability of the geographic position allows to directly select the best next hop without the need to build up a global structure such as a DODAG.

GPSR [KK00] is one example for such a routing protocol. In its original version it greedily selects the node closest to the destination. In case of holes in the topology that can lead to a dead end, GPSR switches to perimeter routing to go around the hole. The greedy selection of the next hop can, however, lead to an uneven distribution of traffic within the network. This is demonstrated in [Lüb15] and an alternative is proposed that chooses the node with the closest distance to the straight line between the original sender and the destination that still advances the progress towards the destination, leading to a much better distribution of the traffic and by this significantly increases the scalability of GPSR.

Further routing approaches for large-scale and dense networks are proposed and analyzed in detail in [Unt14]. They are investigated in the context of a solar tower power plant application, too. One major achievement is the development of a technique for distributed construction of connected dominating sets to determine cluster heads in dense wireless networks.

3.5 Conclusion

This chapter has presented a roundup of the foundations of IEEE 802.15.4 relevant for this thesis and existing slot scheduling solutions developed in this context. It identifies the medium access layer as an important building block of a reliable wireless network, but also indicates the significance of the other layers. Especially in the context of 6TiSCH many promising techniques were developed. While they are certainly adequate for many applications, they are still prone to slot collisions and heavily rely on 6LoWPAN and RPL. However, in some applications such as the presented solar tower power plant there is a need to reduce the overhead as much as possible. Furthermore, specialized routing algorithms such as geographic routing can be required. This calls for a more flexible approach as developed in the following on the basis of DSME.

Chapter 4

Constructing a Reliable Medium Access Layer using DSME

For building truly scalable and reliable wireless multi-hop networks based on time slots, the following features are essential for constructing a schedule that are only partly addressed by the existing methods presented before.

- Collision-free slot scheduling since even low collision probabilities will multiply up in a large network.
- Fully decentralized construction of the schedule to avoid excessive recalculations that get increasingly frequent with a higher number of links.
- Traffic-awareness and efficient usage of the available resources to mitigate the funneling effect that results in large amounts of traffic even for low generation rates in large networks.

This motivates the consideration of IEEE 802.15.4 DSME since it already provides a slot allocation handshake that promises collision-free slot scheduling. This promise is formally analyzed in this chapter and builds the basis for the proposed traffic-aware and prediction-based slot scheduling.

While there are several attempts to implement DSME for a simulator, for example for Cooja in [VBP17], for QualNet in [LC16] and for OPNET in [CBR14], there exists, to the best of my knowledge, no publicly available implementation of DSME that can be executed in a simulator as well as on hardware such as wireless sensor nodes. Even more, [GBA16] explicitly identifies the lack of a complete implementation as the limiting factor for the application of DSME in real environments. Thus, openDSME is presented as implementation of DSME that is useful both as research tool in a simulator as well as for real-world applications.

4.1 Related Work

A simple formula for the throughput of DSME networks is given in [JL12] where DSME with and without CAP reduction is compared to slotted CSMA/CA in a star and square

grid configuration. For DSME, a higher throughput and less energy consumption can be achieved than for CSMA/CA, especially for large networks. Still, the need to enable the receiver during the CAP phase can increase the energy consumption if it is scheduled too often. Thus, [CBR14] proposes to not enable the receiver if the Frame Pending field in the enhanced beacon frame indicates no pending packets. Of course, is only possible if the coordinator is the sole initiator of traffic in the CAP and all other transmissions use the CFP.

In [LJ12] the performance of DSME is analyzed under IEEE 802.11 interference. Since one IEEE 802.11b channel covers four IEEE 802.15.4 channels, thus a quarter of the available IEEE 802.15.4 channels, the performance drops to only 75% when using channel hopping, while for CSMA/CA is not able to circumvent the interference autonomously. A method for handling mobility in wireless sensor networks using DSME is proposed in [LC16]. It requires to distribute scheduling information, such as queue lengths, in the network.

For dense networks, relying on beacon collision notifications can still lead to beacon collisions as shown in [HN14]. It is proposed to extend DSME with permission notifications, denoted as E-DSME. Other proposals to improve the network formation include [VBP17] by means of parameterization of the CSMA/CA during the CAP and by enabling reception during backoff that was already identified as important feature in [WT15] for conventional CSMA/CA. Furthermore, [LLS13] proposes to reduce the time taken for association, denoted as enhanced fast association. The main idea is to spread the association requests over a longer time period to reduce collisions in the CAP. Similarly, [SPW17] proposes to reduce collisions by distributing the channel accesses of nodes evenly over the CAPs per multi-superframe.

4.2 Formal Analysis of the DSME Slot Allocation Handshake*

The three-way handshake of DSME promises the avoidance of collisions between guaranteed time slots within a given neighborhood. In this section, the promise is analyzed in detail under consideration of communication and node failures. Since the schedule is not managed by a central entity, inconsistencies of the slot schedule are not impossible per se. In particular, the slot allocation itself is conducted via CSMA/CA, so it possibly prone to message collisions and race-conditions. Since inconsistencies in the slot schedule might thwart the original requirement of reliable communication, it is crucial to investigate the existence of inconsistencies and to reduce their influence on the performance of the overall network.

First, the slot allocation handshake is presented and the selection of appropriate parameters for the CAP is introduced in order to keep disturbing collisions during the handshake at a minimum. However, even events that occur with a very low probability, for example every few hours, might eventually have severe impacts on the overall performance. Therefore, just reducing the probability is not enough, but definitive statements about the behavior in

*This section is an extension of the previously published "F. Kauer, M. Köstler, T. Lübkert, and V. Turau, *Formal Analysis and Verification of the IEEE 802.15.4 DSME Slot Allocation* in Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), Valletta, Malta, November 2016. https://doi.org/10.1145/2988287.2989148" [KKL16].

the worst possible situation are required. For this, a testbed experiment or a simulation is not sufficient since this situation might occur too rarely to be observed within a reasonable timespan, so a formal verification method is used instead.

4.2.1 Formal Analysis of Distributed Systems

Due to asynchronicity as well as the occurrence of partial failures, distributed systems have a high potential for failures [McC16]. According to this article, formal verification of distributed systems is tedious and composing provably correct systems to larger systems does not necessarily result in a correct overall system. Still, multiple examples are presented where formal verification greatly helped to build stable systems, including [NRZ15] that is a description of how Amazon Web Services were improved by formal methods.

In order to support formal verification, multiple tools and frameworks have been developed, including generic frameworks such as TLA+ [Lam99] and UPPAAL [LPY97], as well as frameworks that are tailored to the needs of distributed systems such as DS2 [AGS16] and Verdi [WWP15].

Wireless connections are in particular challenging for distributed networks due to the unreliability and unpredictability of the wireless channel. Nonetheless, formal verification can prove conformance to industrial standards even for wireless applications [FWD14]. In wireless mesh networks, formal verification was in particular used for routing protocol analysis such as finding loops in AODV [YGK17] and suggesting boundary conditions for fixing these problems [BGH16].

A more abstract verification of ad hoc networks was conducted in [DSZ12] leading to fundamental statements about the decidability of the coverability problem under node and communication failures. In [BAS06] a formal model of the IEEE 802.11 Point Coordination Function and the Distributed Coordination Function is presented to support applications with real-time requirements. It also includes an elaborate introduction about how to model wireless networks in UPPAAL. Proving adherence to time limits in real-time systems is the motivation for many other formal models, too, such as in [SRR16] where wired bus standards such as TTA are surveyed under the occurrence of faults.

In [Fru11], a formal analysis of the IEEE 802.15.4 CSMA/CA procedure is conducted. Furthermore, the performance of a simple abstract slot allocation method is analyzed using formal methods, while a more sophisticated channel allocation procedure was analyzed in [BBD14]. In [TGD17], the mCRL2 language is used to formally model and verify a cluster-tree formation protocol for TSCH. The analysis identifies weaknesses in the protocol such as situations where no proper topology is built and a bad scalability of the algorithm leading to 27 minutes to setup a network of 500 nodes.

4.2.2 Slot Allocation Handshake

The message exchange for allocating slots as given in the IEEE 802.15.4 standard is depicted in Fig. 4.1.

1. For reserving one or multiple slots for usage on a link from node A to B, node A sends out a **DSME GTS Request** to B, including available slots and the preferred

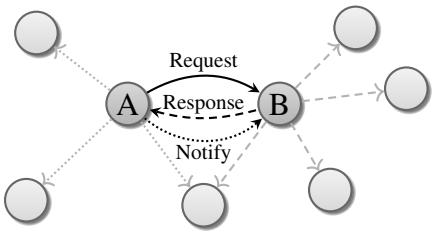


Figure 4.1: GTS allocation handshake.

selection. This takes place during the CAP and since the message is sent as unicast, it will be acknowledged.

2. After node B received the request, it selects an available slot and sends a **DSME GTS Response** (formerly *Reply* in IEEE 802.15.4e) back to node A. By using a link-layer broadcast, all neighbors of B will be informed about the selection, depicted by the dashed arrows. They store this information for their next allocations.

3. Finally, the response is acknowledged by sending a **DSME GTS Notify**. This is also sent as broadcast, indicated by the dotted arrows, because A and B do not necessarily share the same neighbors.

If a slot collision occurs, especially when a response or notify was not correctly received by a node that afterwards initiates an own allocation procedure for the same slot, a **DSME GTS Request** with management type **Duplicate allocation notification** is sent to invalidate the handshake. The GTS handshake is also applied when a slot shall be deallocated, for example if the scheduler requests a reduction in the number of slots or that no successful communication took place in this slot for *macDsmeGtsExpirationTime* multi-superframes in a row. This implies that either no packets were scheduled to be sent or the transmission was not successful, for example due to external interference.

MAC Services and Data Structures

Several MAC services and data structures are defined to manage the handshake. Fig. 4.2 shows a detailed representation of these. In order to send out a request, a MLME-DSME-GTS.request is sent via the MAC Sublayer Management Entity (MLME) interface. It includes a slot allocation bitmap (SAB) that indicates the node's available time slots and channels where all slot-channel combinations used by neighbors are marked as unavailable as well as all channels for the time slots used by A. This is because A is not able to serve two simultaneous slots on different channels. The request also includes a preferred superframe and a slot ID that might optionally be considered during the slot selection procedure at B.

After B received the GTS Request, the MAC layer will generate an indication primitive to be sent to the higher layer. With the information from the SAB, the higher layer chooses a slot that matches both the received and the own SAB. If a matching slot is found, a response is generated that is sent via the MAC layer to the originator. A will inform the upper layer about the response. Furthermore, it issues a notify command with the main purpose of

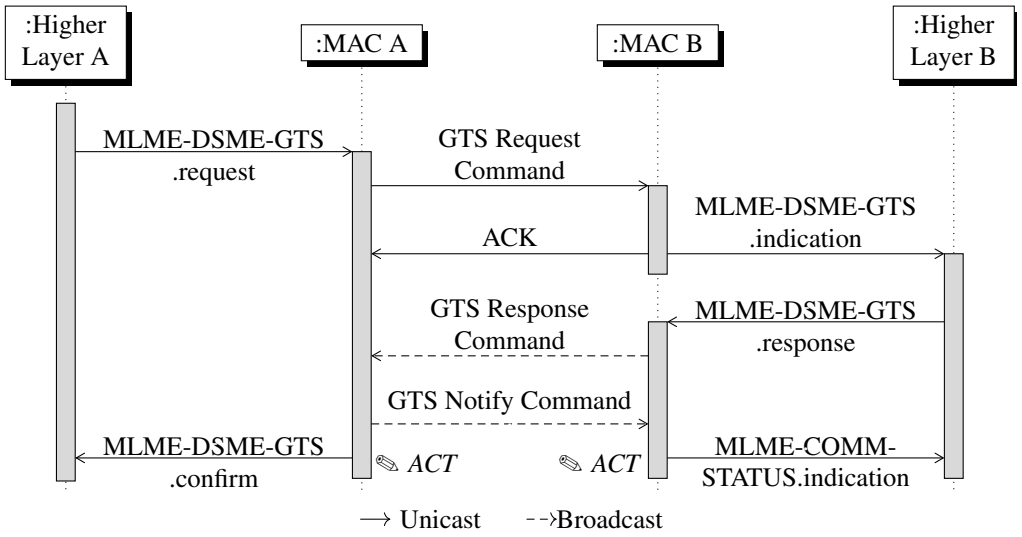


Figure 4.2: Sequence diagram of a successful slot allocation.

notifying the nodes that did not receive the response because they are not in the transmission range of the responder. Moreover, the responder is now aware that the response was received by the originator. In case of a lost response or notify, the nodes will return to the idle state after a timeout of

$$T_w = macResponseWaitTime \cdot aNumSuperframeSlots \cdot \frac{T_s}{2^{SO}}. \tag{4.1}$$

If all messages arrive, the GTS is considered allocated on both sides and will be written to the Allocation Counter Table (ACT), a local data structure on every node that describes which slots are allocated with other nodes. In contrast to the SAB, the ACT does not include information about allocated slots in the neighborhood, but includes further information about the own allocated slots, such as the address of the remote device. The allocated slot will be used for transmissions in the subsequent multi-superframes.

4.2.3 Optimizing the Communication in the CAP

Since the handshake is conducted during the CAP, it is quite possible that transmissions from different devices overlap. It is thus necessary to use CSMA/CA to keep these collisions at a minimum. In the following, unslotted CSMA/CA during the CAP is assumed, though slotted CSMA/CA would be possible since beacons are available. However, the complexity of slotted CSMA/CA is higher and it usually does not provide better performance, as shown for example in [WLZ09].

Special considerations have to be made for DSME due to the CFP and the beacon slots. For example when issuing a transmission towards the end of the CAP, it might not be possible to complete the backoff, clear channel assessment (CCA), data transmission and potentially ACK transmission within the remainder of a CAP. Therefore, the CCA is postponed to the

SO	Slot Duration		CAP Duration	
	Symbols	ms	Symbols	ms
1	120	1.92	960	15.36
2	240	3.84	1920	30.72
3	480	7.68	3840	61.44
4	960	15.36	7680	122.88
5	1920	30.72	15360	245.76
6	3840	61.44	30720	491.52

m_0	Max. Initial Backoff	
	Symbols	ms
3	140	2.24
4	300	4.80
5	620	9.92
6	1260	20.16
7	2540	40.64
8	5100	81.60

Table 4.1: Comparing the slot and CAP duration to the maximum length of the initial backoff phase for unslotted CSMA/CA. For example, for $SO = 3$ and $m_0 = 5$, the CCAs of transmissions issued at the beginning of the CAP will take place in the first 16% of the CAP, while for $m_0 = 8$, this phase will even continue in the subsequent CAP.

next CAP. In fact, the backoff is increased by the duration of the intermediate CFP and the beacon slot to avoid that all CCAs take place directly at the beginning of the CAP, leading to a higher probability of collisions. This will be repeated if the backoff was selected to be even longer than a single CAP.

For minimizing collisions, the selection of the *macMinBE* m_0 is of high importance. It determines the interval of $[0, (2^{m_0} - 1) \cdot S_b]$ symbols from which the first backoff is randomly selected, where

$$S_b := aUnitBackoffPeriod \cdot \text{Symbol duration} = 20 \cdot 16\,\mu\text{s}. \tag{4.2}$$

Since the CAP takes place at most half of the available time, CAP messages are often delayed and many backoffs start at the beginning of the CAP. If the backoffs are too short, the probability of collisions is very high, especially in hidden node situations.

In Table 4.1, the length of the CAP is compared to the length of the maximum initial backoff for various settings. A value of m_0 of at least 5 should be chosen to spread out the beginning of the transmissions in the first 16% of the CAP for SO = 3 in order to mitigate this problem.

Finally, the CSMA/CA settings influence the selection of the *macResponseWaitTime* that is the maximum time until a response, for example a DSME GTS Response or Notify, is considered as lost. It is given in multiples of 960 symbols (*aBaseSuperframeDuration*). The maximum time in symbols it can take until a transmission consisting of S symbols (see Sect. 3.1.1) is successfully transmitted is calculated as

$$n \cdot \left(\left(\sum_{i=0}^{m} 20 \cdot W_i \right) + aCcaTime + S + macAckWaitDuration \right), \tag{4.3}$$

where n is *macMaxFrameRetries*, m is *macMaxCsmaBackoffs*, $aCcaTime = 8$, *macAckWaitDuration* $= 54$ as defined in the IEEE 802.15.4 standard and

$$W_i = \begin{cases} 2^{m_0} & i = 0 \\ 2^i \cdot W_0 & 0 < i \leq m_b - m_0 \\ 2^{m_b - m_0} \cdot W_0 & i > m_b - m_0 \end{cases}, \tag{4.4}$$

where m_b is *macMaxBE*.

With $n = 3$, $m = 4$, $m_0 = 5$, $m_b = 7$ and $S = 50$, this results in 29106 symbols, that is about 7.6 CAP phases for SO $= 3$, so without CAP reduction it takes up to 931 ms until the message is delivered, resulting in a *macResponseWaitTime* of 61. For CAP reduction, this value is multiplied by the number of superframes per multi-superframe, so we use 244 for the evaluation. Of course, it rarely happens that the full time is taken, but the default value of 32 and even the maximum of 64 given by the standard are too low for many practical applications, especially when using large SO and MO and since this calculation does not even consider processing delay for parsing the request and assembling the response.

4.2.4 UPPAAL Model

Since definitive statements about the worst-case behavior of the slot allocation procedure are aspired, the scenario is modeled as a network of timed automata. This is done by means of the UPPAAL tool environment [LPY97] that allows for convenient modeling and formal verification of such a system. An overview of the involved automata is given in Fig. 4.3. The model includes a three layer network stack for the nodes as well as an omniscient observer and a global clock.

The UPPAAL model files are available at
https://github.com/koalo/DSMEVerification
DOI 10.5281/zenodo.3148148

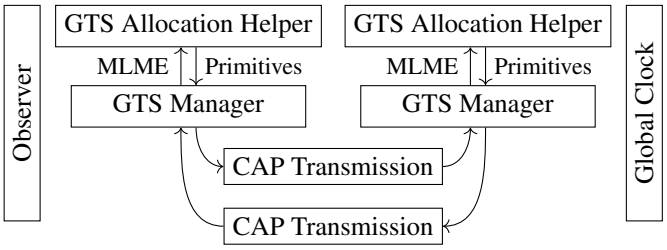


Figure 4.3: Overview of the system model.

- The **GTS Manager** implements the core functionality of the DSME protocol, including the generation of MLME messages for the higher layers as well as MAC frames for transmission via radio.

- The **GTS Allocation Helper** implements the functionality that is dedicated to the higher layers according to the standard. That mainly covers the decision if a slot can be allocated and choosing a slot, as well as generating requests for new slots.
- A **CAP Transmission** is dedicated for the transmission of a message from one GTS Manager to another. It delays the message to be sent and decides the outcome of the transmission.
- The **Observer** regularly checks for inconsistencies, and measures the duration until an inconsistency is recognized by the nodes where applicable.
- The **Global Clock** generates events for every superframe. In a real implementation, the time synchronization is done via beacons, but since this is out of the scope of this model, a global clock simplifies the model and thus allows for faster verification.

UPPAAL allows for a graphical specification of the automata as shown in Fig. 4.4-4.8. Apart from the syntax, the implementations of the GTS Manager and the GTS Allocation Helper are very similar to an implementation for hardware or simulation where an equivalent state machine could be specified. However, the channel model is quite different since its purpose diverges from the one in a simulation where it is advised to closely model the probabilities for the events such as a lost packet. Instead of determining the probability that a certain event occurs, a formal analysis assumes that a certain event occurred and thereupon decides about the soundness of certain statements, for example whether the system is still in a valid state *if* a packet was lost. This is evaluated for all possible sequences of events. On the downside, this requires traversing a very large state space. On the upside, it is not required to model the probabilities, but a specification of all possible events as follows is sufficient.

Once a device has a message to transmit via the CAP phase, it senses the channel. If it is unable to sense a free channel several times in a row, the packet is not sent and an ACCESS_FAILURE is signaled. Otherwise, the packet is sent, but might not be received by the receiver due to collisions or other sources of disturbances. If it was sent as broadcast, the transmitter does not know if the packet was received and will signal SUCCESS. If it was sent as unicast, an acknowledgment is expected. In case of such a failed transmission, NO_ACK is signaled after a timeout. This is also signaled if the transmission was properly received, but the acknowledgment is lost. Otherwise, SUCCESS is signaled if the acknowledgment is received.

4.2.5 Checking for Consistency

The Allocation Counter Table (ACT) for storing the information about the allocated slots is denoted as

$$ACT : N \times S \to N \cup \{\circ\}, \tag{4.5}$$

with N being the set of nodes, $S = \{s \in \mathbb{N}_0 \,|\, 0 \leq s < N_s\}$ being the set of possible slots and $\circ$ denoting that a slot is not allocated. For this, we assume the use of the channel

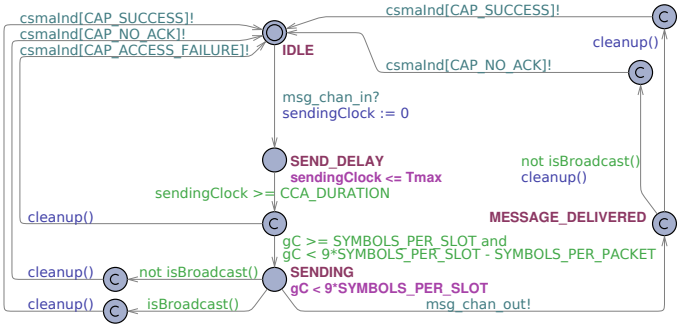


Figure 4.4: State diagram of the CAP Transmission.

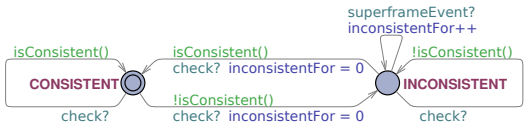


Figure 4.5: State diagram of the Observer.

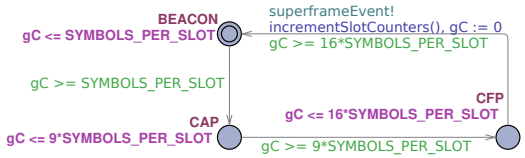


Figure 4.6: State diagram of the Global Clock.

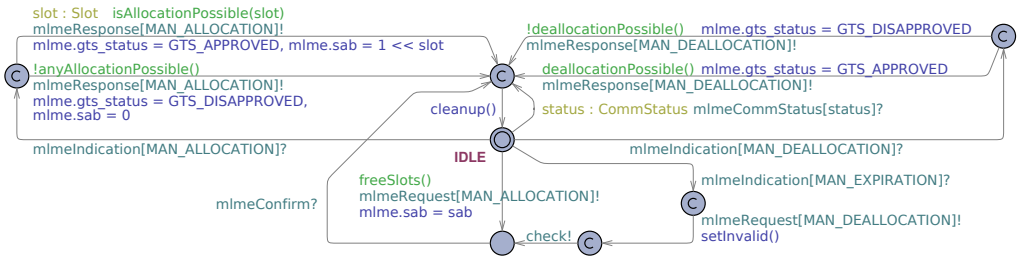


Figure 4.7: State diagram of the GTS Allocation Helper.

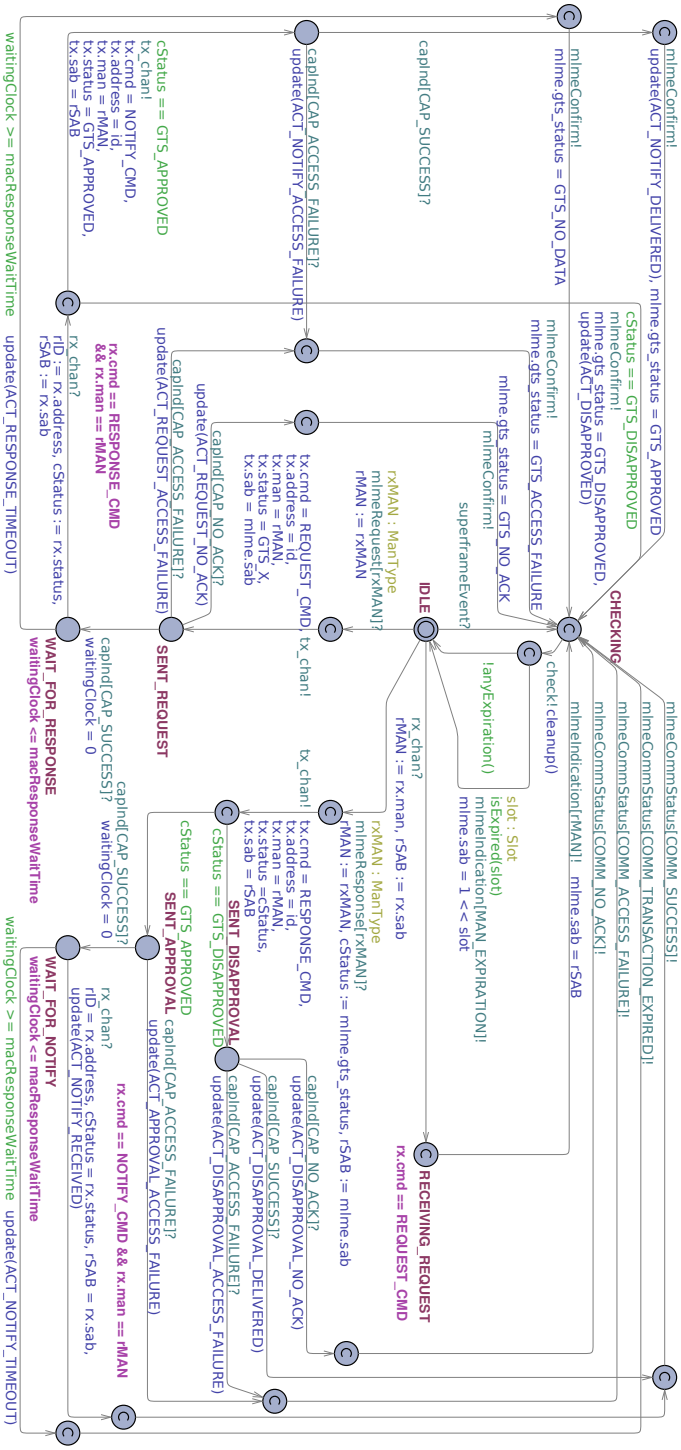


Figure 4.8: State diagram of the GTS Manager.

adaption mode with the maximum number of 16 channels and no CAP reduction, so the number of allocatable slots is

$$N_s = 16 \cdot 7 \cdot 2^{\mathrm{MO}-\mathrm{SO}}, \tag{4.6}$$

but a device can only use one channel at a time. The consistency predicate is defined as

$$\begin{aligned} \phi : \ & \forall n, m \in N \wedge \forall s \in S : \\ & ACT(n,s) = m \Rightarrow ACT(m,s) = n. \end{aligned} \tag{4.7}$$

The requirement for a permanently consistent system, that is that ϕ holds for all points in time, is denoted as

$$A\Box\,\phi. \tag{4.8}$$

The entry in the ACT is not absolutely synchronized, so (4.8) does not hold, but since the state of the slot allocation is only relevant during the CFP, this expression can be relaxed to

$$A\Box\,(\mathrm{CFP} \Rightarrow \phi)\,, \tag{4.9}$$

where CFP is satisfied if and only if the global clock is currently in the CFP state.

The Lost Notify

When checking the model against (4.9), UPPAAL classifies it as not satisfied and gives a sequence as shown in Fig. 4.9 as a counterexample. If the notify is not received by B, it will never mark the slot as allocated in its ACT. Now A assumes the slot is allocated, while B

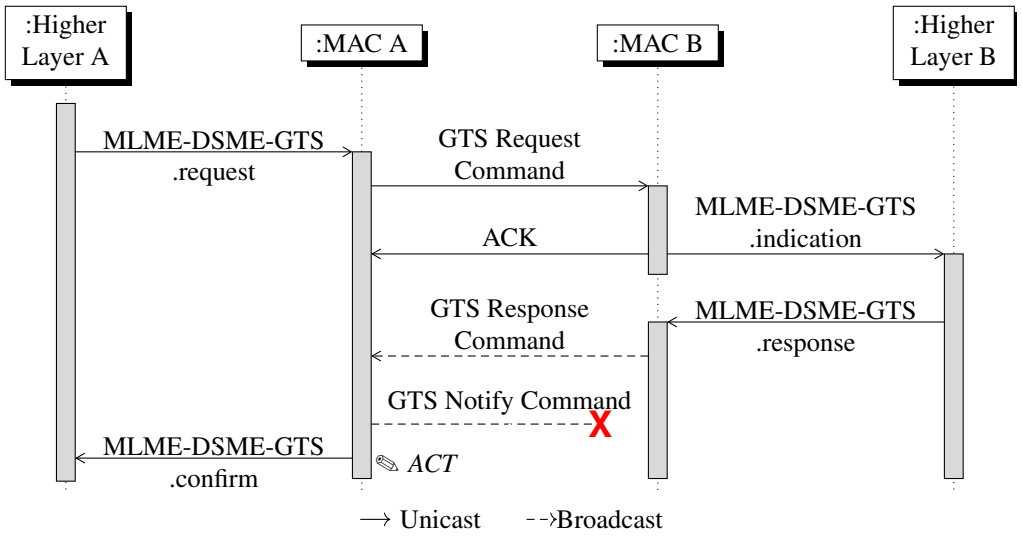


Figure 4.9: Sequence diagram of a failed slot allocation.

assumes the slot is not allocated, leading to an inconsistent state. It is interesting to note that in the original IEEE 802.15.4e standard the slot is written to the ACT already after sending the response, but this would not change the result because in that case the inconsistent state would occur after a lost response instead of a lost notify. The reason for the modification is not mentioned in the standard, however, both versions can lead to an inconsistent state.

While this is an undesirable result, the underlying problem of attaining consensus is of general nature. Attaining common knowledge in a distributed system, such as agreeing on a common GTS, is a well-studied concept and is actually proven to be impossible in the strict sense [HM90]. This proof also refers to the well known coordinated attack problem where two divisions of an army try to agree on a common time to attack but can only communicate via messengers that might get caught. Even with multiple messengers sent back and forth, the sender of the last messenger can not be sure that the messenger arrived and thus can not be sure that the other division will attack. Correspondingly in the DSME protocol, the sender of the notify can not know if the counterpart will eventually set the slot to allocated.

Node Failures

A similar, but more obvious problem occurs when nodes reset during the course of the slot allocation or at a later point in time. This might be due to energy shortages in energy-harvesting applications or bugs in hard- or software. This is modeled by adding an additional transition from every state to the initial state that is fired optionally for all automata of one device at the same time. These transitions are omitted in Fig. 4.4-4.8 for clarity. As soon as a node resets, the ACT is cleared, while the other node might have the slot still allocated or while it even waits for a pending message. For the latter case UPPAAL verifies that due to the timeout T_w, the other node returns to the idle state, too, and does not block. However, if the node resets after the other node has written the slot to the ACT, ϕ is no longer satisfied.

4.2.6 DSME GTS Expiration

While it is not possible to establish common knowledge in a strict sense as defined by Halpern and Moses [HM90], they and others (e.g. [GD92][PT92]) made suggestions to weaken the requirements. For example, if a single division attacks and notices that the other division is not there as well, it might be able to escape, possibly losing some troops. Generalized, if a party will eventually recognize an inconsistency in the common knowledge and act accordingly, the system will eventually return to a consistent state. For the DSME application, this is formalized as

$$A\Box\left(\neg\left(\text{CFP} \Rightarrow \phi\right) \Rightarrow A\Diamond\left(\text{CFP} \Rightarrow \phi\right)\right)$$

with $A\Diamond$ denoting that the statement will eventually hold.

In fact, the standard includes a mechanism to guarantee this property called DSME GTS expiration. If the transmitter has transmitted *macDsmeGTSExpirationTime* (M_x) frames and has not received any acknowledgment, it will deallocate the slot with a procedure that is very similar to the slot allocation. The receiver, on the other hand, will start a deallocation if no frame was received for *macDsmeGTSExpirationTime* opportunities. UPPAAL actually

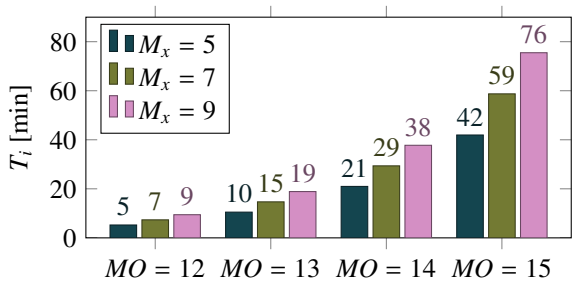


Figure 4.10: Duration for returning from an inconsistent to a consistent state.

verifies that this mechanism eventually leads to a consistent state again, both for the case of a lost notify or of a node failure at an arbitrary point in time.

So what is the price for this, how many troops are lost? Until the slot is recognized as invalid, all communication over this slot will fail. In order to quantify this effect, the observer measures the time $\neg\phi$ holds. However, UPPAAL can only give the results *satisfied* or *not satisfied*. Therefore, it is necessary to verify the property that this clock never gets larger than a fixed value T_i. By testing multiple values, for example by using a binary search, it is possible to determine the maximum duration $\neg\phi$ holds. The result is dependent on the parameters as shown in Fig. 4.10. Thereby, the parameters $M_x = 7$ and $\mathrm{MO} = 15$ are the ones given as default by the standard, resulting in an inconsistency duration of about one hour.

Of course, it is possible to decrease the MO and even achieve less overhead since slots are repeated more often, so less slots have to be allocated to maintain the same throughput. However, this also reduces the time interval until an expiration if M_x is kept constant. For practical applications, especially with large sending interval or a high variance in the traffic, this would lead to repeated deallocations and allocations that induce a high overhead again. Therefore, M_x has to be increased significantly to cancel this out, leading to the same problem of a late repair of an inconsistent state.

4.2.7 Improvement of the Slot Allocation Procedure

While this result shows that in fact the system returns to a consistent state in finite time, the times are very long. Especially in configurations with a large MO, the timeouts are way too long for many applications. Reducing the parameters to decrease the duration is often not possible as elaborated in Sect. 3.2.3. A possible solution would be to generate additional dummy traffic in applications with low traffic demand to avoid the slot expiration for actually valid slot allocations, but of course, this leads to unnecessary channel utilization and boosts the energy consumption. Furthermore, reducing M_x would increase the probability of premature deallocation in case of short-term channel disturbance.

However, by investigating the possible outcomes of a slot allocation, it got evident that at least one of the participants can recognize a potential invalid state much earlier and can

anticipatorily send a deallocation. In order to model this mechanism and also as a suggestion for an actual implementation, the ACT is extended with an invalid flag

$$ACT_n : N \times S \to (N \times \{\text{OK}, \text{INVALID}\}) \cup \{\circ\}.$$

This flag indicates that the node shall deallocate the slot again as soon as possible. The predicate is changed to

$$\begin{aligned}\psi :\ & \forall n, m \in N \wedge \forall s \in S : \\ & (ACT_n(n,s) = (m, OK) \Rightarrow ACT_n(m,s) = (n, OK)) \\ & \vee\ ACT_n(m,s) = (n, \text{INVALID}).\end{aligned}$$

Note that ψ is weaker than ϕ, because it does not imply that *both* nodes have recognized the invalid state. In the unfavorable case, the anticipatory deallocation will not reach the other node. However, this will be recognized since no acknowledgment is received and the deallocation can be repeated immediately. Furthermore, this improvement only helps in case of communication failures, but not node failures if the device sending the deallocation resets before the deallocation request was sent. Even in those cases, the expiration procedure will guarantee a consistent state as determined in Sect. 4.2.6. In any case, it is reasonable that the receiver also turns on the transceiver during invalid and unconfirmed slots, but the transmitter only transmits in slots with OK flag. Thus, packets sent during the transition period have a higher chance to get delivered even though an inconsistency exists.

In order to identify the possible events that can indicate an inconsistent state provoked by the allocation procedure, Fig. 4.11 shows the possible outcomes from the view of the initiator and Fig. 4.12 from the responding device. The possible outcomes are marked as explained in the following.

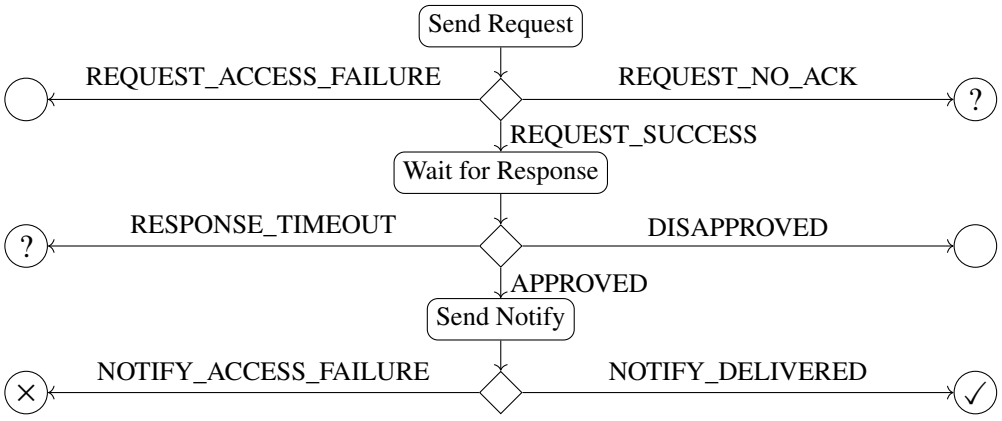


Figure 4.11: Possible results for the requesting device.

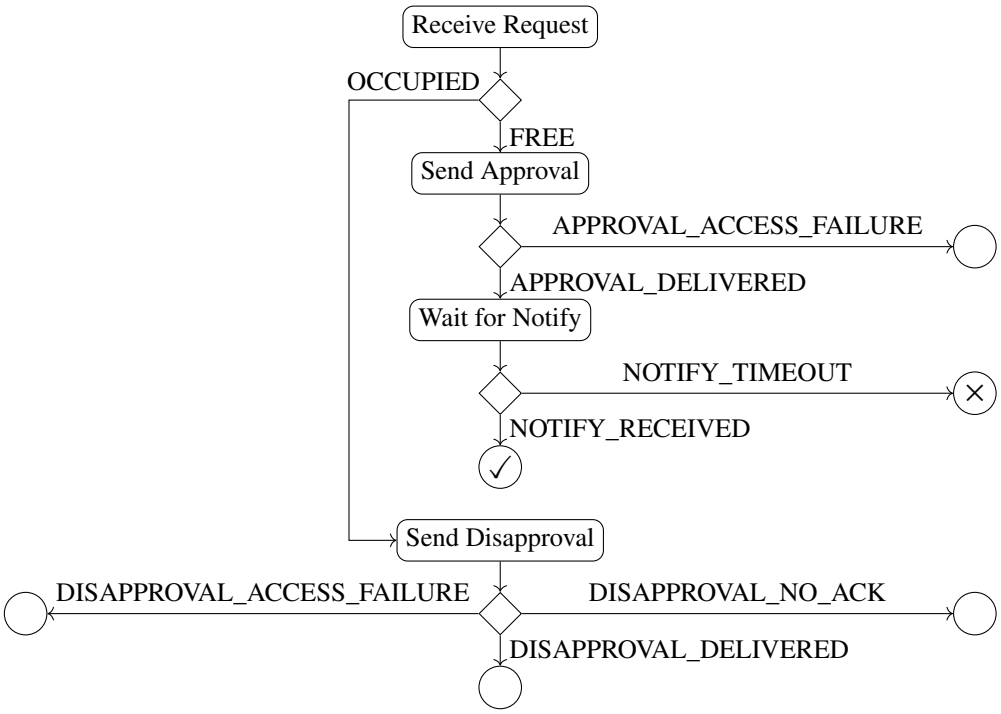


Figure 4.12: Possible results for the responding device.

- ✓ A slot was successfully associated, at least from the perspective of the respective device.
- ○ No positive response was or will be sent to the requesting device. Thus, no allocation or deallocation is necessary.
- ? The request was sent, but no answer was received. Even if a response was sent, there is no indication for the requesting device which slot was chosen, so there is no possibility to allocate or deallocate it. In the worst case, the responder will run into a timeout when waiting for the notify. Either way, no device will allocate the slot.
- ⊗ The notify was not successfully delivered to the responder. The slot is actually known by both sides, but this fact is not known by the responder as explained in Sect. 4.2.5. As soon as this case is recognized, the slot should be marked as INVALID and the devices should deallocate it properly.

By marking a slot as INVALID, the inconsistency case can be easily recognized by at least one of the devices. This is the case if either a timeout occurs when waiting for the notify or the requester senses a busy channel while sending the notify. Note that even though the slot was not allocated in the NOTIFY_ACCESS_FAILURE case, the requesting device should not rely on the deallocation due to the timeout on the responding device. Otherwise, the slot would no longer be known to the requesting device and no proper deallocation will take place, therefore, the neighborhood would not be informed about the inconsistency.

The maximum duration until an inconsistent state is recognized is now mainly dependent on the *macResponseWaitTime* (W_r), resulting in a different T_w which is the timeout for waiting for a notify. As shown in Fig. 4.13, the durations are lower than the ones resulting from the DSME GTS expiration by orders of magnitude. It can be minimized by choosing a low W_r, but this might lead to false negatives in case the notify is delayed, for example due to a high network load. Choosing the default $W_r = 32$ is sufficient for most cases.

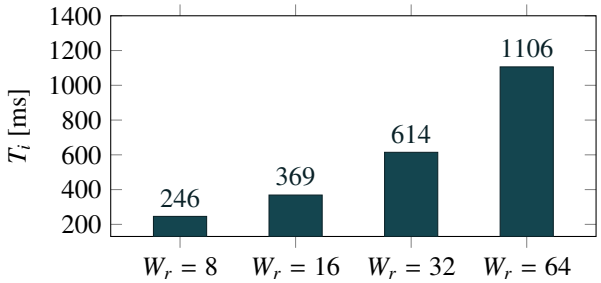


Figure 4.13: Duration for recognizing an inconsistent state for the improved handling.

4.2.8 Further Open Issues

In the previous example only two nodes are considered. With more nodes in the network, other problems can arise that are briefly presented in the following.

Removing Occupied Slots from SAB

One such situation is depicted in Fig. 4.14. In this situation two pairs of devices A/B and C/D use the same time and frequency slot. Since they are not in range of each other, this is no problem as such and even improves spatial reuse. Node E was informed during the slot allocation handshake that this slot shall not be used.

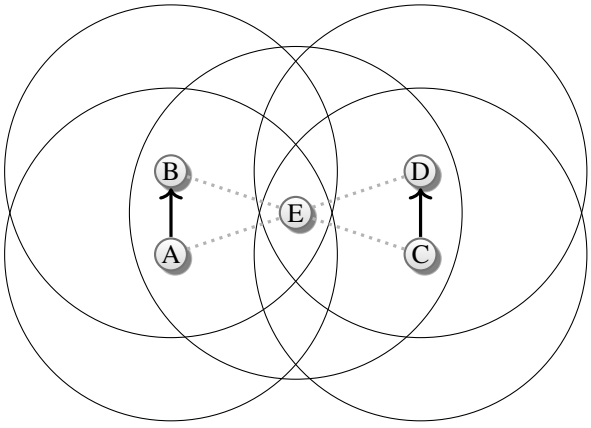


Figure 4.14: Situation where a slot might be incorrectly removed from the SAB.

If now for example C decides to deallocate this slot, they perform a deallocation handshake. By doing so they inform node E about the deallocation. Since node E does not store the information how many neighbors reserved the slot, it will be marked as free, leading to an inconsistent slot allocation bitmap (SAB).

A solution would be to use a counter instead of a single bit, but since the situation is assumed to occur rarely and B will send a duplicate allocation message anyway if E tries to use the slot, this is not worth the additionally required memory.

Eternal Blocking of Slots

A more severe problem that is especially relevant for long running networks with a volatile topology is described in the following. There are multiple variants of this problem, but all have in common that slots are not removed from the SAB even though no pair of nodes is still using them. Over time, this will fill up the SAB until no more free slots are available and no new allocation can take place.

This happens for example if the radio conditions between two nodes deteriorate significantly. Even if this is recognized by the nodes (see Sect. 4.3.3), they are no longer able to perform a proper deallocation handshake. In particular, no response and no notify message will be sent to inform the neighbors that the slot is no longer in use.

The same situation will occur in case of mobility or if a battery powered node runs out of energy and shuts down before allocated slots can be deallocated. While the latter could be mitigated by storing the ACT in persistent memory, a general solution is in demand to avoid the eternal blocking of slots.

A possible solution might be to remove occupied slots from the SAB that were not modified for a long time, indicated by a single additional bit per slot. In regular intervals of maybe about an hour, this flag will be set for every occupied slot. If no deallocation took place after another hour, the bit is still set and the slot will be marked as free in the SAB. Of course, as for the previous issue, one has to rely on the duplicate allocation message to avoid inconsistencies in this case. This could be improved by leasing slots only for a certain time span after which they have to be reallocated.

4.2.9 Conclusion

In this section, the distributed slot allocation procedure is analyzed by means of formal verification. It was shown that in fact inconsistencies in the slot allocation can occur and are not avoidable. However, these inconsistencies are not persistent and will always be resolved. In the original standard, this is done by counting failed transmission during the contention free phase. Since this can induce delays in the order of one hour, it is proposed to exploit the supervision of the message transmission during the slot allocation procedure to recognize the inconsistency much earlier. As a suggestion for a sound implementation, an additional flag is added to the allocation counter table to mark invalid slots. It is shown by formal verification that using the proposed optimization reduces the time until an inconsistency is recognized by several magnitudes.

4.3 Traffic-Aware and Prediction-Based Slot Scheduling

In this section, a distributed traffic-aware scheduling technique is presented that can easily adapt to changing conditions without complete recalculation of the schedule. It is based on the prediction of the future traffic demand over a link. While it is expected that the presented approach can cope with a lot of diverse requirements, it is optimized for scenarios where a high amount of traffic that follows a stationary random distribution has to be delivered reliably. In the simplest case this is a fixed transmission interval that is common for tracking machine parameters such as the state-of-health. One example is the previously mentioned solar tower power plant where a constant monitoring of the steerable mirrors is required in order to maintain the power output of the plant. In other applications, random distributions such as a Poisson distribution might occur. Random processes are for example common for mass customization applications where workpieces on conveyor belts are monitored. In the presented approach, topology changes such as fluctuating wireless conditions will be handled by a decentral recalculation of the schedule. Still high mobility remains problematic where slot allocations are only valid for a short time-span.

4.3.1 Specification of the Scheduling Technique

The proposed technique requires no explicit information from the routing layer, in contrast to [HWW17], and it also does not require any additional message exchange apart from the slot allocation handshake of DSME. This makes it both scalable and versatile so it can be used with any routing layer, including RPL [WTB12] and GPSR [KK00] as shown in the upcoming evaluations. Nevertheless, it can dynamically adapt to the current traffic load by predicting the future amount of traffic per link. It shares some similarities with [PWW16], especially the utilization of overprovisioning, but in contrast, the approach presented in the following is not prone to slot collisions due to the properties of the DSME slot allocation handshake. Furthermore, more light will be shed on the characteristics of the traffic predictor.

In the following, every outgoing link of a node is considered separately. The number of packets pushed to the sending queue during one multi-superframe p_t is counted. This includes the traffic generated at that node and the traffic to be forwarded. An exponentially weighted moving average filter is applied to this value to predict the traffic load λ_t in future multi-superframe according to

$$\lambda_t = \alpha_\lambda \cdot p_{t-1} + (1 - \alpha_\lambda) \cdot \lambda_{t-1}, \text{ with } \lambda_0 = 0. \tag{4.10}$$

This corresponds to the average number of transmission time slots per multi-superframe for the considered neighbor. Accordingly, DSME GTS Requests can be sent to either allocate or deallocate slots to match the actual number of slots $c_{act,t}$ with the required number of slots $c_{req,t}$. However, directly setting $c_{req,t} = \lceil \lambda_t \rceil$ would lead to many repeated slot allocations and deallocations as it will be demonstrated in the evaluation in Fig. 5.14 on page 74.

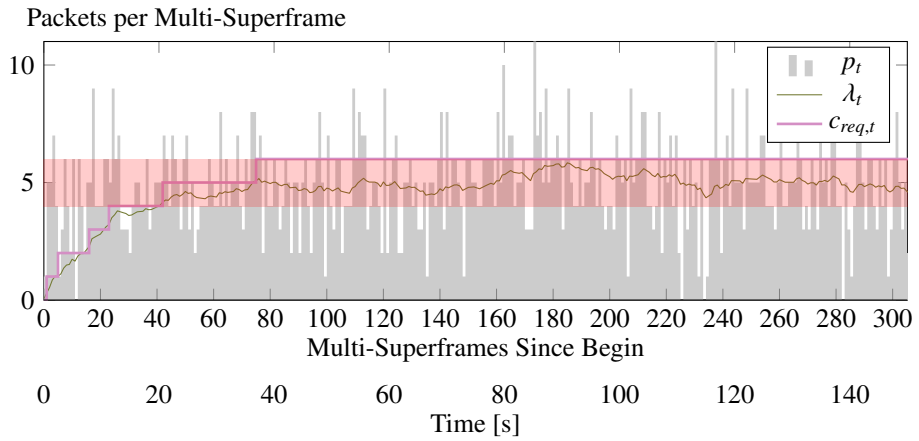


Figure 4.15: An exemplary course of the proposed slot management.

Therefore, a hysteresis is applied, so the required number of slots is calculated as

$$c_{req,t} = \begin{cases} \lceil \lambda_t \rceil & \text{for } \lambda_t - c_{act,t-1} > 0 \\ \lceil \lambda_t \rceil + 1 & \text{for } \lambda_t - c_{act,t-1} < -2 \\ c_{act,t-1} & \text{else.} \end{cases} \tag{4.11}$$

An exemplary course is depicted in Fig. 4.15. The gray bars exemplarily represent Poisson distributed traffic with mean $\overline{p} = 5$ packets per multi-superframe with SO $= 3$ and MO $= 5$. λ_t initially increases and then fluctuates around $\overline{p}$. For the plot, $\alpha_\lambda = 0.05$ is chosen, so λ_t stays, at least in the given time section, within the hysteresis interval of $\overline{p} \pm 1$ depicted by the horizontal bar after about 20 s. A stable configuration is reached after 40 s.

4.3.2 Influence of the Smoothing

The parameter α_λ is used to control the amount of smoothing. For a α_λ close to 1, the current value has a large influence. The system reacts faster to changes in the amount of traffic, but λ_t has a high fluctuation, leading to many GTS allocations and deallocations. For a α_λ close to 0, the opposite holds. Due to the hysteresis, no GTS allocations or deallocations take place if λ_t stays within the respective interval. This effect is assessed in the following for a generic traffic distribution. In order to model the p_t as introduced in the previous section, the random variable X describes the number of incoming packets per multi-superframe and follows a probability distribution with mean $\overline{p}$ and has a given probability mass function

$$f_X : \mathbb{N}_0 \to \{x \in \mathbb{R} \,|\, 0 \leq x \leq 1\}\,. \tag{4.12}$$

The random variable Y_t models the value of λ_t and has a probability mass function

$$f_{Y_t} : K_t \to \{x \in \mathbb{R} \,|\, 0 \leq x \leq 1\} \tag{4.13}$$

with the preimage K_t depending on the time step as derived in the following. Since $\lambda_0 = 0$, Y_0 will surely be 0, so

$$K_0 = \{0\}, f_{Y_0}(0) = 1. \tag{4.14}$$

Analogous to Eq. (4.10), the following Y_t are given by

$$Y_t = \alpha_\lambda \cdot X_{t-1} + (1 - \alpha_\lambda) \cdot Y_{t-1}. \tag{4.15}$$

Therefore, all possible values for Y_t are in

$$K_t = \{\alpha_\lambda \cdot m + (1 - \alpha_\lambda) \cdot c \mid m \in \mathbb{N}_0, c \in K_{t-1}\} \tag{4.16}$$

and for a given $m \in \mathbb{N}_0$, a $k \in K_t$ only occurs in the case the previous value $c \in K_{t-1}$ was

$$c = \frac{k - \alpha_\lambda \cdot m}{1 - \alpha_\lambda}. \tag{4.17}$$

Thus, the probability that Y_t has the value k is given by

$$P(Y_t = k) = \sum_{m \in \mathbb{N}_0} P(X_{t-1} = m) \cdot P\left(Y_{t-1} = \frac{k - \alpha_\lambda \cdot m}{1 - \alpha_\lambda}\right) \tag{4.18}$$

and therefore the probability mass function of Y_t is given by

$$f_{Y_t}(k) = \sum_{m \in \mathbb{N}_0} f_X(m) \cdot \begin{cases} f_{Y_{t-1}}\left(\frac{k-\alpha_\lambda \cdot m}{1-\alpha_\lambda}\right) & \text{for } \frac{k-\alpha_\lambda \cdot m}{1-\alpha_\lambda} \in K_{t-1} \\ 0 & \text{else} \end{cases}. \tag{4.19}$$

From this, the probability mass function after the initial settling time is calculated as $f_{Y_\infty} = \lim_{t \to \infty} f_{Y_t}$ and finally

$$P_{\text{int}} = P(\overline{p} - 1 \leq Y_\infty \leq \overline{p} + 1) = \sum_{k \in K_\infty} \begin{cases} f_{Y_\infty}(k) & \overline{p} - 1 \leq k \leq \overline{p} + 1 \\ 0 & \text{else.} \end{cases} \tag{4.20}$$

is the probability that λ_t is within the interval $[\overline{p} - 1, \overline{p} + 1]$ at a given point in time after the initial settling. However, the actual evaluation is computationally infeasible, so from this an algorithm to calculate an upper and a lower boundary for P_{int} is derived that is presented in Appendix A.1.

The second effect to consider is the delay induced by the smoothing. As derived in Appendix A.1, for a fixed packet rate $\overline{p}$, the number of multi-superframes it takes until $\overline{p} - 1$ is reached is given by

$$t = \frac{\log\left(\overline{p}^{-1}\right)}{\log\left(1 - \alpha_\lambda\right)}. \tag{4.21}$$

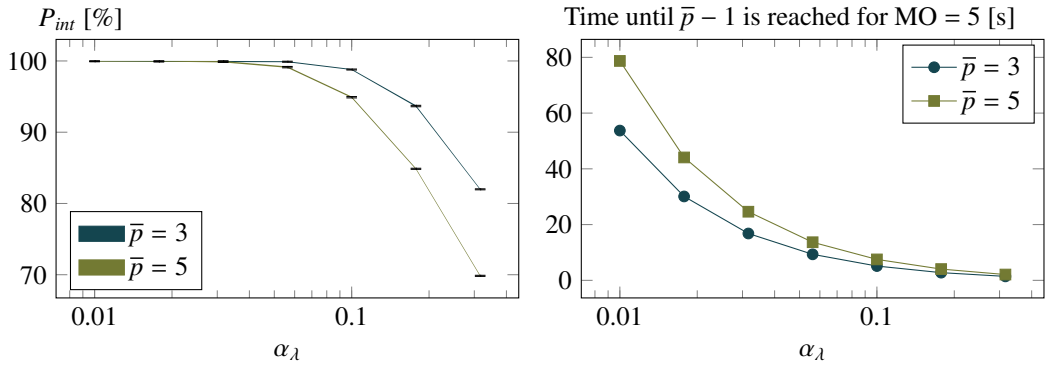


Figure 4.16: Trade-off for choosing the α_λ parameter.

For a fixed transmission interval, the calculation always yields $P_{\text{int}} = 100\%$. This is obvious since λ_t stays constant after the initial settling. For a Poisson traffic distribution, the results are more interesting as shown in Fig. 4.16. With $\alpha_\lambda \leq 0.05$, the probability of staying within the hysteresis interval is larger than 99%, but decreases for larger α_λ. In the same range, it takes more than 10 s to react to significant traffic increases, either at the initialization or later, for example when routes change. If not specified otherwise, $\alpha_\lambda = 0.05$ is applied throughout this thesis, but larger α_λ should be chosen for a more reactive system or a smaller α_λ to reduce the management traffic.

4.3.3 Depreciate Links

While the presented approach leads to a stable slot assignment under static conditions and can still adapt to changes in the traffic load or the radio environment, its hysteresis leads to an unwanted effect if the connection between two nodes is lost. This might be due to changing channel conditions or mobility. If the network layer detects that a link does no longer exist, no more traffic will be routed over this link ($p_t = 0$). According to Eq. (4.10), however, $\lambda_t > 0$ will always hold and thus $c_{req,t}$ will be at least 1 even if the link is no longer available. Therefore, the node will repeatedly try to allocate at least a slot for this link. Is is especially unwanted if the link is too bad to allow for a successful slot allocation handshake that will therefore be retried again and again.

To avoid this effect, a counter is introduced that counts the number of multi-superframes since the last multi-superframe with packets to be sent out ($p_t > 0$). If this counter reaches a threshold, $c_{req,t}$ is forced to 0. The threshold depends on the traffic distribution and the stability of the network, but if the amount of traffic per time is not too low, setting it to the same value as *macDsmeGtsExpirationTime* is usually a good choice. The normal operation is resumed as soon as packets are available again.

4.4 An Open-Source Implementation of IEEE 802.15.4 DSME*

In order to verify the developed methods under realistic conditions, a comprehensive implementation of IEEE 802.15.4 DSME is developed, denoted as openDSME. For enabling broad usage by researchers and in real-world applications, it is published as open-source software under the terms of a 3-clause BSD-style license.

The openDSME software is available at opendsme.org. In particular:
Core: github.com/openDSME/openDSME DOI 10.5281/zenodo.3147550
For OMNET++/INET: github.com/openDSME/inet-dsme DOI 10.5281/zenodo.3147131
For Contiki: github.com/openDSME/contiki DOI 10.5281/zenodo.3147774

In contrast to, for example, many existing implementations of TSCH, such as the one for Contiki or the full stack implementation OpenWSN [WVK12], openDSME is not restricted to a given platform. In contrast, it only requires basic services such as a timer and provides a simple interface to the physical and network layer for easy integration into existing systems. This basic design principle of high portability already enabled the integration in various frameworks such as the OMNeT++ simulator [Var01], CometOS [UWT12] including support for the ATmega256RFR2 and Contiki [DGV04] including the Cooja simulator and the M3OpenNodes provided by the FIT/IoT-LAB [ABF15].

This approach furthermore enables the deployment of the same software for simulation and wireless hardware. This allows for convenient development and debugging in reproducible and controlled simulated environments and short development cycles without the hassle of real-world deployments. Afterwards, the same software that has already proven to provide the anticipated functionality in the simulator can be used with wireless hardware and the testing can then focus on the platform-specific peculiarities.

As programming language of openDSME C++ is used, but it does not require features provided by the STL that would constraint its portability. Currently, it consists of about 18000 lines of C++ source code (including comments). While this is a debatable metric for the complexity of software, the comparison, for example, with the 2000 lines of codes required for the CSMA/CA implementation of the INET framework [INET] illustrates the high effort required for efficient time-slotted medium access.

The MAC is fully implemented in software, including the generation of acknowledgments. On the upside, this allows for a very flexible implementation. Especially under consideration of inefficient hardware MAC layers [WT15], this is a large bonus and furthermore allows for easy adaptation to other platforms. On the downside, for hardware with little computational power, timing issues become very relevant. DSME poses real time requirements for delays such as the maximum wait time for an ACK that have to be handled appropriately for hardware platform. For simulations this is not an issue, because the simulated time is decoupled from the real-world time.

*Extended version, including a tutorial, published in "F. Kauer, M. Köstler, and V. Turau, *openDSME - Reliable Time-Slotted Multi-Hop Communication for IEEE 802.15.4* in Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem, ed. by A. Virdis and M. Kirsche, ISBN 978-3-030-12841-8, Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-12842-5_15" [KKT19].

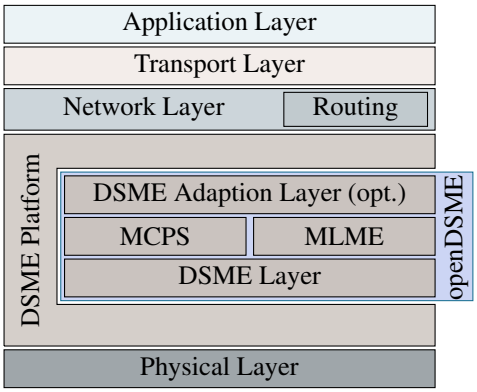


Figure 4.17: Integration of openDSME in the network stack.

Fig. 4.17 depicts how openDSME is embedded into the network stack. The inner structure can be coarsely separated into the `DSMELayer` that is the actual implementation of IEEE 802.15.4 DSME and the `DSMEAdaptionLayer` that provides helpers that are not specified in the standard, but are important for a seamless integration in a network stack. These layers are interconnected by the standardized Medium Access Control Common Part Sublayer (MCSP) and Medium Access Control Sublayer Management Entity (MLME) Service Access Points (SAP). This separation makes it possible to use openDSME either as pure DSME implementation and controlling it via the MCPS and MLME interfaces or to utilize the simple packet-based interface of the `DSMEAdaptionLayer` and use openDSME as drop-in replacement for existing link-layer implementations. Every platform only has to provide a thin wrapper, called `DSMEPlatform`, that adapts the interface of openDSME to the one of the platform.

4.4.1 The openDSME Stack

Fig. 4.18 shows a more detailed view on the software architecture of openDSME that will be presented by tracing the path of a message. If the network layer wants to send a message, it forwards this message to the the `MessageHelper` that decides if any actions have to be performed before the message can be sent. If the node is not even associated, a scan will be issued via the `ScanHelper` and after a successful synchronization, an association will take place via the `AssociationHelper`.

The `GTSHelper` is informed about the incoming message and will consult the scheduling to find out if an allocation or deallocation if needed. A slot is automatically allocated if a message arrives towards a node to which no GTS is currently allocated. The current implementation does not try to optimize the delay and thus randomly selects a slot out of the available slots. The number of incoming messages during a multi-superframe is counted and this measure is used to control the number of required GTSs with the method proposed in Sect. 4.3. Other slot scheduling approaches can be easily plugged into the system to analyze their performance.

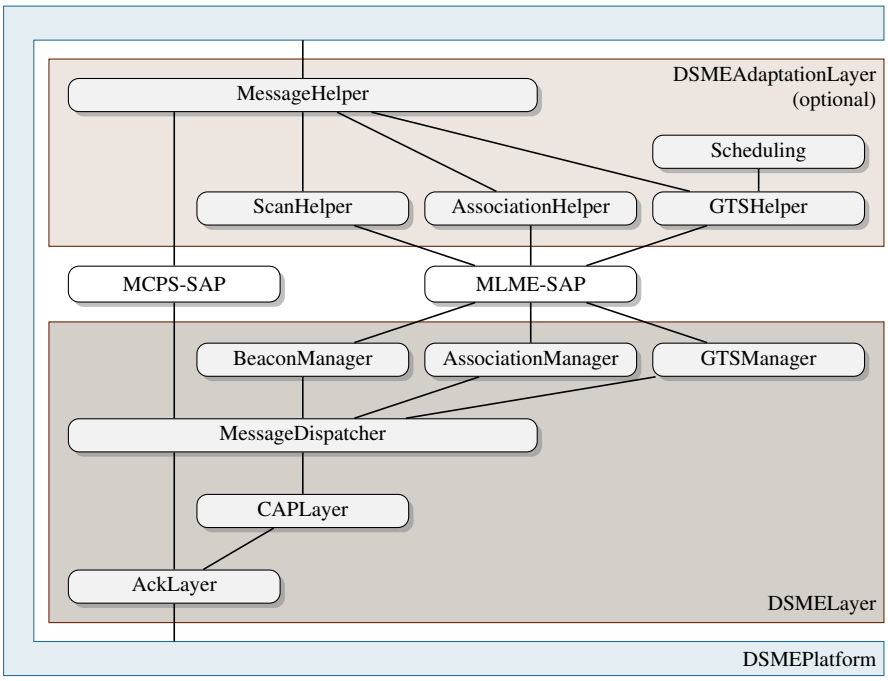


Figure 4.18: Software structure of the openDSME implementation.

The `MessageHelper` then forwards the data messages via the MCPS interface to the `MessageDispatcher`. It also specifies if the message shall be sent in the CAP or the CFP. Currently, all link-layer broadcasts of data packets are sent in the CAP and all other data packets are sent in the CFP. The `MessageDispatcher` is responsible for maintaining the DSME time structure and to send the correct messages at the given point in time and the corresponding frequency channel. Messages to be sent in the CAP are forwarded to the `CAPLayer` that is responsible for the CSMA-CA mechanism of the CAP. As soon as a message shall be sent, it is forwarded to the `AckLayer` that will send the message to the physical layer and also manages the reception of the acknowledgment, if requested. The `AckLayer` is also responsible for receiving messages from the physical layer and sending out acknowledgments. The message is then forwarded to the `Message-Dispatcher`. Data messages will be sent to the upper layer via the MCPS interface and the `MessageHelper`. DSME specific packets, such as beacons, association messages or GTS commands are dispatched to the respective manager modules.

4.4.2 Auxiliary Modules

In openDSME, several auxiliary modules are included that are not part of the stack. The most important ones are presented in the following.

The `NeighborQueue` is responsible for maintaining a list of the neighbors as well as to queue the messages to be sent. The neighbor list itself is implemented as red-black tree to allow for an efficient retrieval by the MAC address. Instead of using the map

implementation included in the C++ STL, an own implementation is provided to allow the usage of openDSME on hardware platforms without default STL support such as the Atmel microcontrollers. A neighbor is added to this list when the first message is issued by an upper layer to be sent to this neighbor. The queue itself is implemented as a set of linked lists, one for every neighbor and one for maintaining a set of free slots. This allows to operate on a fixed amount of message slots and thus avoids the need of dynamic memory. Still, it provides the oldest message pending for a given neighbor in $O(1)$ (after getting the neighbor iterator in $O(logn)$ where n is the number of neighbors). This is especially important for DSME, because the message to be sent in a slot does not only depend on the age, but also on the neighbor that listens in a given slot.

The `DSMEAllocationCounterTable` stores the currently allocated slots and is also implemented as red-black tree. This allows for efficient access to the state of every slot without the need to store the state of the idle slots explicitly. This would waste a lot of memory, because, especially when using a large MO, usually a fraction of the slots per multi-superframe are actually active for a given node. Related are the `DSMESlotAllocationBitmap` that stores the slots occupied by all neighbors and the `BeaconBitmap` that stores the occupied beacon slots in the neighborhood.

The operation of the modules in the stack itself, such as the `GTSManager` is specified as finite state machines. For this, the `DSMEBufferedMultiFSM` class is provided that allows to spawn multiple parallel transactions (for example if multiple GTS transactions run in parallel) and can buffer pending events. Other modules such as the `AckLayer` do not require parallel transactions and thus only use a subset of the provided functionality.

The `TimerMultiplexer` multiplexes single timer to the different timers that are used in the `DSMEEventDispatcher` to dispatch the various events required for the operation of openDSME. This includes the slot event that fires at every slot (in use) and a pre-slot event that is fired shortly before the slot event to prepare the transceiver (e.g. tune it to the correct channel). Furthermore, a CSMA timer is used for controlling the backoffs in the CAP and an ACK timer for noticing a lost acknowledgment. All these events are derived from a single 62.5 kHz timer that has to be provided by the platform. This frequency corresponds to the duration of a IEEE 802.15.4 symbol duration of 16 μs and all other relevant times can be specified as multiples of the symbol duration.

4.4.3 openDSME Platform Abstraction

A platform has to specify the following three interfaces. For supported platforms these are already provided, but for porting openDSME to a new (simulation or hardware) platform, these are most relevant. The `IDSMERadio` interface provides the access to the (simulated or actual) transceiver. This includes setting the channel or starting a CCA, but also the handover of messages in both directions. Since for DSME an accurate timing, especially for the beacons, is required, the transmission of messages is a two phase process where the message is first *prepared*, i.e. written to the transceiver, and at the correct point in time, a *sendNow* instruction is issued.

The second interface is the `IDSMEMessage` that is a container for the message content. This approach allows the platform to use its own representation of messages. For hardware,

one usually wants to implement a fixed pool of fixed sized messages to avoid the use of dynamic memory, while for OMNeT++, it is possible to wrap an INET packet so there is no need to repack the message content of upper layers. All other relevant interfaces are provided by the `IDSMEPlatform`. This includes for example an interface to the timer used by the `TimerMultiplexer` or the generation of random numbers.

In the OMNeT++ port, the `DSMEPlatform` provides most of the platform specific code that also includes the interface to the upper layers and the initialization. From the point of view of OMNeT++, this class is a monolithic module that implements the `inet::IMACProtocol` interface. Since it is implemented as a single OMNeT++ module the message exchange between the different parts of openDSME are not traceable as OMNeT++ messages as it is the case for example in [KS14], but it enables an easy transition to hardware implementations.

4.5 Conclusion

The proposed medium access layer promises fully decentralized construction of traffic-aware slot schedules to be used in large-scale networks. To enable this, the slot allocation handshake of DSME is analyzed by formal methods identifying weaknesses such as the impossibility to recognize lost notifies. A method is proposed to significantly mitigate the negative influence on the schedule's consistency.

For selecting the appropriate number of slots for each link, the arising traffic is measured and an exponentially weighted moving average filter is used for smoothing. Together with a hysteresis this allows for reduced management traffic. The smoothing factor α_λ trades a lower probability of required slot (de)allocations for a lower reactivity to changing traffic demands as shown analytically. The improvements to the slot allocation handshake and the proposed slot scheduling are incorporated in the open-source implementation openDSME providing a versatile software module to be integrated in frameworks such as the OMNeT++ simulator or the Contiki OS. This will provide the basis for the evaluations in the following chapters.

Chapter 5

Discrete Event-Based Simulation

The use of a discrete event-based simulator is common for the simulation of wired and wireless communication networks. A popular choice is the OMNeT++ simulation environment [Var01] in combination with the INET framework [INET] that provides a large set of predefined components to model communication networks. In OMNeT++, the main entities are messages and modules. Messages model transmissions on the physical channel, but also communication within a node that is composed of modules such as applications and implementations of the different layers. Modules generate messages with associated timestamp and destination that will be collected in an event queue. The simulation core will then select the message with the lowest timestamp, advance the global simulation time to this timestamp and deliver the message to the destination module, possibly leading to the generation of new messages.

The advantage compared to a testbed is the possibility to represent a large number of nodes with a single computer and without the need of development and deployment of dedicated hardware. Furthermore, memory and computation constraints are less, amongst other reasons since the simulation time is decoupled from the real world time. The simulation core can skip times with no events, thus speeding up the simulation, while during other phases where extensive calculations have to be executed, the simulation time is paused without influencing the processing of future events.

The major disadvantage is the inability to accurately represent the physical layer with acceptable computational effort. This is, however, required for determining if a given transmission arrives successfully. Physical layer models such as outlined in Sect. 3.1.2 can only provide an approximation to the multitude of real-world effects and are nevertheless expensive to compute.

In this chapter, simulations of wireless networks are presented, focused on the performance of the medium access control layer. After introducing the setup, simulations of CSMA/CA are evaluated and the influence of the parameters is determined. It is followed by an evaluation of DSME by means of the openDSME implementation and a comparison with CSMA/CA.

5.1 Simulation Setup

The topology used for simulation resembles the application of a wireless network in a solar tower power plant as illustrated in Fig. 5.1. While in the real application, the position of the heliostats is optimized to provide maximum power output of the plant, it is assumed for simplicity that the heliostats are arranged in concentric circles around a central node. In the following evaluation, 4 circles of nodes are used, corresponding to 62 nodes including the sink as shown in Fig. 5.2. The distance between the circles is 130 m. According to the path loss calculations in Sect. 3.1.2, this approximately implies that communication is only possible between adjacent nodes (cf. Fig. 5.2). However, in reality there is, in principle, no fixed maximum range that prevents reception or interference of a transmission, it only gets increasingly improbable. Thus, for maximum accuracy, every message should be delivered to every node in the network and the reception probability has to be calculated individually according to the physical layer model as presented in Sect. 3.1.2.

Since this leads to a high computational effort, especially for large networks, a minimum reception power and minimum interference are specified. According to Fig. 3.2b, -3.3 dB SINR leads to a PER of 98% for a PSDU of 20 octets that is used as cut-off. Thus, for the calculated noise power of -100.44 dBm, the minimum reception power is set to -103.74 dBm. The same is set as minimum interference power. As energy detection level, -90 dBm is chosen corresponding to the properties of the ATmega256RFR2.

For simulating the traffic in the network and for determining the amount of properly delivered packets, every node generates packets to be sent to the central node. If not specified otherwise, the maximum PSDU of 127 Bytes is used. The packets are generated either with a fixed interval of I_{up} or according to a Poisson traffic model with exponentially distributed intervals with mean I_{up}. This data is then routed via GPSR with straight path node selection (see Sect. 3.4) to the central node where the received messages are counted and duplicates

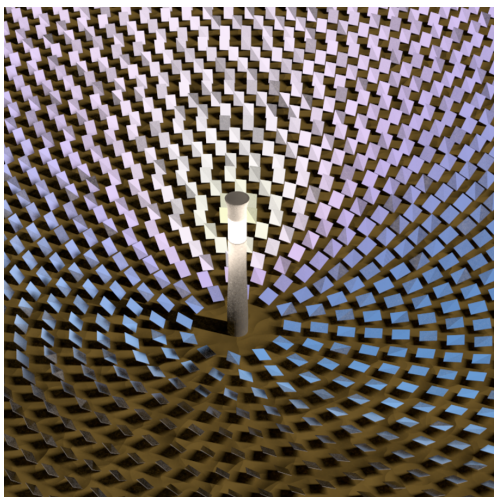


Figure 5.1: Illustration of a solar tower power plant.

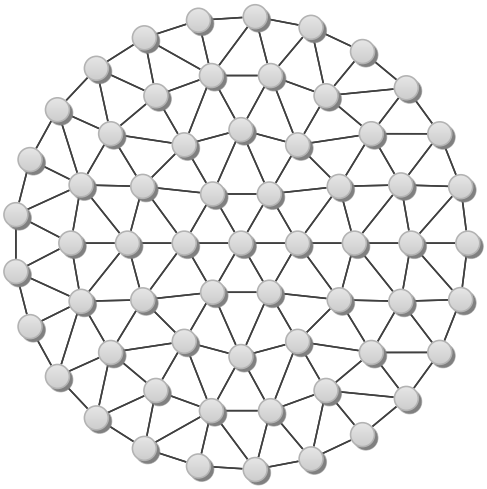


Figure 5.2: Nodes and stable links in the applied topology.

are discarded by using sequence numbers. Since we are interested in the results in the steady state and want to avoid effects originating from starting and stopping the simulation, the following procedure is applied:

1. Traffic generation starts with the simulation, but packets are marked as warm-up traffic.
2. After a setup time of 15 min, the generated packets are flagged as measurement packets.
3. When $N_p = 100$ packets are sent with the given rate, the generated packets are flagged as cool-down packets. If no measurement packet was received for 15 s, the run is stopped.

The end-to-end packet delivery ratio PDR_i for node i is calculated from the number of received packets $N_{r,i}$ at the central node as $\text{PDR}_i = \frac{N_{r,i}}{N_p}$. In the following figures, the PDR is shown as the mean over all PDR_i together with the 95% confidence intervals for this mean for 5 runs. It is depicted as shaded area or error bars in the plots.

5.2 Performance of CSMA/CA

In this section, CSMA/CA is evaluated with the previously presented simulation setup. The main goal is to optimize the parameters for maximum reliability and to get the most competitive contention-based medium access for the following evaluation of DSME.

In Fig. 5.3, the end-to-end packet delivery ratio (PDR) is shown for different per-node sending rates for Poisson traffic generation. For ${I_{up}}^{-1} = 0.2$ Hz, corresponding to $I_{up} = 5$ s, most packets are successfully delivered to the sink. For an increasing packet generation rate, corresponding to a decreasing average packet interval, the probability of successful delivery decreases. The main reason is the higher probability of collisions between transmissions. Since longer packets have a higher probability for collisions, the PDR drops faster for a

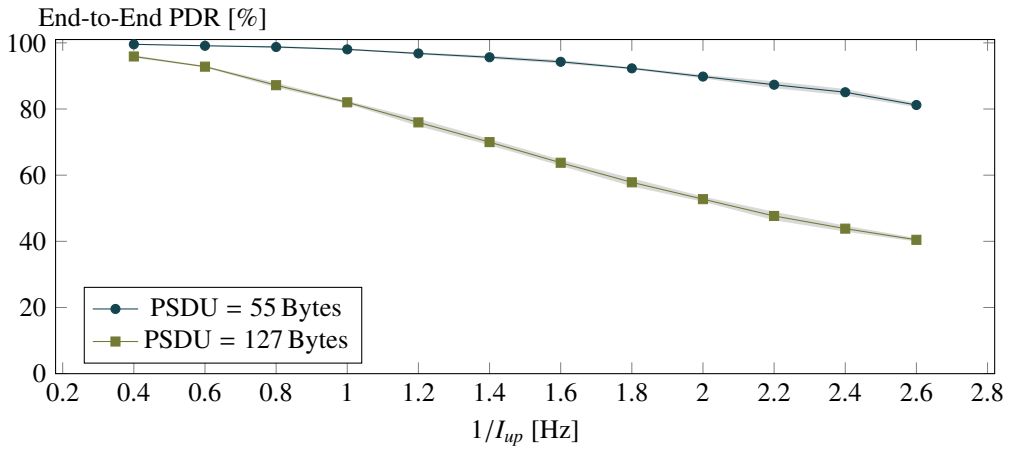


Figure 5.3: End-to-End PDR for Poisson traffic generation for the default settings according to the IEEE 802.15.4 standard ($m = 4$, $n = 3$, $m_0 = 3$, $m_b = 5$).

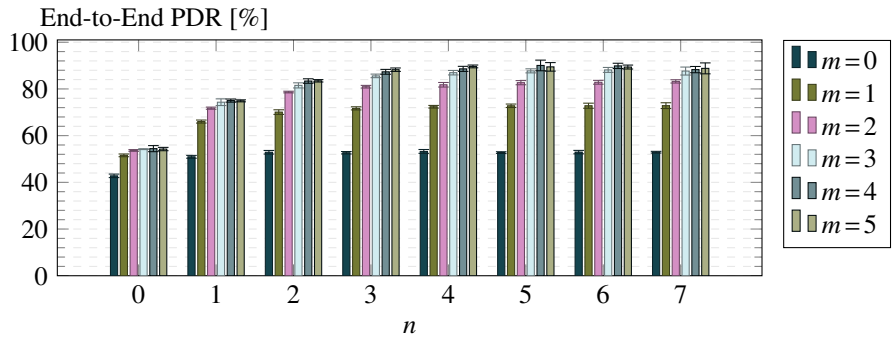


Figure 5.4: PDR for different *macMaxCsmaBackoffs* (m) and *macMaxFrameRetries* (n) with $m_0 = m_b = 7$, maximum queue length $K = 30$ and $I_{up} = \frac{1}{1.2}$ s for Poisson traffic generation.

PSDU of 127 Bytes than for 55 Bytes. An in-depth analysis of the relevant effects and their influence on the PDR can be found in Sect. 6.3, but for now we will take a more high-level approach and search for the parameters that lead to the best PDR for a packet generation rate of 1.2 Hz in the given scenario.

For this, Fig. 5.4 shows the PDR when changing *macMaxCsmaBackoffs* (m) and *macMaxFrameRetries* (n). In the given scenario, using the maximum parameters $m = 5$ and $n = 7$ results in the highest PDR. In general, retransmissions can improve the reliability, but also lead to a higher channel utilization and are less effective as generally assumed due to the simultaneous retransmission effect (see Sect. 6.3.6), therefore the gain for the higher number of retransmissions is negligible.

Fig. 5.5 depicts the corresponding plot for fixed packet generation intervals. It shows the same trend, but the results are much worse for the lower parameters of m and n. This can be explained by the fact that all nodes generate the packets at the same time so the probability of collisions with CSMA/CA is highly increased. While it is quite common for sensors in industrial applications to generate sensor readings at synchronized points in time, this should be avoided when using CSMA/CA or at least mitigated by additional artificial randomized delays.

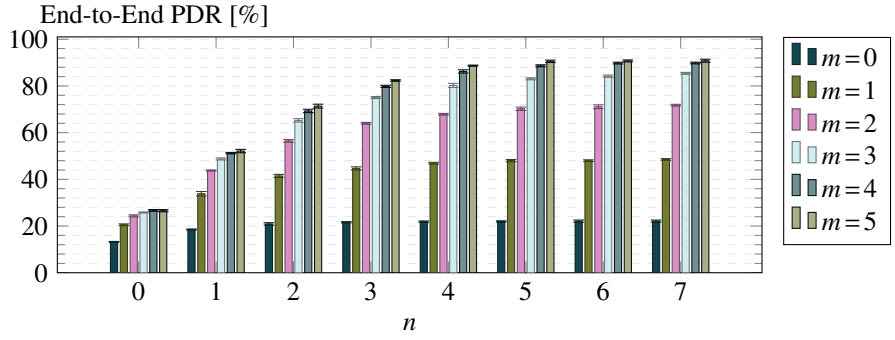


Figure 5.5: PDR for different *macMaxCsmaBackoffs* (m) and *macMaxFrameRetries* (n) with $m_0 = m_b = 7$, $K = 30$ and $I_{up} = \frac{1}{1.2}$ s for fixed interval traffic generation.

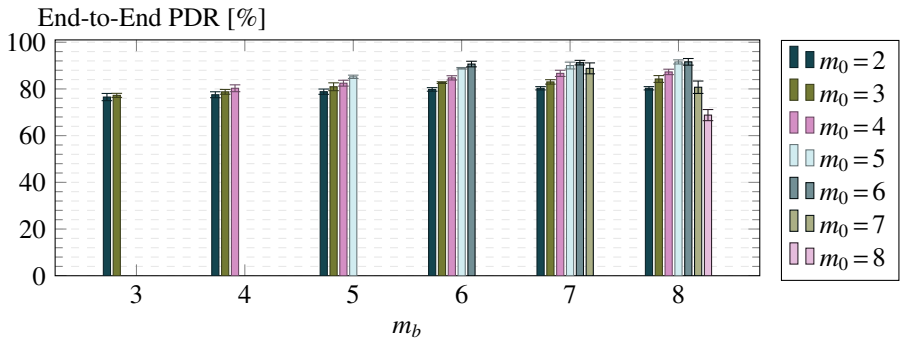


Figure 5.6: PDR for different backoff exponents *macMinBE* (m_0) and *macMaxBE* (m_b) with $m = 5$, $n = 7$, $K = 30$ and $I_{up} = \frac{1}{1.2}$ s for Poisson traffic generation.

One option for this is the use of a large backoff exponent. With a large backoff exponent, the backoff interval is randomly selected from a larger interval. Thus the transmission starts are spread over a larger time interval, too, reducing the probability of collisions at the cost of a larger transmission delay. This is shown in Fig. 5.6 and Fig. 5.7. Especially the scenario with fixed sending interval benefits a lot from larger backoff exponents. However, the PDR decreases again for very large values. In the following we choose $m_0 = m_b = 7$.

Third, the queue length K is evaluated in Fig. 5.8. Since DSME has a higher RAM overhead, the CSMA/CA stack could use a larger queue and still requires less RAM, while avoiding queue losses. In general, a longer queue increases the PDR, but only up to about $K = 14$ for the given scenario. For larger values of K, the maximum queue length is rather irrelevant, because packets are mainly dropped due to collisions and not due to queue overflow.

From this parameter study, it is concluded that the parameters $m = 5$, $n = 7$, $m_0 = m_b = 7$, $K = 30$ achieve the best reliability in the given scenario. The corresponding results are shown in Fig. 5.9. In these plots, still, the same trend of decreasing PDR for

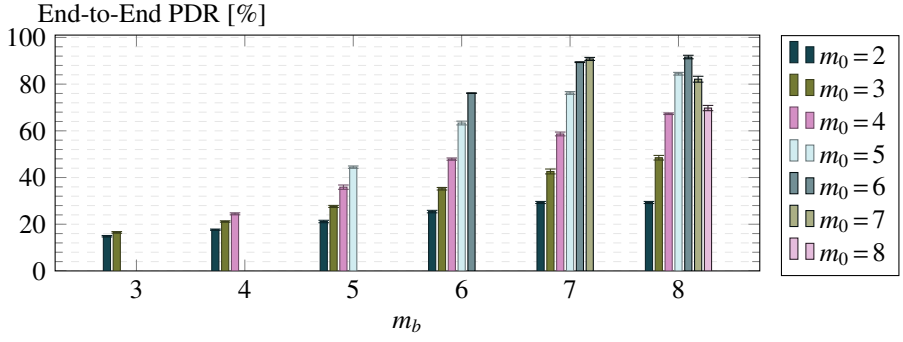
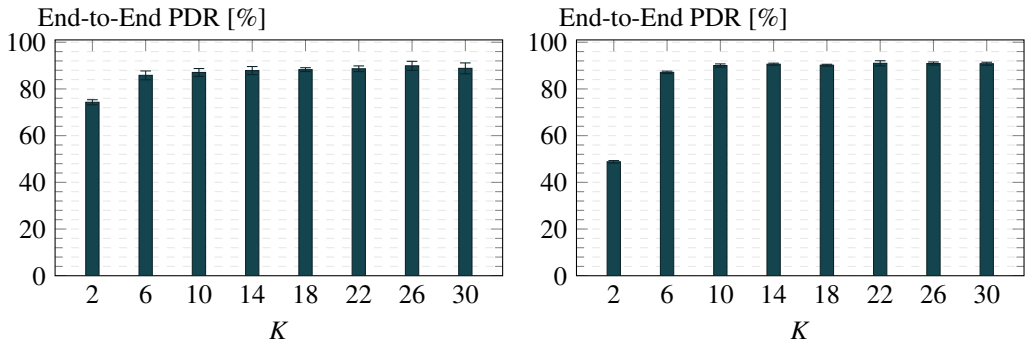


Figure 5.7: PDR for different backoff exponents *macMinBE* (m_0) and *macMaxBE* (m_b) with $m = 5$, $n = 7$, $K = 30$ and $I_{up} = \frac{1}{1.2}$ s for traffic generation with a fixed interval.

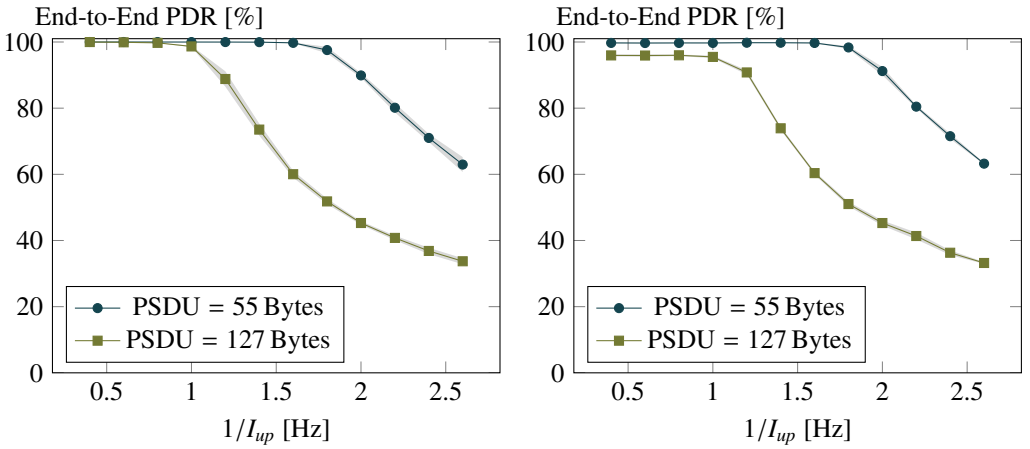


(a) Poisson traffic generation

(b) Fixed sending interval

Figure 5.8: PDR for different maximum queue lengths with $m = 5$, $n = 7$, $m_0 = m_b = 7$ and $I_{up} = \frac{1}{1.2}$ s.

higher rates is visible, but the point where reliable message transmission is possible is shifted towards higher rates. When comparing the results of the Poisson traffic with those of the fixed generation interval, we see that for a PSDU of 55 Bytes the differences are marginal, but for 127 Bytes, the PDR for the fixed sending interval never goes above 96%, demonstrating again the disadvantage of synchronized message generation when using CSMA/CA. This parameter selection may not be optimal for other scenarios, especially since they increase the congestion, delay and are not useful for volatile networks, but they are most competitive in terms of end-to-end PDR for the considered topology and traffic.



(a) Poisson traffic generation

(b) Fixed sending interval

Figure 5.9: End-to-End PDR for the determined set of parameters with maximum reliability ($m = 5$, $n = 7$, $m_0 = 7$, $m_b = 7$, $K = 30$).

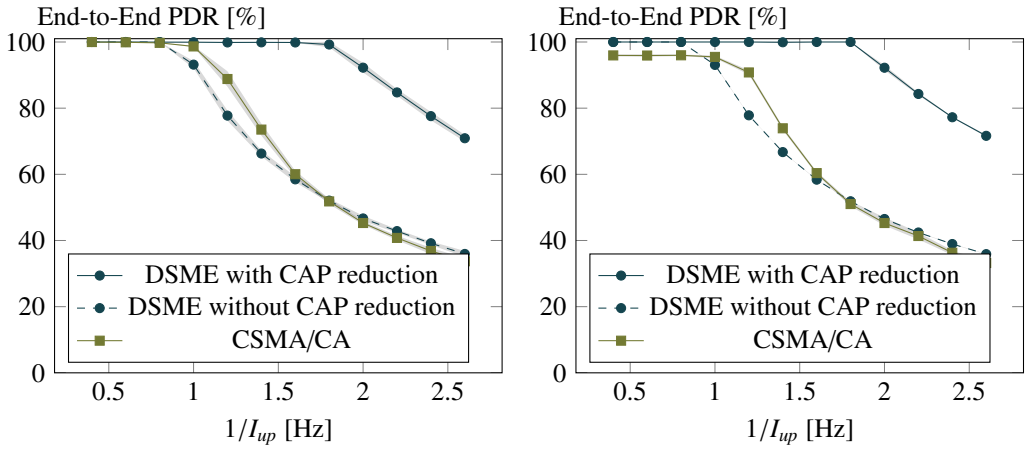
5.3 Simulative Evaluation of openDSME

In order to demonstrate the performance of the implementation of openDSME and the approach for managing slots proposed in Sect. 4.3, this section presents a simulative evaluation of openDSME in terms of reliability, end-to-end delay and generated GTS management traffic.

5.3.1 Reliability

In Fig. 5.10, the packet delivery ratios (PDR) of DSME with MO = 6 and CSMA/CA are compared. For the Poisson traffic generation in Fig. 5.10a, the traffic can successfully be delivered for a packet sending rate of 0.8 Hz or lower for all scenarios. For higher rates the PDR decreases due to the increasing amount of traffic in the network. Without CAP reduction, less than half of the time is used to transmit data packets (the other time is reserved for beacons and the CAP). Still, the performance is nearly as good as CSMA/CA that suffers from collisions for rates over 0.8 Hz. DSME with CAP reduction, however, can still deliver the packets for twice as much traffic of 1.6 Hz, where CSMA/CA only delivers about 61% of the packets. It is obvious that the higher complexity of DSME compared to CSMA/CA pays of by providing a much higher reliability. Fig. 5.10b for the fixed packet interval shows that, in contrast to CSMA/CA, the distribution has no influence on the results for DSME since packet collisions are avoided.

Fig. 5.11 demonstrates the influence of the MO parameter. Only the results for the Poisson traffic generation are shown since the differences are not significant. Without CAP reduction, the influence of the MO parameter is small as it only controls the granularity of the schedule. Therefore, the corresponding graphs are grouped closely. With CAP reduction, however, the effect is significant since it determines the overall share of the CAP



(a) Poisson Traffic

(b) Fixed Generation Interval

Figure 5.10: Comparison of the end-to-end packet delivery ratio of CSMA/CA and DSME.

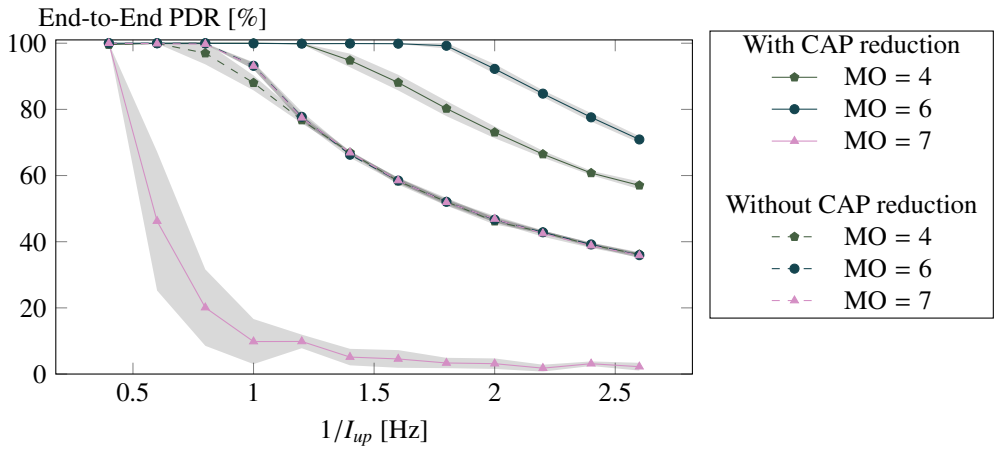


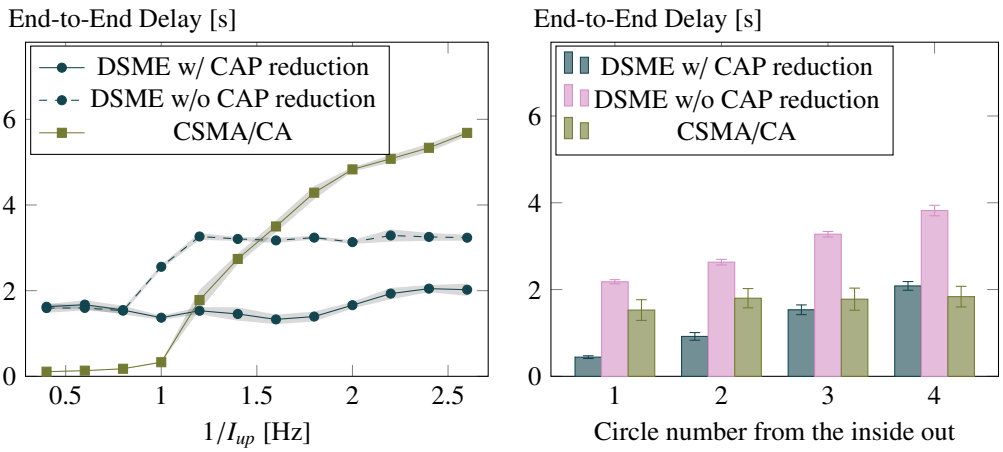
Figure 5.11: Packet delivery ratio for different DSME settings.

thus increases the number of GTS per time. By means of equations (3.8) and (3.9), the share of the CFP in a multi-superframe is calculated as approximately 69% for MO $= 4$, 88% for MO $= 6$ and 91% for MO $= 7$. Therefore, the throughput, and with it the PDR, increases with increasing MO. However, since the share of the CAP is reduced to only about 3% for MO $= 7$ and at the same time, more GTS handshakes are required to allocate the increasing number of GTS, the CAP can no longer handle the amount of management traffic for MO $= 7$. So the PDR drops early and sharply since the nodes are not longer able to acquire enough slots in time. Lowering α_λ and increasing the warm-up period can mitigate this problem at the cost of a less reactive network.

5.3.2 Delay

Many industrial applications do not only come with strict reliability constraints, but also require timely delivery of data packets. Fig. 5.12a compares the average end-to-end delay of the packet delivery. In this plot, packets that are not delivered are not included in the statistic. Clearly, DSME has a disadvantage compared to CSMA/CA for this metric at low data rates since packets are delayed until the next matching transmission slot. This waiting time is often much longer than the backoff of CSMA/CA. In particular, this is due to the scheduling algorithm that does not try to optimize the delay, so it is expected that this leaves a lot of room for improvements. For larger data rates, however, where CSMA/CA is no longer able to deliver all packets and even those that get through face a much higher delay. This is due to retransmissions and queuing so that using DSME is advantageous again. Also in DSME the queuing delay increases when the network gets saturated. Without CAP reduction, this effect can be seen at around 1 Hz. The corresponding plot for the fixed packet generation intervals, depicted in Fig. 5.13a, shows a lower delay for DSME due to the lower variance of the traffic, but the qualitative statements still hold.

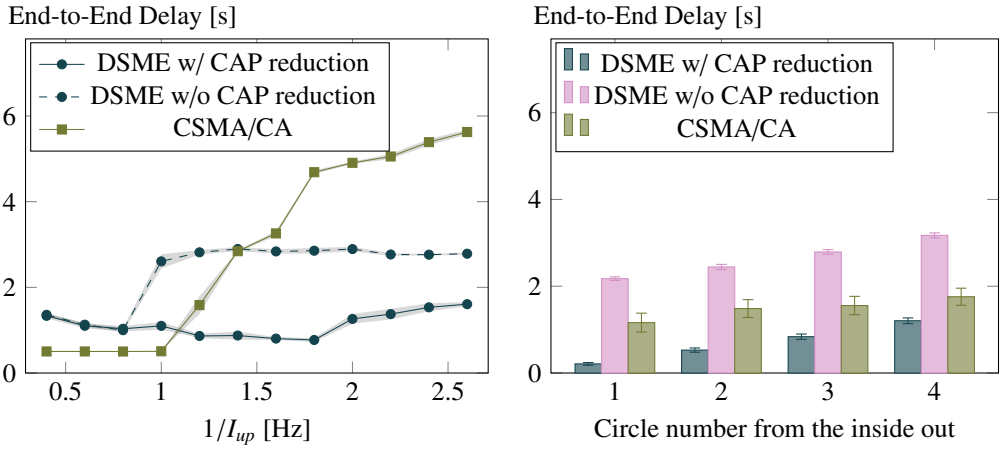
For looking at this in more detail, Fig. 5.12b and Fig. 5.13b distinguish between the different circles of the network for a rate of 1.2 packets per second. Obviously, the end-



(a) End-to-end delay over the generation rate. (b) End-to-end delay for every circle ≡ number of hops, for $I_{up} = \frac{1}{1.2}$ s.

Figure 5.12: Comparison of the average end-to-end delay of CSMA/CA and DSME for Poisson traffic generation.

to-end delay increases with the number of hops. Since the DSME network with CAP reduction is not saturated, the additional delay per hop is approximately constant. Without CAP reduction, the network is saturated in the center at this packet rate so the delay is significantly higher already for the innermost circle. This indicates that the packets fill up the queues of the nodes in the innermost circle, while the outer circles are not yet saturated.



(a) End-to-end delay over the generation rate. (b) End-to-end delay for every circle ≡ number of hops, for $I_{up} = \frac{1}{1.2}$ s.

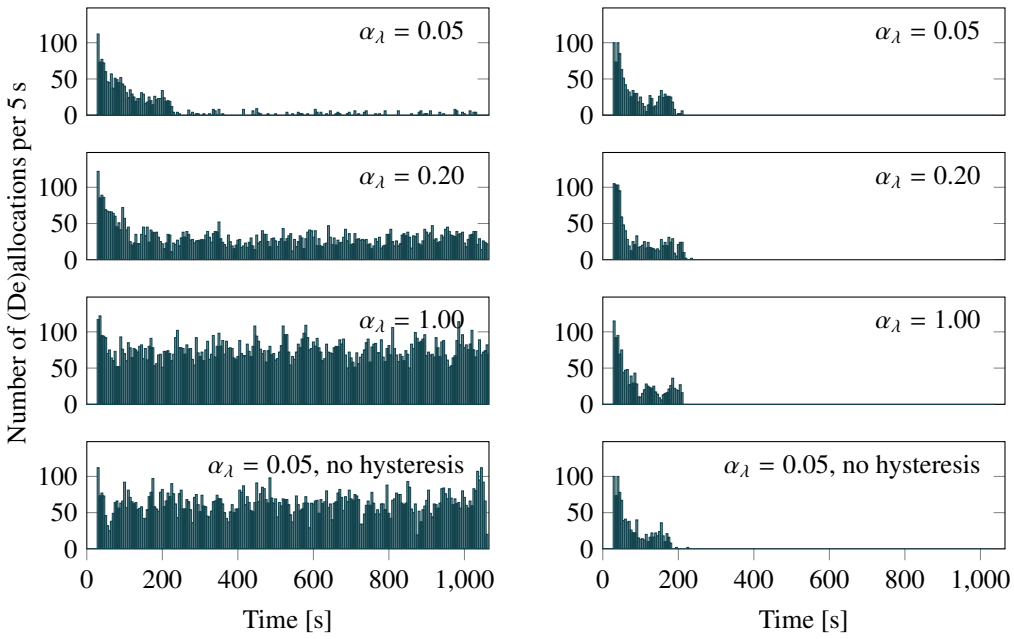
Figure 5.13: Comparison of the average end-to-end delay of CSMA/CA and DSME for fixed generation intervals.

For the saturated CSMA/CA scenario, the largest part of the delay occurs on the innermost circle, too, while the differences of the outer circles are even within the confidence intervals, indicating that the network is not saturated in the outer cycles, again.

5.3.3 GTS Management

As already mentioned before, the influence of the management traffic can not be neglected. Though it takes place in the CAP and therefore does not directly influence the transmission in the CFP, a smooth GTS management is required for a efficient and error-free schedule. If the CAP is congested, GTS handshakes can not be performed in time as required (cf. MO = 7 with CAP reduction in Fig. 5.11). Furthermore, errors during the GTS handshake can lead to an inconsistent schedule, even though the procedure tries to mitigate these (for details see Sect. 4.2).

Fig. 5.14 depicts the influence of the proposed slot management on the amount of allocations and deallocations over time for $I_{up} = 1\,\text{s}$. For every 5 s, the total number of GTS handshakes in the network is summed up. With the value of $\alpha_\lambda = 0.05$ we see an initialization phase until about 200 s in Fig. 5.14a. In this phase, the nodes associate to the network and reserve the required amount of slots along the multi-hop paths. After that only very few changes in the schedule take place as predicted in Sect. 4.3.2. With increasing α_λ, the amount of management traffic increases to account for the higher volatility of the traffic prediction. This can lead to a better adaption of the current requirements, but at the



(a) Poisson Traffic

(b) Fixed Generation Interval

Figure 5.14: Number of allocations and deallocations per 5 s over time for different settings of the proposed scheduling mechanism.

previously mentioned costs. For $\alpha_\lambda = 1$, that is without any smoothing, the management traffic does not significantly decrease over time and no stable state can be reached. The same holds for the last scenario where $\alpha_\lambda = 0.05$, but no hysteresis is applied (i.e. $c_{req,t} = \lceil \lambda_t \rceil$). Note that the first bars match for the scenario with and without hysteresis, because it is not relevant when λ_t is much smaller than p_t. For a fixed generation interval as shown in Fig. 5.14b, α_λ has a negligible influence since after the initialization phase, the traffic and thus the required number of slots over every link is constant and no further (de)allocations are required.

5.4 Conclusion

In this section, the application of the discrete event-based simulator OMNeT++ for the evaluation and comparison of different medium access techniques is presented. The results show that in fact CSMA/CA suffers from significant packet loss in the given scenario, even for optimized parameters. DSME on the other hand, can achieve a much higher reliability. If set up appropriately, that is in particular the use of the CAP reduction mode and an adequate MO setting, it can reliably handle twice the amount of traffic than CSMA/CA. Furthermore, the traffic generation pattern has no significant influence on the performance of DSME, in contrast to CSMA/CA, where synchronized packet generation can be devastating. The disadvantage of DSME is the much higher delay, especially for scenarios with low traffic, and the higher complexity.

Chapter 6

Analytical Modeling of Steady State Performance

Discrete event-based simulations as presented in the previous chapter are convenient for testing and evaluating protocols and their implementations. These simulations perform a step-by-step execution of events. In order to get representative results to make statements about their performance, a large number of events is required. This is especially true for large networks or for the analysis of rare events, leading to very long execution times.

A suitable alternative is the usage of analytical models based on probability theory. In the context of this thesis, an analytical model is a system of equations that can be solved by numerical means. Solving an adequate system of equations can often be performed much faster than the generation and processing of a sufficient number of events in a corresponding simulation. The results do not depend on the selection of random numbers and can thus be reproduced and verified more easily.

The main benefit for research, however, is the modeling process itself in parallel with the implementation of a simulation. While in a simulation, effects, such as the hidden node problem, arise implicitly, they require explicit handling by an analytical model. Discrepancies between the results of the model and the simulation therefore either indicate a bug in the simulation or an effect that is not properly represented in the analytical model. This, eventually, leads to a more trustworthy simulation and a better overall understanding.

This chapter presents two analytical models for evaluation of large wireless multi-hop networks. After presenting related work in Sect. 6.1, the considered topology and the resulting potentially disturbing transmissions are determined in Sect. 6.2. From this, an analytical model for CSMA/CA is presented in Sect. 6.3 and a model for multi-hop TDMA is developed in Sect. 6.5 that can be applied for example to TSCH or DSME. In order to generate static schedules for better comparability, Sect. 6.4 introduces a notation for valid and conflict-free schedules and three traffic-aware tree-based scheduling algorithms.

The calculations are conducted by means of the Portable, Extensible Toolkit for Scientific Computation (PETSc) [BGM97, BAA18] and the implementation of the analytical models is available as open-source software at https://github.com/koalo/AnalyticalMultiHop

DOI 10.5281/zenodo.3148187

6.1 Related Work

Mathematical modeling of wireless networks is a broadly studied topic. In [GK00], Gupta and Kumar have determined an upper bound for the throughput in a randomly generated ad-hoc network. Their model was extended to other traffic patterns and under consideration of IEEE 802.11 in [LBC01].

A major factor in the performance of a wireless network is the medium access technique. Thus, in [Bia00], Bianchi has proposed a Markov chain model for accurate representation of CSMA/CA as used in IEEE 802.11. This approach has motivated a large number of improvements, including [WPL02] that introduced a finite number of retransmissions. It was also applied to IEEE 802.15.4 in [MSM06] for single-hop networks and in [MPF12] for multi-hop networks.

Besides the development of models for CSMA/CA, a lot of research focuses on TDMA. For example, in [GNY04] and [BVD03], the effect of spatial reuse for ad-hoc networks is analyzed analytically. TSCH in particular is modeled in [GSA14], with focus on network formation and in [SPW17] an analytical model for DSME is proposed with focus on the contention access period (CAP). The latter is very similar to the analysis of conventional CSMA/CA as presented in Sect. 6.3 because it does not take guaranteed time slots into account.

A major aspect of network analysis is the influence of the queue filling levels. The fundamentals of queuing theory are well-established [CB08]. Especially the M/D/1/K model is of particular interest for TDMA networks, describing Poisson distributed arrival, deterministic service time, a single transmitter and a queue of length K. In [Seo14] closed-form expressions are given for the blocking probability and other properties. Many variants were analyzed including service times following a general distribution [Smi11]. In [KP98] a generalized queuing model for TDMA is developed that is also applicable for traffic that does not follow a Poisson distribution.

6.2 Topology and Analysis of Conflicting Transmissions

In the following, a static topology is considered with N nodes including the gateway v_0 and $N-1$ clients. In contrast to the simulation, the routing is statically precalculated. For the evaluations in this chapter, an optimal static routing tree $\mathcal{T}$ is computed by using Dijkstra's algorithm. The weight

$$-\log\left(1-\mathrm{BER}_{(v,w)}\right)+10^{-3} \tag{6.1}$$

is used to minimize the bit error along the path, but still minimize the hop count if the BER is negligible. The BER calculation is given in Sect. 3.1.2. Instead of this approach, any procedure that yields a suitable routing tree $\mathcal{T}$ can be used as basis for the computation. The tree is described by the predicate $T(p,c)$, that is true if and only if p is a parent of c in

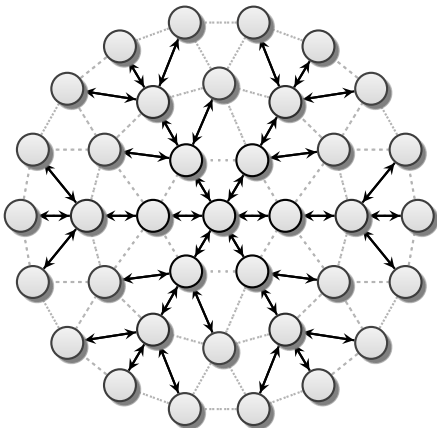


Figure 6.1: Topology with routing tree (solid) and other nodes in range (dashed).

$\mathcal{T}$, yielding the set of active links

$$\mathbf{L} := \mathbf{L}_{\text{up}} \cup \mathbf{L}_{\text{down}}, \tag{6.2}$$

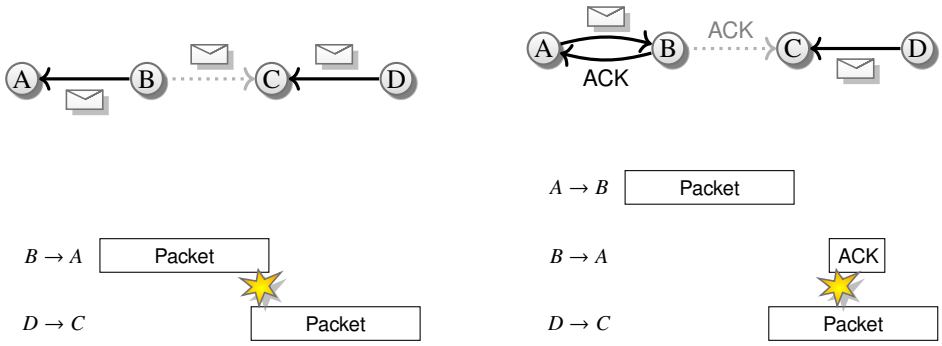
$$\mathbf{L}_{\text{up}} := \{(c,p) \mid T(p,c)\}, \quad \mathbf{L}_{\text{down}} := \{(p,c) \mid T(p,c)\}. \tag{6.3}$$

Similar to the simulation setup presented in Sect. 5.1 a minimum interference power P_{dist} is specified that is strong enough to disturb an ongoing reception at w, so

$$D(v,w) := P_{rx}(v,w) > P_{dist}. \tag{6.4}$$

As simplification, all simultaneous transmissions by nodes for which this predicate holds are assumed to prevent the reception of a packet. In fact, the successful reception also depends on the order of reception. If the receiver has synchronized to the preamble of one packet, it can, due to the channel coding, suppress signals with slightly higher signal power, while for the same reason weaker signals transmitted earlier can suppress the reception of later stronger transmissions. Also, due to the related possibility to receive packets with a negative SINR, it is not possible to recognize all potentially disturbing transmissions by measuring the power on the channel, because the energy detection level has to be larger than the noise power or otherwise carrier sensing would always indicate a busy channel. These effects are not included in the model. The resulting routing tree is depicted in Fig. 6.1 by solid lines, while together with the dashed lines, they depict the predicate $D(v,w)$.

In order to represent packets (sent along a link) as well as acknowledgments (sent in the reverse direction), even under the hidden node problem, an enhanced conflict graph is introduced that considers all constellations of senders (in the following: sends a packet, but receives an acknowledgment) and receivers (receives a packet, but sends an acknowledgment). The significance of this approach is illustrated in Fig. 6.2. In Fig. 6.2a a classical hidden node constellation is depicted where node D is not able to recognize the transmission from B to A and thus the transmissions collide at C. Fig. 6.2b shows how the same situation can occur when transmitting acknowledgments. Even though the transmission from D to C is not disturbed by the transmission from A to B, the later transmitted acknowledgment by B collides at C.



(a) Collision of two packets

(b) Collision of a packet and an ACK

Figure 6.2: Illustration of two different hidden node constellations.

Many other constellations are possible as we will see in Sect. 6.3.5. For this and also for defining conflict-free schedules in the TDMA model, the following basic constellations are defined. For a link $(v_1, w_1) \in \mathbf{L}$ the possibly disturbing links $(v_2, w_2) \in \mathbf{L}$ are depicted in Fig. 6.3 and formulated as

$$\mathcal{S}_{\mathcal{S},(v_1,w_1)} = \{(v_2, w_2) \in \mathbf{L} \,|\, D(v_1, v_2) \wedge v_1 \neq v_2\}\,, \tag{6.5}$$

that is the sender of one link receives transmissions from the sender of another link,

$$\mathcal{R}_{\mathcal{S},(v_1,w_1)} = \{(v_2, w_2) \in \mathbf{L} \,|\, D(w_1, v_2) \wedge v_1 \neq v_2\}\,, \tag{6.6}$$

that is the receiver of one link receives transmissions from the sender of another link,

$$\mathcal{S}_{\mathcal{R},(v_1,w_1)} = \{(v_2, w_2) \in \mathbf{L} \,|\, D(v_1, w_2) \wedge v_1 \neq v_2\}\,, \tag{6.7}$$

that is the sender of one link receives transmissions from the receiver of another link,

$$\mathcal{R}_{\mathcal{R},(v_1,w_1)} = \{(v_2, w_2) \in \mathbf{L} \,|\, D(w_1, w_2) \wedge v_1 \neq v_2\}\,, \tag{6.8}$$

that is the receiver of one link receives transmissions from the receiver of another link.

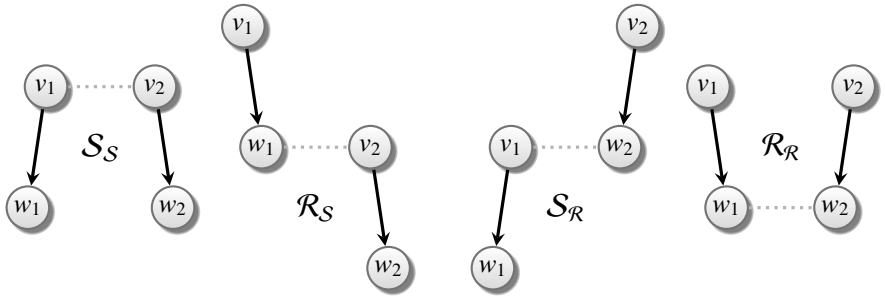


Figure 6.3: The four basic relations for conflicts of two links.

6.3 CSMA/CA Model*

This section proposes an analytical model for the CSMA/CA medium access technique of the IEEE 802.15.4 wireless standard [802.15.4-2015]. The starting point is the model presented by Di Marco et al. [MPF12]. This model is considerably enhanced as well as optimized to allow the evaluation of large scale networks. While the model in [MPF12] includes multi-hop traffic to a gateway (upstream), the proposed model also considers traffic from the gateway to the nodes (downstream) by taking the topology of the network into account. In addition, it contains a more sophisticated modeling of the conflict graph for representing collisions in consideration of hidden nodes. This provides the possibility to consider the influence of acknowledgments to the transmission probability with low computational effort. The results imply a much higher significance of this effect as supposed before. Furthermore, previous models assume that the probability of packet loss falls exponentially with the number of retransmissions. This is a considerable underestimation as also shown in [ZL07] by simulation and this section proposes an explanation based on simultaneous retransmissions by multiple nodes.

Fig. 6.4 shows the different modules of the analytical model and how they are linked. The topology is given as an input to the model and describes how the nodes are linked and how they route the traffic. This information is used for calculating how the actual traffic is distributed within the network (Sect. 6.3.2) as well as for determining which transmissions might lead to busy channels and packet collisions in the neighborhood model (Sect. 6.3.5).

The traffic distribution yields the probability q_l that a node has a pending transmission to send over link l. Though, it does not directly send the packet, but uses a CSMA/CA technique with retransmissions to increase the success probability. This is modeled in Sect. 6.3.3 and yields the probability τ_l that it actually senses the channel to start the transmission. For this, it also depends on information from the neighborhood model, such as the probability α_l that the channel is sensed free and the probability $P_{\text{noACK},l}$ that no acknowledgment arrives and a retransmission should be started.

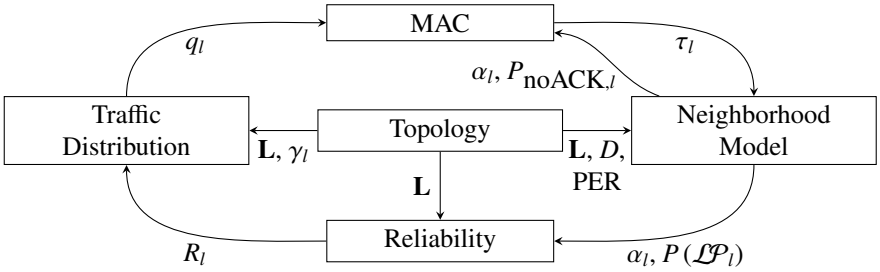


Figure 6.4: Relationships between the different parts of the model.

*A shorter version of this section was previously published as "F. Meier and V. Turau, *An Analytical Model for Fast and Verifiable Assessment of Large Scale Wireless Mesh Networks* in Proceedings of the 11th International Conference on the Design of Reliable Communication Networks (DRCN), Kansas City, MO, USA, March 2015. https://doi.org/10.1109/DRCN.2015.7149011" [MT15].

The information from the neighborhood model and the topology is also used to calculate the final reliability of a link R_l, that is the probability that a packet will eventually be transmitted over the link and not dropped because of a busy channel or repeated collisions. R_l is again used to calculate the traffic distribution.

This closes a circle, so it is evident that the equations of the outer modules are interlinked. They built up a non-linear equation system that can be computed iteratively by numerical means. The topology, and thereby also the analog model, are not in the loop, so they can be calculated offline before the actual computation as covered in Sect. 6.2.

6.3.1 Traffic Generation

A Poisson traffic generation model is applied, so the intervals between two packet generations are exponentially distributed. The mean interval for traffic from a node to the gateway (upstream) is denoted as I_{up}, while for each client, the gateway generates packets with mean interval I_{down}. All times are defined as multiples of

$$S_b := aUnitBackoffPeriod \cdot \text{Symbol duration} = 20 \cdot 16\,\mu s. \tag{6.9}$$

According to Sect. 3.1.1, 6 octets are transmitted in addition to the PSDU, so taking into account 4 bits per symbol for the O-QPSK PHY, a transmission of a packet takes

$$T_p = \frac{8 \cdot (L_{\text{PSDU}} + 6) \cdot 16\,\mu s}{4 \cdot S_b} = \frac{L_{\text{PSDU}} + 6}{10} \tag{6.10}$$

time units. An acknowledgment has a fixed PSDU of 5 octets, so it takes $T_{ack} = \frac{11}{10}$ time units. Together with the specified supplementary values of the long interframe spacing $IFS = 40$ and the time to wait before sending the ACK $t_{ack} = 12$, an acknowledged transmission takes

$$T_{succ} = T_p + T_{ack} + \frac{IFS + t_{ack}}{20} = T_p + T_{ack} + 2.6 \tag{6.11}$$

time units. The number of symbols $t_{m,ack}$ to wait for a lost acknowledgment (*macAckWaitDuration*) is calculated according to the standard

$$\begin{aligned} t_{m,ack} &= aUnitBackoffPeriod \\ &\quad + aTurnaroundTime + phySHRDuration \\ &\quad + \lceil 6 \cdot phySymbolsPerOctet \rceil \\ &= 20 + 12 + 10 + 6 \cdot 2 = 54. \end{aligned} \tag{6.12}$$

This gives a total time for an unsuccessful transmission of

$$T_{fail} = T_p + \frac{t_{m,ack}}{20} = T_p + 2.7. \tag{6.13}$$

time units. In relation to S_b, the generation rate on each client in upstream direction is

$$g_{\text{up}} = \frac{S_b}{I_{up}} \tag{6.14}$$

and since the gateway v_0 generates packets for each client its packet generation rate is

$$g_{\text{down}} = \frac{(N-1) \cdot S_b}{I_{down}}. \tag{6.15}$$

6.3.2 Traffic Distribution

The distribution of the traffic is directly calculated from the tree. λ_l denotes the packet sending rate of the sender v of a link $l = (v, w) \in \mathbf{L}$, including generated and forwarded traffic. At the receiver side μ_l is the effective packet rate to be forwarded by w from v. Furthermore, γ_n denotes the number of proper descendants of a nodes as shown in Fig. 6.5. Since we are not interested in the actual content of the messages, but in overall performance measures, and the generated traffic is equally distributed over the nodes, it is not necessary to model the flow of a single message through the network, but instead express the traffic load as fraction of the incoming messages.

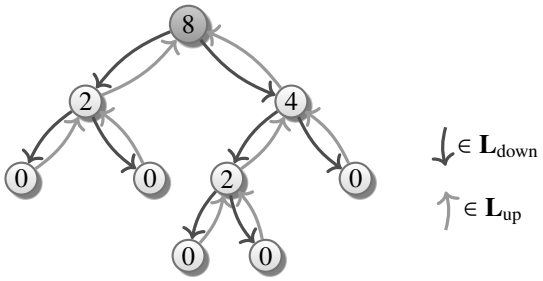
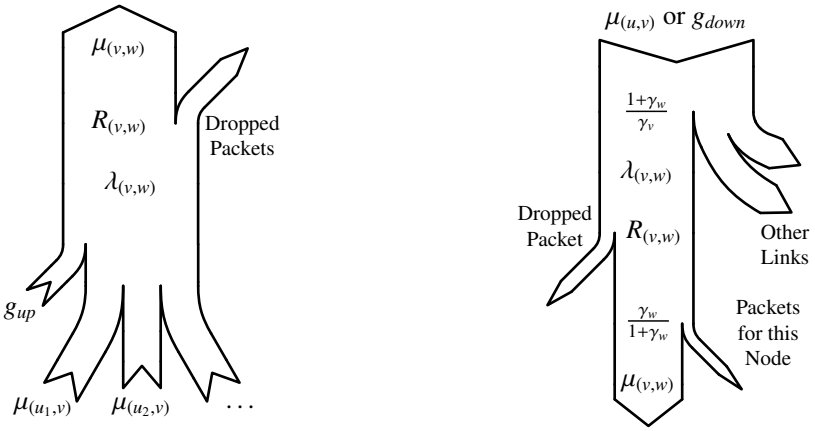


Figure 6.5: An example for the set of active links and the values of γ_n in the nodes.

Fig. 6.6 depicts the possible inflows and drains of traffic. For upstream, $\lambda_{(v,w)}$ is the sum of the traffic of all proper descendants and the traffic generated in this node. Some packets will be dropped with a probability of $R_{(v,w)}$, because it was not possible to transmit them successfully.

For downstream, the gateway is the only node that autonomously generates packets with rate g_{down}. The respective receivers v of a link (u, v) transmit traffic to be forwarded to their proper descendants with rate $\mu_{(u,v)}$. This traffic is split up between the outgoing links. Each link (v, w) going out of v gets that fraction of the traffic $\lambda_{(v,w)}$ that corresponds to the number of nodes reachable via this link $(1 + \gamma_w)$ in relation to the total number of nodes reachable via v, namely γ_v. The second difference to upstream is the fact that each node will consume the fraction of the traffic intended for this node. Altogether, this results in the



(a) Upstream (b) Downstream

Figure 6.6: The traffic flow of a link.

recursive expressions

$$\lambda_{(v,w)} = \begin{cases} \frac{1+\gamma_w}{\gamma_v} \cdot g_{\text{down}} & v = v_0 \wedge (v,w) \in \mathbf{L}_{\text{down}} \\ \frac{1+\gamma_w}{\gamma_v} \cdot \mu_{(u,v)}\, T(u,v) & \wedge (v,w) \in \mathbf{L}_{\text{down}} \\ g_{\text{up}} + \sum_{(u,v) \in \mathbf{L}_{\text{up}}} \mu_{(u,v)} & (v,w) \in \mathbf{L}_{\text{up}} \end{cases}, \tag{6.16}$$

$$\mu_{(v,w)} = \lambda_{(v,w)} \cdot R_{(v,w)} \cdot \begin{cases} \frac{\gamma_w}{1+\gamma_w} & (v,w) \in \mathbf{L}_{\text{down}} \\ 1 & (v,w) \in \mathbf{L}_{\text{up}} \end{cases}. \tag{6.17}$$

Since the probability of a pending packet is modeled as Poisson distributed, the probability of at least one pending packet in a given time interval of length S_b is given by

$$q_l = 1 - \frac{\lambda_{n,i}^{0}}{0!} e^{-\lambda_l} = 1 - e^{-\lambda_l}. \tag{6.18}$$

The model neglects the probability that two or more packets are pending at the same time at the same node and handles them as a single packet, so it will underestimate the traffic for very short transmission intervals.

6.3.3 Model for IEEE 802.15.4 CSMA/CA

The CSMA/CA MAC layer is modeled by a Markov chain on every link. This deviates from the reality, where usually only a single CSMA/CA state machine exists for a node and not for every link. This approach, however, allows to model collisions more accurately since, for example, the probability of retransmissions also depends on the receiver and its neighborhood. The Markov chain is illustrated in Fig. 6.7 and consists of the following states.

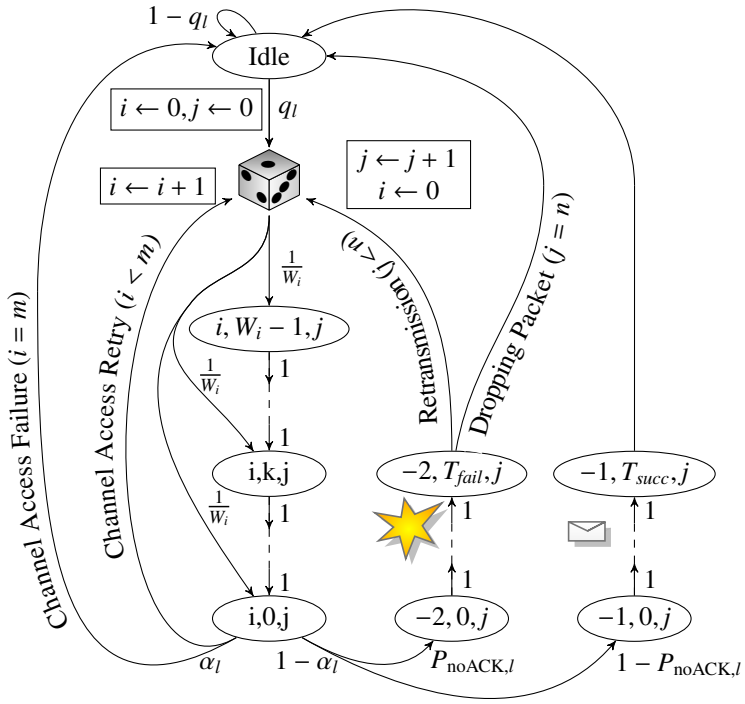


Figure 6.7: States and transitions of the presented CSMA/CA model.

θ_{idle} No transmission pending.

$\theta_{i,k,j}$ In the i^{th} backoff stage after j transmission attempts. CCA in k steps.

$\theta_{-1,q,j}$ During a successful transmission after q time steps and j preceding transmission attempts.

$\theta_{-2,q,j}$ During a colliding transmission after q time steps and j preceding transmission attempts.

$P(\xi \to \zeta)$ is the probability of going from ξ to ζ in one step. The probability that the sender will leave the idle state within one time unit is given by q_l as calculated in the previous section. Therefore, the probability of staying in the idle state is

$$P(\theta_{idle} \to \theta_{idle}) = 1 - q_l. \tag{6.19}$$

Otherwise, the MAC layer starts the CSMA/CA procedure as described in Sect. 3.2.1. The maximal backoff time span $W_i - 1$ increases with the number $i \in [0, m]$ of failed channel accesses, with

$$W_i = \begin{cases} 2^{m_0} & i = 0 \\ 2^i \cdot W_0 & 0 < i \leq \overline{m} = m_b - m_0 \\ 2^{\overline{m}} \cdot W_0 & i > \overline{m} \end{cases}. \tag{6.20}$$

The die is symbolic for the associated choice. It is not a state in itself, but just bundles the different ways to go into the backoff period. The probability of choosing a specific state is W_i, therefore

$$P\left(\theta_{idle} \rightarrow \theta_{0,k,0}\right) = \frac{q_l}{W_0}, \qquad 0 \leq k \leq W_0 - 1. \tag{6.21}$$

Within one time unit, the backoff duration is decreased by one

$$P\left(\theta_{i,k+1,j} \rightarrow \theta_{i,k,j}\right) = 1, \qquad 0 \leq i \leq m \wedge 0 \leq k < W_i - 1 \wedge 0 \leq j \leq n. \tag{6.22}$$

As soon as the state $\theta_{i,0,j}$ is reached, channel sensing takes place. If the channel is sensed busy with probability $1 - \alpha_l$ as calculated in Sect. 6.3.5, the channel access retry counter i is incremented and the backoff procedure is started again

$$P\left(\theta_{i-1,0,j} \rightarrow \theta_{i,k,j}\right) = \frac{\alpha_l}{W_i}, \qquad 0 < i \leq m \wedge 0 \leq k \leq W_0 - 1 \wedge 0 \leq j \leq n. \tag{6.23}$$

If i has reached m, the packet will be dropped

$$P\left(\theta_{m,0,j} \rightarrow \theta_{idle}\right) = \alpha_l, \qquad 0 \leq j \leq n. \tag{6.24}$$

Otherwise, if the channel is sensed free, the transmission takes place. If the packet or the associated acknowledgment sent out by the receiver will eventually collide, $\theta_{-2,0,j}$ is chosen as next state

$$P\left(\theta_{i,0,j} \rightarrow \theta_{-2,0,j}\right) = (1 - \alpha_l) \cdot P_{\text{noACK},l}, \qquad 0 \leq i \leq m \wedge 0 \leq j \leq n. \tag{6.25}$$

Of course, the real hardware does not know if the transmission will collide, but in the analytical model the probability $P_{\text{noACK},l}$ can be computed from the collision graph as presented in the next section. In case of a upcoming successful transmission, the $\theta_{-1,0,j}$ state is chosen

$$P\left(\theta_{i,0,j} \rightarrow \theta_{-1,0,j}\right) = (1 - \alpha_l) \cdot \left(1 - P_{\text{noACK},l}\right), \qquad 0 \leq i \leq m \wedge 0 \leq j \leq n. \tag{6.26}$$

During the transmission, the counter h is increased until the time is over

$$P\left(\theta_{-1,h,j} \rightarrow \theta_{-1,h+1,j}\right) = 1, 0 \leq h < T_{fail} \wedge 0 \leq j \leq n, \tag{6.27}$$

$$P\left(\theta_{-2,h,j} \rightarrow \theta_{-2,h+1,j}\right) = 1, 0 \leq h < T_{succ} \wedge 0 \leq j \leq n. \tag{6.28}$$

After a collision, the retransmission counter $j \in [0, n]$, with n being the maximum number of retries *macMaxFrameRetries*, is incremented, the channel access retry counter i is reset to zero and a new backoff period starts

$$P\left(\theta_{-2,T_{fail},j-1} \rightarrow \theta_{0,k,j}\right) = \frac{1}{W_0}, \qquad 0 < j \leq n. \tag{6.29}$$

Finally, the idle state is reached after a successful transmission or as soon as the maximum number of retransmissions is reached and the packet is dropped

$$P\left(\theta_{-1,T_{succ},j} \to \theta_{idle}\right) = 1, \qquad 0 \leq j \leq n, \tag{6.30}$$

$$P\left(\theta_{-2,T_{fail},n} \to \theta_{idle}\right) = 1. \tag{6.31}$$

All transition probabilities not specified before are zero. The stationary distribution of the Markov chain yields the probability that a given link is in a specific state as derived in Appendix A.2.1. In particular, the probability of being in a generic sensing state is given by

$$\tau_l = b_{0,0,0} \frac{1-{\alpha_l}^{m+1}}{1-\alpha_l} \frac{1-{y_l}^{n+1}}{1-y_l}, \tag{6.32}$$

with the shortcut $y_l := P_{\text{noACK},l} \cdot \left(1-{\alpha_l}^{m+1}\right)$ and $b_{0,0,0}$, the probability of being in the first channel sensing state, given by

$$\begin{aligned}\frac{1}{b_{0,0,0}} = \frac{1}{2}\Bigg(& W_0 \frac{1-(2\cdot\alpha_l)^{\min(m,\overline{m})+1}}{1-2\cdot\alpha_l} + \frac{1-{\alpha_l}^{\min(m,\overline{m})+1}}{1-\alpha_l} \\ & + \frac{\left(2^{m_b}+1\right){\alpha_l}^{\overline{m}+1}\left(1-{\alpha_l}^{\max(0,m-\overline{m})}\right)}{1-\alpha_l}\Bigg) \frac{1-{y_l}^{n+1}}{1-y_l} \\ & + \left(1-{\alpha_l}^{m+1}\right)\frac{1-{y_l}^{n+1}}{1-y_l}\left(T_{succ}\cdot\left(1-P_{\text{noACK},l}\right) + T_{fail}\cdot P_{\text{noACK},l}\right) \\ & + \frac{1}{q_l}\left({y_l}^{n+1} + \frac{1-{y_l}^{n+1}}{1-y_l}\left({\alpha_l}^{m+1} + \frac{1-P_{\text{noACK},l}}{\left(1-{\alpha_l}^{m+1}\right)^{-1}}\right)\right).\end{aligned} \tag{6.33}$$

6.3.4 Simultaneous Transmissions

The probability that the sender of a link $j \in L$ does not start a transmission in a given time unit is given by

$$\tau_j\alpha_j + \left(1-\tau_j\right). \tag{6.34}$$

with τ_j, the probability that the sending node of j is trying to access the channel and α_j, the probability that the channel would be sensed busy (see Sect. 6.3.3). So either the sender tries to access the channel, but senses it busy $\left(\tau_j\alpha_j\right)$ or it does not try to access the channel at all $\left(1-\tau_j\right)$. For a given set of links $\mathbf{S} \subset \mathbf{L}$, the probability that at least one $j \in \mathbf{S}$ starts a transmission in a given time unit is therefore given by the complementary event of the event that all nodes are not sending

$$Q(1,\mathbf{S}) = 1 - \prod_{j\in\mathbf{S}} \left(\tau_j\alpha_j + \left(1-\tau_j\right)\right). \tag{6.35}$$

The same expression can be derived from the expression $\Pr[\mathcal{A}_l]$ in [MPF12] as conducted in Appendix A.2.2, but is less complex and takes much less computation time. For multiple time units, so $t \geq 1$, the probability that a transmission is started during any of these is given by

$$Q(t, \mathbf{S}) = 1 - (1 - Q(1, \mathbf{S}))^t . \tag{6.36}$$

6.3.5 Analyzing the Neighborhood

The formula derived in the previous section can be used to calculate collisions that occur if at least two transmissions arrive at the same receiver at the same time. In [MPF12], only collisions between transmissions of packets, but not acknowledgments are considered, assuming the acknowledgments are too short to influence the reception probability. In the following, the influence of adjacent links on the interference probability with CSMA/CA under consideration of acknowledgments is analyzed using the sets of potentially disturbing links determined in Sect. 6.2 on page 78.

For this, $\mathcal{CP}_l$ denotes the event that a packet collides on a link l, while $\mathcal{CA}_l$ denotes the event that an acknowledgment collides. The constellation that the sender v_2 of the link $j = (v_2, w_2)$ is in the range of both the sender v_1 and the receiver w_1 of a link $l = (v_1, w_1)$ can be expressed as

$$j \in \mathcal{R}_{\mathcal{S},l} \cap \mathcal{S}_{\mathcal{S},l}. \tag{6.37}$$

In this constellation, both senders will wait for each other if they sense an ongoing transmission. Although, a sender might sense within the turnaround time of the other one and vice versa and they therefore start transmitting within an interval of 2 time units. So the probability of a collision of two packets on links that obey this constellation is expressed as

$$P(\mathcal{CP}_{l,0}) = Q(2, \mathcal{R}_{\mathcal{S},l} \cap \mathcal{S}_{\mathcal{S},l}) . \tag{6.38}$$

If receiver w_1 might receive a packet from sender v_2, but the transmission of v_2 is not recognized by v_1, the channel sensing has basically no effect. This is called the hidden node problem. Two transmissions of length T_p might overlap in an interval of $2T_p$. This is calculated as

$$P(\mathcal{CP}_{l,1}) = Q\big(2T_p, \mathcal{R}_{\mathcal{S},l} \setminus \mathcal{S}_{\mathcal{S},l}\big) . \tag{6.39}$$

The transmission can not only collide with another packet, but also with an acknowledgment. If all involved nodes are in range, the new packet might collide with an acknowledgment if the carrier sensing takes place between the arrival of the original packet and the beginning of the acknowledgment

$$P(\mathcal{CP}_{l,2}) = Q(1, \mathcal{S}_{\mathcal{S},l} \cap \mathcal{S}_{\mathcal{R},l} \cap \mathcal{R}_{\mathcal{R},l}) . \tag{6.40}$$

If the senders can not hear each other, the sensing might also take place at the end of the packet,

$$P\left(\mathcal{CP}_{l,3}\right) = Q(2, (\mathcal{S}_{\mathcal{R},l} \cap \mathcal{R}_{\mathcal{R},l}) \setminus \mathcal{S}_{\mathcal{S},l}) \,. \tag{6.41}$$

If it were to occur earlier, the transmission would collide with the packet, so it should not be counted here, too. If the acknowledgment can not be recognized, the packet transmission might take place during the whole acknowledgment

$$P\left(\mathcal{CP}_{l,4}\right) = Q(T_{ack}, (\mathcal{S}_{\mathcal{S},l} \cap \mathcal{R}_{\mathcal{R},l}) \setminus \mathcal{S}_{\mathcal{R},l}) \,. \tag{6.42}$$

This case is extended in the following constellation by the sensing before the pause (cf. $P\left(\mathcal{CP}_{l,3}\right)$)

$$P\left(\mathcal{CP}_{l,5}\right) = Q(T_{ack}+1, (\mathcal{R}_{\mathcal{S},l} \cap \mathcal{R}_{\mathcal{R},l}) \setminus \mathcal{S}_{\mathcal{S},l} \setminus \mathcal{S}_{\mathcal{R},l}) \,. \tag{6.43}$$

Finally, the packet might overlap with an acknowledgment during the whole transmission, if the sensing of both senders is not effective

$$P\left(\mathcal{CP}_{l,6}\right) = Q\big(T_p + T_{ack}, \mathcal{R}_{\mathcal{R},l} \setminus \mathcal{S}_{\mathcal{S},l} \setminus \mathcal{S}_{\mathcal{R},l} \setminus \mathcal{R}_{\mathcal{S},l}\big) \,. \tag{6.44}$$

In addition, the acknowledgment itself might be affected by collision with a packet

$$P\left(\mathcal{CA}_{l,0}\right) = Q(1, \mathcal{S}_{\mathcal{S},l} \cap \mathcal{R}_{\mathcal{S},l}) \,, \tag{6.45}$$

and in particular if the acknowledgment can not be recognized

$$P\left(\mathcal{CA}_{l,1}\right) = Q(T_{ack}, \mathcal{S}_{\mathcal{S},l} \setminus \mathcal{R}_{\mathcal{S},l}) \,. \tag{6.46}$$

The probability that a packet collides with at least one other packet or acknowledgment is

$$P\left(\mathcal{CP}_l\right) = P\left(\bigcup_{i=0}^{6} \mathcal{CP}_{l,i}\right) . \tag{6.47}$$

Note that the events are not mutually exclusive, so inclusion-exclusion principle has to be applied for the calculation. Furthermore, a transmission might be dropped because of a bad link with probability $\mathrm{PER}_{L_{\mathrm{PSDU}},l}$ as given in (3.4). Taking this into account, the overall probability that a packet is not successfully received is

$$P\left(\mathcal{LP}_l\right) = P\left(\mathcal{CP}_l\right) + \left(1 - P\left(\mathcal{CP}_l\right)\right) \cdot \mathrm{PER}_{L_{\mathrm{PSDU}},l}. \tag{6.48}$$

The combined probability of a lost acknowledgment is

$$P\left(\mathcal{CA}_l\right) = P\left(\bigcup_{i=0}^{1} \mathcal{CA}_{l,i}\right) . \tag{6.49}$$

Taking a possible bad link into account, the probability of losing the acknowledgment is

$$P\left(\mathcal{LA}_l\right) = P\left(\mathcal{CA}_l\right) + \left(1 - P\left(\mathcal{CA}_l\right)\right) \cdot \mathrm{PER}_{5,l}. \tag{6.50}$$

since the PSDU for an ACK is 5 Bytes. Finally, the overall probability that no acknowledgment arrives at the sender is

$$P_{\mathrm{noACK},l} = P\left(\mathcal{LP}_l\right) + \left(1 - P\left(\mathcal{LP}_l\right)\right) \cdot P\left(\mathcal{LA}_l\right). \tag{6.51}$$

The probability α_l that the channel is sensed busy is calculated along the same line as the probability that either a packet transmission or an acknowledgment arrives at the sender

$$\alpha_{pkt,l\,pkt,l} = Q\left(T_p, \mathcal{S}_{\mathcal{S},l}\right), \tag{6.52}$$

$$\alpha_{ack,l\,ack,l} = Q\left(T_{ack}, \mathcal{S}_{\mathcal{R},l}\right), \tag{6.53}$$

$$\alpha_l = \alpha_{pkt,l\,pkt,l} + \alpha_{ack,l\,ack,l} - \alpha_{pkt,l\,pkt,l} \cdot \alpha_{ack,l\,ack,l}. \tag{6.54}$$

Note that all sets can be computed offline, so the actual computation is linear with the number of neighbors.

6.3.6 Simultaneous Retransmission Effect

If the collision probability is assumed to be independent of the current transmission attempt, the packet arrival probability is given by

$$\begin{aligned} R_{l,indep} = 1 &- \left(P\left(\mathcal{LP}_l\right)\left(1 - {\alpha_l}^{m+1}\right)\right)^{n+1} \\ &- {\alpha_l}^{m+1} \sum_{j=0}^{n} \left(P\left(\mathcal{LP}_l\right)\left(1 - {\alpha_l}^{m+1}\right)\right)^{j}, \end{aligned} \tag{6.55}$$

according to [MPF12]. After comparison with the simulation (cf. Fig. 6.8), it turned out that the reliability of packet transmission is not independent of the current retransmission attempt and this effect has a major impact on the results. Therefore, the model contains an elaborate handling of this matter.

The probability of a mutual disturbance of at least two hidden nodes is calculated as

$$P\left(\mathcal{CB}_{l,2}\right) = Q\left(2T_p + 2, \left(\mathcal{R}_{\mathcal{S},l} \cap \mathcal{S}_{\mathcal{R},l}\right) \setminus \mathcal{S}_{\mathcal{S},l}\right). \tag{6.56}$$

After such a mutual disturbance took place, those nodes will issue a retransmission. The backoff exponent is reset to the initial backoff counter m_0, so there are $W_0 = 2^{m_0}$ possible backoff time spans. The event that the retransmission collides again, given i other nodes with pending retransmissions is denoted as ${\mathcal{CR}_{l,2}}^{i}$. For two nodes the probability is

$$\begin{aligned} \omega &= \min\left(W_0 - T_p - 1, 0\right) \\ P\left({\mathcal{CR}_{l,2}}^{1}\right) &= 1 - \sum_{h=1}^{\omega} \frac{2h}{{W_0}^2} = 1 - \frac{\omega + \omega^2}{{W_0}^2}. \end{aligned} \tag{6.57}$$

Of course, this expression is only a lower bound of the actual probability that there will be a repeated collision. For example, there might be several initial collisions at the same time leading to a higher probability of a repeated collision. It can be calculated by summing over all possible combinations of mutual disturbance. However, the price for this gain of accuracy is too high considering the exponential growth in computational complexity, in particular for large scale networks. Analogous to (6.38), even senders which can mutually sense their transmissions might be affected by packet collision. The probability that at least two senders which can mutually sense their transmissions are affected by mutual disturbance is

$$P\left(\mathcal{CB}_{l,1}\right) = Q(2, \mathcal{R}_{\mathcal{S},l} \cap \mathcal{S}_{\mathcal{R},l} \cap \mathcal{S}_{\mathcal{S},l}) \,. \tag{6.58}$$

${\mathcal{CR}_{l,1}}^{j}$ is defined analogous to ${\mathcal{CR}_{l,2}}^{i}$ and the corresponding probability of a repeated collision of two nodes is calculated as

$$P\left({\mathcal{CR}_{l,1}}^{1}\right) = {W_0}^{-1}. \tag{6.59}$$

With these quantities, an absorbing Markov chain is built with the states

- κ_{succ} Successful transmission
- κ_{cf} Channel access failure
- $\kappa_{0,0}$ No preceding mutual disturbance
- $\kappa_{1,0}$ Hidden node(s) with pending retransmission
- $\kappa_{0,1}$ Visible node(s) with pending retransmission
- $\kappa_{1,1}$ Hidden node(s) and visible node(s) with pending retransmission

In the following, all unspecified transition probabilities as well as $P\left({\mathcal{CR}_{l,2}}^{0}\right)$ and $P\left({\mathcal{CR}_{l,1}}^{0}\right)$ are zero. κ_{succ} and κ_{cf} are the absorbing final states, so

$$P\left(\kappa_{succ} \rightarrow \kappa_{succ}\right) = 1, \tag{6.60}$$

$$P\left(\kappa_{cf} \rightarrow \kappa_{cf}\right) = 1. \tag{6.61}$$

The probability of a channel access failure is always the same

$$P\left(\kappa_{p,q} \rightarrow \kappa_{cf}\right) = \alpha_l^{m+1}. \tag{6.62}$$

All following transitions need to take this into account with

$$\beta_l = 1 - \alpha_l^{m+1}. \tag{6.63}$$

The probability of a successful transmission is given by the probability that neither a conventional collision takes place with $P\left(\mathcal{LP}_l\right)$ nor a repeated collision

$$P\left(\kappa_{p,q} \rightarrow \kappa_{succ}\right) = \beta_l \cdot \left(1 - P\left(\mathcal{LP}_l \cup {\mathcal{CR}_{l,2}}^{p} \cup {\mathcal{CR}_{l,1}}^{q}\right)\right). \tag{6.64}$$

Note that $P(\mathcal{LP}_l)$ is used instead of $P_{\text{noACK},l}$, because even if the acknowledgment does not arrive at the sender, the packet itself might be transmitted successfully. The transmission to $\kappa_{0,0}$ takes place if no repeated collision takes place, but the packet collides anyway with another transmission

$$P(\kappa_{p,q} \to \kappa_{0,0}) = \beta_l \cdot (P(\mathcal{LP}_l) - P(\mathcal{CB}_{l,2} \cup \mathcal{CB}_{l,1})) \cdot (1 - P(\mathcal{CR}_{l,2}{}^p \cup \mathcal{CR}_{l,1}{}^q)). \tag{6.65}$$

All remaining state transitions indicate that any repeated collision takes place

$$P(\kappa_{p,q} \to \kappa_{1,0}) = \beta_l \cdot P(\mathcal{CB}_{l,2} \cup \mathcal{CR}_{l,2}{}^p) \cdot (1 - P(\mathcal{CB}_{l,1} \cup \mathcal{CR}_{l,1}{}^q)), \tag{6.66}$$

$$P(\kappa_{p,q} \to \kappa_{0,1}) = \beta_l \cdot (1 - P(\mathcal{CB}_{l,2} \cup \mathcal{CR}_{l,2}{}^p)) \cdot P(\mathcal{CB}_{l,1} \cup \mathcal{CR}_{l,1}{}^q), \tag{6.67}$$

$$P(\kappa_{p,q} \to \kappa_{1,1}) = \beta_l \cdot P(\mathcal{CB}_{l,2} \cup \mathcal{CR}_{l,2}{}^p) \cdot P(\mathcal{CB}_{l,1} \cup \mathcal{CR}_{l,1}{}^q). \tag{6.68}$$

The probability R_l that a packet was successfully transmitted is described by the probability of reaching κ_{succ} from $\kappa_{0,0}$ in at most $n+1$ steps, calculated by the corresponding power of the transition matrix.

6.3.7 Path Reliability

The end-to-end reliability in upstream direction $\text{PDR}_{\text{up},n}$ is the probability that a packet sent by a node n arrives at the gateway. It is given by the product of the reliabilities along the routing path

$$\text{PDR}_{\text{up},c} = \begin{cases} \text{PDR}_{\text{up},p} \cdot R_{(c,p)} & c \neq v_0 \wedge (c,p) \in \mathbf{L}_{\text{up}} \\ 1 & c = v_0 \end{cases}. \tag{6.69}$$

By construction, there is always exactly one p with $(c,p) \in \mathbf{L}_{\text{up}}$ unless c is the gateway. The reliability in downstream direction $\text{PDR}_{\text{down},n}$, that is the probability that a packet sent by the gateway arrives at its destination, is given correspondingly by multiplication of the reliabilities in downstream direction

$$\text{PDR}_{\text{down},c} = \begin{cases} \text{PDR}_{\text{down},p} \cdot R_{(p,c)} & c \neq v_0 \wedge (p,c) \in \mathbf{L}_{\text{down}} \\ 1 & c = v_0 \end{cases}. \tag{6.70}$$

6.3.8 Network Throughput

For upstream traffic, the throughput of the network is given by the amount of traffic arriving at the sink v_0 per time and is calculated as

$$\frac{1}{S_b} \cdot \sum_{(u,v) \in \mathbf{L}_{\text{up}}} \begin{cases} \lambda_{(u,v)} \cdot R_{(u,v)} & v = v_0 \\ 0 & \text{otherwise} \end{cases}. \tag{6.71}$$

6.3.9 Evaluation

For evaluating the analytical model it is implemented by means of the PETSc suite [BAA18, BGM97]. As base point the following settings are used. A topology of concentric circles with $N = 19$, a PSDU of 60 Bytes, $I_{up} = 250\,\text{ms}$, $P_{tx} = 3.5\,\text{dBm}$, $P_N = -100.44\,\text{dBm}$, $P_{dist} = -100\,\text{dBm}$, $m = 4$, $n = 3$, $m_0 = 3$ and $m_b = 5$. The results are averaged over all nodes in the network. The results are compared to an OMNeT++ simulation as introduced in Sect. 5.1. In order to apply the same physical model as presented in Sect. 6.2, the minimum SINR is set to 0 dB in the simulation and correspondingly, the energy detection level, minimum reception power and minimum interference power are all set to $-100\,\text{dBm}$. Also, artificial internal processing delays are disabled for the simulation. In order to get insights about the probability distribution of the results, the simulation has to be repeated several times with different seeds and thus, the plots for the simulation show the 95% confidence interval for the mean of 5 runs. The model always yields the same result for the same input, so no repetitions are required.

Fig. 6.8 shows the resulting end-to-end packet delivery ratio for different *macMaxFrameRetries* settings for upstream traffic. They are compared with the simulation and a version of the model that assumes a collision probability that is independent of the retransmission attempt. While all resulting PDRs increase with increasing n, it is apparent that the simulation and the improved model result in much lower PDRs. This demonstrates the significance of the simultaneous retransmission effect. The effectiveness of retransmissions is not as high as commonly expected for IEEE 802.15.4. The hidden node problem and the low initial backoff that is used after each failed transmission lead to a high probability of repeated collisions of the same pair of packets.

In Fig. 6.9 the gain of the improved neighborhood modeling is apparent. The PDR for upstream traffic decreases with increasing PSDU for all setups, but when considering ACK collisions in the model or the simulation, the resulting PDRs are lower. Thus, even acknowledgments that are short in comparison to the data packets have a considerable influence on the probability of collisions.

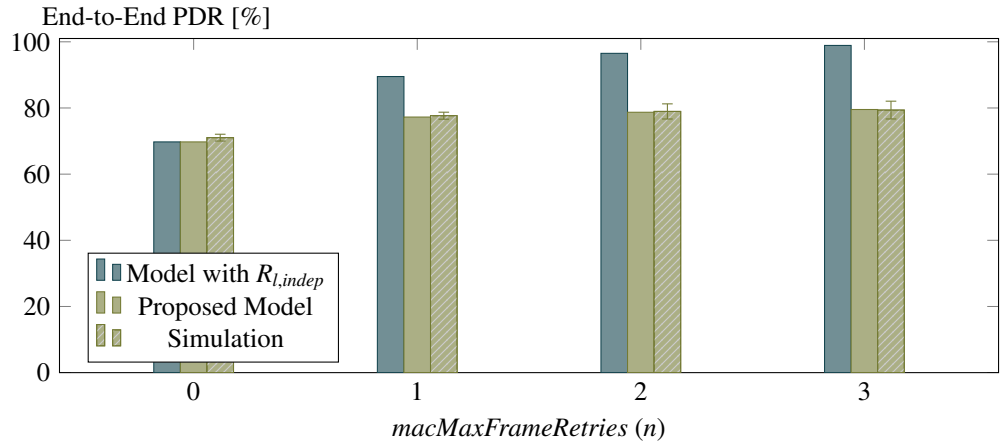


Figure 6.8: Comparison of the PDR assuming independent collision probabilities for the retransmission attempts with the proposed model and the simulation.

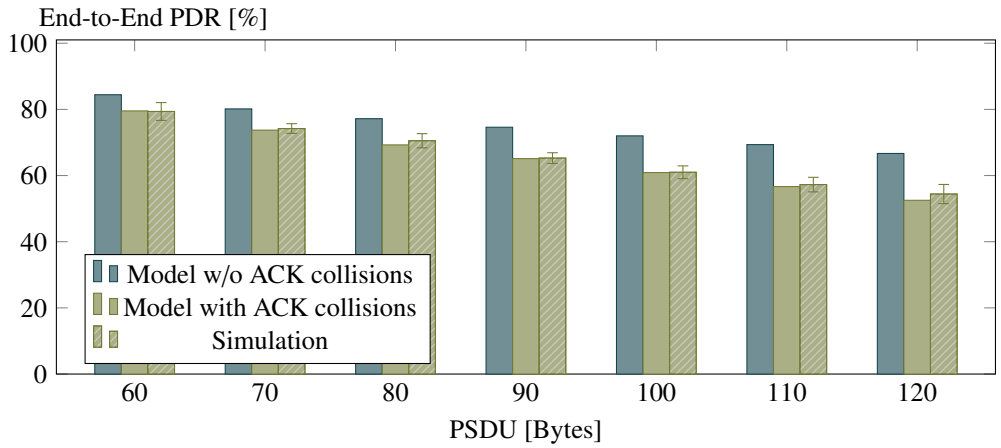


Figure 6.9: Comparison of the PDRs with and without the consideration of ACK collisions for different PSDUs.

The influence of the packet generation rate on the PDR is shown in Fig. 6.10a. The reliability clearly increases with longer intervals, while upstream traffic is significantly less reliable than downstream traffic, because the generated packets agglomerate towards the center. In contrast to the upstream traffic for which model and simulation match quite good, the PDR for the simulation drops significantly for downstream traffic at 9 Hz. This can be explained by the fact that the model does not consider queue drops. Since all packets are generated by v_0, the packets can not be sent out fast enough for high rates. This is demonstrated in Fig. 6.10b that shows the ratio of the total number of queue drops during the simulation at v_0 in relation to the total number of sent packets.

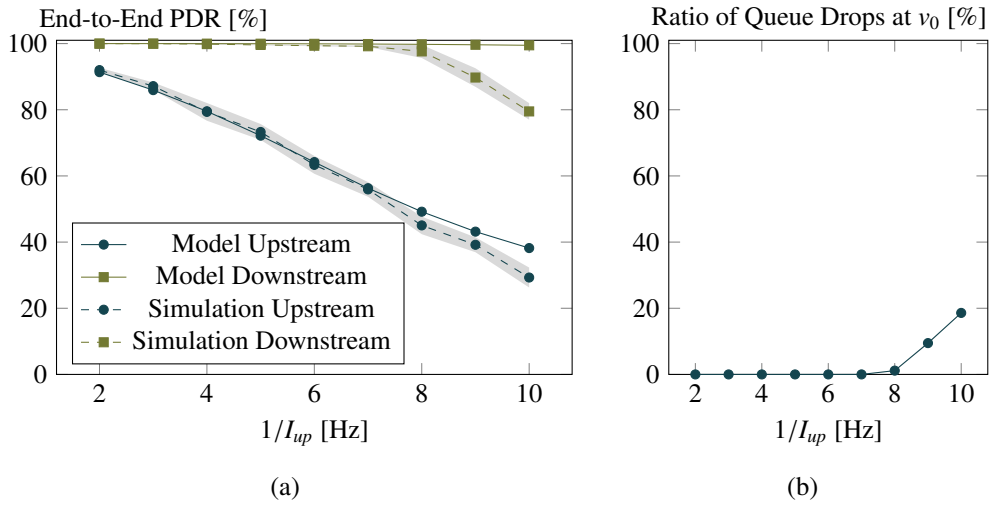
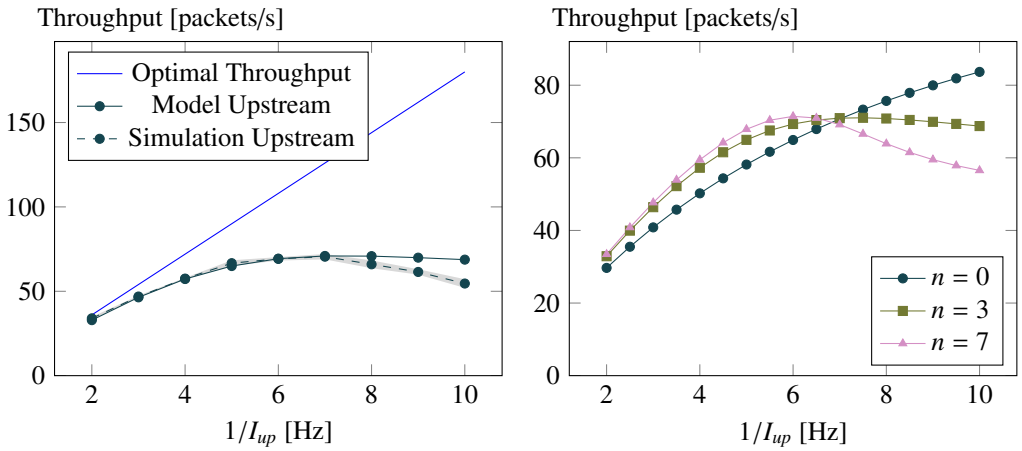


Figure 6.10: PDR for increasing packet generation rate for up- and downstream.



(a) Calculated throughput compared to the optimal throughput and the simulation.

(b) Calculated throughput for different *macMaxFrameRetries*.

Figure 6.11: Average number of packets arriving at the sink per second.

But also the results for upstream traffic deviate for rates larger than 7 Hz. This is also apparent when plotting the throughput as shown in Fig. 6.11a. The optimal throughput is given by $(N-1) \cdot {I_{up}}^{-1}$ and represents the throughput if no packets are lost. At a rate of 7 Hz the throughput has a maximum and then drops for both model and simulation, but faster in the simulation. This drop indicates an overload situation in the network and is in accordance with the simulative results in [PRM06, Fig. 8]. For further analysis, Fig. 6.11b shows how the throughput course depends on the maximum number of retransmissions. While retransmissions help to increase the reliability for lower rates, the higher number of packets in the network lead to congestion and thus the maximum throughput reached at a lower rate. It can be concluded that using a high number of retransmissions is not sufficient

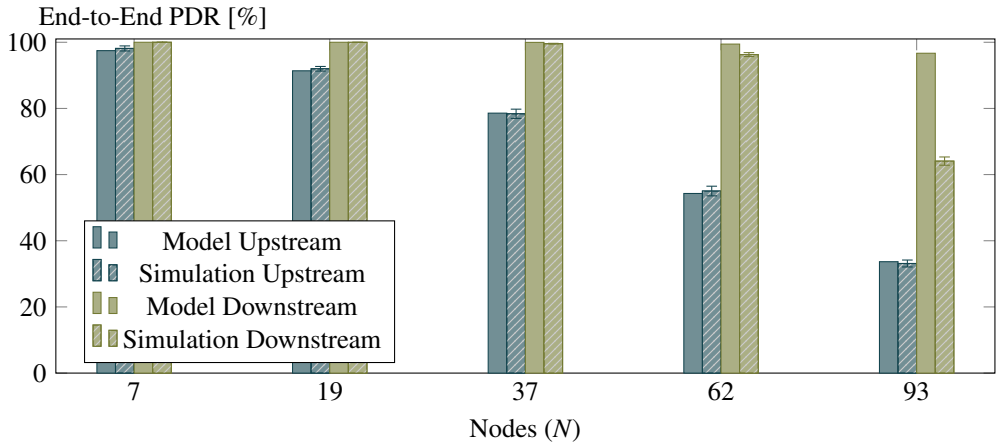


Figure 6.12: Reliability decreases with increasing network size and downstream is more reliable than upstream. $I_{up} = I_{down} = 500$ ms.

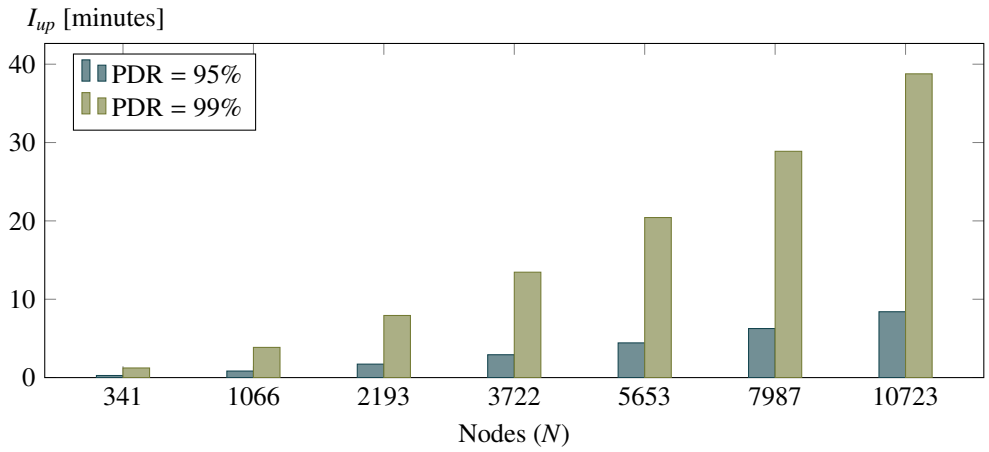


Figure 6.13: The minimum packet generation interval to maintain a given end-to-end PDR in the presented setup.

for building very reliable networks, but other techniques such as time-slotted medium access should be considered, too.

Finally, we will have a look at the scalability of CSMA/CA. In Fig. 6.12, the PDR is shown for an increasing number N of nodes in the network. Clearly, the PDR decreases with increasing N, while for downstream traffic the same effect can be seen as before. The question about the scalability can also be stated the other way around like "How many packets can be sent per time, while maintaining a given reliability?" and with a simulation such a question can only be answered by iterating over large sets of parameters. An analytical model can often be solved for several variables, yielding the result in a single step of computation. In this way an analytical model can even support simulations by giving adequate initial parameters for subsequent simulation runs. This additional advantage of an analytical model is demonstrated in Fig. 6.13 that shows the minimum sending interval for given network sizes. Apparently, for very large networks, long sending intervals are required for maintaining high reliability.

6.4 Token-Based Slot Scheduling*

In this section, several distributed algorithms for building schedules are presented that are based on the principle of token-passing, extending the work in [TW07]. These approaches are based on an existing tree in the network that is traversed in a depth-first search and one node after another builds its slot schedule before passing the control on to the next node, hence the name. In contrast to centralized approaches, the overhead is reduced since there is no need to collect the information about the network state on at a central entity, but a stable schedule is achieved after a finite number of steps, assuming all messages arrive

*The first two schedules were previously published in "F. Kauer and V. Turau, *An Analytical Model for Wireless Mesh Networks with Collision-Free TDMA and Finite Queues*, EURASIP Journal on Wireless Communications and Networking 2018:149, June 2018. https://doi.org/10.1186/s13638-018-1146-x" [KT18a].

correctly and in order. This makes this approach very suitable for analytical analysis and to demonstrate the advantage of traffic-aware schedules. As hybrid scheduling algorithms they are, however, not directly applicable to large-scale real-world applications due to their inability to cope with changes in the network topology and the traffic load without a complete recalculation as outlined in Sect. 3.3.9. After the introduction of the notation and considered traffic distribution, two traffic-aware schedules for TSCH are introduced and one that is also suitable for DSME with its less flexible superframe structure.

6.4.1 Valid and Conflict-Free Schedules

In this section, the notation used in the following is introduced. A network of N nodes $v_0, \ldots, v_{N-1}$ with a fixed schedule is considered. All slots have an equal time duration T_s. For TSCH it is usually 10 ms and for DSME it depends on the superframe order and is calculated according to (3.5) on page 27. A slotframe consists of l_S slots and is repeated with an interval of $T_s \cdot l_S$. The schedule for node v_n is given as a tuple of transmission slots $\mathcal{T}_n = (t_{n,j})_{j=0,\ldots,|\mathcal{T}_n|-1}$, sorted in ascending order, where $t_{n,j} \in \mathcal{T}_n$ if and only if the slot with the zero-based index $t_{n,j}$, that is $0 \leq t_{n,j} < l_S$, is assigned to n for conflict-free transmission. Correspondingly, $\mathcal{R}_n = (r_{n,j})_{j=0,\ldots,|\mathcal{R}_n|-1}$ describes the reception slots. A slot is never in both $\mathcal{T}_n$ and $\mathcal{R}_n$, because a node can not transmit and receive at the same time. Furthermore, $\mathcal{K}_n(i)$ gives the respective reception node if $i \in \mathcal{T}_n$ or the respective transmission node if $i \in \mathcal{R}_n$. Fig. 6.14 presents a simple example for these definitions.

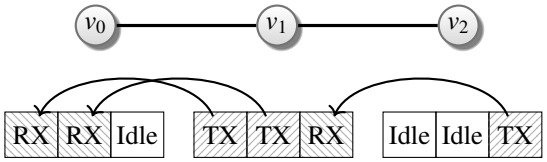


Figure 6.14: An exemplary schedule corresponding to $N = 3$, $l_S = 3$, $\mathcal{T}_0 = ()$, $\mathcal{R}_0 = (0,1)$, $\mathcal{K}_0(0) = 1$, $\mathcal{K}_0(1) = 1$, $\mathcal{T}_1 = (0,1)$, $\mathcal{R}_1 = (2)$, $\mathcal{K}_1(0) = 0$, $\mathcal{K}_1(1) = 0$, $\mathcal{K}_1(2) = 2$, $\mathcal{T}_2 = (2)$, $\mathcal{R}_2 = ()$ and $\mathcal{K}_2(2) = 1$.

For modeling frequency diversity $\mathcal{C}_{v_n}(i)$ is introduced. It specifies the channel to use in slot i by node v_n. The channel is an element of the set of channels $\mathcal{C}$. For IEEE 802.15.4 in the 2.4 GHz band, it consists of the numbers 11 to 26.

To check that a schedule is conflict-free and valid, the set

$$\mathbf{L}_i = \{(n, \mathcal{K}_n(i)) \,|\, i \in \mathcal{T}_n, \forall n\} \tag{6.72}$$

of all links that are active during slot i has to be considered for all slots. Each node can only be transmitter *or* receiver during a slot

$$(i \in \mathcal{T}_n \Rightarrow i \notin \mathcal{R}_n) \wedge (i \in \mathcal{R}_n \Rightarrow i \notin \mathcal{T}_n)\,. \tag{6.73}$$

Algorithm 1 Determine proper descendants

```
1: initialization  mark all children as unvisited ∀ v_n
2: on message FORWARD at v_n from v_s
3:     v_n.γ ← 0
4:     HANDLENODE(v_n)
5: on message BACKTRACK(j) at v_n from v_s
6:     v_n.γ_s ← j { Store proper descendants of children }
7:     v_n.γ ← v_n.γ + j
8:     HANDLENODE(v_n)
9: procedure HANDLENODE(v_n)
10:    if ∃ unvisited child v_u then
11:        mark v_u as visited
12:        send FORWARD to v_u
13:    else if v_n ≠ v_0 then { leaf or sub tree fully handled }
14:        send BACKTRACK(v_n.γ + 1) to v_{p_n}
```

Furthermore, the links have to be unique and consistent, $\forall i \in \mathcal{T}_n$ it must hold

$$i \in \mathcal{R}_{\mathcal{K}_n(i)} \wedge (\mathcal{K}_n(i) = k \Leftrightarrow \mathcal{K}_k(i) = n)\,. \tag{6.74}$$

However, this alone is insufficient when taking collisions of both data packets and acknowledgments into account as already introduced in Sect. 6.2. According to that section, with $D(v,w)$ denoting that a transmission of v might disturb a reception at w, the potentially disturbing links for a link $(v_1, w_1) \in \mathbf{L}_i$ are

$$\begin{aligned}\boldsymbol{\mathcal{D}}_{i,(v_1,w_1)} = &\{(v_2,w_2) \in \mathbf{L}_i \,|\, D(v_1,v_2) \wedge v_1 \neq v_2\} \\ &\cup \{(v_2,w_2) \in \mathbf{L}_i \,|\, D(w_1,v_2) \wedge v_1 \neq v_2\} \\ &\cup \{(v_2,w_2) \in \mathbf{L}_i \,|\, D(v_1,w_2) \wedge v_1 \neq v_2\} \\ &\cup \{(v_2,w_2) \in \mathbf{L}_i \,|\, D(w_1,w_2) \wedge v_1 \neq v_2\}\,.\end{aligned} \tag{6.75}$$

One possibility to ensure a conflict-free schedule is to have

$$\boldsymbol{\mathcal{D}}_{i,\cup} = \bigcup_{l \in \mathbf{L}_i} \boldsymbol{\mathcal{D}}_{i,l} \tag{6.76}$$

empty for all slots. Hence, no concurrent transmissions take place in the neighborhood. By exploiting frequency diversity it is, however, sufficient to make sure that no concurrent transmissions take place on the same frequency channel at the same time, i.e. one has to make sure that for all slots $0 \leq i < l_S$ holds

$$\mathcal{C}_{v_1}(i) \neq \mathcal{C}_{v_2}(i), \forall (v_2,w_2) \in \boldsymbol{\mathcal{D}}_{i,(v_1,w_1)}. \tag{6.77}$$

6.4.2 Traffic Distribution

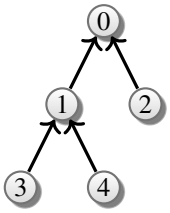
It is assumed in the following that every node generates the same amount of traffic that shall be routed to the root v_0 of the tree. The main idea is that every node has enough slots for the traffic generated at that node as well as for forwarding the traffic of its children, denoted as $\mathbf{C}_n$. More formally, if γ_n denotes the number of proper descendants of node v_n, that is the number of nodes whose traffic is routed through this node, every node will get $\gamma_n + 1$ transmission slots towards the sink. Algorithm 1 describes a distributed algorithm to calculate γ_n for every node and also to make every node aware of the number of proper descendants of their children. For this, every node keeps the variables $v_n.\gamma$ for the number of proper descendants of this node n and for each child s its number of proper descendants $v_n.\gamma_s$. The index of the parent of a node v_n is denoted as p_n.

The depth-first search is started by sending FORWARD to v_0. After initializing $v_0.\gamma$, FORWARD messages are sent until a leaf is reached. After this a BACKTRACK message is sent back to the parent. The parent increments its counters before continuing the depth-first search. After a subtree is fully handled, the number of nodes in this subtree is sent back to the parent in a BACKTRACK message.

6.4.3 Traffic-Aware Schedule Single-Channel (TASC)

The first traffic-aware scheduling algorithm as shown in Algorithm 2 is based on the scheduling algorithm Type III from [TW07]. As for Orchestra's Sender-based Dedicated Slots, only one node in the entire network is sending at every given point in time, so again $\mathcal{D}_{i,\cup} = \emptyset$. It is therefore denoted as single-channel scheduling, though, similar to Orchestra, channel hopping could be used to mitigate external interferences.

The algorithm is started by sending TRACK(1) to v_0 after every node has performed the initialization. The algorithm is a depth-first search again, but instead of sending back the number of nodes in the subtree, multiple transmission slots are reserved towards the parent, one for handling the traffic of every proper descendant and one for the traffic generated at that node ($\gamma_n + 1$). The transmission slot is recorded in the local $\mathcal{T}_n$ and a message is sent to the parent v_{p_n} to record the reception slot in its $\mathcal{R}_{p_n}$. When sending the TRACK message back to the parent, the number of already assigned slots is sent along as an offset for the next slot assignment.



Node \ Slot	0	1	2	3	4	5	6
0		Idle	Idle	RX	RX	RX	RX
1		RX	RX	TX	TX	TX	Idle
2		Idle	Idle	Idle	Idle	Idle	TX
3		TX	Idle	Idle	Idle	Idle	Idle
4		Idle	TX	Idle	Idle	Idle	Idle

Figure 6.15: Traffic-Aware Schedule Single-Channel (TASC).

The overall number of slots l_S in a slotframe for this schedule is calculated as

$$l_S = 1 + \sum_{n=1}^{N-1} \gamma_n + 1, \tag{6.78}$$

since every node (apart from v_0) performs $\gamma_n + 1$ slot allocations. The additional slot is the first slot of the slotframe as specified by the minimal 6TiSCH configuration [VPW17]. It is usually left free for shared communication, such as enhanced beacons and DODAG Information Objects (DIOs) for RPL. Fig. 6.15 shows a simple example for the application of this schedule. At any given point in time, only a single node in the whole network transmits and one receives and the nodes get assigned slots according to their traffic demand.

6.4.4 Traffic-Aware Schedule Multi-Channel (TAMC)

In general, it is not required that only one node is sending at every point in time. Two pairs of nodes that are sufficiently apart, can send at the same time without interference. Secondly, nodes can communicate on different channels to avoid interference. This spatial and frequency diversity can be exploited to shorten the slotframe and therefore increase the throughput and lower the latency.

A corresponding distributed algorithm is given in Algorithm 3. In contrast to the previous algorithm, where the child determines the transmission slots towards the parent, in this algorithm the parent determines the reception slots in which it will expect the child to send. It is started by sending TRACK to v_0. As in the previous algorithm, every node requires

Algorithm 2 Traffic-Aware Schedule Single-Channel (TASC)

1: **initialization**
2: $\mathcal{T}_n \leftarrow () \ \forall \ v_n$
3: $\mathcal{R}_n \leftarrow () \ \forall \ v_n$
4: mark all children as unvisited $\forall \ v_n$

5: **on message** TRACK(z) **at** v_n **from** v_s
6: **if** $\exists$ unvisited child v_u **then**
7: mark v_u as visited
8: **send** TRACK(z) to v_u
9: **else if** $v_n \neq v_0$ **then** { leaf or sub tree fully handled }
10: **for** $i \leftarrow z, \ldots, z + v_n.\gamma$ **do**
11: $\mathcal{T}_n \leftarrow \mathcal{T}_n^\frown (i)$ { append to $\mathcal{T}_n$ }
12: $\mathcal{K}_n(i) \leftarrow p_n$
13: **send** ASSIGNRX(i) to v_{p_n}
14: **send** TRACK($z + v_n.\gamma + 1$) to v_{p_n}

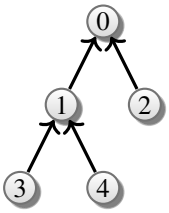
15: **on message** ASSIGNRX(i) **at** v_n **from** v_s
16: $\mathcal{R}_n \leftarrow \mathcal{R}_n^\frown (i)$
17: $\mathcal{K}_n(i) \leftarrow s$

$\gamma_n + 1$ transmission slots. This is also calculated in line 9 for a child u when a node is first handled by its parent. Afterwards, $\sigma = v_n.\gamma_u + 1$ slot allocations are performed and the loop is always finished in line 20 if we assume there are always enough channels (see below, otherwise line 14 would fail) and l_S is chosen as follows. For every allocation, a previously unused time slot is searched, recorded at sender and receiver and the used channel is blocked in the 2-hop neighborhood. For this, every node maintains a set of blocked channels for every slot $\mathcal{B}_n(i)$. This search will always be successful if we choose the slotframe length as

$$l_S = 1 + \max_{n=0,\ldots,N-1} \begin{cases} 2 \cdot \gamma_n + 1 & v_n \neq v_0 \\ \gamma_n & v_n = v_0 \end{cases}, \tag{6.79}$$

because the root requires that the slotframe has at least one slot for receiving (potentially forwarded) traffic from every proper descendant. For all other nodes this holds, too, but in addition the same number of slots is required for forwarding and one slot for transmitting the traffic generated at that node. Again, the additional slot is the shared first slot in the slotframe. Since we assume enough channels are available to avoid conflicts, the slotframe length corresponds to the requirement of the node with the largest required slotframe length.

In contrast to the other algorithms, transmissions can take place simultaneously without or with very little interference by assigning a dedicated channel or a non-conflicting channel hopping sequence to every conflicting link in a neighborhood. In the presented algorithm, this is ensured by signaling every slot assignment to all neighbors $\mathbf{N}_n$ of the sender and the receiver as well as the parents of the neighbors. Thereby, the four interference constellations in Fig. 6.3 on page 80 are avoided. This idea is similar to the slot allocation handshake of DSME as well as the DeBraS scheduling algorithm [ML16] and is especially important for dense networks.



Slot / Node	0	1	2	3	4	5
0		RX on 11	RX on 11	RX on 11	RX on 15	Idle
1		TX on 11	TX on 11	TX on 11	RX on 11	RX on 11
2		Idle	Idle	Idle	TX on 15	Idle
3		Idle	Idle	Idle	TX on 11	Idle
4		Idle	Idle	Idle	Idle	TX on 11

Figure 6.16: Traffic-Aware Schedule Multi-Channel (TAMC).

Algorithm 3 Traffic-Aware Schedule Multi-Channel (TAMC)

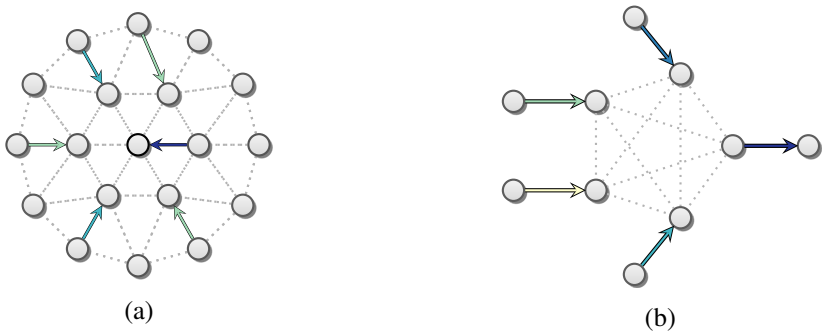
initialization
  $\mathcal{T}_n \leftarrow ()\ \forall\ v_n$
  $\mathcal{R}_n \leftarrow ()\ \forall\ v_n$
  $\mathcal{B}_n(i) \leftarrow\ \forall\ 0 \leq i < l_S,\ \forall\ v_n$
  mark all children as unvisited $\forall\ v_n$

on message TRACK **at** v_n **from** v_s
  if $\exists$ unvisited child v_u **then**
    mark v_u as visited
    $\sigma \leftarrow v_n.\gamma_u + 1$
    for $i \leftarrow 1, \ldots, l_S - 1$ **do** { slot 0 is reserved }
      if $i \notin \mathcal{R}_n \cup \mathcal{T}_n$ **then** { slot i is idle }
        $\mathcal{R}_n \leftarrow \mathcal{R}_n{}^\frown (i)$ { append to $\mathcal{R}_n$ }
        $\mathcal{K}_n(i) \leftarrow u$
        select $c \in \mathcal{C} \setminus \mathcal{B}_n(i)$
        $\mathcal{C}_{v_n}(i) \leftarrow c$
        send ASSIGNTX(i, c) to v_u
        send BLOCKCHANNEL(i, c, true) to all $v \in \mathbf{N}_n \setminus \{v_u\}$
        $\sigma \leftarrow \sigma - 1$
        if $\sigma = 0$ **then**
          break { all slots are assigned }
    send TRACK to v_u
  else if $v_n \neq v_0$ **then** { leaf or sub tree fully handled }
    send TRACK to v_{p_n}

on message ASSIGNTX(i, c) **at** v_n **from** v_s
  $\mathcal{T}_n \leftarrow \mathcal{T}_n{}^\frown (i)$
  $\mathcal{K}_n(i) \leftarrow s$
  $\mathcal{C}_{v_n}(i) \leftarrow c$
  send BLOCKCHANNEL(i, c, true) to all $v \in \mathbf{N}_n \setminus \{v_s\}$

on message BLOCKCHANNEL$(i, c,$ forward$)$ **at** v_n **from** v_s
  $\mathcal{B}_n(i) \leftarrow \mathcal{B}_n(i) \cup c$
  if forward **then send** BLOCKCHANNEL(i, c, false) to p_n

In general, the assignment of channels is a graph coloring problem where $\mathbf{L}_i$ is the set of vertices and $\boldsymbol{\mathcal{D}}_{i,\cup}$ is the set of edges. Executing the presented heuristic algorithm for the first slot results in the coloring shown in Fig. 6.17a. Here, only three channels are required, so the 16 channels available for IEEE 802.15.4 in the 2.4 GHz band are more than sufficient. In general, however, no upper bound can be given for the number of required channels, because the conflict graph is not necessarily planar. Fig. 6.17b shows an example that requires five channels and could be extended to an arbitrary number of channels. While in general finding a valid coloring with at most 16 channels is not ensured, it is usually



(a) (b)

Figure 6.17: A possible coloring for a set of conflicting links (a) and a topology that requires a coloring with five colors (b).

more than sufficient in real-world applications. Fig. 6.16 shows an example for a generated schedule. It is shorter than the one generated by TASC, since in slot 4 both the link between node 2 and node 0 as well as the one between node 3 and node 1 are active.

6.4.5 Fair Assignment of Excess Slots (FTAS)

While the previous algorithms are very suitable for TSCH, they will perform worse for DSME. This is due to the fixed slot structure of DSME. It can be adapted to different needs as described in Sect. 3.2.3, but only in certain increments. Therefore, Algorithm 4 presents a distributed algorithm for determining a schedule for a fixed number of available slots. The basic idea is that, apart from the sink v_0, no more than half of the available slots shall be used for reception to balance outgoing and incoming traffic. For most topologies no more than half the slots will be used per child at v_0, too, but for the case that most traffic is routed over a single neighbor of v_0 the checks in lines 23 and 18 are introduced. Secondly, every child of a node shall get at least one slot. This also requires that MO is selected high enough that the number of guaranteed time slots is at least twice as high as the number of children for every node.

The remaining slots are distributed according to the D'Hondt apportionment method [BKK16] also used in the context of allocating seats after elections in politics. Lines 16 to 24 correspond to Listing 2 in [BKK16], where the ϕ_u are the divisors, v_ϵ is the currently elected node and the mandates correspond to the number of assigned slots. The method is weighted by the number of descendants of a child $v_n.\gamma_u$, so that children with a higher number of descendants get more slots than those with fewer descendants.

The unusable slots, that is the beacon and CAP slots, are given by the set

$$\mathcal{U} = \bigcup_{i=0}^{2^{\mathrm{MO-SO}}-1} \{16i, 1+16i, 2+16i, \ldots, 8+16i\} \tag{6.80}$$

without CAP reduction and with CAP reduction

$$\mathcal{U} = \{0, 16, \ldots, 16 \cdot 2^{\mathrm{MO-SO}} - 1\} \cup \{1, 2, \ldots, 8\}\,. \tag{6.81}$$

Algorithm 4 Fixed-Size Traffic-Aware Schedule (FTAS)

1: **initialization**
2: $\mathcal{T}_n \leftarrow (),\ \mathcal{R}_n \leftarrow ()\ \forall\ v_n,\ \mathcal{B}_n(i) \leftarrow\ \forall\ 0 \leq i < l_S,\ \forall\ v_n$

3: **procedure** SELECTRECEPTIONSLOT(v_u)
4: **for** $i \leftarrow 0, \ldots, l_S - 1$ **do**
5: **if** $i \notin \mathcal{R}_n \cup \mathcal{T}_n \wedge i \notin \mathcal{U}$ **then** { slot i is idle }
6: select any $c \in \mathcal{C} \setminus \mathcal{B}_n(i)$
7: $\mathcal{R}_n \leftarrow \mathcal{R}_n{}^\frown (i)\ ,\ \mathcal{K}_n(i) \leftarrow u,\ \mathcal{C}_n(i) \leftarrow c$
8: **send** ASSIGNTX(i, c) to v_u
9: **send** BLOCKCHANNEL($i, c, true$) to all $v \in \mathbf{N}_n \setminus \{v_u\}$
10: **break**

11: **on message** FORWARD **at** v_n **from** v_s
12: **for all** $v_u \in \mathbf{C}_n$ **do** { one slot for every child }
13: SELECTRECEPTIONSLOT(v_u)
14: **if** $|\mathbf{C}_n| > 0$ **then** { assign remaining slots }
15: $\omega \leftarrow \begin{cases} \left\lfloor \frac{l_S - |\mathcal{U}|}{2} \right\rfloor - |\mathbf{C}_n| & v_n \neq v_0 \\ l_S - |\mathcal{U}| - |\mathbf{C}_n| & v_n = v_0 \end{cases}$
16: $\phi_u \leftarrow 1\ \forall\ v_u \in \mathbf{C}_n$ { initialize divisors }
17: select any $v_\epsilon \in \mathbf{C}_n$
18: **while** $\omega > 0$ **and** $\exists\, v_u \in \mathbf{C}_n$: $\phi_u < \left\lceil \frac{l_S - |\mathcal{U}|}{2} \right\rceil$ **do**
19: **for all** $v_u \in \mathbf{C}_n \setminus \{v_\epsilon\}$ **do**
20: **if** $\phi_u \cdot (v_n.\gamma_\epsilon + 1) < \phi_\epsilon \cdot (v_n.\gamma_u + 1)$ **then**
21: $v_\epsilon \leftarrow v_u$ { currently elected node }
22: $\phi_\epsilon = \phi_\epsilon + 1$
23: **if** $\phi_\epsilon < \left\lceil \frac{l_S - |\mathcal{U}|}{2} \right\rceil$ **then**
24: SELECTRECEPTIONSLOT(v_ϵ), $\omega = \omega - 1$
25: HANDLENODE(v_n)

26: **on message** BACKTRACK **at** v_n **from** v_s
27: HANDLENODE(v_n)

28: **procedure** HANDLENODE(v_n)
29: **if** $\exists$ unvisited child $v_u \in \mathbf{C}_n$ **then**
30: mark v_u as visited
31: **send** FORWARD to v_u
32: **else if** $v_n \neq v_0$ **then** { leaf or sub tree fully handled }
33: **send** BACKTRACK to v_{p_n}

34: **on message** ASSIGNTX(i, c) **at** v_n **from** v_s
35: $\mathcal{T}_n \leftarrow \mathcal{T}_n{}^\frown (i)\ ,\ \mathcal{K}_n(i) \leftarrow s,\ \mathcal{C}_n(i) \leftarrow c$
36: **send** BLOCKCHANNEL($i, c, true$) to all $v \in \mathbf{N}_n \setminus \{v_s\}$

37: **on message** BLOCKCHANNEL(i, c, forward) **at** v_n **from** v_s
38: $\mathcal{B}_n(i) \leftarrow \mathcal{B}_n(i) \cup c$
39: **if** forward **then send** BLOCKCHANNEL($i, c, false$) to v_{p_n}

Slot / Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0										RX on 11	RX on 11	RX on 11	RX on 11	RX on 11	RX on 11	RX on 11
1										TX on 11	RX on 15	TX on 11	TX on 11	TX on 11	RX on 15	RX on 15
2										Idle	TX on 11	Idle	Idle	Idle	TX on 11	TX on 11
3										Idle	TX on 15	Idle	Idle	Idle	Idle	TX on 15
4										Idle	Idle	Idle	Idle	Idle	TX on 15	Idle

Figure 6.18: FTAS.

As in TAMC, the channel used for a slot is selected on a first-come, first-serve basis and blocked in their neighborhood and at the parents of the neighbors by filling the sets $\mathcal{B}_n(i)$. Thus, the selection in line 6 could fail and an implementation has to be aware of this case, but as elaborated before this is unlikely due to the high number of available slots. The number of sent messages can be slightly reduced by tracking the already forwarded blocking messages. For starting the algorithm, TRACK is sent to v_0. Fig. 6.18 shows a possible result for the topology in Fig. 6.16 for a single superframe per multi-superframe. Since, compared to TAMC, 2 more slots have to be filled, these are distributed over the available links according to their traffic demand.

The schedules presented in this section will be used in the evaluation of the following analytical model for TDMA.

6.5 TDMA Model*

In this section, an analytical model for multi-hop TDMA networks is proposed. Besides the general advantages of analytical models as presented in the introduction of this chapter, this model is in particular interesting for analyzing new scheduling algorithms without the need to implement them for a full simulation stack. The main reasons for packet loss in wireless multi-hop networks can be coarsely categorized as follows.

- Collisions of transmissions of the same network.
- External interferences, for example between IEEE 802.15.4 and IEEE 802.11 transceivers or with another unsynchronized IEEE 802.15.4 network.

*Partly previously published in "F. Kauer and V. Turau, *An Analytical Model for Wireless Mesh Networks with Collision-Free TDMA and Finite Queues*, EURASIP Journal on Wireless Communications and Networking 2018:149, June 2018. https://doi.org/10.1186/s13638-018-1146-x" [KT18a].

- High path loss, for example due to a large distance or fading.
- Queue drops if the rate of packets that are generated or have to be forwarded is higher than the rate at which packets can be transmitted. This can be permanent or during a burst.

To mitigate these losses, the introduced TDMA data link layers TSCH and DSME provide the following features that allow for reliable multi-hop communication.

- A slot structure with predefined timing to arrange the transmissions in the time domain to avoid collisions within the same network.
- Time synchronization to ensure an aligned timing throughout the multi-hop network.
- Channel adaption (only DSME) and channel hopping (both) to arrange transmission in the frequency domain to increase the number of transmissions per time as well as to mitigate external interferences.
- Procedures for setting up time and frequency schedules in a distributed fashion to ensure conflict-free transmissions. This is only integrated in DSME, but attempts exist to achieve this in TSCH networks as presented in Sect. 3.3.

To calculate the performance in steady-state, it can be assumed that all nodes are already associated to the network, are properly synchronized to the global notion of time, and have negotiated a conflict-free and valid multi-hop schedule. In such a schedule, a combination of a time slot and a frequency channel is either free or assigned to exactly one transmitter and one receiver in every neighborhood. By this, packet collisions can be avoided, even in hidden node constellations. In DSME networks this constraint is inherent, given that no failures happen during the negotiation (cf. Sect. 4.2). For TSCH networks, such a fixed assignment is not necessarily required, but sharing will obviously lead to collisions, even though the probability can be reduced depending on the assignment [DNL15, Fig. 4]. Furthermore, the schedule should avoid using links with high path loss or highly fluctuating channel conditions to avoid occasional packet loss on the physical layer by means of an appropriate neighborhood management as presented in [STW15].

In large industrial plants, the frequencies of the used radio components are usually coordinated and monitored to avoid cross-interferences and as a security measure [KDM09]. However, external interferences can often not be avoided completely, so TSCH and DSME can dynamically use multiple frequency channels. Apart from mitigating external interferences, this can also be used to increase the throughput by assigning the same time slot to multiple transceiver pairs on different frequency channel and is therefore also modeled in this paper. It can either be implemented as channel adaption where a fixed frequency channel is assigned at a given time or channel hopping where the channel to be used is iterated over a given sequence. Yet, in this thesis external interferences are not considered, so channel hopping with non-overlapping hopping sequences is equivalent to a fixed assignment as for channel adaption.

Since only conflict-free schedules are considered, it is therefore concluded that in properly constructed TSCH and DSME networks without a lot of external disturbances, the occurrence of queue drops is the main factor that determines the maximum achievable

network performance. Yet, in real-world networks other losses can still occur. Sect. 6.5.9 gives an outlook about how to consider these in the model.

As noted in Sect. 6.1, the M/D/1/K model is often used to model queues in the context of TDMA networks. However, it has two major weaknesses in the given scenario: Even if the traffic generation itself is modeled as Poisson distributed, this distribution does not necessarily hold for forwarding nodes, so the M/D/1/K model is not suitable for multi-hop networks. Furthermore, the service times are not necessarily deterministic but can diverge significantly, due to two effects: First, the schedule itself might be irregular, for example if multiple subsequent transmission slots are followed by a large idle phase. Secondly, even if the schedule is regular, the service time of a packet that arrives at an empty queue is not constant, but depends on the time left until the next transmission slot. This effect is especially relevant in scenarios with low traffic. A proper model has to take these effects into account.

6.5.1 System Model

In the presented system model as illustrated in Fig. 6.19 the queue of every node is modeled as an instance of a Markov chain and they are linked to form a model of the full network. Packets are either forwarded as received or generated at the node. In both cases, the packets are pushed to a queue of fixed length or dropped if the queue is full. If there is at least one packet in the queue at the beginning of a transmission slot, a transmission attempt takes place. The calculations are based on fixed multi-hop schedules. A schedule can be calculated offline with algorithms such as the ones presented in Sect. 6.4. Alternatively, a schedule can be extracted from a real-world deployment or a simulator. In the following, the notation for such a schedule as introduced in Sect. 6.4.1 is used.

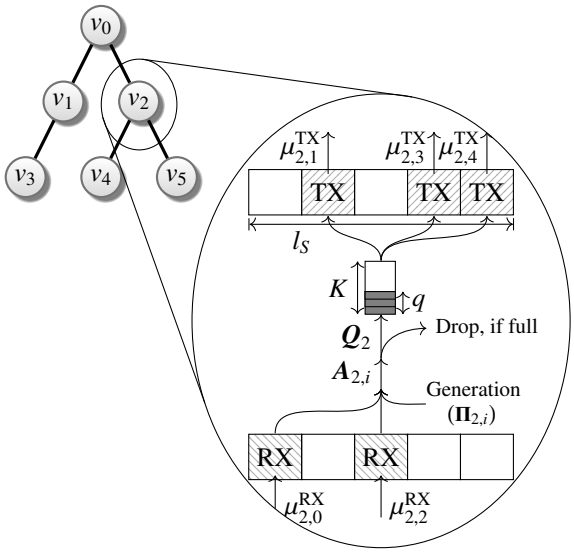


Figure 6.19: Illustration of the proposed model annotated with the notation introduced in the following sections.

6.5.2 Queue Model

Before modeling multi-hop communication, this section considers the viewpoint of a single node v_n and its local behavior. In the multi-hop model, every node of the network will get its own instance of this model. The main focus is the finite queue, so after this section we will be able to calculate packet drops due to a full queue as well as the expected queuing delay. The inputs into this model are the probabilities for incoming traffic and the distribution of the reception slots $\mathcal{R}_n$ and transmission slots $\mathcal{T}_n$. Beyond the usage in a multi-hop model as introduced in the next section, the model in this section is also applicable in other setups, such as single-hop networks or for evaluating the performance of a single node as presented in Sect. 6.5.3.

The following policy is used to model the queue:

- The queue can hold at most K packets.
- The number of packets in the queue at the beginning of a slot is denoted as q.
- New packets can arrive at any time during a slot. At most $K - q$ packets are accepted during a slot.
- A packet is removed from the queue at the end of a slot if and only if it is a transmission slot and the packet was already in the queue at the beginning of the slot, i.e. $q > 0$, modeling the transmission process itself.

The queue of a node is modeled as a discrete-time Markov chain with the states

$$\mathbf{\Sigma}_n = \left\{ \sigma^n_{q,i} \,\middle|\, 0 \leq q \leq K, 0 \leq i < l_S \right\}. \tag{6.82}$$

In order to account for the irregular schedule and for modeling the service time more accurate, the state of the queue $\sigma^n_{q,i}$ does not only account for the number q of packets in the queue as for usual M/D/1/K models, but also the current position within the slot schedule i. As shown in Fig. 6.20, the model presented here is a super set of the M/D/1/K model for $l_S = 1$ and $\mathcal{T}_n = (0)$. Fig. 6.21 depicts a more complex schedule, the corresponding states and the transitions as presented in the following.

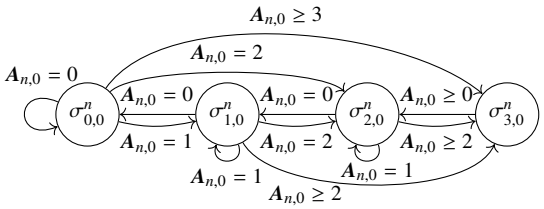


Figure 6.20: A M/D/1/K model in the syntax introduced in this thesis with $K = 3$, $l_S = 1$ and $\mathcal{T}_n = (0)$.

Traffic Model

The number of arriving packets during a time slot i, that is the generated and received traffic, is described by the random variable $\boldsymbol{A}_{n,i}$. For the purpose of the queuing model it can follow an arbitrary distribution that can optionally depend on the current state.

In the following, a traffic model composed of a Poisson distribution and a Bernoulli distribution is presented. While a Poisson distribution is popular for modeling traffic generation, as also used in the CSMA/CA model, a Bernoulli distribution is more suitable for traffic forwarding in time-slotted networks since it allows for at most one packet per slot.

For the Poisson part, the random variable $\boldsymbol{\Pi}_{n,i}$ is introduced with the probability distribution given by

$$P\left(\boldsymbol{\Pi}_{n,i} = k\right) = \frac{{\lambda_{n,i}}^k}{k!} e^{-\lambda_{n,i}}, \tag{6.83}$$

with the mean packet rate $\lambda_{n,i}$. The probability for *at least* k packets is given by

$$P\left(\boldsymbol{\Pi}_{n,i} \geq k\right) = 1 - \sum_{j=0}^{k-1} P\left(\boldsymbol{\Pi}_{n,i} = k\right). \tag{6.84}$$

This is extended with a Bernoulli distribution with the probability $\beta_{n,i}$ that a single packet is arriving in addition to the Poisson traffic in slot i, so we get

$$P\left(\boldsymbol{A}_{n,i} = k\right) = \begin{cases} (1-\beta_{n,i}) \cdot P\left(\boldsymbol{\Pi}_{n,i} = 0\right) & k = 0 \\ (1-\beta_{n,i}) \cdot P\left(\boldsymbol{\Pi}_{n,i} = k\right) & \\ \quad + \beta_{n,i} \cdot P\left(\boldsymbol{\Pi}_{n,i} = k-1\right) & \text{otherwise} \end{cases} \tag{6.85}$$

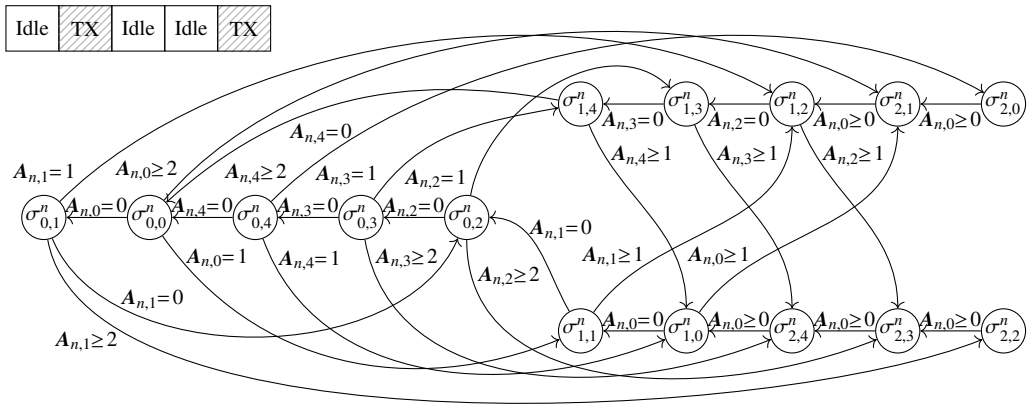


Figure 6.21: Schedule for $l_S = 5$, $\mathcal{T}_n = (1,4)$ and corresponding state diagram for $K = 2$.

and

$$P\left(\boldsymbol{A}_{n,i} \geq k\right) = \begin{cases} 1 & k = 0 \\ (1-\beta_{n,i}) \cdot P\left(\boldsymbol{\Pi}_{n,i} \geq k\right) & \\ \quad + \beta_{n,i} \cdot P\left(\boldsymbol{\Pi}_{n,i} \geq k-1\right) & \text{otherwise.} \end{cases} \tag{6.86}$$

Since all l_S slots occur with the same probability, the expected total number of packets per slotframe is calculated as the sum of the expected number of the $\boldsymbol{A}_{n,i}$ as derived in Appendix A.3.1 as

$$\mathcal{A}_n = \sum_{i=0}^{l_S-1} E[\boldsymbol{A}_{n,i}] = \sum_{i=0}^{l_S-1} (1-\beta_{n,i}) \cdot \lambda_{n,i} + \beta_{n,i} \cdot (\lambda_{n,i}+1)\,. \tag{6.87}$$

Queuing Probability

The random variable $\boldsymbol{Q}_n$ is the number of accepted packets that are inserted into the queue per slot. Since at most $K-q$ packets can be inserted into the queue, its probability distribution, i.e. the probabilities of queuing k packets, is calculated as

$$P\left(\boldsymbol{Q}_n = k \,\middle|\, \sigma_{q,i}^n\right) = \begin{cases} P\left(\boldsymbol{A}_{n,i} = k\right) & k < K-q \\ P\left(\boldsymbol{A}_{n,i} \geq K-q\right) & k = K-q \\ 0 & \text{otherwise.} \end{cases} \tag{6.88}$$

Obviously, the difference of $\boldsymbol{A}_{n,i}$ and $\boldsymbol{Q}_n$ is the number of packets dropped due to a full queue.

Transition Probabilities

$P\left(\xi \to \zeta\right)$ with $\xi, \zeta \in \boldsymbol{\Sigma}_n$ is the probability of going from ξ to ζ in one step. Transition probabilities not listed are zero, in particular for transitions with non-consecutive slots. Furthermore, % denotes the modulo operation. So for $0 \leq q \leq K$ and $0 \leq i < l_S$, all possible transitions are specified by

$$P\left(\sigma_{q,i}^n \to \sigma_{\max(q-\tau_{n,i},0)+k,\ (i+1)\%l_S}^n\right) = P\left(\boldsymbol{Q}_n = k \,\middle|\, \sigma_{q,i}^n\right) \tag{6.89}$$

where

$$\tau_{n,i} = \begin{cases} 1 & i \in \mathcal{T}_n \\ 0 & i \notin \mathcal{T}_n \end{cases} \tag{6.90}$$

and k is the number of arriving or newly generated packets during the time slot that iterates from 0 to $K-q$. While this gives a complete description of the model, two corner cases should be mentioned explicitly for clarity. With an empty queue and $\boldsymbol{A}_{n,i} = 0$, that is no arriving packets, the model stays in the states with $q = 0$. With a full queue, it is not

possible to insert further packets into the queue so $\boldsymbol{Q}_n = 0$ and thus the single possible transition out of the states with $q = K$ has the probability

$$P\left(\boldsymbol{Q}_n = 0 \,\middle|\, \sigma_{K,i}^n\right) = P\left(\boldsymbol{A}_{n,i} \geq 0\right) = 1, \tag{6.91}$$

either to a state with $q = K - 1$ if the current slot is a transmission slot or else to a state with $q = K$.

As shown in Appendix A.3.2, these probabilities can be used to calculate the stationary distribution of this Markov chain. This results in the probability $c_{n,q,i}$ of being in state $\sigma_{q,i}^n$ in the stationary case.

Transmission Probability

From the $c_{n,q,i}$, the probability that a successful transmission takes place within a given slot is calculated from the complementary probability of being in a state with empty queue ($q = 0$) as

$$\mu_{n,i}^{\mathrm{TX}} = \tau_{n,i} \cdot \left(1 - \frac{c_{n,0,i}}{\sum_{q=0}^{K} c_{n,q,i}}\right). \tag{6.92}$$

Packet Acceptance Probability

If q packets are in the queue, $K - q$ packets can be inserted into the queue, so the expected number of *accepted* packets in state $\sigma_{q,i}^n$ is

$$E\left[\boldsymbol{Q}_n \,\middle|\, \sigma_{q,i}^n\right] = P\left(\boldsymbol{A}_{n,i} \geq K - q\right) \cdot (K - q) + \sum_{k=0}^{K-q-1} P\left(\boldsymbol{A}_{n,i} = k\right) \cdot k. \tag{6.93}$$

Since the events of being in a state are mutually exclusive and exhaustive, $E[\boldsymbol{Q}_n]$ can be calculated from this according to the law of total expectation as

$$E[\boldsymbol{Q}_n] = \sum_{q=0}^{K} \sum_{i=0}^{l_S - 1} c_{n,q,i} \cdot E\left[\boldsymbol{Q}_n \,\middle|\, \sigma_{q,i}^n\right] \tag{6.94}$$

and the overall expected number of packets per slotframe is thus $l_S \cdot E[\boldsymbol{Q}_n]$. Together with the overall expected number of packets arriving at the queue, the overall packet acceptance probability is calculated as

$$P_{\mathrm{accept},n} = \frac{l_S \cdot E[\boldsymbol{Q}_n]}{\mathcal{A}_n}. \tag{6.95}$$

Queuing Delay

The number of time steps it takes until an arriving packet is transmitted is described by the random variable $\boldsymbol{D}_n$. For convenience, we define

$$\phi_n(i) = \begin{cases} |\mathcal{T}_n| - 1 & i \leq t_{n,0} \ \vee \ i > t_{n,|\mathcal{T}_n|-1} \\ g : t_{n,g} < i \leq t_{n,g+1} & \text{else,} \end{cases} \tag{6.96}$$

as the index of the transmission slot preceding i and

$$\delta(i,j) = \begin{cases} j - i & j \geq i \\ j - i + l_S & \text{otherwise,} \end{cases} \tag{6.97}$$

as the number of slots when going from i to j. Then, given the event of being in state $\sigma^n_{q,i}$ after accepting a packet and pushing it in the queue, $\boldsymbol{D}_n$ is calculated as

$$\boldsymbol{D}_n\left(\sigma^n_{q,i}\right) = f_q l_S + 1 + \delta\left(i,\ t_{n,(\phi_n(i)+q)\%|\mathcal{T}_n|}\right) \tag{6.98}$$

where $f_q = \left\lceil \frac{q}{|\mathcal{T}_n|} - 1 \right\rceil$ is the number of iterations over the full slotframe, the summand 1 accounts for the transmission slot itself and the last summand for the remaining packets after $f_q \cdot |\mathcal{T}_n|$ packets were transmitted. The expected delay for a packet arriving at an arbitrary point in time can be calculated from this as

$$E[\boldsymbol{D}_n] = \sum_{q=0}^{K} \sum_{i=0}^{l_S-1} c_{n,q,i} \cdot \boldsymbol{D}_n\left(\sigma^n_{\max(q-\tau_{n,i},0)+1,\ (i+1)\%l_S}\right) \tag{6.99}$$

considering the additional transition to account for the arriving packet itself.

6.5.3 Single Node Evaluation

At first, the results for the queuing model on a single node are analyzed without considering the effects of multi-hop networks. The purpose of this section is to demonstrate the benefit of the refined model in contrast to a simple M/D/1/K model by comparing them with an event-based simulation. For this, an exemplary schedule is shown in Fig. 6.22 together with the two approximations presented in the following. The schedule is construed in a way that on average the same amount of traffic enters and leaves the system. However, due to a finite queue length of $K = 5$, packet drops can still occur. In Fig. 6.23 the following four evaluations are compared:

- In the **M/D/1/K** model, there is no option to specify multiple slots and only an overall packet rate rate can be specified.
- In the **Distributed** approximation, the transmission and idle slots are modeled correctly, but incoming traffic is modeled to follow a Poisson distribution and not a Bernoulli distribution. Furthermore, it is distributed over all slots. To get the same

Full

	TX	RX	Idle	Idle	Idle
$\beta_{n,i}$	0	0.5	0	0	0
$\lambda_{n,i}$	0.1	0.1	0.1	0.1	0.1

Distributed

	TX	Idle	Idle	Idle	Idle
$\beta_{n,i}$	0	0	0	0	0
$\lambda_{n,i}$	0.2	0.2	0.2	0.2	0.2

M/D/1/K

	TX
$\beta_{n,i}$	0
$\lambda_{n,i}$	1

Figure 6.22: Exemplary schedule for evaluating the queuing model together with two approximations, all with the same $\mathcal{A}_n = 1$.

$\mathcal{A}_n$ of 1 as in the actual scenario, all $\lambda_{n,i}$ are set to $\frac{1}{l_S} = 0.2$. This scenario allows to assess the deficiency of not including the $\beta_{n,i}$ in the model.

- The **Full Model** scenario additionally models the reception slot correctly as presented.
- The **Simulation** is a discrete event simulation of the queue based on SimPy [SimPy] that implements the policy as presented at the beginning of Sect. 6.5.2. The mean and 95% confidence interval are shown of 10 runs with 10000 packets.

While the full model matches the simulation very well, the probability of dropping packets due to a full queue is higher in the distributed scenario, since the variance of the packet reception is higher, even though the average reception rate is equal. In fact, for $g_{\text{up}} = 0$

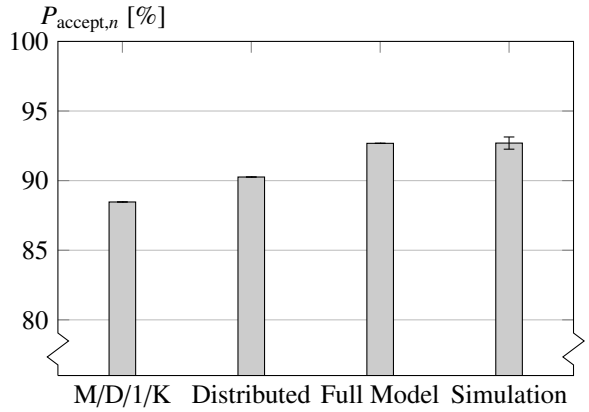


Figure 6.23: Comparing the full presented model with a simulation and simpler models: An M/D/1/K model and a model that considers the TX and Idle slots, but distributes the reception over all slots.

and at least as much TX slots as RX slots, the full model will never indicate a packet loss ($P_{\text{accept},n} = 100\%$), while in the simplified models there is a non-zero probability that more packets are received in a slotframe than sent, eventually leading to packet loss.

The M/D/1/K model gives even less accurate results, because the service time is constant and equal to $l_S \cdot T_s$, while in the distributed and the full model case, packets might be processed faster if the queue is empty and they arrive during the later slots.

6.5.4 Analysis of the Queue Distribution

In this section, we take a deeper look into the distribution of the queue level and its influence on the probability of accepting a packet for different traffic loads. For this, a scenario with $K = 10$, $l_S = 5$ and a single transmission slot is considered.

In Fig. 6.24, the probability distribution of the queue level is plotted for different traffic loads $\mathcal{A}_n$. For this, the probability of being in a state with q packets in the queue is calculated as

$$c_{n,q,\star} = \sum_{i=0}^{l_S-1} c_{n,q,i}. \tag{6.100}$$

In the first scenario, the traffic load is distributed over the $\lambda_{n,i}$, corresponding to a node that does only generate but does not forward traffic and in the second scenario, it is distributed only over the $\beta_{n,i}$, corresponding to a node that only forwards traffic generated by other nodes.

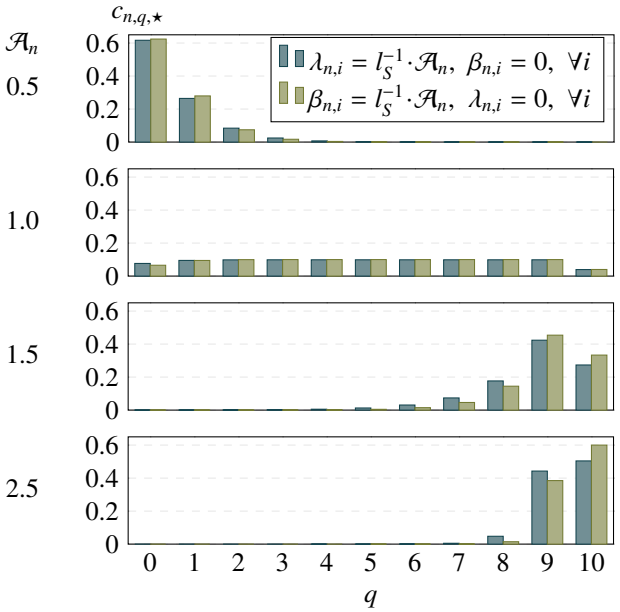


Figure 6.24: Distribution of queue levels for $K = 10$, $l_S = 5$ and a single transmission slot.

For a low rate of $\mathcal{A}_n = 0.5$, the queue is empty most of the time with a decaying probability of having more packets in the queue. The probabilities for queue levels larger than 3 are negligible, so the probability of accepting new packets is $P_{\text{accept},n} = 1.00$ for both scenarios.

For a medium rate of $\mathcal{A}_n = 1$, where on average one packet is sent and received every slotframe, the distribution is nearly constant. This leads to a packet acceptance probability of $P_{\text{accept},n} = 0.95$ for the first and $P_{\text{accept},n} = 0.96$ for the second scenario. Interestingly, the probability for $q = K$ is significantly smaller than the others.

This is due to the policy that in a transmission slot with $q = K$ no new packets can be pushed to the queue, but certainly a packet will be sent, i.e. $q = K - 1$ in the next slot. However, for all other queue levels, it is possible to maintain the same queue level by sending and receiving one packet. That is even more apparent for $\mathcal{A}_n = 1.5$ where a queue level larger than $q = 5$ most of the time, leads to a packet acceptance probability of $P_{\text{accept},n} = 0.67$ for both scenarios. Having $q = K$ more often than other queue levels is only possible for very high rates of $\mathcal{A}_n = 2.5$. Here, the node is highly congested and a $P_{\text{accept},n} = 0.40$ is achieved in both scenarios.

When comparing the scenarios, we see that the difference is not very large, but in the second scenario, the probabilities are shifted to the boundaries. This evaluation also shows that often only a small part of the queue is actually utilized. For $\mathcal{A}_n = 0.5$ a maximum queue size of $K = 3$ would be sufficient to prevent packet loss. Also for $\mathcal{A}_n = 2.5$, this would not change the packet acceptance probability. The probability distribution would shift to the left, but does not change its overall shape relative to the upper bound. Of course, the queuing delay would be reduced by this. The bottom line is that the maximum queue length K has the highest influence in scenarios where the amount of incoming and outgoing traffic is similar.

6.5.5 Multi-Hop Model

To model multi-hop communication, every node in the network gets its own instance of the previously presented model. They are then linked to get a full network for allowing to determine network-wide metrics. In a data-collection scenario, nodes are not influenced by the behavior of nodes towards the sink for collision-free TDMA. This is in contrast to CSMA/CA, where messages sent out by the latter can lead to collisions. Therefore, for the TDMA model, it is sufficient to calculate the equations of the nodes from the leafs towards the sink and no iterative calculation is required.

Traffic Generation and Forwarding

For the purpose of this thesis, we assume a data-collection scenario with sink v_0. Every node, except v_0, generates packets with exponentially distributed intervals with mean I_{up} that are to be forwarded to v_0 via a routing tree. The base time unit is the slot length T_s, so the generation rate is

$$g_{\text{up}} = \frac{T_s}{I_{up}}. \tag{6.101}$$

In the following, only homogeneous traffic generation is considered, so we set

$$\lambda_{n,i} = \begin{cases} g_{\text{up}} & v_n \neq v_0 \\ 0 & v_n = v_0, \end{cases} , \forall 0 \leq i < l_S. \tag{6.102}$$

but other traffic patterns can be obtained by choosing individual values for $\lambda_{n,i}$.

Besides the traffic generation itself, there is a probability $\mu_{n,i}^{\text{RX}}$ of receiving a packet from a neighbor in a reception slot to be forwarded to the sink. This value is calculated from the probability $\mu_{\mathcal{K}_n(i),i}^{\text{TX}}$ that the neighbor $\mathcal{K}_n(i)$ is transmitting in the given slot as

$$\mu_{n,i}^{\text{RX}} = \begin{cases} \mu_{\mathcal{K}_n(i),i}^{\text{TX}} & i \in \mathcal{R}_n \\ 0 & \text{else.} \end{cases} \tag{6.103}$$

The forwarding is modeled by the Bernoulli part of the traffic model since at most one packet can be received per slot, so $\beta_{n,i} = \mu_{n,i}^{\text{RX}}$.

Network Throughput

The throughput of the network corresponds to the number of packets arriving at the sink v_0 per time. Since the sink does not generate traffic itself, the throughput can be calculated as

$$\frac{\mathcal{A}_0}{T_s} = {T_s}^{-1} \cdot \sum_{i=0}^{l_S-1} \mu_{0,i}^{\text{RX}}. \tag{6.104}$$

End-to-End Metrics

The packet delivery ratio $\text{PDR}_{\text{up},n}$, that is the probability that a packet originating from node v_n is finally received by the sink v_0, is given by the product of $P_{\text{accept},n}$ over the path

$$\text{PDR}_{\text{up},n} = \begin{cases} \text{PDR}_{\text{up},p_n} \cdot P_{\text{accept},n} & v_n \neq v_0 \\ 1 & v_n = v_0, \end{cases} \tag{6.105}$$

where p_n is the parent of v_n in the routing tree.

Similarly, the end-to-end delay $\mathfrak{D}_{\text{up},n}$ is calculated as

$$\mathfrak{D}_{\text{up},n} = \begin{cases} \mathfrak{D}_{\text{up},p_n} + E[\boldsymbol{D}_n] & v_n \neq v_0 \\ 0 & v_n = v_0. \end{cases} \tag{6.106}$$

6.5.6 Comparing Multi-Hop Model and Simulation

After evaluating the model for a single node only, we now build up a full multi-hop network to analyze its performance. For this the model is first used to compare its results to the Contiki implementation of the Orchestra scheduling [DNL15] for TSCH running in the COOJA simulator [ODE06] with a maximum queue length of $K = 16$. It is evaluated for a data-collection scenario in a concentric topology with 19 nodes as used in Sect. 6.3.

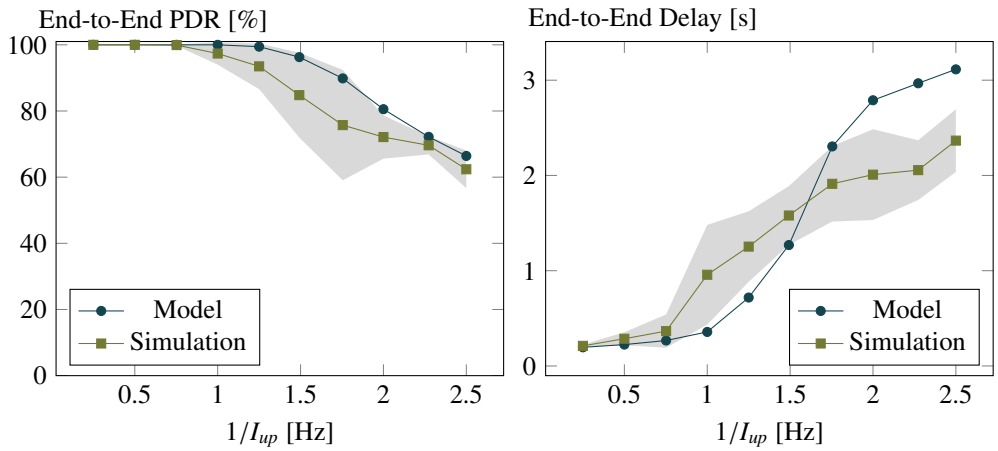


Figure 6.25: Comparing a Sender-based Dedicated Orchestra Slot Schedule in the Cooja Simulator and the model with $N = 19$ and $K = 16$.

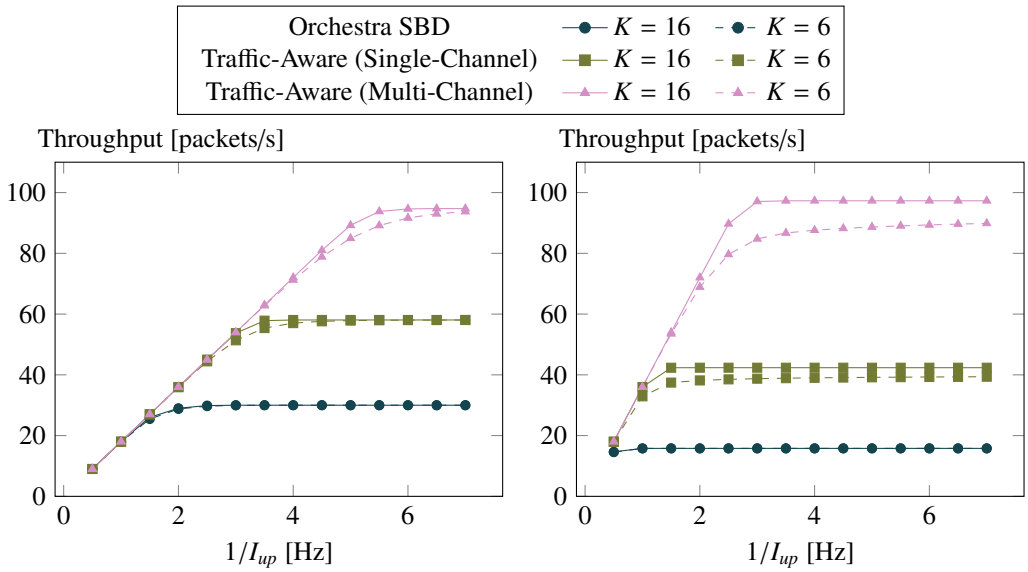
Sender-based Dedicated (SBD) Orchestra Slots are used, where every node has a dedicated transmission slot and thus, $\mathcal{D}_{i,\cup} = \emptyset$ and so the schedule is conflict-free and valid. Furthermore, the slotframe length is minimized to maximize the throughput under the given constraint.

The results in Fig. 6.25 show the end-to-end packet delivery ratio as well as the end-to-end delay averaged over all nodes in the network (except v_0). In the simulation, every node sends packets to the center with exponentially distributed intervals. After a warm-up phase of 15 minutes, 100 packets are monitored if they arrive and how long they take. The resulting mean and the 95% confidence interval over 5 runs are shown in the plot.

While the simulation uses the Routing Protocol for Low power and Lossy Networks (RPL) [WTB12], the analytical model uses a predefined tree routing as described in Sect. 6.2. The packet delivery ratio $\mathrm{PDR}_{\mathrm{up},n}$ and the end-to-end delay $\mathfrak{D}_{\mathrm{up},n}$ are shown in the plot. Obviously, no confidence intervals are shown for the analytical results because no randomness is included in the calculation. The PDR results show a good conformance, while for the delay discrepancies around 1 Hz and above 2 Hz indicate the existence of not properly considered effects. The existing differences can be mainly traced back to the varying routing trees in the simulation runs. Still it demonstrates the applicability of the analytical model for multi-hop networks as well as the validity of the Contiki simulation for the given scenario.

6.5.7 Evaluation and Comparison of TSCH Schedules

In Fig. 6.26, the achievable throughput for Orchestra SBD is compared to the single-channel and multi-channel traffic-aware schedules presented in Sect. 6.4. Multiple scenarios are considered, that is a network of $N = 19$ and $N = 37$ nodes in a concentric topology and a maximum queue sizes of $K = 6$ (dashed) and $K = 16$ (solid). For this, the number of packets received by the sink v_0 is plotted over the packet generation rate of every node. For low rates, the network is able to handle the complete traffic, so the throughput rises



(a) $N = 19$ (b) $N = 37$

Figure 6.26: Throughput for the Sender-based Dedicated Orchestra Schedule and the two presented traffic-aware schedules over the packet sending interval for a network of $N = 19$ and $N = 37$ nodes aligned in concentric circles. The solid lines are for $K = 16$, the dashed for $K = 6$..

	$N = 19$			$N = 37$		
Schedule	l_S	RX slots	RX ratio	l_S	RX slots	RX ratio
Orchestra SBD	20	6	30%	38	6	16%
Traffic-Aware (Single-Channel)	31	18	58%	85	36	42%
Traffic-Aware (Multi-Channel)	19	18	95%	37	36	97%

Figure 6.27: Schedule properties at the root node v_0.

linearly. For high rates, the networks is saturated and increasing the rate does not increase the throughput anymore. This is in contrast to the plots in Fig. 6.11 on page 95 where no fixed saturation level is reached for CSMA/CA.

In general, the traffic-aware schedules provide significantly more throughput than Orchestra SBD, while the multi-channel schedule has an even higher throughput than the single-channel schedule. The higher throughput can be explained by considering that nodes higher in the tree that need to handle more traffic have more slots compared to nodes with low traffic demand. It is also apparent in Fig. 6.26 that the maximum queue length K is most significant in the transition section, in conformance to Sect. 6.5.4. Compared to the applied schedule, it has a lower impact on the throughput, but the difference is larger for $N = 37$ than for $N = 19$ due to the higher number of hops.

Also, when comparing different network sizes, we see that the saturation is reached for smaller rates. This is expected, because the same packet generation rate applied at

more nodes leads to more overall traffic. Secondly, the single-channel schedule and even more Orchestra SBD show a significantly lower throughput, because having more nodes requires a higher number of slots during which the sink is idle. This does not hold for the multi-channel schedule, because spatial reuse is possible.

In Fig. 6.27, the slotframe lengths are given together with the number and ratio of reception slots at the sink v_0. For Orchestra SBD, every neighbor of v_0 has only one slot, so v_0 has a long phase of inactivity. For the traffic-aware schedules, the inner nodes get more slots because of their higher traffic load. Thus, the sink can receive more packets per slotframe resulting in a higher throughput. The missing slot in the last row is due to the shared slot. In the multi-channel schedule, the sink can even potentially receive data traffic in every slot, apart from the first one. This comparison again explains the difference in throughput for the different network sizes, because comparative to the network with 19 nodes, the RX ratio is significantly lower for $N = 37$ and the Orchestra SBD and single-channel schedules, while it is even larger for the multi-channel schedule where the shared slot has a lower impact due to the longer slotframe length.

Fig. 6.28 finally shows the saturation level of the throughput achieved by the proposed Fixed-Size Traffic-Aware Schedule (FTAS) schedule as presented in Algorithm 4 on page 104. For short slotframes, the achievable throughput is lower that the multi-channel version of the traffic-aware schedule, but larger than for the single-channel version. For increasing slotframe length it converges to the multi-channel version and can even outperform it slightly for very long slotframes due to the lower impact of the first slot. This is, however, not adequate for practical applications since too few time is reserved for shared traffic.

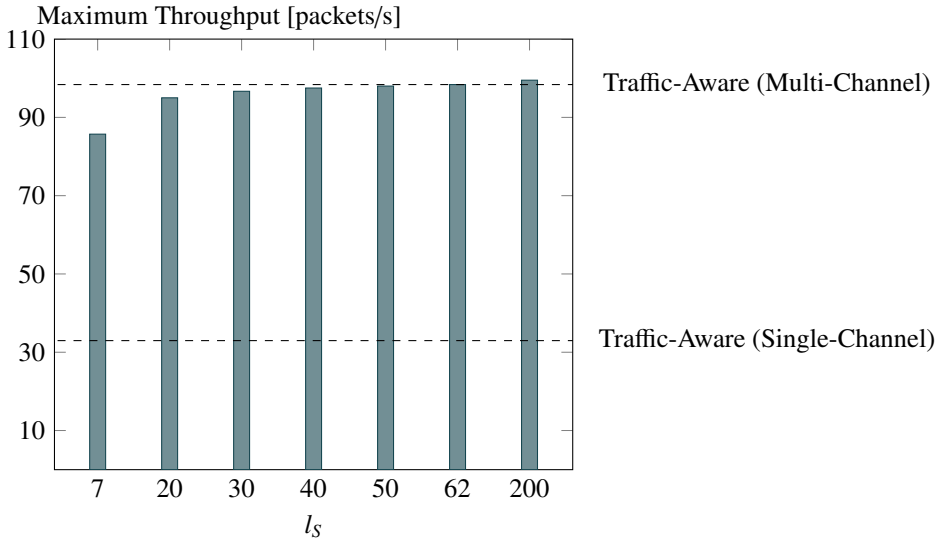


Figure 6.28: Fixed-Size Traffic-Aware Schedule applied to TSCH.

6.5.8 Comparison with DSME

The main difference between TSCH and DSME is the availability of distributed management procedures for setting up schedules in DSME, while they are left for the upper layers in TSCH. However, for the presented model that is irrelevant since it analyzes the schedule in a steady state. Thus, the only major difference is the slot structure. In TSCH, the time slots can be arbitrarily dedicated as contention-free and contention-access slots. DSME has a less flexible structure as presented in Sect. 3.2.3. The contention-access phase is usually used for management, similar to, but longer than the extra slot used for TSCH in the evaluation above. Secondly, the slots are usually shorter, because TSCH requires some extra time in every slot due to its time synchronization procedure, while DSME uses a dedicated beacon slot (see [JAG16] for a detailed comparison of the impact). Therefore, the timing of the model has to be adapted and the schedule has to account for the slots not used for contention free communication.

Furthermore, due to the less flexible structure, from the presented token-based scheduling techniques only the Fixed-Size Traffic-Aware Schedule (FTAS) can reasonably be applied to DSME since it can distribute excess slots evenly that would otherwise be left idle and thus reduce the throughput. For the following evaluation, a concentric topology with $N = 62$ nodes is applied. The PDR and throughput is analyzed for DSME with SO $= 3$, MO $= 5$ and a maximum queue length $K = 16$. It is compared to TSCH with multi-channel traffic-aware scheduling in Fig. 6.29.

As expected from Sect. 5.3.1, the network saturates faster when not using CAP reduction, due to the lower number of available time slots per time. With CAP reduction, the performance is slightly better than TSCH with the traffic-aware multi-channel schedule. For further analysis, the saturation level for different settings is compared in Fig. 6.30. The throughput in terms of packets is better for lower SO and thus shorter slots, but it does

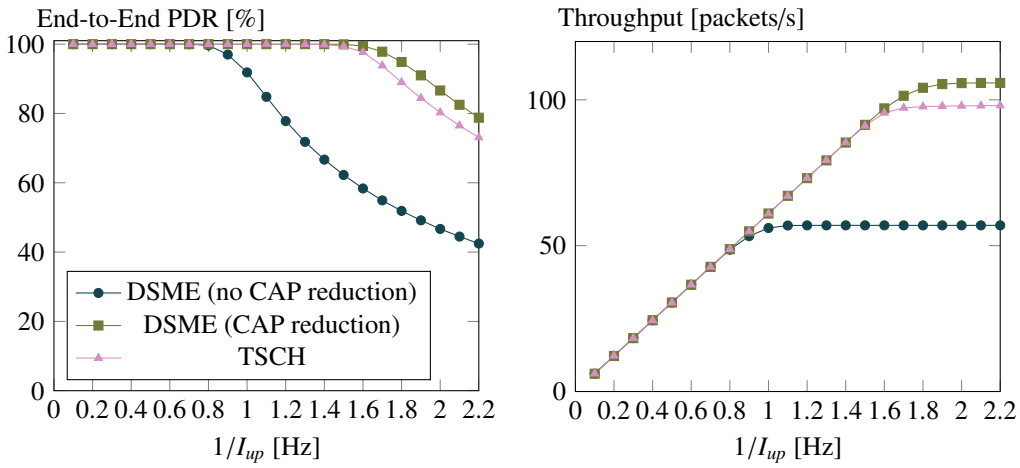


Figure 6.29: End-to-end packet delivery ratio and throughput for $N = 62$ and $K = 16$. For DSME the FTAS schedule is applied with SO $= 3$ and MO $= 5$. For TSCH, the traffic-aware multi-channel schedule is applied.

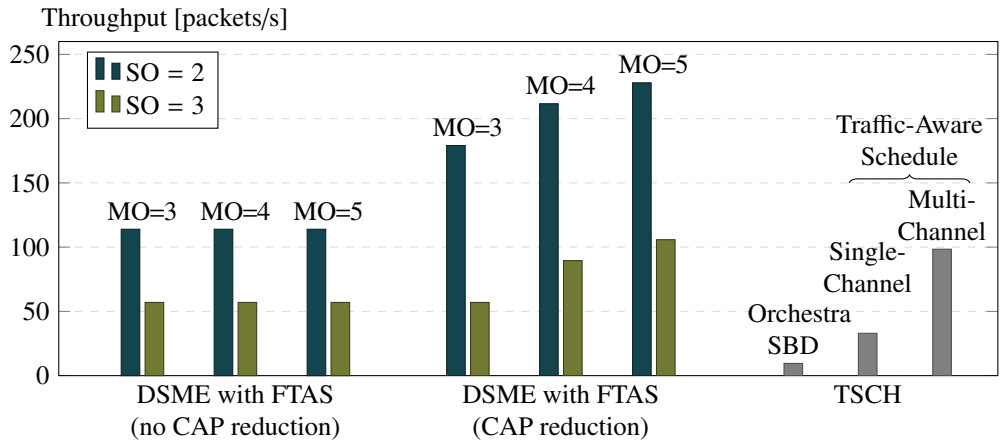


Figure 6.30: Saturation throughput for various DSME and TSCH schedules.

not allow for using the full packet length. Without CAP reduction, the MO has nearly no influence since the number of slots per time stays constant. However, with CAP reduction, the CAP consumes less overall time per multi-superframe for higher MO, leading to higher throughput. In contrast, TSCH has no explicit CAP, so while it can achieve comparatively high throughput in the given scenario, management traffic that is not considered in this evaluation will finally lower the throughput.

6.5.9 Extension to Other Sources of Packet Loss

As outlined before, most packet losses in TSCH and DSME networks are queue drops. However, in practice other sources of packet loss are inevitable. The consideration of packet loss during the transmission can be integrated in the presented model by replacing equation (6.103) with

$$\mu_{n,i}^{\mathrm{RX}} = \begin{cases} \mu_{\mathcal{K}_n(i),i}^{\mathrm{TX}} \left(1 - \mathrm{PER}_{L_{\mathrm{PSDU}},(\mathcal{K}_n(i),n)}(t)\right) & i \in \mathcal{R}_n \\ 0 & \text{else,} \end{cases} \tag{6.107}$$

where $\mathrm{PER}_{L_{\mathrm{PSDU}},(\mathcal{K}_n(i),n)}(t)$ is the probability that a packet transmission with a PSDU of L_{PSDU} bytes fails at time t.

In (3.4) on page 23, an example for a time-independent calculation of $\mathrm{PER}_{L_{\mathrm{PSDU}},(m,n)}$ can be found and in [MFS13] a model for Rayleigh-lognormal fading in IEEE 802.15.4 networks is given. Time dependence is required for accurate modeling of external interferences or fading channels, but that is out of the scope of this thesis.

Furthermore, since the model assumes no losses on the physical layer, retransmissions are not considered. If other sources of packet loss are considered, retransmissions have to be integrated in the model by adding transitions in the Markov chain for maintaining the queue level after a failed transmission. For the considered scenario, these transitions would never be taken and thus not change the results.

6.6 Discussion

This chapter proposes two analytical models for the analysis of medium access in multi-hop networks based on probability theory. Both a model for CSMA/CA and one for TDMA, including TSCH and DSME, are presented. While the first focuses on accurate modeling of collisions, the second one assumes a collision-free schedule and thus queue drops are the main source of packet loss.

The comparison with a simulation helps to identify shortcomings of either the model or the simulation. By this approach several interesting effects were identified. For CSMA/CA, it got evident that retransmissions fail with a much higher probability than the initial transmission, due to mutual disturbance of multiple nodes, leading to repeatedly overlapping retransmissions. It is expected that not resetting the backoff counter after a collision (as performed by the IEEE 802.11 DCF) might help a lot and improves the reliability. Also collisions of and with acknowledgments are important for accurate modeling. For TDMA, the probability distribution of forwarded traffic is identified to be significantly different than the traffic generation and thus the slot arrangement should be taken into account.

A further advantage of an analytical model is the possibility to get answers to a range of questions faster and more easily without the need to iterate over large parameter spaces. Thus it was possible to analyze a network with 10723 nodes to identify a per-node minimum packet generation interval of 38.8 minutes to achieve an end-to-end PDR of 99%.

When comparing the throughput behavior of CSMA/CA and TDMA we see that with TDMA a constant saturation level is reached where increasing the traffic generation will only lead to more queue drops but does not change the throughput. In contrast, for CSMA/CA no constant saturation level is reached. Instead the channel gets increasingly congested leading to more collisions and the overall throughput can even drop when increasing the rate. Due to the same reason, retransmissions only increase the reliability in unsaturated networks, while on the other hand they lead to faster saturation for lower traffic generation rates.

The comparison of different scheduling algorithms by means of the TDMA model highlights the importance of traffic-awareness to achieve maximum throughput. On the other hand, simpler schedules such as Orchestra SBD can be realized without the need for coordinating the schedule via message exchange. Furthermore, the analysis shows that the exploitation of spatial reuse and frequency diversity improves the performance significantly. The comparison of TDMA and DSME points out that the latter is less flexible and thus requires a more elaborate schedule, but can achieve comparable or even better performance by using shorter slots.

When comparing to a real world deployment, the results should be seen as an upper bound, since there are a lot more effects that are not captured by the model. This includes a very variable nature of the physical layer as well as limitations by the routing layer. Therefore, computing the model should be seen as the first step of deploying a large-scale wireless mesh network. If the reliability computed by the model is too low, the IEEE 802.15.4 standard is not appropriate for the intended application.

Overall, the proposed analytical models, together with their software implementation, are a useful tool for testing new ideas while developing new slot schedules and for practitioners who want to estimate the performance of wireless mesh networks.

Chapter 7

Experiments in Physically Deployed Networks

The use of testbeds consisting of actual wireless hardware is of major importance for development and evaluation of algorithms and protocols for wireless networks. While analytical considerations and simulations come with less initial investment, they can only partly reproduce the real world. The main reason is that propagation of radio waves is highly complex and even the most complex models can only cover parts of the actual mechanisms and are computationally expensive.

However, the deployment of wireless hardware comes with a high effort, especially when targeting large-scale multi-hop networks to be used in applications such as industrial plants [OBB13, PRM14]. Valuable overviews over the many expected and unexpected problems that have to be solved before starting the first measurement are given in [BIS08] and [LBV06] and even after that a constant maintenance is required. Especially providing sufficient electrical power and the possibility for updates (see 8.2) are non-trivial tasks that can constrain the convenient usage of testbeds for research.

Instead of setting up a new testbed it is therefore often advisable to resort to existing testbeds that provide convenient remote control interfaces. One popular example is the FIT/IoT-LAB [ABF15] that consists of multiple wireless network deployments ranging from 41 to 928 nodes of different types. Other remotely controllable testbeds include WISEBED [CFK09] and the MoteLab [WSW05].

This chapter presents experiments in the FIT/IoT-LAB to assess the applicability of openDSME in a physically deployed network. Since it is most interesting to analyze networks with a high number of hops, the chapter starts with the description of a technique for generating multi-hop networks in dense testbeds in a constraint area.

7.1 Constructing Customized Multi-Hop Topologies*

The number of nodes in a testbed such as the IoT-LAB is sufficient for many experiments. However, most deployments of the IoT-LAB span a comparatively small area where every transceiver can reach every other node in one hop, which does not allow evaluating complex routing algorithms, such as RPL [WTB12], or data link layers such as IEEE 802.15.4 DSME. The evaluation of such protocols requires dedicated topologies with specific properties. Furthermore, during debugging, very specific topologies can be helpful.

One obvious solution is to reduce transmission power and receiver sensitivity of the transceivers or to select a subset of nodes to generate more sparsely connected networks. By this approach it is also possible to setup specific conditions to induce certain phenomena. This is especially interesting to provoke and debug weaknesses of protocols. Relevant topology properties include density – to regulate channel utilization and spatial reuse –, a high number of hops or the (non-)existence of weak or asymmetric links. Building long chains is interesting to model tunnels [CCD11], grids for potato fields [LBV06] or tree-like structures for large-scale data collection [MMH12].

The main contribution of this section is an approach to construct topologies in existing sensor network deployments satisfying given conditions for evaluating multi-hop protocols based on link quality measurements. The focus is on the construction of subgraphs with uniform density as illustrated in Fig. 7.1 as well as on tree-like topologies with specified depth and breadth. Also applications of this method for deployments in the FIT/IoT-LAB are presented.

The implementation is published at https://github.com/koalo/iotlab_topologies to enable other researchers to generate suitable topologies for their experiments in the testbed of their choice. DOI 10.5281/zenodo.3148096

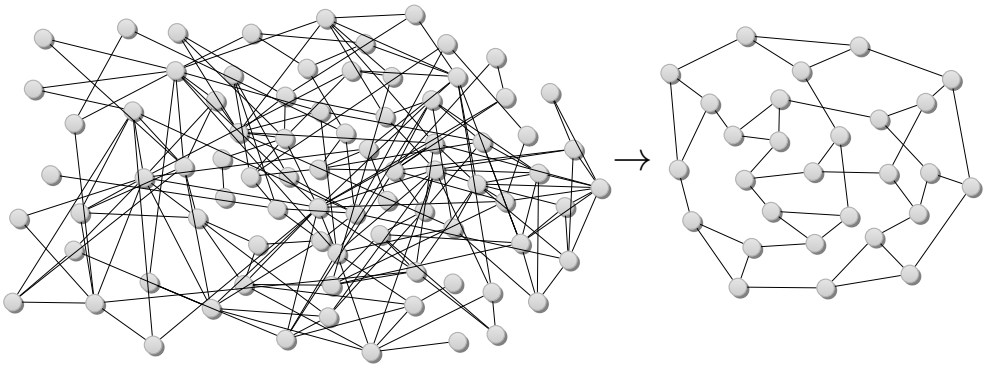


Figure 7.1: Construction of an induced subgraph with uniform density.

*This section slightly extends the previously published "F. Kauer and V. Turau, *Constructing Customized Multi-hop Topologies in Dense Wireless Network Testbeds* in Ad-hoc, Mobile, and Wireless Networks, Lecture Notes in Computer Science 11104, presented at the 17th International Conference on Ad Hoc Networks and Wireless (AdHoc-Now), St Malo, France, September 2018. https://doi.org/10.1007/978-3-030-00247-3_28" [KT18b].

7.1.1 Related Work

Several publications cover channel characterization in the IoT-LAB [BAR13, WAV15], including work on the applicability of RSSI measurements for localization [HV12] and with the focus on the repeatability of experiments [PGS16, PGS17]. The latter covers many aspects that are also relevant in the following, including transmission power selection for controlling the density and the selection of quality radio links. For this, a strategic approach is important, because the topology has a large impact on the performance, for example on delivery ratio and energy consumption, as discussed in [DHM14].

Outside the IoT-LAB, channel characterizations were conducted for example in complex factory environments [TWH07] and by evaluating the influence of antenna, mutual alignment and distance on the transmission between wireless sensor nodes [MMB07]. In the same publication, the influence of transmission power adjustment is discussed to minimize interferences. The adjustment of the transmission power for topology control is a broadly studied topic [RR00, LLV13, San05]. The latter also covers homogeneous transmission power adjustment for topology control and is therefore similar to the approach presented in the following, though the objective is different.

7.1.2 The Transceivers of the FIT/IoT-LAB

The primary node type in the IoT-LAB is the M3 Open Node. It consists of an ARM Cortex M3, an Atmel AT86RF231 [Atm09], a 2.4 GHz chip antenna and several other peripherals. For the purpose of the presented approach, the most important settings of the AT86RF231 are the transmission power and the sensitivity.

For the following, one should recall the transmission chain presented in Fig. 2.1 on page 7. For a AT86RF231 transceiver, the amplification can be set to one of 16 steps by the TX_PWR bits of register PHY_TX_PWR. It results in a transmission power ranging non-linearly from -17 dBm to 3 dBm. When the signal arrives at the receiver, the signal level at the LNA is estimated and this estimate is denoted as Receive Signal Strength Indicator (RSSI). The demodulation depends on the sensitivity setting RX_PDT_LEVEL. It controls a minimum RSSI level before the LNA that has to be reached in order for the signal to be processed. It ranges from -48 dBm to -90 dBm in 15 steps, excluding the special case 0, where all decodable packets are received and the sensitivity is stated as -101 dBm. The influence of the received signal power on the packet reception ratio is described in (3.4) on page 23.

The overall reduction of signal power between the PA of node a and the LNA of node b is denoted as $\Lambda_{a,b}$ in the following. For example, given a link with $\Lambda_{a,b} = 63$ dB, a communication will be possible for a RX_PDT_LEVEL of 9, corresponding to a sensitivity of -66 dBm if the transmission power is chosen to be larger than -3 dBm. If the sensitivity is not reduced, a good communication is even possible for the lowest transmission power of -17 dBm. For reference, Fig. 7.2 gives the full table of possible settings.

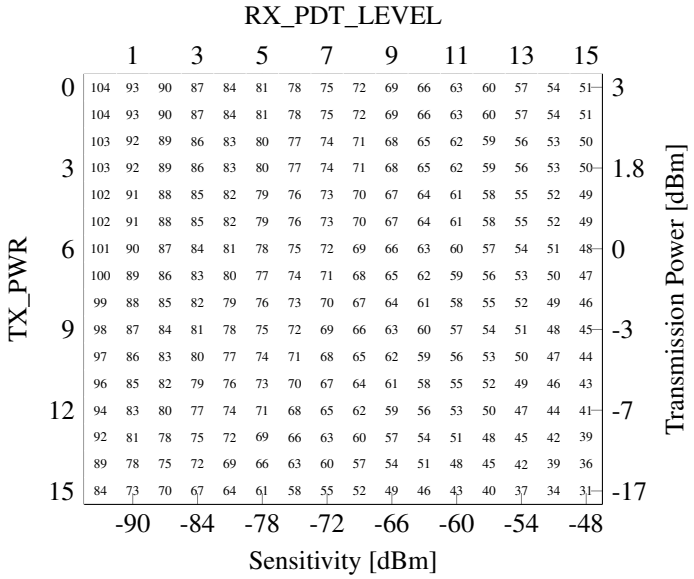


Figure 7.2: Path loss bound in dB depending on the TX_PWR and RX_PDT_LEVEL.

7.1.3 Measurements

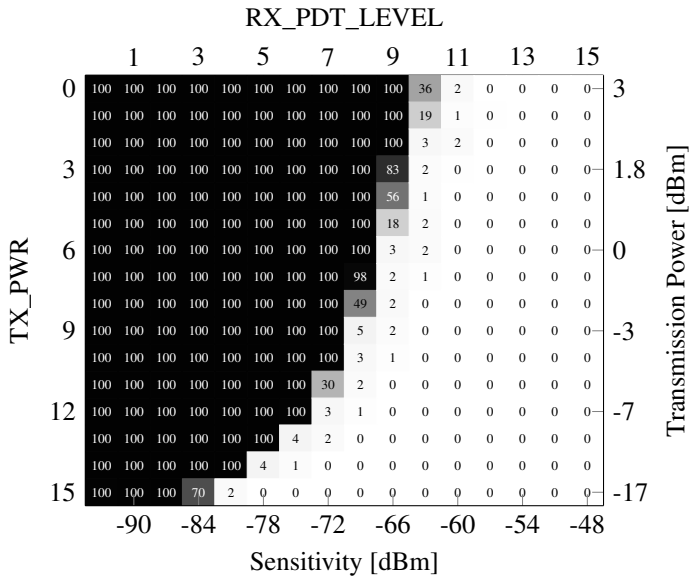
As first step, the influence of the transmission power and sensitivity on the transmission between two exemplary nodes is evaluated. One node repeatedly sends out packets, while the other one is in reception mode. Fig. 7.3a shows the ratio of received packets depending on the TX_PWR and the RX_PDT_LEVEL. Two distinct areas with nearly perfect and no reception are apparent, though a small transition area exists.

In Fig. 7.3b, the measured RSSI values are shown. Clearly, for the area of no reception, no RSSI values can be given. Apart from that, they only depend on the transmission power and are not influenced by the sensitivity. However, one can see clearly the functioning of the RX_PDT_LEVEL to avoid the reception of packets with a too low reception power.

7.1.4 Topology Generation Procedure

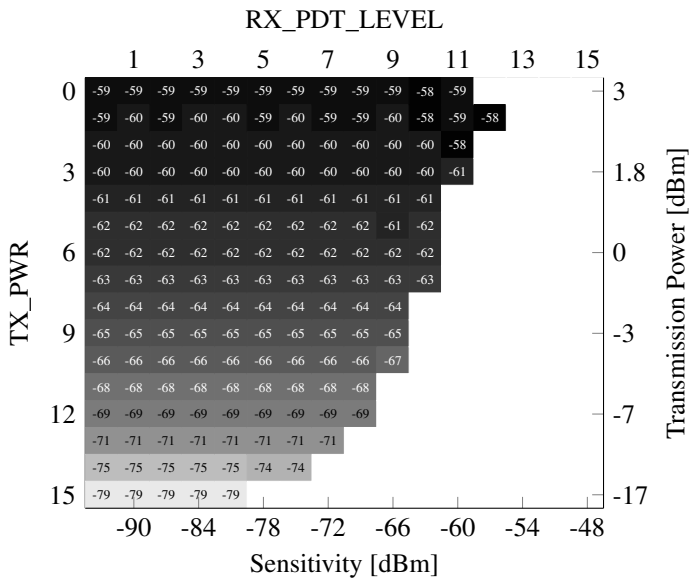
In the following, the proposed procedure for generating multi-hop topologies is explained and exemplarily conducted for the Saclay testbed of the IoT-LAB, more specifically the 12 M3 Open Nodes aligned in a 3x4 grid in the so called Digiteo 2 room. Compared to the other testbeds of the IoT-LAB it is relatively small and compact, so it allows for better traceability of the results in the following. Full results for the other testbeds are given in Sect. 7.1.5. The general procedure is as follows:

1. Measure $\Lambda_{a,b}$ between every pair of nodes.
2. Estimate the neighborhood graphs depending on the transmission power and sensitivity setting.
3. By means of these graphs, construct a topology by selecting a subset of the nodes and appropriate settings.
4. Verify this selection in the real testbed.



(a) Packet Reception Ratio [%]

,



(b) RSSI [dBm]

Figure 7.3: Measured packet reception ratio and RSSI between two nodes depending on the TX_PWR and RX_PDT_LEVEL. Note the non-linear dependency between TX_PWR and the resulting transmission power in dBm!

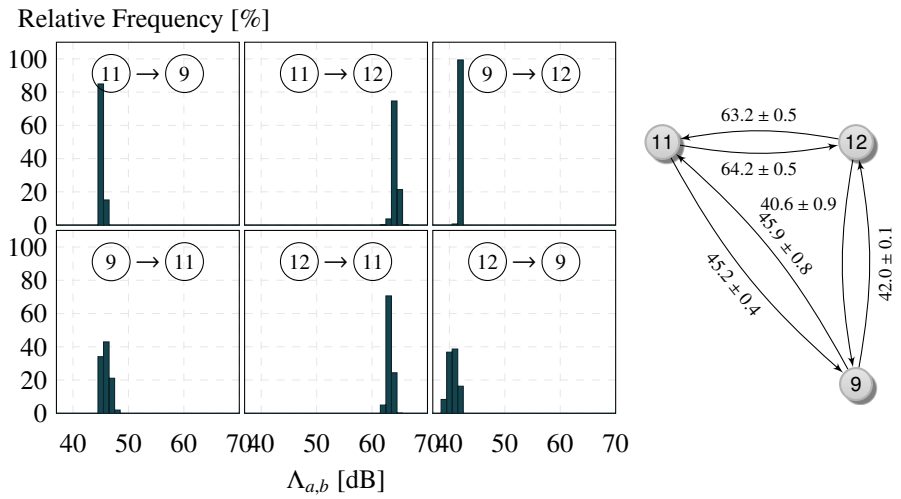


Figure 7.4: The distribution of the pairwise RSSI measurements for three nodes. The right part shows the associated graph of the nodes together with the resulting mean and standard deviation of the measurements.

Pairwise $\Lambda_{a,b}$ Measurements

The RSSI is used in the following to estimate $\Lambda_{a,b}$ between each pair of nodes in the deployments of the IoT-LAB. Please be advised that wireless conditions change rapidly and are hardly reproducible, especially when the conditions change, such as a window or a door that is opened. Therefore, to update the results recent measurements are required, but still the topology generation algorithm itself has to take these fluctuations into account. The measurement is conducted as follows. Every node repeatedly sends out packets with full transmission power. To reduce the probability of collisions, random intervals between the transmissions are chosen and CSMA/CA is used according to the IEEE 802.15.4 standard.

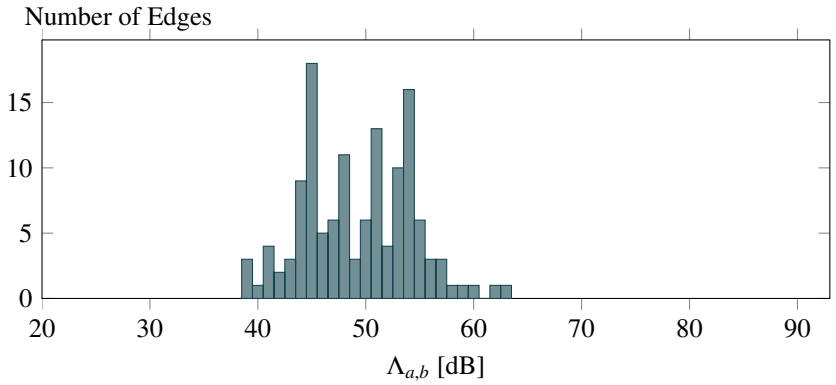


Figure 7.5: The distribution of the measured $\Lambda_{a,b}$ for the Saclay testbed.

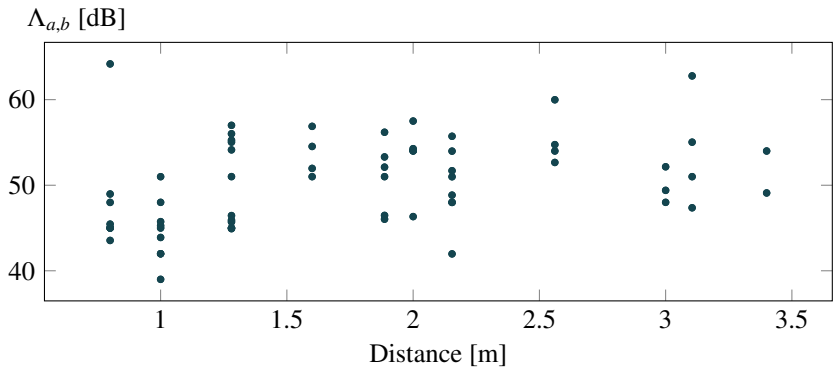


Figure 7.6: Measured $\Lambda_{a,b}$ in relation to the euclidean distance between the nodes showing only a slight correlation (Pearson correlation coefficient 0.38).

Every node in the neighborhood where the signal is strong enough to be received, measures the RSSI of the packet. By subtracting the RSSI from the constant and known transmission power, we get an estimate for $\Lambda_{a,b}$ between the respective nodes. This is repeated so that finally every node has sent at least 250 packets. It is important to measure on the same frequency channel as the one used in the final experiment. When multiple channels will be used such as in DSME, all relevant channels have to be taken into account.

Fig. 7.4 exemplarily shows the results for three nodes in the Saclay deployment. The left part of the figure depicts the distribution of the measured value for every pair of nodes, while the right part shows the mean and standard deviation of these measurements on the graph. Fig. 7.5 shows a histogram over all links between the 12 M3OpenNodes and in Fig. 7.6 the losses over the distance between the respective nodes are plotted, showing only a slight correlation with a Pearson correlation coefficient of 0.38. This is also the reason why generating a topology by picking seemingly fitting nodes in the map of physical locations rarely works well.

Neighborhood Graphs

Based on the data from the last section, the existence of usable bidirectional links for various link budgets can be estimated. An edge is added to the neighborhood graph if the measured path loss between two nodes for both directions is smaller than the given link budget. This gives a family $\boldsymbol{G}$ of graphs $G_{\hat{\lambda}} = \left(V, E_{\hat{\lambda}}\right)$, with $\hat{\lambda}_{min} \leq \hat{\lambda} \leq \hat{\lambda}_{max}$ and $E_{\hat{\lambda}} = \left\{(a,b) \,|\, \Lambda_{a,b} \leq \hat{\lambda} \wedge \Lambda_{b,a} \leq \hat{\lambda}\right\}$. Here, $\hat{\lambda}_{min} = 31\,\text{dB}$ is the smallest and $\hat{\lambda}_{max} = 104\,\text{dB}$ the largest bound that is possible with the AT86RF231. Fig. 7.7 shows the resulting graphs G_{42}, G_{46}, G_{50} and G_{104} for the Saclay testbed. Obviously, the number of edges increases with $\hat{\lambda}$. G_{104} is fully meshed, every node can reach every other node in a single hop. It is also apparent that there exist good links of long euclidean distance (e.g. $5 \leftrightarrow 12$), but also bad links with small euclidean distance (e.g. $11 \leftrightarrow 12$). Reasons for this include nonuniform antenna patterns, obstacles and reflections.

The analysis of these graphs will finally help to decide which transmission power and which sensitivity should be chosen for a suitable topology for evaluation of multi-hop

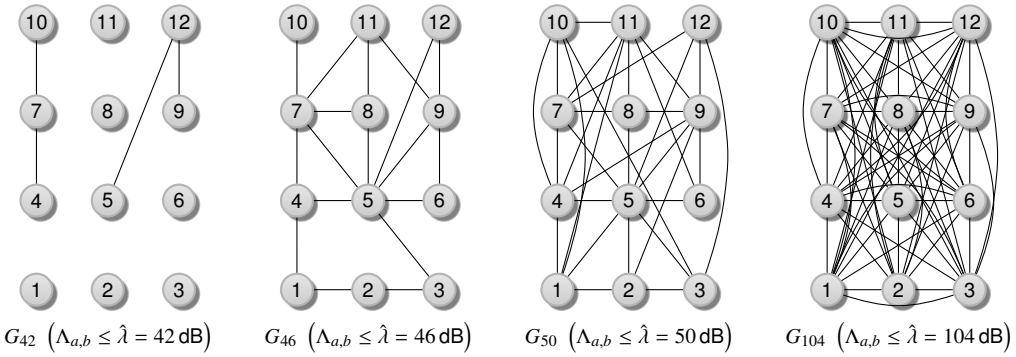


Figure 7.7: Selected graphs from $\boldsymbol{G}$ for the Saclay testbed.

topologies. One approach to this task is the analysis of the degrees $d(v)$ of the nodes in the network. In Fig. 7.8, the distribution of the node degrees is shown for each $\hat{\lambda}$ for the Saclay testbed. Beyond $\hat{\lambda} = 56$, all nodes have at least 9 neighbors, while below $\hat{\lambda} = 39$, no edges exist. The intermediate section is the most interesting for the given application. For example, in G_{46}, that is also shown in Fig. 7.7, one node has only one neighbor, five have 2 neighbors, four have degree 3 or 4, and for two it is between 5 and 8.

Constructing a Topology with Given Density

The task of this step is to decide for a subset of the nodes that form a topology suitable for the respective requirements at hand and a bound $\hat{\lambda}$, corresponding to the respective transmission power and sensitivity settings. While different settings per node may be feasible for some scenarios, a homogeneous $\hat{\lambda}$ is chosen to simplify the final realization and to avoid unexpected effects from heterogeneous transmission powers, such as asymmetric links. As stated previously, the requirements for this selection can be very diverse. While very dense graphs with few hops can be realized easily by selecting as many nodes as possible and using high transmission power and sensitivity, the construction of the longest chain of nodes requires solving a well known NP-complete problem, referred to as *induced path* [GJ79].

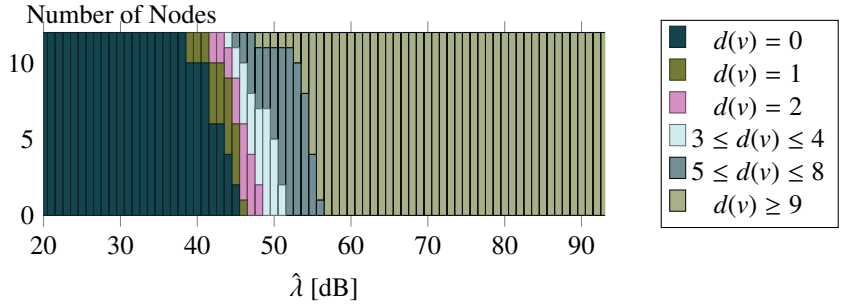


Figure 7.8: The distribution of the node degree over the bound $\hat{\lambda}$.

At first we consider topologies with constant node degree c, i.e., every node has c neighbors. This problem can be formulated and solved by integer linear programming (ILP). Let $\boldsymbol{x} \in \{0,1\}^{|V|}$ such that $\boldsymbol{x}(u) = 1$ if node u is selected and $\boldsymbol{x}(u) = 0$ if not. The set of neighbors of node u with a loss at most $\hat{\lambda}$ is given by $\mathcal{N}_{u,\hat{\lambda}} = \left\{ v \,\middle|\, (u,v) \in E_{\hat{\lambda}} \right\}$. For a given $\hat{\lambda}$, the ILP can be formulated as:

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{maximize}} && \sum_{u \in V} \boldsymbol{x}(u) \\ &\text{subject to} && c \cdot \boldsymbol{x}(u) \leq \sum_{v \in \mathcal{N}_{u,\hat{\lambda}}} \boldsymbol{x}(v) \leq c + (m-c) \cdot (1 - \boldsymbol{x}(u)), \forall u \in V. \end{aligned}$$

Here, $m = \max_{w \in V} \left(\left| \mathcal{N}_{w,\hat{\lambda}} \right| \right)$, so for $\boldsymbol{x}(u) = 0$ the condition reduces to $0 \leq \sum_{v \in \mathcal{N}_{u,\hat{\lambda}}} \boldsymbol{x}(v) \leq m$ and therefore always holds, while for $\boldsymbol{x}(u) = 1$, it is equivalent to $\sum_{v \in \mathcal{N}_{u,\hat{\lambda}}} \boldsymbol{x}(v) = c$. This ILP can be solved by a solver such as COIN-OR Cbc [Lou03] that is interfaced with the Python PuLP frontend in the implementation. By iterating over the $\hat{\lambda}$, we get a set of (not necessarily connected) subgraphs and can then select, for example, the largest connected component from these. Fig. 7.9 shows a topology for the Saclay testbed that is generated by this procedure. Also Fig. 7.1 is the result of applying this technique to the largest connected component for $\hat{\lambda} = 47\,\text{dB}$ and $c = 3$ in the Lille testbed.

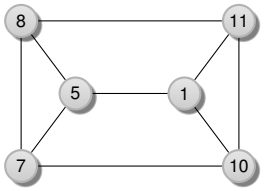


Figure 7.9: Topology generated by the first technique with $\hat{\lambda} = 49\,\text{dB}$ and $c = 3$. The positions do not represent the physical locations (cf. Fig. 7.7).

Constructing a Tree Topology

For many evaluations, especially for analyzing tree routing techniques such as RPL, a tree topology with a large depth is useful. The following properties allow for a versatile, yet easy to compute, construction of such tree topologies.

1. The graph is connected.
2. A node v_0 is designated as root, e.g., to serve as a RPL DODAG root.
3. The number of nodes that are reachable from v_0 over exactly δ hops, referred to as breadth in this paper, is at least $\kappa(\delta)$, being a predefined function. For example, with $\kappa(\delta) = \delta + 1$, the number of nodes per level increases, while $\kappa(\delta) = 1$ also allows for linear topologies.
4. There exist no links with $\Lambda_{a,b} \leq \hat{\lambda} + \Delta$ that would change the topology if the conditions change slightly; Δ is a margin to account for fluctuations. The robustness of the topology increases with increasing Δ.

Algorithm 5 Algorithm to construct a leveled subgraph for a given function κ

1: **procedure** MONITOREDBFS($V, E, v_0, \hat{\lambda}, \Delta, \kappa$)
2: $V_{sub}(0) \leftarrow \{v_0\}, \delta \leftarrow 0$
3: **do**
4: $V_{sub}(\delta+1) \leftarrow \{\}$
5: **for all** $u \in V_{sub}(\delta)$ **do**
6: **for all** $v \in \mathcal{N}_{u,\hat{\lambda}}$ **do**
7: **if** $v \notin \bigcup_{i=0}^{\delta+1} V_{sub}(i)$ **then**
8: **if** $\mathcal{N}_{v,\hat{\lambda}+\Delta} \cap \bigcup_{i=0}^{\delta-1} V_{sub}(i) = \emptyset$ **then**
9: $V_{sub}(\delta+1) \leftarrow V_{sub}(\delta+1) \cup \{v\}$
10: $\delta \leftarrow \delta + 1$
11: **while** $|V_{sub}(\delta)| \geq \kappa(\delta)$
12: $V_{sub}(\delta) \leftarrow \{\}, \delta_{\text{best}} \leftarrow \delta - 1$ { remove partially filled layer }
13: **return** $(\delta - 1, V_{sub})$

Algorithm 5 generates a subsets of nodes for given $\hat{\lambda}$ and v_0 that lead to the largest number δ_{best} of hops towards v_0. In the algorithm, $V_{sub} : \mathbb{N}_0 \rightarrow \mathcal{P}(V)$ associates a depth value with the selected nodes of this depth. A node in $V_{sub}(\delta)$ may have several neighbors in $V_{sub}(\delta-1)$ but none in $V_{sub}(l)$ with $l < \delta - 1$.

Procedure MONITOREDBFS is basically a breadth-first search starting from v_0 and thus guarantees requirement 1 and 2. The condition in line 11 monitors the number of nodes per level and aborts the search when it can not be continued without violating requirement 3. Finally, line 8 ensures requirement 4 by excluding nodes that would have been visited earlier when $\hat{\lambda}$ would have been selected slightly larger. It has to be noted that this algorithm does not necessarily find *the* best possible topology, because it might be possible to generate subgraphs with a larger depth by removing nodes a priori. Though, this again would lead to the longest induced path problem. MONITOREDBFS is called for all values of $\hat{\lambda}$ and all nodes v_0 to get a set of subgraphs which can be used to select, for example, a subgraph with the largest depth.

Node Reduction

Some applications require a large number of nodes, but usually it is advisable to reduce the size of the network to ease the analysis of a particular phenomenon as long as the behavior is unchanged. It should also not be forgotten that the FIT/IoT-LAB is shared with other researchers, so less nodes means less hindrance for others. Therefore, this section presents an optional procedure to reduce the number of nodes of the previously found subgraph, while still maintaining the requirements. It basically strips away all nodes that are not on a path to a higher depth with the additional constraint of maintaining $\kappa(\delta)$ nodes for depth δ. For a given generated layered subset of nodes V_{sub} with associated δ and $\hat{\lambda}$, this can again

be specified as ILP where we want to

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{u \in V} x(u) \\ \text{subject to} \quad & \sum_{u \in V_{sub}(i)} x(u) \geq \kappa(i) \quad \forall 1 \leq i \leq \delta \\ & \forall 1 \leq i \leq \delta, \forall u \in V_{sub}(i) : \\ & x(u) \leq \sum_{v \in V_{sub}(i-1) \cap \mathcal{N}_{u,\hat{\lambda}}} x(v). \end{aligned}$$

Verification

The output of the algorithm is based on measurements of the pairwise RSSI values which are fluctuating due to changes in the environment. Therefore, current measurements are necessary if a high accuracy is required and it is important to verify that the constructed topology in fact fulfills the given requirements before starting the actual experiment.

For this, the RPL TSCH example for the IoT-LAB is used that is included in the IoT-LAB Contiki fork. A topology found for the given measurements of the Saclay testbed and $\kappa(\delta) = \delta + 1$ consists of the nodes $\{4, 5, 7, 8, 10, 12\}$ with root $v_0 = 8$ for the bound $\hat{\lambda} = 46\,\text{dB}$. Due to the constraint size of the testbed, it has depth $\delta_{\text{best}} = 2$. It can for example be achieved by choosing a transmission power of $-17\,\text{dBm}$ and a sensitivity of $-63\,\text{dBm}$, since $-17\,\text{dBm} - (-63\,\text{dBm}) = 46\,\text{dB}$.

It is, however, important to consider the transition region between perfect and no reception. Thus, it may be necessary to increase the transmission power slightly or to improve the sensitivity. For this evaluation, the sensitivity was improved to $-66\,\text{dBm}$. Finally, we get the routing tree by requesting the RPL parent for each node. This results in the tree shown in Fig. 7.10 fulfilling the given requirements. Links to candidate neighbors that are not parents are not shown.

7.1.5 Testbed Comparison

After focusing on the Saclay testbed we consider other testbeds in this section. Tbl. 7.1 shows the result for applying the second approach to all testbeds. The larger testbeds,

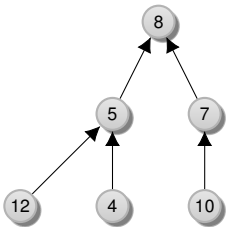


Figure 7.10: RPL routing tree for a transmission power of $-17\,\text{dBm}$ and a sensitivity of $-66\,\text{dBm}$. Topology generated by the second technique.

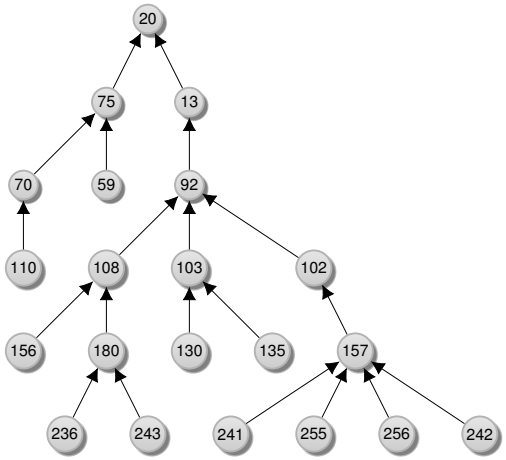


Figure 7.11: A RPL routing tree with depth 5 at the Lille site for a transmission power of 3 dBm and a sensitivity of −81 dBm.

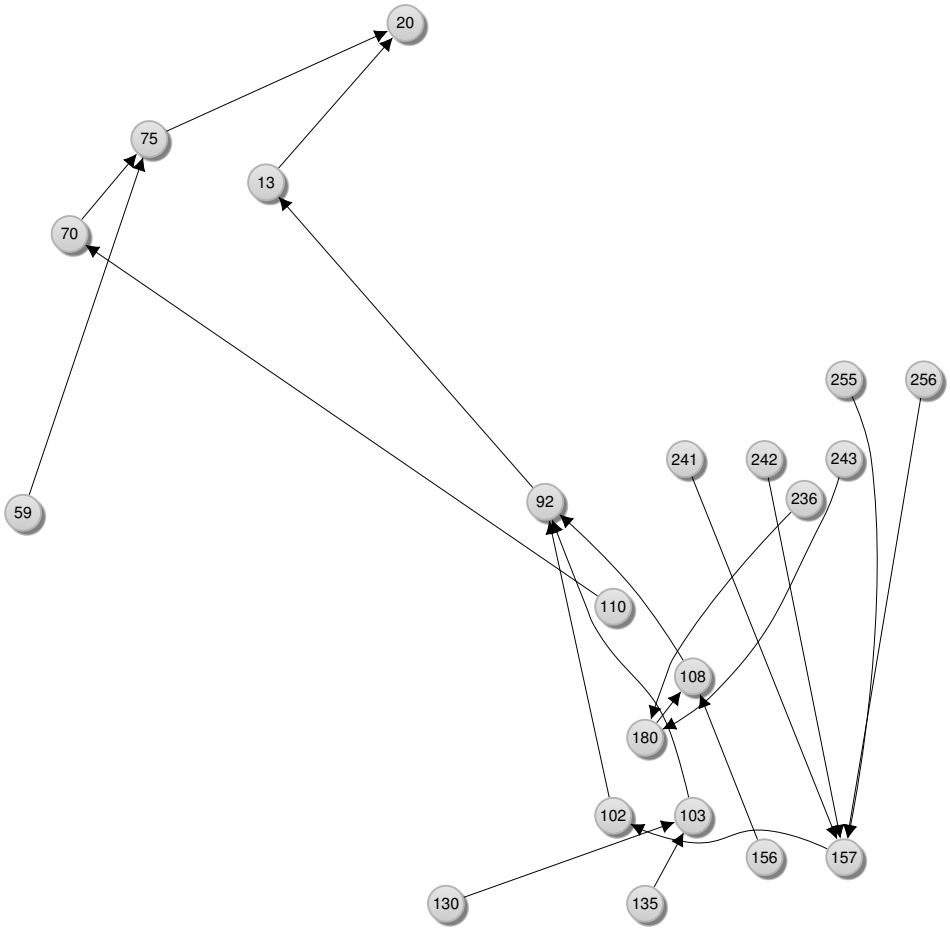


Figure 7.12: The tree of Fig. 7.11 with the physical positions of the nodes.

Testbed	Nodes (available)		$\kappa(\delta) = \delta + 1$			$\kappa(\delta) = 1$		
			$\Delta =$ 15 dB	$\Delta =$ 10 dB	$\Delta =$ 5 dB	$\Delta =$ 15 dB	$\Delta =$ 10 dB	$\Delta =$ 5 dB
Grenoble	364	δ_{best}	8	13	15	15	37	38
		$\hat{\lambda}$	73-74	70	69-73	71	69	61
Lille	229	δ_{best}	5	7	8	8	10	12
		$\hat{\lambda}$	70-74	73-74	64-70	64-74	64-66	67-69
Paris	69	δ_{best}	2	3	4	3	5	9
		$\hat{\lambda}$	46-66	49-53	48-51	46-49	49	48
Strasbourg	63	δ_{best}	2	3	4	3	4	7
		$\hat{\lambda}$	48-74	50-57	49-56	50-53	46-52	51-52
Lyon	17	δ_{best}	-	2	2	2	2	4
		$\hat{\lambda}$		54-58	49-63	45-60	45-65	51
Saclay	12	δ_{best}	-	2	2	2	3	4
		$\hat{\lambda}$		46	46-51	45-51	46	46

Table 7.1: Maximal achievable depth δ and the associated $\hat{\lambda}$ range in dB for the different testbeds of the FIT/IoT-LAB. In Lyon and Saclay, no topologies with at least two hops are possible for $\Delta = 15\,\text{dB}$ and $\kappa(\delta) = \delta + 1$.

Grenoble and Lille are in fact the only ones that are not fully meshed at full transmission power and sensitivity. With the previously stated requirements and $\Delta = 15$, the maximum achievable depth for $\kappa(\delta) = \delta + 1$ according to the measurements is $\delta_{\text{best}} = 8$ when setting $\hat{\lambda}$ to 74 dB in Grenoble. With a reduced $\Delta = 5$, the depth increases to 15, but the topology is less robust. Also, with $\kappa(\delta) = 1$ more hops can be achieved. Finally, Fig. 7.11 shows a resulting RPL routing tree in the Lille testbed and Fig. 7.12 depicts the physical positions of the nodes in this experiment with some links that would not be obvious based on the locations, demonstrating the benefit of the presented approach.

7.2 Evaluation of openDSME in the FIT/IoT-LAB

The main purpose of the evaluation in this section is to demonstrate that openDSME as introduced in Sect. 4.4 is in fact suitable for execution on wireless nodes as well as to present some properties that are best shown on a real system, such as power consumption. The approach presented in the previous section is used generating a line and a tree topology used for the following evaluation.

As operating system, Contiki with 6LoWPAN stack is used where the CSMA MAC is replaced by openDSME. This entails the use of RPL for routing. Since we target maximum reliability and stable network conditions, RPL is configured to use the Minimum Rank with Hysteresis Objective Function (MRHOF) with ETX metric.

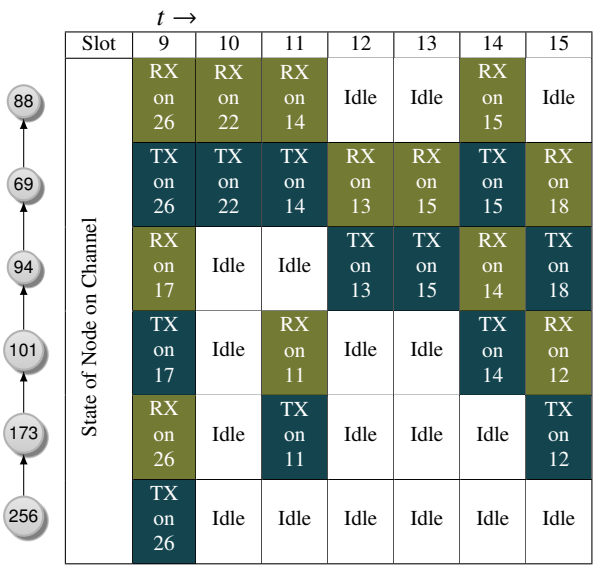
7.2.1 Schedule

In Fig. 7.13 a line topology with 6 nodes in the Lille testbed is shown together with an emerging schedule. Every node (apart from the sink 88) sends packets on average every 500 ms towards the sink. As introduced in Sect. 4.3, the nodes allocate slots based on the amount of traffic over the link. No CAP reduction is applied and SO = MO = 4, so there is only one superframe per multi-superframe with a duration of $2^4 \cdot 960$ symbols $\cdot\, 16 \frac{\mu s}{\text{symbol}} = 245.8$ ms according to (3.6) on page 27. So one slot is sufficient for transmitting on average about 4 packets per second. When applying the hysteresis to account for times of momentarily higher traffic, the shown schedule emerges. The higher the number of nodes that route over a given link, the more slots are allocated.

It is also apparent that slots are reused. For example, time slot 9 (the first slot in the CFP), is used on channel 26 for the link 69 → 88, while it is used at the same time on channel 17 for link 101 → 94 as well as for 256 → 173, again on 26. This is valid, since the transmission from 256 is out of range for node 88.

7.2.2 Network Formation

As explained in Sect. 3.2.3, the nodes first need to synchronize to a beacon, then associate to the network. In the second phase, they potentially become coordinators on their own to pass on the time synchronization. In openDSME, the following procedure is implemented to maintain an even distribution of the coordinators. An associated node that is not a coordinator (so far), will count the number of coordinators it recognizes in the neighborhood. If less than two beacons were received, it randomly decides to become coordinator itself with a chance of 1/3 every beacon interval. Also, when a node was performing a passive scan for



Slot	9	10	11	12	13	14	15
88	RX on 26	RX on 22	RX on 14	Idle	Idle	RX on 15	Idle
69	TX on 26	TX on 22	TX on 14	RX on 13	RX on 15	TX on 15	RX on 18
94	RX on 17	Idle	Idle	TX on 13	TX on 15	RX on 14	TX on 18
101	TX on 17	Idle	RX on 11	Idle	Idle	TX on 14	RX on 12
173	RX on 26	Idle	TX on 11	Idle	Idle	Idle	TX on 12
256	TX on 26	Idle	Idle	Idle	Idle	Idle	Idle

Figure 7.13: Line topology with associated emerging schedule in the hardware experiment.

too long without receiving any beacon, it will switch to active scan and send out a beacon request. A node receiving a beacon request will also turn into a coordinator. This procedure minimizes the number of coordinators and allows for very dense networks while lowering the energy consumption and ensuring a connected network.

In the shown line topology, all nodes have to become coordinators to enable the functioning of the network. Fig. 7.14 demonstrates the associated timing. Since the parent of a node has to become coordinator before its child can synchronize to it, the shown waterfall shape emerges. After about 40 seconds all nodes are associated. This time is influenced by several parameters such as the scanning duration and the beacon interval.

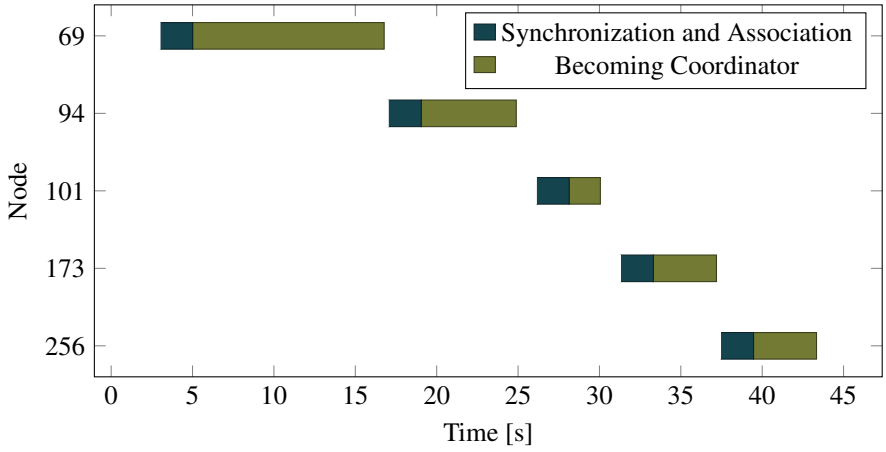


Figure 7.14: Network formation in the FIT IoT-LAB line topology experiment.

7.2.3 Energy Consumption

Besides the high reliability as demonstrated in Sect. 5.3, DSME has the advantage of consuming less energy than an always-on CSMA/CA MAC, because the transceiver can be turned off during unused slots in the CFP. While it is not explicitly targeted to consume as few power as possible, such as low-power listening approaches [PHC04], the consumption can be reduced significantly as shown in this section.

For this experiment, a tree topology with 10 nodes is built. Every node sends a packet on average every 2 seconds to the PAN coordinator. Fig. 7.15 shows one resulting RPL routing tree, annotated with the average power consumption of each node for an exemplary run with $SO = 4$, $MO = 6$ and enabled CAP reduction. Of course, since RPL is a dynamic routing protocol and the environmental conditions can change, the routing tree is different for other runs. The stated power is the additional power used by the transceiver, excluding the power for the CPU and other peripherals. For getting these values, the idle consumption of each the M3OpenNodes as published in [Kau18] is subtracted from the total power consumption. While the figure only represents a single run and the results are thus quite diverse, it is already apparent that the nodes with a high number of connections have a higher power consumption due to the higher number of reserved slots.

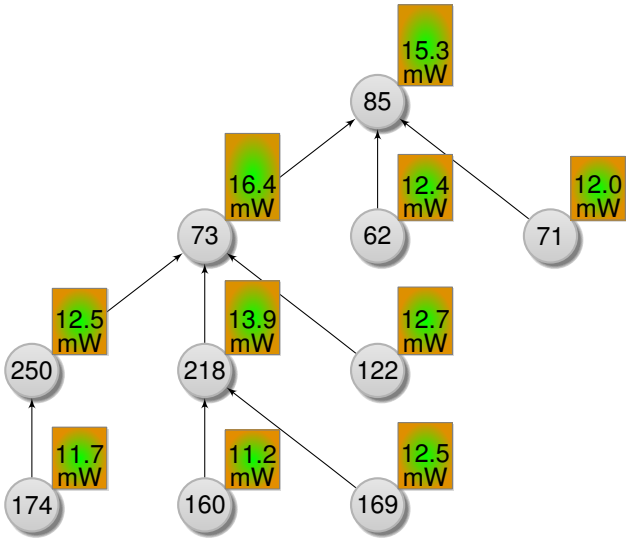


Figure 7.15: Tree topology and results for a single scenario.

In order to get more significant results, this experiment was repeated for various MO settings, with and without CAP reduction as well as for an always-on CSMA/CA MAC and every setup was repeated ten times. Fig. 7.16 shows the resulting power consumptions, averaged over the network, together with the 95% confidence intervals. The power consumption in the DSME experiments without CAP reduction is about 10 mW smaller than the one for the CSMA experiment. However, the influence of MO is small and can be explained by the higher granularity. For $MO = 4$, the nodes consume the same amount of power, regardless of the CAP reduction, because there is only one superframe per multi-superframe anyway. Though, the skipped CAP for the other MO settings leads to a significant reduction in power consumption.

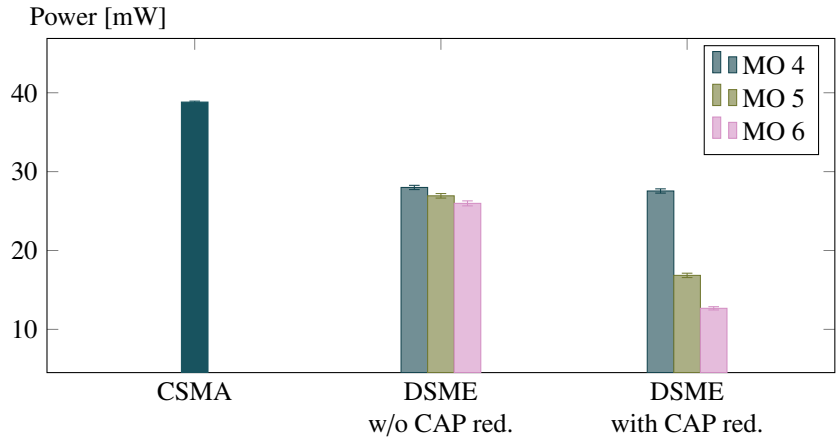


Figure 7.16: Measured power consumption for different scenarios.

7.3 Conclusion

This chapter presents experiments in a physically deployed wireless testbeds, namely the FIT/IoT-LAB. First, an approach is proposed to generate multi-hop topologies in dense wireless network testbeds. The well-known fact that in common wireless sensor network settings, the received signal strength only correlates slightly with the distance can be verified for the FIT/IoT-LAB testbed. This makes it difficult to handpick reasonable nodes and settings for executing experiments. Therefore, channel condition measurements are conducted to estimate neighborhood graphs depending on the transmission power and sensitivity settings. As expected, the density of the resulting graph decreases with lower transmission power and reduced sensitivity.

These measurements form the starting point for a constructive algorithm to generate tree topologies with a custom minimum number of nodes per depth as well as an ILP for reducing the number of nodes afterwards. Finally, it is shown that by the proposed approach the number of hops can be increased significantly in contrast to the default settings, where - except for Grenoble and Lille - only single-hop topologies are possible.

This approach is used to generate a line and a tree topology for evaluating openDSME. For this, openDSME is integrated in Contiki and combined with the 6LoWPAN stack, demonstrating its portability. An exemplary schedule that is generated by the traffic-aware and prediction-based slot scheduling is shown, also demonstrating the ability of spatial reuse of slots. Furthermore, the network formation process is analyzed and the reduced power consumption of openDSME compared to an always-on MAC is demonstrated. Thus, IEEE 802.15.4 DSME is not only interesting for achieving high throughput in large networks, but also for applications with very few traffic where long superframe structures can be used to significantly prolong the lifetime of battery powered wireless nodes.

Chapter 8

Realization of Real-World Deployments

After analyzing the proposed system formally, with analytical models, by using a simulator and a physically deployed testbed, the last step is the deployment in an actual application. In Sect. 8.1, the setup of the research project AutoR[1] is presented. Also, further aspects important for industrial applications are covered. This includes the ability to reprogram the deployed hardware over the wireless interface addressed in Sect. 8.2. Furthermore, while the comparatively high delay of the multi-hop communication is not an issue for monitoring and calibration tasks in the solar tower power plant, the requirement to shut down the plant in case of emergencies calls for further attention leading to a supplementary radio system presented in Sect. 8.3.

8.1 System Integration

One of the main pillars of the AutoR project was the implementation of a wireless field control for solar tower power plants with up to several hundreds of thousands of solar-powered heliostats.[2] According to [KJD07, Fig. A-6], the wiring costs for a single heliostat are in the order of several hundreds of dollars, so using wireless technology promise a large cost reduction, provided that the costs for the wireless hardware are small and the energy consumption of the heliostats is low enough so they can be powered autonomously, for example by photovoltaic cells. Other topics of the AutoR project include the analysis of wind loads and improvements of the design of heliostats as presented in [PRM14]. In this section some challenges are presented to demonstrate the step from the theoretical analysis towards a real-world application.

The main task of a heliostat is to focus sunlight on a central tower, where the heat is used to power a turbine. For this, the sun has to be tracked and the mirrors have to be aligned correspondingly. In order to adapt to changing conditions, such as wind and clouds, the heliostats can also be defocused to reduce the absorbed heat at the receiver. This is

[1] Full German title: "Autonomer Leichtbau-Heliostat mit Radkranzantrieb"

[2] The author of this thesis was mainly involved in the construction of the radio hardware and the implementation of the related software.

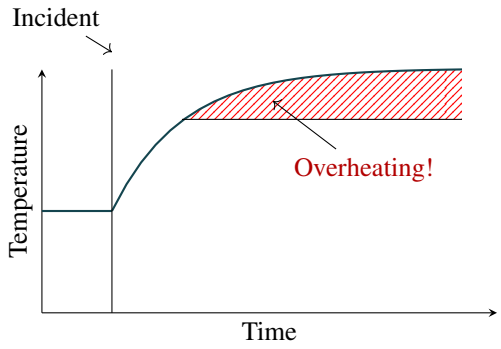


Figure 8.1: Overheating of a thermal receiver after an incident according to [FMF15].

especially important after certain incidents such as component failures, when the heat can not be dissipated fast enough, so the receiver overheats, leading to possible damages [FMF15]. Fig. 8.1 shows an exemplary temperature course that has to be mitigated by moving the heliostats to defocus the sunlight within seconds. A similar requirement occurs for emerging storms where the heliostats have to take a horizontal position to minimize the area exposed to the wind.

From the experiences from the preceding project, HelioMesh [KRB11], it was clear that a wireless mesh network will not be able to deliver individual motor control commands to every heliostat with a high rate. Thus, a decentralized approach was implemented where every heliostat is able to autonomously track the sun. By this approach it was possible to reduce the requirements to the following four communication patterns:

Individual Heliostat Control for setting and requesting parameters of heliostats. This includes calibration and status notifications, primarily for predictive maintenance. These commands and notifications require only small payloads of up to 40 byte and require bidirectional communication.

Field Control for controlling the whole field at once. This is mainly used for power control of the plant. This communication is unidirectional and only requires small payloads.

Emergency Shutdown for defocusing of the mirrors in case of an emergency condition, such as a broken valve, reliably within two seconds.

Firmware Update in case the software running on the embedded systems of the heliostats needs to be fixed or improved. The size of a firmware image can be in the range of several hundreds of kilobytes.

These requirements are quite diverse. For the individual heliostat control and the field control, a wireless mesh network using IEEE 802.15.4 is very promising according to Sect. 2.7 because it supports bidirectional communication, has a low power consumption and can be implemented cost-efficiently. Furthermore, 40 bytes is an adequate payload for such a network. Due to the very high number of heliostats, reliable channel access is essential. For this, the medium access layer proposed in Chapter 4 is promising.

Also, the firmware update task can be performed with this network. While a firmware image is far too large for a single IEEE 802.15.4 frame, there are proven and tested mechanisms for fragmenting and disseminating firmware images in wireless mesh networks. The application of such a technique to the given scenario is presented in Sect. 8.2.

However, with such a network it would be difficult to ensure the requirements for the emergency shutdown. Especially the comparatively high latency of IEEE 802.15.4 DSME is obstructive, but even with CSMA/CA the network would be fully loaded with the emergency shutdown task as described in [UT12b] and can not reliably perform other tasks.

8.1.1 Dual-Radio Approach

In order to tackle these opposing requirements, the usage of two independent radio technologies is proposed consisting of a IEEE 802.15.4 mesh network for bidirectional communication as well as a unidirectional radio system that can reach every node in a single hop. The mesh network operates in the 2.4 GHz band, while the unidirectional system operates in a Sub-GHz band to achieve a longer range as well as preventing interferences between the systems. The application of the dual-radio approach is illustrated in Fig. 8.2. The usage of two radio technologies is profitable, since the requirements of an emergency shutdown, especially the unidirectional communication and low data rate, are opposed to other tasks that require high data rate and bidirectional communication with lower timing requirements, such as collecting measurement data, that are more suitable for a wireless mesh network.

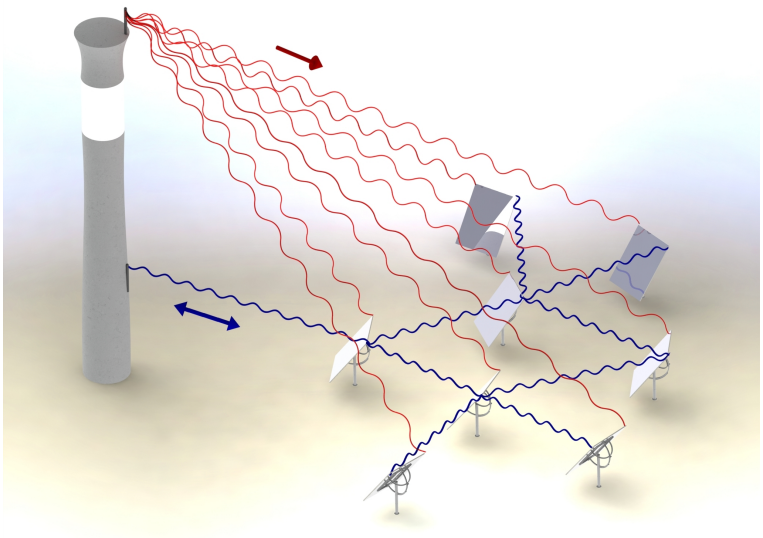


Figure 8.2: Illustration of the dual-system approach in a solar tower power plant. The lower antenna interfaces an IEEE 802.15.4 mesh network that allows for bidirectional communication, while the upper one regularly transmits a keep-alive signal that is received by all components.

The combination of multiple radios was already suggested in [SBH08]. In this paper an IEEE 802.11 radio was combined with a 2.4 GHz low-power transceiver that wakes up the energy-intensive IEEE 802.11 radio in case of incoming traffic. Several other hardware platforms combine IEEE 802.15.4 transceivers with transceivers operating in Sub-GHz band, for example BTnode [BDH04], or two IEEE 802.15.4 transceivers, for example Opal [JKK11]. While these devices are similar to the receivers presented in this paper, they use the multiple radios mainly for reducing energy consumption and increasing throughput. In contrast, the main goal of the presented approach is to provide dependability and low latency in very large deployments for critical applications. Details and evaluations of the emergency shutdown system are presented in Sect. 8.3.

8.1.2 Hardware Architecture

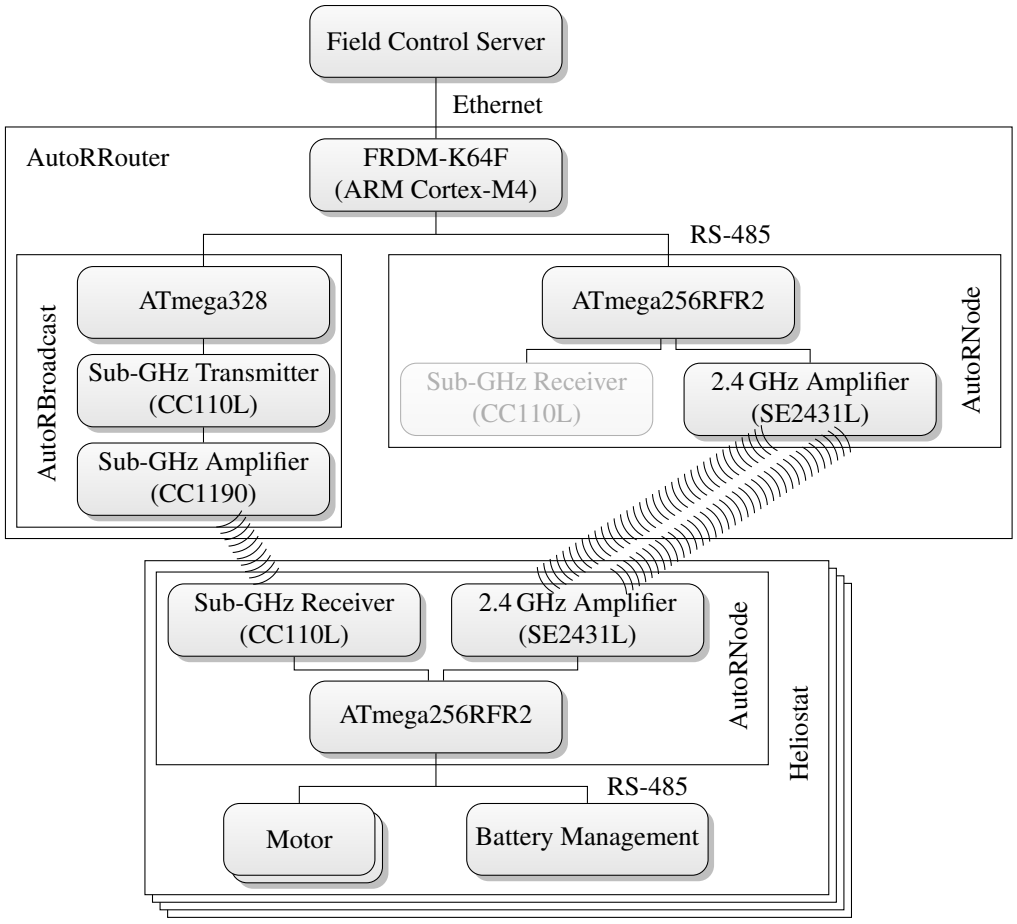


Figure 8.3: Communication Hardware Architecture of the AutoR Project.

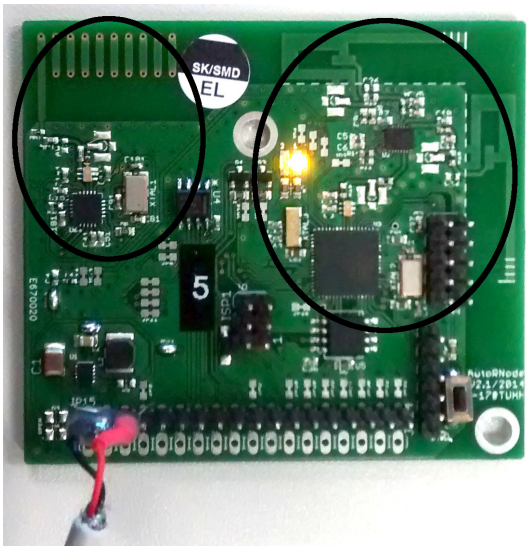


Figure 8.4: An AutoRNode with IEEE 802.15.4 transceiver (right part) and Sub-GHz receiver (upper left part).

In this section, the communication hardware architecture of the AutoR project is briefly introduced. It is depicted in Fig. 8.3 and consists of the AutoRRouter that is connected via Ethernet to the field control server and wirelessly to the individual heliostats. The main communication component on every heliostat is the AutoRNode shown in Fig. 8.4. It includes an ATmega256RFR2 system on chip [Atm14] with an 8-bit microcontroller and an IEEE 802.15.4 transceiver similar to the AT86RF231 used in the M3 Open Nodes (see Sect. 7.1.2). A CC110L transceiver by Texas Instruments is used for the emergency signalization as described in Sect. 8.3.4 and a Skyworks SE2431L is used as 2.4 GHz RF frontend. Furthermore, external Flash memory is provided for supporting firmware updates. The AutoRNode communicates with the other components on the heliostat via RS-485.

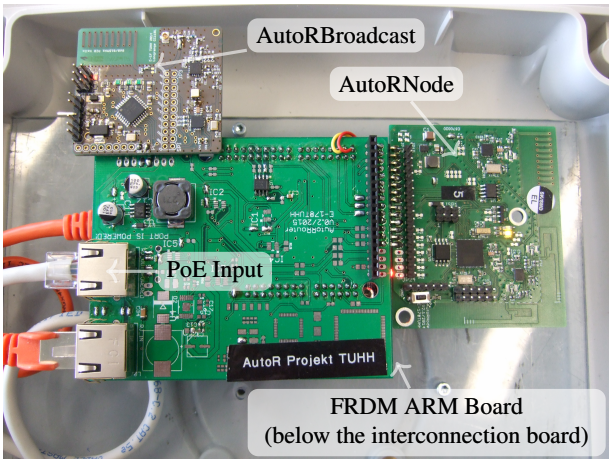


Figure 8.5: The interior of the AutoRRouter.

The central gateway, denoted as AutoRRouter and shown in Fig. 8.5, is also equipped with an IEEE 802.15.4 transceiver for communication with the mesh network. For simplicity, this functionality is provided by an AutoRNode integrated in the AutoRRouter with disabled CC110L. The Sub-GHz transmissions for the emergency signalization are performed by another component, denoted as AutoRBroadcast, consisting of an ATmega328 microcontroller, a CC110L and a CC1190 RF amplifier. These components are coupled via RS-485 with an ARM-based hardware component that is responsible for the communication with the field control server.

The higher costs for the Sub-GHz transmitter including the amplifier, compared to the Sub-GHz receiver, is only needed once, while the receiver can be realized cost efficiently because most functionality such as control by the microcontroller and power supply can be shared with the IEEE 802.15.4 part. Thus, the costs for the Sub-GHz receiver including peripherals are only 20% of the overall costs of an AutoRNode. The same holds for the occupied PCB area. Therefore, the additional cost per deployed node compared to a pure IEEE 802.15.4 network is comparatively low but comes with a huge gain in terms of functionality.

8.2 Over-the-Air Software Updates*

In this section, a concept for remote software updates of all involved components in a heliostat is presented. Three major problems exist in the considered scenario. The maximum data rate is in the order of 250 kB/s. Therefore, updating one device at a time is not feasible. Secondly, an IEEE 802.15.4 data frame can not consist of more than 127 Byte. This is sufficient for most control and sensing messages, but a firmware image in the order of 150 kB has to be fragmented. Last, but not least, when using CSMA/CA as MAC layer, the reliability of packet transmissions is significantly reduced due to the high probability of packet collisions. This can be tackled by techniques that avoid collisions on the channel, for example by using DSME, but nevertheless algorithms for retransmitting corrupted parts on a higher layer are beneficial.

Several algorithms exist that support fragmentation and retransmissions for firmware distribution in mesh networks. The most popular is Deluge [HC04], also due to its implementation in the TinyOS framework that found widespread usage. Many improvements were suggested, for example MNP [KW05] that targets lower energy consumption or REACTIVE [CPC14] that optimizes the interplay of Deluge and Low Power Listening. Other approaches use forward error correction for achieving less retransmissions [UT12a]. A comprehensive survey of the existing algorithms is provided in [XHS17].

However, firmware updates include more than just bulk data dissemination as pointed out in [BS13]. Providing an overall framework that supports convenient user interaction as well as handling of failures and recovery is equally important. A corrupted firmware update can

*An extended version was previously published as "F. Kauer, F. Meyer, and V. Turau, *A Holistic Solution for Reliable Over-the-Air Software Updates in Large Industrial Plants* in Proceedings of the 13th Workshop on Intelligent Solutions in Embedded Systems (WISES), Hamburg, Germany, June 2017. https://doi.org/10.1109/WISES.2017.7986928" [KMT17], also analyzing the influence of the time for writing to the flash memory on the performance.

render the complete network useless. Apart from providing forward error correction, the main focus of [UT12a] is to provide a fail-safe fallback for such a situation.

This section focuses in particular on the ability to conduct firmware updates on complex equipment such as a heliostat that consists of several interconnected microcontrollers. This is especially important in real-world applications where not every microcontroller can or should support a full communication stack due to performance reasons or proprietary devices. The requirement to update third party devices also restricts the usage of modular reprogramming as in [MAL15].

8.2.1 System Overview

The system is optimized to support a huge number of homogeneous devices. However, updates of individual devices or devices consisting of multiple microcontrollers is fully supported since the actual update process is separate from the dissemination and under full control of the control room.

The basic procedure is depicted in Fig. 8.6. It consists of three phases: Upload of the new software to the IEEE 802.15.4 gateway, that is the AutoRNode inside the AutoRRouter, data dissemination and local firmware update. At first, the heliostats have to be set to maintenance mode. This is important since while updating a device in the plant, for example a motor controller, it will not be able to maintain its operation of controlling the heliostat alignment. Wind gusts can, for example, lead to uncontrolled movement of the heliostats,

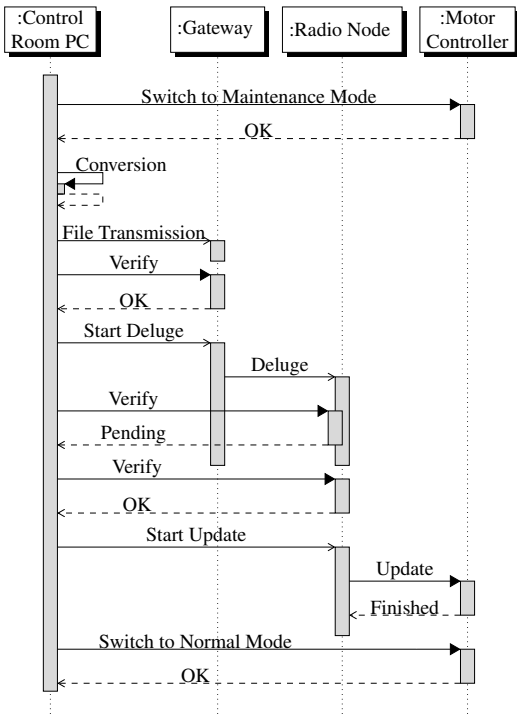


Figure 8.6: Sequence Diagram of a Firmware Update Procedure.

inducing a risk for workers in the field and the integrity of the mirrors. In order to enter this mode, the heliostat will move to a horizontal position and fasten a remote-controlled, mechanical lock to prevent unintentional movements. Afterwards, the user loads a new program file into the system at the control room. It is then converted into a more compact format that will be distributed in the network.

The characteristics of the connection between the control room and the gateway can be different from the mesh network. For example, it might be a wired connection or even a remote connection over the Internet. Thus, a two-step transmission is used where the firmware is first transmitted point to point from the control room PC to the gateway of the IEEE 802.15.4 network. The file is stored on the gateway using a persistent flash storage and the Coffee File System [TDH09]. After the transmission is finished and verified, the Deluge algorithm is started that distributes the file in the network in a broadcast fashion.

The dissemination will take some time as estimated in Sect. 8.2.3. The control room application will actively request the state by issuing a verification, starting from the outermost nodes not before the estimated time is up. Therefore, it has complete control over the amount of status messages transmitted, thus avoiding additional congestion on the channel. After the file is stored and verified on every device in the network, the IEEE 802.15.4 node will be commanded to update the respective devices. If the node itself is updated, it will use the procedure described in [UT12a]. Otherwise, it will use the local, wired data bus on the heliostat to update the external components. From the point of view of the external component, the update works just as it would work with an In-System-Programmer attached to the local bus. After a while, the user will set the system back to normal operation. Failure to do so, indicated by a respective message, will be handled by restarting the update, either for the whole network or a single device.

8.2.2 Building Blocks

File Format and Verification

The first step of the procedure is the conversion of the provided firmware image from the Intel HEX format into a format that consists of the following parts:

4 Byte File Signature (0x7ddda4d5) for file identification
4 Byte Start address of the firmware to be programmed in the target memory
4 Byte End address
2 Byte Device address of the internal heliostat data bus
2 Byte Device specific checksum for verifying the update
x Byte Firmware image in binary format
1 Byte End byte (0xac)

The main motivation of this conversion is the more compact representation of the firmware. A 400 kB Intel HEX file can be reduced to a file of about 150 kB, without explicit compression, just by using a binary representation instead of an ASCII representation. The length **x** can be calculated by subtracting the start address from the end address. The end byte can not be used for finding the end, because it might also be present in the firmware itself. However, the Coffee File System requires a non-zero byte at the end of

every file, so the end byte ensures this property. The checksum is used to verify the correct execution of the local firmware update and depends on the update procedure of the device.

A second checksum is used to verify the correct transmission of the overall file from the control room to the radio node. It is calculated as a cyclic redundancy check with the polynomial $x^{16} + x^{12} + x^5 + 1$ during the file conversion and displayed separately. A verification procedure is started by sending the checksum from the control room to the gateway or radio node. The same checksum calculation will then be applied to the locally stored file and the result is compared. It will be signaled back if the checksums match. Also, if the node knows the completion of the transmission is pending, for example because the Deluge algorithm indicates missing pages, a respective message is sent back.

Deluge

In the following, a short introduction in Deluge Algorithm is given as presented in [HC04]. It is a data dissemination algorithm that utilizes broadcast messages to ensure a quick and reliable distribution and updating of large files in a network. The nodes operate on local state machines, without the need of explicit synchronization. To increase efficiency, large files are divided into multiple pages which are updated one by one. The main idea of the algorithm is that every node periodically broadcasts a *summary* to all other nodes in the area. This summary contains the version of the local file and the number of the highest available page. If a node S with a newer version receives a summary, sent by a node R, it responds with its *object profile*, containing more detailed information about the versions of the single pages. From this information, R can calculate which pages need to be updated and start requesting them from one of its neighbors which advertised the page's availability through a summary. The requested page is then sent per link-layer broadcast to enable multiple nodes to update the page at the same time. Finally, the requesting node sends its summary to propagate the new version of the file.

On top of this simple algorithm, Deluge provides a variety of techniques to reduce network traffic and increase the performance of the transmission. Therefore, multiple summaries with the same version and highest page and multiple requests for the same page are suppressed. Additionally, the algorithm can cope with varying data rates and asymmetric links by keeping track of transmission timers and limiting the maximum number of request to a single node.

8.2.3 Analytical Model for the Data Dissemination

In order to estimate the time it takes until a file is distributed in a given network with CSMA/CA, an analytical model is derived in the following. It consists of three parts:

1. Estimating the per-link channel utilization from a given topology and traffic distribution.

2. Calculating the duration for transmitting a single page over a single hop.

3. Determining the time to transmit a complete file over multiple hops.

There already exist several analytical models for Deluge, the most popular was built by Dong et al. [DCL09]. However, their model uses the TOSSIM simulator to estimate the channel utilization for calculating the packet reception ratio. Instead, the presented model does not require a simulator or real-world measurements, but relies purely on analytical calculations by using the IEEE 802.15.4 model from Sect. 6.3. Furthermore, previous models did not consider the time to persistently write the data that has a significant influence on the performance. Thus, instead of assuming data packets are sent directly one after another, an inter-packet interval of $D_{data} = 100\,\text{ms}$ is used to account for the time for writing to the flash memory (see [KMT17] for details). For calculating the worst-case channel utilization in the currently active region, every node is assumed to send out data packets in average intervals of D_{data}. These packets consist of 5 Bytes for the Deluge header and 92 Bytes Deluge payload, so together with the FCS and the MAC header, it has a PSDU of 108 Bytes. According to (6.10), this results in a transmission duration of $T_{\text{data}}^{\text{Deluge}} = 11.4$ backoff periods of length S_b. A request consists of 4 Bytes, only, so it has a PSDU of 15 Bytes and thus takes $T_{\text{req}}^{\text{Deluge}} = 2.1$ backoff periods.

Since the model from Sect. 6.3 considers only multi-hop unicasts, the following adaption was made to accurately model the link layer broadcasts of the data packets without retransmissions and acknowledgments. The packet rate for a given down link (i,j) was set to

$$\lambda_{(i,j)} = \frac{S_b}{D_{data} \cdot |\mathbf{C}_i|}$$

where S_b is the base time unit of the model and $|\mathbf{C}_i|$ the number of direct children of node i. Solving the resulting system of non-linear equations yields the following probabilities.

- $R_{(i,j)}$, the probability of a successful transmission of a data packet on every link that considers collisions due to simultaneous carrier sensing and hidden node collisions.
- $\boldsymbol{\beta}_{(i,j)}$, the probability that the channel is sensed free before the maximum backoff counter is reached.
- $Q(t,\mathbf{S})$, the probability that over at least one out of a given set of links $\mathbf{S}$, a packet transmission takes place during t backoff periods of length S_b.

Assuming that the requests themselves have only a marginal influence on the channel utilization, the probability that a request can be sent over a link (i,j) and does not collide due to a hidden node situation can be calculated from this as

$$R_{(i,j)}^{R} = \boldsymbol{\beta}_{(i.j)} \cdot \left(1 - Q\left(T_{\text{data}}^{\text{Deluge}} + T_{\text{req}}^{\text{Deluge}}, \mathbf{H}\right)\right),$$

where $\mathbf{H}$ is the set of links where the sender is in range of j, but not of i.

For calculating the duration of transmitting a page, we consider the situation where a node i has already received a page that is missing at node j. After receiving an advertisement, j tries to send a request to i per unicast. If no acknowledgment is received, it is retried $n = 3$

times, otherwise j has to wait for an advertisement again. Therefore, the expected time until a request was successfully received by i is given by

$$T^{req}_{(i,j)} = \frac{D_{adv}}{R_{(i,j)} \cdot \left(1 - \left(1 - R^{R}_{(j,i)}\right)^{n+1}\right)}.$$

The expected time it takes until every data packet was transmitted successfully is given by

$$T^{data}_{(i,j)} = \frac{N_{pkts} \cdot D_{data}}{R_{(i,j)}},$$

with $N_{pkts} = 32$, the number of packets per page. Missing packets will be directly requested by the receiver, so there is no need for another $T^{req}_{(i,j)}$ if packets are lost. Summing these two times up results in the expected overall time $T^{pg}_{(i,j)}$ for a complete page transmission.

To get the duration to transmit p pages over multiple hops from node 0 to node j, a recursive expression is defined as

$$T^{mh}_{p,j} = T^{pg}_{(\mathbf{p}_j,j)} + \max\left(T^{mh}_{p,\mathbf{p}_j}, T^{mh}_{p-1,j}, T^{mh}_{p-1,\mathbf{c}_{j,1}}, \ldots, T^{mh}_{p-1,\mathbf{c}_{j,|\mathbf{C}_j|}}\right),$$

where $\mathbf{p}_j$ is the parent of j in the dissemination tree and $\mathbf{C}_j = \left(\mathbf{c}_{j,1}, \ldots, \mathbf{c}_{j,|\mathbf{C}_j|}\right)$ are the

Algorithm 6 Calculation of T^{mh} for N_{pg} pages and N nodes

```
1:  procedure TRANSMISSIONDURATION(T^mh, N_pg, N)
2:      for j ← 0, ..., N − 1 do
3:          T^mh_{0,j} ← 0
4:      for p ← 0, ..., N_pg do
5:          T^mh_{p,0} ← 0
6:      for p ← 1, ..., N_pg do
7:          for all c ∈ C_0 do
8:              push c onto stack
9:          while stack not empty do
10:             j ← pop from stack
11:             T^mh_{p,j} ← max(T^mh_{p,p_j}, T^mh_{p−1,j})
12:             for all c ∈ C_j do
13:                 push c onto stack
14:                 if T^mh_{p,j} < T^mh_{p−1,c} then
15:                     T^mh_{p,j} ← T^mh_{p−1,c}
16:             T^mh_{p,j} ← T^pg_{(p_j,j)} + T^mh_{p,j}
```

children. The children have to be considered, because usually j transmits page $p-1$ first to its children and can meanwhile not receive new data for page p. This expression can be efficiently computed for all nodes and pages with dynamic programming as shown in Algorithm 6. Finally, the overall duration to distribute a file in the whole network is the maximum over all $T^{mh}_{p,j}$.

8.2.4 Scalability Evaluation

In the following, the analytical model is evaluated for a concentric topology, where packets are transmitted from one ring to another in a multi-hop fashion. It is compared to an OMNeT++ simulation of the presented system. The results as shown in Fig. 8.7 depicts the time until every node in the network has successfully received a file of 128 kB.

It is apparent that the Deluge algorithm can largely benefit from spatial multiplexing so that 10 minutes are sufficient for a network of 1185 nodes, a topology with 19 concentric circles. Furthermore, the analytical model is a very good estimate for the performance of the simulated network.

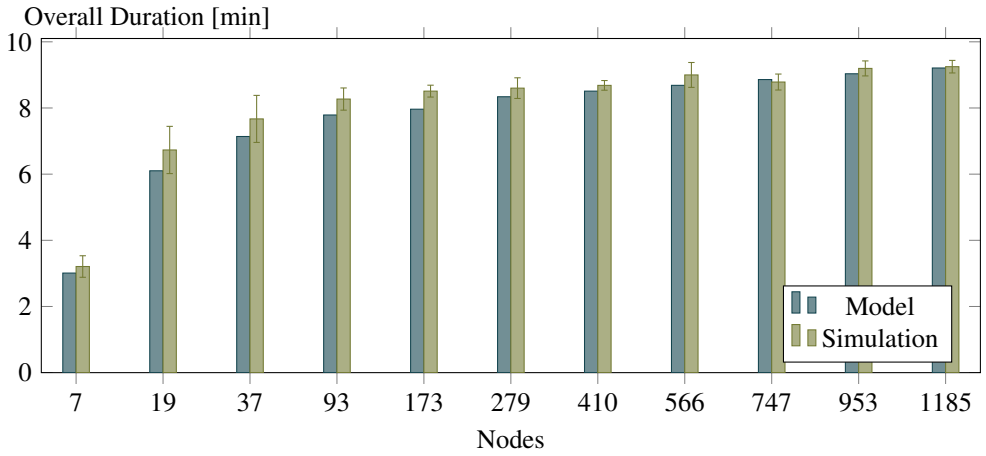


Figure 8.7: Comparison of the analytical model to the simulation for various network sizes and a file of 128 kB.

8.2.5 Conclusion

In this section, a holistic approach for remote software updates in large industrial plants is presented, based on a wireless field bus using the IEEE 802.15.4 standard. The data dissemination itself utilizes the Deluge algorithm and the files are persistently stored using the Coffee File System. The evaluation by using an analytical model and a simulation show that a network of 1185 nodes can be updated in under 10 minutes, demonstrating the efficient pipelining of Deluge.

8.3 Reliable Emergency Signalization*

In industrial applications, a high reliability of a wireless transmission system is essential to achieve maximum performance of the plant. This soft requirement is, however, often accompanied by a hard reliability requirement when it comes to safety relevant communication. In order to avoid danger to life and limb as well as damage to the facility, certain actions have to be performed reliably in a timely manner. Examples include closing valves of oil terminals in case of an earthquake [Has15], stopping automated guided vehicles (AGVs) at seaports [GK05] or defocusing reflected sunlight in a solar tower power plant to avoid overheating as presented previously. In particular, damage to critical infrastructure has to be avoided, because long downtimes can lead to cascading failures and negative impacts to the economy and the society. In the aforementioned examples, the number of devices can range up to several hundreds or thousands, an often neglected fact that has a severe impact on the reliability of wireless networks.

In order to perform a certain task in an emergency condition, such as an earthquake or an overheating receiver, a signal has to be issued from the central control to all devices. The amount of information is very low, in principle only a single bit, and only a single source of information is present, so a unidirectional transmission is sufficient. This concept is very common in many areas where optical (flashlights) or acoustical (sirens) emitters signal emergency conditions over a large area.

Thus, channel access is not an issue and a single transmitter is sufficient as long as it can cover the whole plant. Such an approach is presented and evaluated in this section. It can be easily combined with another wireless system and adds real-time capabilities to ensure operational safety. By using this system, the traffic load for the primary radio system and the latency are reduced and as a by product, an accurate time synchronization is provided.

8.3.1 Existing Solutions

Wireless emergency shutdown systems are already widely used in the industry [SRH98, Wie03], but usually they control only a single machinery and they are not able to control a large number of devices. For example in [GBS14], a wireless emergency stop system for a robot is presented based on a bidirectional communication in the 2.4 GHz band. Some solutions exist that are also capable of signaling emergency conditions to a higher number of nodes. In [Has15], a network with 500 devices is presented that provides update rates of 5 seconds. It utilizes the ISA100 standard for industrial automation that is based on the IEEE 802.15.4 standard. A solution based on the WirelessHART standard presented in [LNW15] investigates a holistic process control application, but even by using complex algorithms and accurate tuning of parameters, the system availability is not guaranteed in a network of only 21 nodes.

*An extended version was previously published as "F. Kauer, E. Kallias, and V. Turau, *A Dual-Radio Approach for Reliable Emergency Signaling in Critical Infrastructure Assets with Large Wireless Networks*, International Journal of Critical Infrastructure Protection 21, February 2018. https://doi.org/10.1016/j.ijcip.2018.02.002" [KKT18], also including the consideration of security aspects and the impact of false alarms on the availability of the plant.

Only one known solution [UT12b] scales to at least 10000 nodes and can provide an emergency shutdown latency of 2 seconds by using an optimized broadcasting algorithm to disseminate keep-alive beacons in the network, while maintaining an 8.6% false alarm probability. Its main disadvantage is that it fully loads the network with this task and thus has a strong negative impact on parallel communication. Furthermore, it was only evaluated by simulations and not in a real world application.

8.3.2 Proposed Solution

In the proposed solution, a dedicated transmitter sends a keep-alive beacon in regular intervals as heartbeat in order to signal normal operating conditions. In case of an emergency condition, the transmitter is shut down and the absence of $\widetilde{m}$ consecutive beacons signals this condition to the nodes. Using keep-alive messages as in [UT12b] is inevitable because packet loss is a problem in wireless networks that can only be mitigated but never completely solved and any failure to indicate an emergency condition might lead to a catastrophic failure. Disturbed transmissions could possibly lead to a wrong indication of an emergency condition, but never to the wrong indication of a normal operating mode. However, the amount of false alarms has to be kept very low to maintain stable and profitable operation (more details in [KKT18, Sect. 8.1]).

Link-layer acknowledgments for the Sub-GHz communication would provide no benefit, since the time to collect the acknowledgments from all nodes by far exceeds the very low transmission interval. Thus a new beacon is sent anyway before a retransmission would be possible. Furthermore, sending acknowledgments from every node would overload the channel completely. Therefore, a unidirectional communication is sufficient.

Beacon Transmission

Each beacon consists of the following four elements as depicted in Fig. 8.8:

Preamble and Sync Word Used for bit and byte synchronization of the receiver. The byte length of the preamble is denoted as l_P and the length of the sync word as l_{sync}. Optimal parameters are given in Sect. 8.3.4.

Timestamp The milliseconds elapsed between 1 January 1970 (Unix epoch) and the start of the transmission with byte length l_T. It is used together with the message integrity code to prevent replay attacks but is also useful for many user applications.

Message Integrity Code (MIC) Used as security measure (see [KKT18] for details). The byte length is denoted as l_M. Its secondary role is to recognize transmission errors, so no additional checksum is needed.

The total length in byte of a single beacon is given by

$$l = l_P + l_{sync} + l_T + l_M \tag{8.1}$$

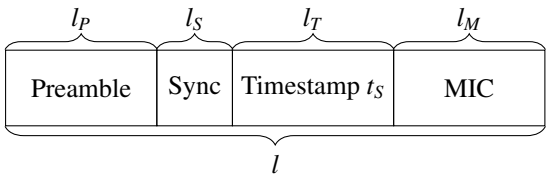


Figure 8.8: Structure of a beacon

that is used together with the bit data rate r to calculate the duration of a beacon transmission

$$T_B = \frac{8 \cdot l}{r}. \tag{8.2}$$

Since the calculation of the MIC takes a certain time T_M, the following procedure is used as illustrated in Fig. 8.9 to prevent timing inaccuracies. By using this approach, a time synchronization accuracy of 1 ms can be accomplished. At time t a timer is started for $t + T_F$, with $T_F > T_M$ and the generation of a message with timestamp $t_s = t + T_F$ is started. This includes the calculation of the MIC and the transmission into the buffer of the transmitter. As soon as the timer fires, the transmission is started. The next message generation is started at $t + I$, with the fixed beacon interval length I determined in Sect. 8.3.4.

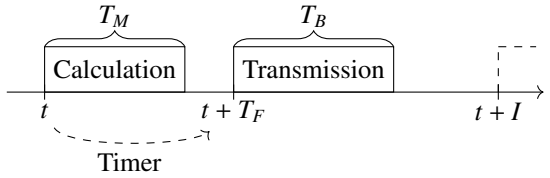


Figure 8.9: Beacon generation procedure

Due to the fact that the traffic is only unidirectional, no sophisticated coordination is needed as in the wireless mesh network. Furthermore, operational regulations can constrain the usage of other systems transmitting in the same band in the area of the infrastructure facility [NERC06, KDM09]. Anyhow, occasional disturbances from other devices or other sources of narrowband noise can not be precluded. Therefore, the transmission channel is changed circularly to one of $|\mathcal{C}|$ channels for each beacon as illustrated in Fig. 8.10. A resulting measured spectrogram is shown in Fig. 8.11. The system can cope with multiple disturbed channels, since only each $\widetilde{m}$-th beacon has to be received to ensure proper operation.

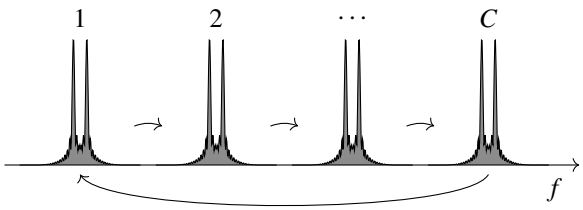


Figure 8.10: Switching the channel for mitigating disturbances

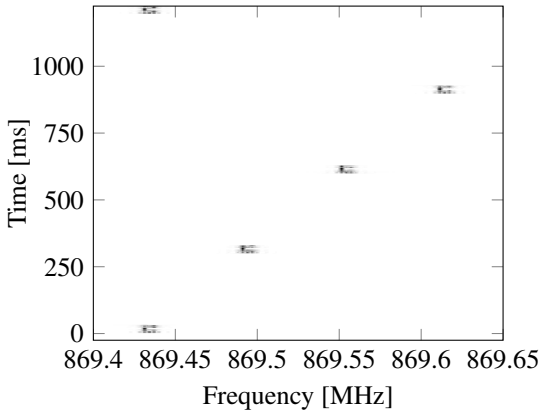


Figure 8.11: Spectrogram of four beacon transmissions in four channels

Beacon Reception

During normal operation, every node receives a beacon in intervals of length I. Correspondingly to the beacon transmission, the following procedure is used to maintain accuracy of the timestamp as depicted in Fig. 8.12: On the node a local time counter runs that counts the milliseconds since start of the node t_l. When a correct sync word has been received, a dedicated digital line is asserted that triggers an interrupt on the main processor. The current local time is saved as $t_{l,r}$ in the interrupt service routine. After the packet was finally received and the integrity is ensured, the shift between local and global time is calculated as

$$T_{shift} = t_s - t_{l,r} + \frac{8 \cdot (l_P + l_S)}{r}. \tag{8.3}$$

The local counter is never adjusted to maintain monotonicity. Instead, the shift is added to the current local time to get the current global time $t_l + T_{shift}$.

After the reception of a beacon, the receiver is set into sleep mode to save energy. Since I is fixed and known by the receiver, the optimal sleeping period can be calculated from the reception time of the previous beacon. The receiver is woken up from sleep mode at time $t_s + I - \varepsilon$, with a small time span ε to account for state transition times. Afterward, the receiver is tuned to the channel of the next expected beacon and switched to reception mode.

Subsequently to a reset of the node, it has to synchronize to the phase and channel

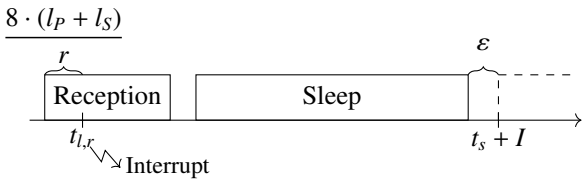


Figure 8.12: Beacon reception procedure

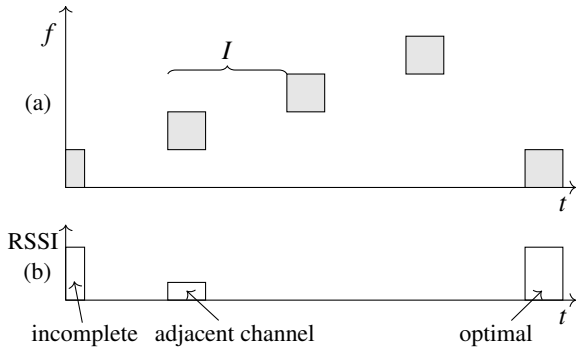


Figure 8.13: The node synchronization process. The transmitter sends out beacons on different channels (a), while the receiver listens on the first channel and searches for the optimal beacon to synchronize to (b).

hopping scheme of the beacon transmission. For this, the receiver is turned on and tuned to the first channel. As soon as a beacon is received and considered valid, the Received Signal Strength Indication (RSSI) is saved. Since the dynamic range of the receiver is larger than the adjacent channel rejection [Tex16] this might have been a beacon originating from another channel, given the distance to the transmitter is small. Therefore, the receiver stays on the same channel for $|\mathcal{C}| + 1$ beacon intervals as shown in Fig. 8.13 and synchronizes to the received beacon with the maximum RSSI value.

By using this approach, the system can be easily extended to multiple transmitters. This can be advantageous to accomplish a larger coverage or to circumvent shadowing. For example, four transmitters with directional antennas can be placed around a tower to provide 360° coverage. In order to avoid overlapping, the beacon transmissions can be shifted in time and channel. Due to the synchronization procedure presented above, every receiver automatically synchronizes itself to the respective transmitter that provides the best reception quality.

8.3.3 Realization

The proposed solution is implemented by using Texas Instruments CC110L [Tex16] transceivers operating in a Sub-GHz band with 2-FSK modulation. A CC110L can act as transmitter and as receiver, but every component requires only one functionality. For cost reduction, the corresponding CC113L can be used as pure receiver and the CC115L as transmitter. Alternatively, many other transceiver/receiver pairs are available that can be adapted for the described application.

The CC110L supports various frequency bands. The used frequency band highly depends on the local regulations and the requirements of the application. A prominent example for a license exempt frequency range in Europe is 869.40 MHz to 869.65 MHz, since ETSI EN 300 220-1 [ETSI12] allows for up to 500 mW radiated power with 10% duty cycle. In the US, the frequency range 902 MHz to 928 MHz is a suitable alternative, according to 47 CFR 15 [FCC15]. The application in licensed bands would be possible, too, with the

advantage of fewer disturbances and, if applicable, more transmission power and higher duty cycles. The obvious disadvantage is the higher organizational and financial effort. Apart from the maximum limits, the general principle is to use the lowest transmission power that enables a satisfactory communication to reduce the disturbances for other users.

Reaction Time

The purpose of the developed system is to signal an occurring emergency situation by interrupting the transmission of the beacons. The latency until this is recognized is mainly dependent on the number $\widetilde{m}$ of missing beacons that signal an emergency condition and the beacon interval, bounded by the data rate according to (8.4). The processing time of the beacon at the receiver causes a small additional delay. Fig. 8.14 depicts the measured latency from the interruption of the beacon transmission until the receiver is aware of the emergency condition. The distribution is shown over 20 measurements for each setting.

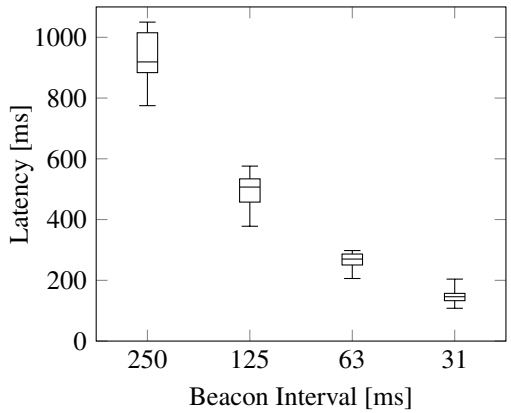


Figure 8.14: Latency for signaling emergency conditions with $\widetilde{m} = 4$

Time Synchronization

The time synchronization procedure can in principle achieve an accuracy of ± 1 ms. This can be even improved by also considering the phase of the transmission. It has to be noted that this is not necessary for the presented emergency shutdown, but this extension to the system can be easily realized if the target application requires time synchronization (for example for highly accurate time stamping of sensor readings).

For implementing this, a timer with higher resolution is required, for example by dividing a millisecond into 250 intervals and starting the beacon transmission always in the first interval. The lower bound is given by the precision of determining the transmission start in sender and receiver as well as the relative deviation of the clocks within a synchronization interval. The latter is not only depending on the accuracy of the crystal but also manufacturing tolerances, temperature influences and aging of the load capacitors and the PCB. Minimizing these effects in hardware is cost intensive, therefore a software approach is used. The ATmega256RFR2 on the receiver boards provides the possibility to calibrate the

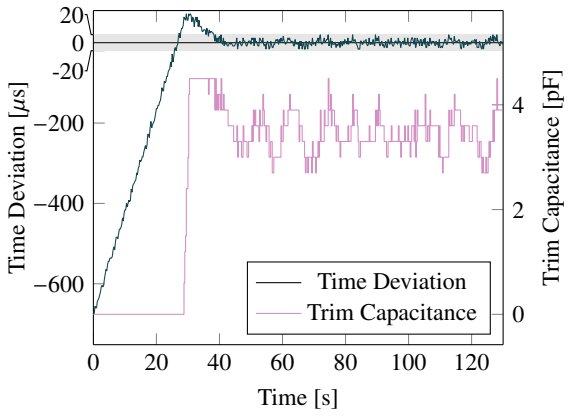


Figure 8.15: Synchronizing the clock phase by regulating the load capacitance

clock by adding additional load capacitance per software. With a PI controller, the clock frequency can be regulated to minimize the time deviation between sender and receiver. Fig. 8.15 shows how the controller regulates the time deviation from initially $-676\,\mu s$ to under $\pm 20\,\mu s$.

Energy Considerations

The nodes of the presented application are powered autonomously by solar panels on the heliostats, buffered by a lithium-titanate battery. Since they are dimensioned to power the drives of the heliostats, too, the total amount of available energy is higher than in many other energy-harvesting applications, but nevertheless, consuming an excessive amount of energy on the nodes can not be tolerated.

The energy consumption is kept low by two facts: The nodes only receive data and never transmit, accordingly no high-powered amplifier is needed. Secondly, the nodes can shut

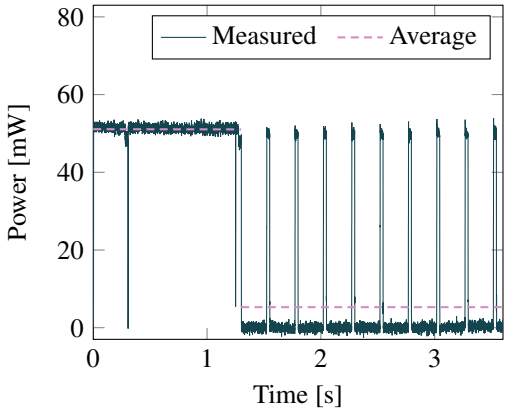


Figure 8.16: Trace of the power consumption during beacon reception with $r = 5768$ bps and a duty cycle of 10%

down the receiver in between the receptions due to the regular transmission pattern. The impact of this technique is evaluated in the following. An Agilent B2901A Source/Measure Unit was used to measure the power consumed by the receiver with a supply voltage of 3.0 V. The constant power of 16.61 mW consumed by the other components on the board was subtracted from the measured power. Fig. 8.16 shows the power consumption during synchronization of 51.09 mW and the power consumption trace during normal operation with an average power of 5.29 mW, resulting in a power reduction of nearly 90%.

8.3.4 Selecting Optimal Parameters

In order to achieve optimal performance, the parameters for the transmission procedure and the settings of transmitter and receiver must be chosen carefully. The most important quality criteria are the beacon interval length I and a reliable beacon reception at larger ranges. The beacon interval length together with $\widetilde{m}$, the number of missing beacons that signal an emergency condition, directly result in the latency $\widetilde{m} \cdot I$ until an emergency condition is recognized by the nodes. Decreasing $\widetilde{m}$ reduces the latency, but makes the system more susceptible for wrong indications of emergency states due to temporary disturbances. In the following, a latency of one second is chosen as target value and $\widetilde{m}$ is set to 4, corresponding to the number of channels. This gives an upper bound for the beacon interval length of $I_{\max} = 250\,\text{ms}$.

A lower bound results from the duty cycle given by the restrictions of the chosen frequency band (in this case $D = 10\%$)

$$I_{\min} = \frac{T_B}{D} = \frac{8 \cdot l}{r \cdot D}, \tag{8.4}$$

as given by (8.2). Hence, higher data rates r should be aspired and smaller packet lengths l, to reduce the reaction time. If nothing else is specified, the length l used in the following is

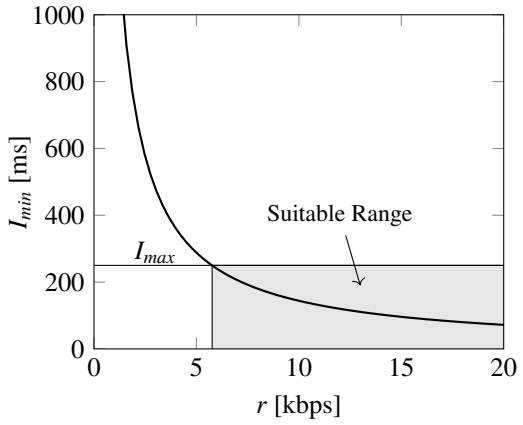


Figure 8.17: Minimum beacon interval length for $D = 10\%$ and $l = 18$

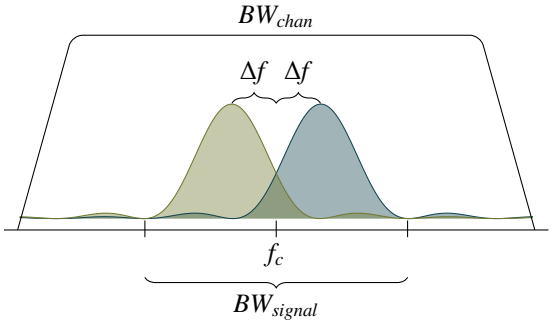


Figure 8.18: 2-FSK Modulation

given by (8.1) as

$$l = l_P + l_{sync} + l_T + l_M = 4 + 2 + 6 + 6 = 18. \tag{8.5}$$

Fig. 8.17 shows the minimum beacon interval lengths over the data rate r according to (8.4). It follows that the minimum suitable data rate is 5760 bps.

On the other hand, increasing the data rate increases the amount of bandwidth consumed and decreases the sensitivity of the receiver and therefore the range. In a binary frequency shift keying (2-FSK), see Fig. 8.18, the signal switches between two frequencies $f_c - \Delta f$ and $f_c + \Delta f$, where f_c denotes the carrier frequency and Δf the frequency deviation. The modulation index is defined as

$$h = \frac{2 \cdot \Delta f}{r \cdot 1\,\text{Bit}^{-1}}. \tag{8.6}$$

According to [Skl01], the minimum tone spacing for noncoherently detected 2-FSK signals that results in orthogonal waveforms is equal to the symbol rate, giving a modulation index of $h = 1$. The null-to-null bandwidth of the signal is given by

$$\begin{aligned} BW_{signal} &= 2 \cdot \Delta f + 2 \cdot r \cdot 1\,\text{Bit}^{-1} \\ &= (h+2) \cdot r \cdot 1\,\text{Bit}^{-1}. \end{aligned} \tag{8.7}$$

This results in a minimum bandwidth requirement of

$$3 \cdot r \cdot 1\,\text{Bit}^{-1}. \tag{8.8}$$

The channel filter bandwidth of the receiver depends on the bandwidth of the signal, but additionally the frequency drift due to temperature and fabrication tolerances of the crystal has to be considered: The crystal used in the analyzed hardware has an accuracy of ± 10 ppm, so according to [Tex18] the filter bandwidth at the receiver is given by

$$\begin{aligned} BW_{chan} &> BW_{signal} + 4 \cdot 10\,\text{ppm} \cdot f_c \\ &= BW_{signal} + 34776\,\text{Hz}. \end{aligned} \tag{8.9}$$

Thus, for the minimum suitable data rate $r = 5760\,\text{bps}$, the optimal parameters are

$$\Delta f = 0.5 \cdot 5760\,\text{Hz} = 2880\,\text{Hz}, \tag{8.10}$$

$$BW_{chan} > 3 \cdot 5760\,\text{Hz} + 34776\,\text{Hz} = 52056\,\text{Hz}. \tag{8.11}$$

Indoor Testbed

In order to quantify the influence of the parameters on the sensitivity of the receiver, an indoor testbed was set up. In contrast to the outdoor testbed as presented in Sect. 8.3.5, this allows to iterate over large parameter sets fully automated whereof the most prominent results are presented in the following. The sender and one receiver were deployed in an office environment at a distance of about 40 m. In order to compensate for the small range, the received signal was attenuated by 18 dB and the transmission power was reduced as specified in the following to provoke a condition where disparities are obvious, so the absolute values do not reflect the actual performance of the system as evaluated in Sect. 8.3.5. During one experiment, the transmitter transmits 100 packets with a known payload with 12 Bytes length and the receiver counts the number of error free receptions and calculates the packet reception ratio (PRR) from this. An experiment was repeated 10 times and the results in the plots show the mean value together with the 95% confidence intervals as shaded area.

The left part of Fig. 8.19 depicts the influence of the modulation index h and the channel filter bandwidth BW_{chan} on the packet reception ratio for a transmission power of $-17\,\text{dBm}$ and a data rate of $r = 5768\,\text{bps}$. The discrepancies from the optimal parameters, as calculated in the previous section, are due to the parameter resolution of the radio chip.

For low frequency deviations, the PRR is low, because two different symbols are difficult to distinguish. It increases until the minimum required modulation index $h = 1$, where a plateau is reached. At about $h = 5$, especially for higher bandwidths, the PRR rises again before it drops to zero when the signal is no longer within the receiver bandwidth. Though, using such high modulation index is unsuitable because (8.9) is violated, so

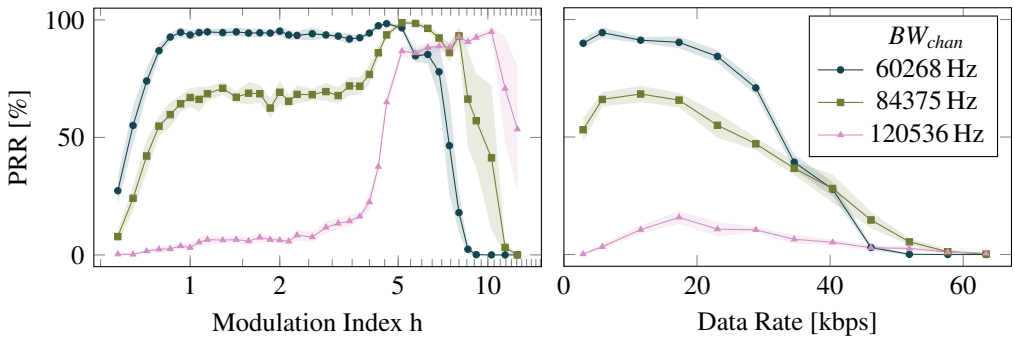


Figure 8.19: Packet reception ratio (PRR) over modulation index (left, with $r = 5768\,\text{bps}$) and data rate (right) for different receiver filter bandwidths at $-17\,\text{dBm}$ transmission power in the indoor testbed

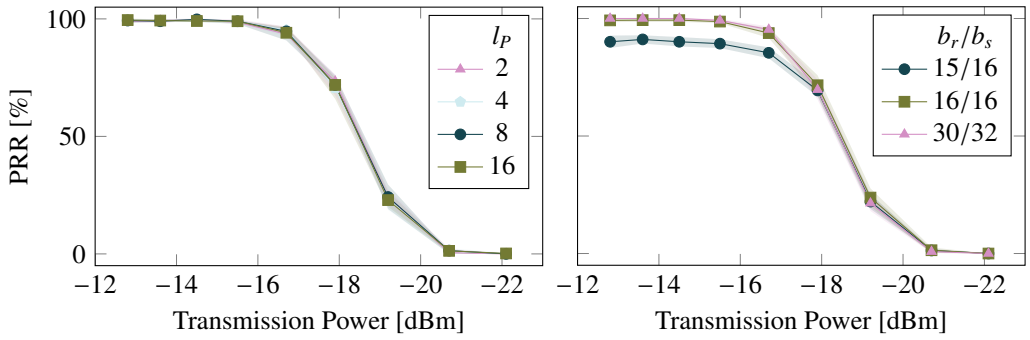


Figure 8.20: Influence of the preamble length and the synchronization sequence on the packet reception ratio for $r = 5768$ bps and $BW_{chan} = 60268$ Hz

frequency tolerances might render the system unusable. In addition, using a higher receiver bandwidth decreases the PRR significantly, because more noise is captured at the receiver side, decreasing the sensitivity.

This is also apparent in the right part of Fig. 8.19 where the PRR is evaluated over the data rate r for a modulation index of $h = 1$. In general, a smaller data rate, as well as a smaller BW_{chan}, is preferred because of the higher packet reception ratio. Though, for higher data rates the receiver bandwidth has to be increased to capture the whole signal properly.

Further parameters that influence the duration of a packet transmission but also the probability of a successful transmission at larger ranges are the length of the preamble and the synchronization sequence. For evaluating the effect of these parameters, the nominal transmission power was varied between -13 dBm and -22 dBm for $r = 5768$ bps, $h = 1$ and $BW_{chan} = 60268$ Hz. In the final application, the transmitter sends with more output power, but the received power will decrease with the distance. Thus, a better PRR at lower transmission power corresponds to a larger range.

The left part of Fig. 8.20 shows the influence of the preamble length l_P. In the given setup, no measurable differences can be determined, while a shorter preamble is preferred because it decreases the duration of a transmission. The results might be different when clock drifts occur due to temperature fluctuations or for higher data rates. For the presented system, $l_P = 4$ was chosen. The synchronization sequence also has a low influence for a transmission power under -18 dBm as depicted in the right part of Fig. 8.20. Thereby, the second number b_s describes the number of bits in the synchronization sequence, while the first number b_r describes how many correct bits are required for recognizing a correct synchronization sequence. Though, the setting $15/16$ shows an unsatisfactory behavior at transmission powers above -18 dBm where a PRR of only 0.9 is reached. Therefore, the setting $16/16$ was chosen.

8.3.5 Evaluation in a Heliostat Field

The performance of the overall system was evaluated in a solar tower power plant. In order to evaluate the reception properties over the distance while being as flexible as possible, an evaluation system was built that is based on a smartphone application. For the evaluation, the transmitter is placed at a fixed position and sends out beacons at fixed intervals with a transmission power of 23 mW. The receiver is connected to a smartphone and signals a successful beacon transmission together with the received signal strength indication (RSSI) as well as a missed beacon reception. The smartphone application records the measurement together with the current position determined via GPS. As antenna, a $\lambda/4$-antenna was used that is mounted at the end of a rod to minimize disturbances by the surveying person as shown in Fig. 8.21.

For the first experiment, the sender was placed at a height of about 2.5 m in front of the heliostat field. Fig. 8.22 shows the measured RSSI at different positions of the heliostat field with the sender at $(0\,\mathrm{m}, 0\,\mathrm{m})$. Obviously, the signal strength decreases with the distance to the transmitter, but even in the last rows it is sufficiently large.

In order to evaluate the performance for larger ranges, the transmitter was placed on the solar power tower at a height of about 20 m over ground, allowing a line-of-sight evaluation of up to 1 km. The RSSI over the distance from the tower is plotted in Fig. 8.23a. Even at a distance of over 1 km, the signal strength is large enough, while large variations are apparent. More interesting for the final application is the amount of false alarms as shown in Fig. 8.23b, that is the probability that an emergency condition is signaled at the receiver because $\widetilde{m}$ successive beacons could not be received, even though beacons are sent out by the transmitter.

Figure 8.21: Evaluation system with a receiver connected to a smartphone at the solar tower power plant

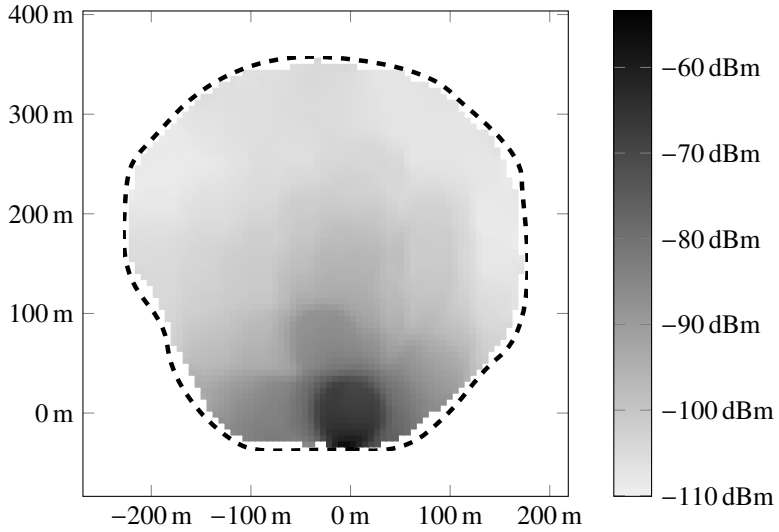
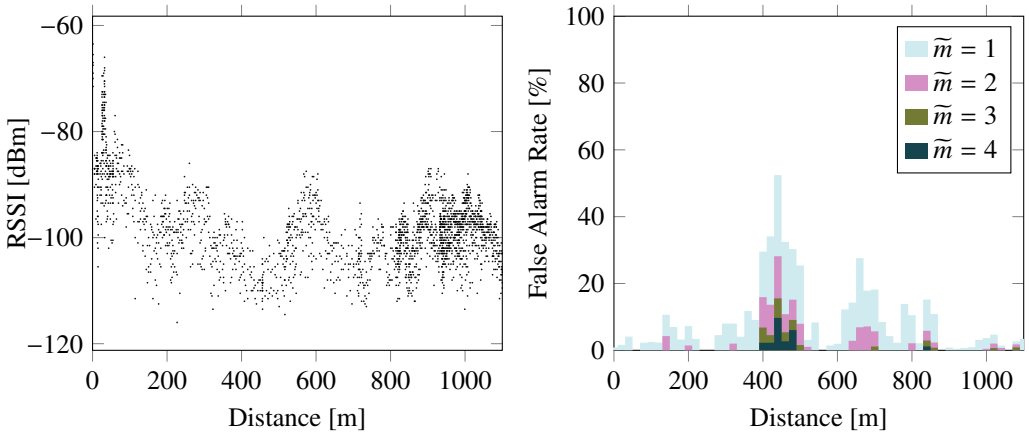


Figure 8.22: Interpolated RSSI distribution within the leaf-shaped heliostat field

The results show that the pure distance is much less important to the reception quality than shadowing of the line-of-sight. Around a distance of 400 m, the line-of-sight was blocked by some buildings and around 650 m by trees. This can be seen in Fig. 8.23a by a drop of the RSSI as well as in Fig. 8.23b by a larger number of false alarms, corresponding to burst packet errors of length $\widetilde{m}$. For other applications, for example in urban environments, more transmission power can be used to compensate this effect. Provided that the line-of-sight is not blocked, $\widetilde{m} = 4$ was sufficient in this experiment to accomplish no false alarms even at a range of 1 km.



(a) RSSI

(b) False alarm rate

Figure 8.23: Evaluation over the distance from the transmitter

8.3.6 Assessment

The presented systems can signal emergency conditions with a latency of one second within a range of 1 km, in compliance with the requirements of the presented application in a solar tower power plant. Though, the latency can be largely reduced at the cost of receiver sensitivity and therefore coverage. By using a keep-alive mechanism, the system maintains safe operation even in case of power outages or external disturbances, such as intentional jamming. Disturbances are further mitigated by obeying a channel hopping scheme. The receivers consume about 5 mW on average making the system suitable for autonomous deployment in an infrastructure facility. As a by-product, the system provides a time synchronization of $\pm 20\,\mu s$. Since the emergency radio system is unidirectional and does not require any routing, mobility of the nodes is not a problem and the scalability is only limited by the number of devices that can be placed within the achievable range.

Chapter 9

Conclusion

This thesis analyzes and improves techniques for enabling the construction of scalable wireless multi-hop networks for industrial applications. The focus is on the coordination of a large number of transceivers to avoid mutual interference. The thesis takes a holistic approach by using formal methods, analytical models, simulation and physically deployed testbeds for leveraging their individual advantages. A recurring topic is the analysis of the Deterministic and Synchronous Multi-Channel Extension (DSME) of IEEE 802.15.4 with its slot allocation handshake for distributed construction of collision-free time and frequency schedules. This leads to the comprehensive open-source implementation openDSME. The thesis is completed by the analysis of other medium access techniques such as CSMA/CA and TSCH as well as considerations for system integration such as the realization of a dual-radio approach for signalization of emergency conditions in industrial plants.

By modeling the slot allocation handshake of DSME as a network of timed automata in the UPPAAL tool environment, weaknesses are uncovered when losses of management messages are considered that can lead to very long phases of inconsistencies in the schedule. A modification is proposed to reduce this time significantly. In order to allow for efficient usage of the available resources, several methods for building schedules in multi-hop TDMA networks are proposed. The token-based approach is useful for building static schedules in a distributed way. This is particularly helpful for testing the limits by analytical analysis. Since static schedules are not suitable for changing traffic demands and volatile topologies, an alternative approach is proposed based on the local prediction of future traffic demands.

In order to get a method for fast, yet accurate, assessment of the performance of medium access methods, two analytical models are developed. In contrast to earlier models, the CSMA/CA model considers collisions of and with acknowledgment as well as the effect of simultaneous retransmissions. Both effects have a much higher influence on the performance as commonly expected. The TDMA model is able to precisely handle a given slot schedule and is therefore more accurate than only applying the M/D/1/K queuing model often used for modeling TDMA networks. It is used to analyze the proposed token-based scheduling techniques and the need for traffic-awareness for achieving high throughput is demonstrated.

For a step towards the practical application of DSME in real-world scenarios, the open-source implementation openDSME is developed. The thesis presents the software architecture of openDSME and a simulative evaluation showing the much higher reliability that can be achieved by using DSME instead of CSMA/CA at the cost of a higher complexity and a

higher delay in unsaturated networks. For demonstrating the applicability of openDSME in physically deployed networks, an evaluation in the FIT/IoT-LAB is conducted showing for example that DSME achieves a much lower energy consumption of the transceiver. In order to conduct this experiment, a method is presented to construct topologies with given density or trees with high depth and given breadth in dense wireless testbeds. Finally, application-specific challenges in large-scale industrial plants are presented. This includes the development and analysis of a method for over-the-air software updates and a concept for signalization of emergency conditions. For this, the communication devices are equipped with a long-range Sub-GHz transceiver and an IEEE 802.15.4 transceiver to combine the best of both worlds.

For increasing the range of possible applications, future research should concentrate on the development of scheduling mechanisms that are explicitly optimized for low delay. Furthermore, the resilience to external disturbances such as Wi-Fi should be analyzed and potentially improved by explicit channel blacklisting or using the channel hopping mode of DSME. Last, but not least, the deployment in an actual industrial application would help to collect valuable long-term experience that is essential for the wide-spread usage in the Industrial Internet of Things.

The presented techniques for efficient utilization of the wireless channel are an important building block to enable a multitude of applications for data-gathering and remote control. Besides conventional industrial control that can often be realized more cost-efficiently using wireless technology, the collection, combination and analysis of previously disregarded process data promises a high potential for innovative use cases.

To support this, the main goal should be to make the wireless transmission as transparent for the user as possible to put the focus on the application. Future research will strive towards this goal by increasing the reliability, lowering the delay and increasing the achievable throughput. Nevertheless, some particularities require special consideration during the development and operation of a wireless communication system. For example, the transmission quality is only loosely correlated with the distance. This has severe consequences for the architecture of a system, especially in multi-hop deployments or when using multiple base stations.

While future research will enable more and more innovative applications, this also increases the utilization of the wireless spectrum. Dealing with coexistence of wireless systems gets increasingly important. Conventional approaches, such as fixed frequency or duty-cycle assignments and listen-before-talk schemes, will not be able to handle the increasing amount of devices. Cognitive radio is a promising area of research to mitigate this bottleneck significantly by dynamically adapting the transmission parameters to the local occupation of the spectrum. Today, this is mostly based on spectrum sensing or preconfigured location-specific databases.

However, as shown in this thesis, decentralized cooperation can provide superior performance over approaches that are based on sensing or preconfiguration. While this thesis only considers coordination within a single wireless system, extending the principle to the coordination of independent wireless systems is both an interesting and challenging research topic that promises efficient utilization of the precious resources. This is vital for future wireless applications in the Internet of Things.

Appendix A

Appendix

A.1 Calculations for Analyzing the Traffic Prediction

In the following, calculations for analyzing the traffic prediction as introduced in Sect. 4.3.2 are presented. Algorithm 7 is applied to calculate an upper bound $P_{\text{int,upper}}$ and a lower bound $P_{\text{int,lower}}$ for P_{int}. This algorithm requires a probability mass function with $\tilde{f}_X(m) = 0 \; \forall m > m_{\max}$ for a fixed $m_{\max} \gg \overline{p}$. This is a minor constraint for most practical applications while popular generic distributions, such as the Poisson distribution, can be approximated by such a probability distribution by applying

$$\tilde{f}_X(m) = \begin{cases} f_X(m) & m < m_{\max} \\ \sum_{k=m_{\max}}^{\infty} f_X(k) & m = m_{\max} \\ 0 & \text{else.} \end{cases} \tag{A.1}$$

Secondly, to bypass the fact that the number of elements in K_t grows exponentially, the interval from 0 to $m_{\max}$ is split into $N = m_{\max} \cdot h$ intervals of size $1/h$ with a chosen integer $h > 1$. A larger h will lead to tighter bounds and a longer execution time. Also a small $\varepsilon > 0$ is to be chosen.

For calculating the number of multi-superframes until $\overline{p} - 1$ is reached, the following calculation is applied. Given is the recursive equation

$$\lambda_t = \overline{p} \cdot \alpha_\lambda + \lambda_{t-1} \cdot (1 - \alpha_\lambda) \text{ with } \lambda_0 = 0.$$

Solving for λ_t as

$$\lambda_t = \left(1 - (1 - \alpha_\lambda)^t\right) \cdot \overline{p}.$$

Setting λ_t equal to target $\overline{p} - 1$ and solving for t finally leads to

$$\overline{p} - 1 = \left(1 - (1 - \alpha_\lambda)^t\right) \cdot \overline{p}$$
$$t = \frac{\log\left(\overline{p}^{-1}\right)}{\log\left(1 - \alpha_\lambda\right)}.$$

Algorithm 7 Probability Bounds for $\overline{p} - 1 \leq \lambda_t \leq \overline{p} + 1$

function STEP(y_{t-1}, v)
  $y_t \leftarrow [0, 0, 0, \ldots, 0]$, with $|y_t| = N$
  for $i \leftarrow 0, \ldots, N-1$ **do**
    for $m \leftarrow 0, \ldots, m_{\max}$ **do**
      $\lambda \leftarrow m \cdot \alpha_\lambda + (i/h + v) \cdot (1 - \alpha_\lambda)$
      $p \leftarrow \tilde{f}_X(m) \cdot y_{t-1}(i)$
      $s \leftarrow \min(\lfloor \lambda \cdot h \rfloor, N-1)$
      $y_t(s) \leftarrow y_t(s) + p$
  return y_t
function INTERVALPROBABILITY
  $\check{y}_0 \leftarrow [1, 0, 0, \ldots, 0]$, with $|\check{y}_0| = N$
  $\hat{y}_0 \leftarrow [0, 0, 0, \ldots, 1]$, with $|\hat{y}_0| = N$
  $t \leftarrow 0$
  do
    $t \leftarrow t + 1$
    $\check{y}_t \leftarrow$ STEP($\check{y}_{t-1}, 0$)
    $\hat{y}_t \leftarrow$ STEP($\hat{y}_{t-1}, 1/h$)
    $P_{\text{int,lower}} \leftarrow 1 - \sum_{i=0}^{\lceil(\overline{p}-1)\cdot h\rceil-1} \check{y}_t(i) - \sum_{i=\lfloor(\overline{p}+1)\cdot h\rfloor}^{N} \hat{y}_t(i)$
    $P_{\text{int,upper}} \leftarrow 1 - \sum_{i=0}^{\lceil(\overline{p}-1)\cdot h\rceil-1} \hat{y}_t(i) - \sum_{i=\lfloor(\overline{p}+1)\cdot h\rfloor}^{N} \check{y}_t(i)$
  while $P_{\text{int,upper}} - P_{\text{int,lower}} > \varepsilon$
  return $P_{\text{int,upper}}, P_{\text{int,lower}}$

A.2 Proofs for the Analytical CSMA/CA Model

A.2.1 Stationary Distribution

In the asymptotic case, the probability b_ζ of being in state θ_ζ is calculated by summing over the probabilities of the preceding states Ξ, weighted by the transition probability.

$$b_\zeta = \sum_{\xi \in \Xi} b_\xi \cdot P\left(\theta_\xi \rightarrow \theta_\zeta\right) \tag{A.2}$$

Therefore, from (6.21) and (6.22) results

$$b_{0,k,0} = \frac{q_l}{W_0} b_{idle} + b_{0,k+1,0}, \qquad 0 \leq k < W_i - 1. \tag{A.3}$$

Since the maximum backoff period is $W_i - 1$, it holds

$$b_{i,W_i,j} = 0 \qquad 0 \leq i \leq m \wedge 0 \leq j \leq n, \tag{A.4}$$

so the recursion resolves to

$$b_{0,0,0} = W_0 \cdot \frac{q_l}{W_0} b_{idle} = q_l \cdot b_{idle}. \tag{A.5}$$

The same for (6.23) results in

$$b_{i,0,j} = W_i \cdot \frac{\alpha_l}{W_i} b_{i-1,0,j} = \alpha_l \cdot b_{i-1,0,j}, \qquad 0 < i < m-1 \wedge 0 \leq j < n, \tag{A.6}$$

$$= {\alpha_l}^i \cdot b_{0,0,j} \tag{A.7}$$

and for (6.29)

$$b_{0,0,j} = W_0 \cdot \frac{1}{W_0} b_{-2,T_{fail},j-1} = b_{-2,T_{fail},j-1}, \qquad 0 < j < n. \tag{A.8}$$

together with (6.25)

$$b_{0,0,j} = \sum_{i=0}^{m} b_{i,0,j-1} \cdot (1-\alpha_l) \cdot P_{\text{noACK},l} \tag{A.9}$$

$$= b_{0,0,j-1} \cdot (1-\alpha_l) \cdot P_{\text{noACK},l} \cdot \sum_{i=0}^{m} {\alpha_l}^i \tag{A.10}$$

$$= b_{0,0,0} \cdot \left((1-\alpha_l) \cdot P_{\text{noACK},l} \cdot \sum_{i=0}^{m} {\alpha_l}^i \right)^j \tag{A.11}$$

and taking the sum as the partial sum of the geometric series

$$= b_{0,0,0} \cdot \left((1-\alpha_l) \cdot P_{\text{noACK},l} \cdot \frac{1-{\alpha_l}^{m+1}}{1-\alpha_l} \right)^j \tag{A.12}$$

$$= b_{0,0,0} \cdot \left(P_{\text{noACK},l} \cdot \left(1-{\alpha_l}^{m+1} \right) \right)^j \tag{A.13}$$

$$= b_{0,0,0} \cdot {y_l}^j. \tag{A.14}$$

For each $k \in [0, W_i - 1] \wedge i \geq 0$ holds with (6.21), (6.22), (6.23) and (6.29)

$$b_{i,k,j} = b_{i,k+1,j} + \begin{cases} \frac{q_l}{W_0} b_{idle} & \text{for } i = 0 \wedge j = 0 \\ \frac{\alpha_l}{W_0} b_{i-1,0,j} & \text{for } 0 < i < m-1 \wedge 0 \leq j < n \\ \frac{1}{W_0} b_{-2,T_{fail},j-1} & \text{for } i = 0 \wedge 0 < j < n \end{cases} \tag{A.15}$$

and taking (A.5), (A.6) and (A.8) into account

$$= b_{i,k+1,j} + \begin{cases} \frac{q_l}{W_0} \frac{1}{q_l} b_{0,0,0} & \text{for } i = 0 \wedge j = 0 \\ \frac{\alpha_l}{W_i} \frac{1}{\alpha_l} b_{i,0,j} & \text{for } 0 < i < m-1 \wedge 0 \leq j < n \\ \frac{1}{W_0} b_{0,0,j} & \text{for } i = 0 \wedge 0 < j < n \end{cases} \tag{A.16}$$

$$= b_{i,k+1,j} + b_{i,0,j} \cdot \frac{1}{W_i}. \tag{A.17}$$

The recursion dissolves to

$$b_{i,k,j} = b_{i,0,j} \cdot \frac{W_i - k}{W_i}, \tag{A.18}$$

and together with (A.7) and (A.14)

$$b_{i,k,j} = b_{0,0,0} \cdot {y_l}^j \cdot {\alpha_l}^i \cdot \frac{W_i - k}{W_i}. \tag{A.19}$$

The probabilities in the stationary distribution have to sum up to one (normalization condition), that is

$$\sum_{i=0}^{m} \sum_{j=0}^{n} \sum_{k=0}^{W_i-1} b_{i,k,j} + \sum_{j=0}^{n} \left(\sum_{h=0}^{T_{succ}-1} b_{-1,h,j} + \sum_{h=0}^{T_{fail}-1} b_{-2,h,j} \right) + b_{idle} = 1. \tag{A.20}$$

From (A.19) results

$$\sum_{i=0}^{m} \sum_{j=0}^{n} \sum_{k=0}^{W_i-1} b_{i,k,j} = \sum_{i=0}^{m} \sum_{j=0}^{n} \sum_{k=0}^{W_i-1} b_{0,0,0} \cdot {y_l}^j \cdot {\alpha_l}^i \cdot \frac{W_i - k}{W_i} \tag{A.21}$$

$$= b_{0,0,0} \sum_{i=0}^{m} \sum_{j=0}^{n} {y_l}^j \cdot {\alpha_l}^i \cdot \sum_{k=0}^{W_i-1} \frac{W_i - k}{W_i} \tag{A.22}$$

$$= b_{0,0,0} \sum_{j=0}^{n} {y_l}^j \cdot \sum_{i=0}^{m} {\alpha_l}^i \cdot \frac{W_i + 1}{2} \tag{A.23}$$

splitting the second sum at $\overline{m}$

$$= b_{0,0,0} \sum_{j=0}^{n} {y_l}^j \left(\sum_{i=0}^{\min(m,\overline{m})} {\alpha_l}^i \cdot \frac{W_i + 1}{2} + \sum_{i=\overline{m}+1}^{m} {\alpha_l}^i \cdot \frac{W_i + 1}{2} \right) \tag{A.24}$$

$$= b_{0,0,0} \sum_{j=0}^{n} {y_l}^j \left(\sum_{i=0}^{\min(m,\overline{m})} {\alpha_l}^i \cdot \frac{2^i W_0 + 1}{2} + \sum_{i=\overline{m}+1}^{m} {\alpha_l}^i \cdot \frac{2^{\overline{m}} W_0 + 1}{2} \right) \tag{A.25}$$

$$= \frac{b_{0,0,0}}{2} \frac{1 - {y_l}^{n+1}}{1 - y_l} \left(\sum_{i=0}^{\min(m,\overline{m})} {\alpha_l}^i \cdot \left(2^i W_0 + 1 \right) + \sum_{i=\overline{m}+1}^{m} {\alpha_l}^i \cdot \left(2^{\overline{m}} 2^{m_0} + 1 \right) \right) \tag{A.26}$$

$$= \frac{b_{0,0,0}}{2} \frac{1 - {y_l}^{n+1}}{1 - y_l} \left(\sum_{i=0}^{\min(m,\overline{m})} \left(2\alpha_l\right)^i \cdot W_0 + \sum_{i=0}^{\min(m,\overline{m})} {\alpha_l}^i + \left(2^{m_b - m_0 + m_0} + 1 \right) \sum_{i=\overline{m}+1}^{m} {\alpha_l}^i \right) \tag{A.27}$$

$$= \frac{b_{0,0,0}}{2} \frac{1 - {y_l}^{n+1}}{1 - y_l} \left(\sum_{i=0}^{\min(m,\overline{m})} \left(2\alpha_l\right)^i \cdot W_0 + \sum_{i=0}^{\min(m,\overline{m})} {\alpha_l}^i \right.$$

$$+ (2^{m_b}+1) \sum_{i=\overline{m}+1}^{m} {\alpha_l}^{\overline{m}+1} {\alpha_l}^{i-\overline{m}-1} \Bigg) \tag{A.28}$$

$$= \frac{b_{0,0,0}}{2} \frac{1-{y_l}^{n+1}}{1-y_l} \Bigg(W_0 \frac{1-(2\alpha_l)^{\min(m,\overline{m})+1}}{1-2\alpha_l} + \frac{1-{\alpha_l}^{\min(m,\overline{m})+1}}{1-\alpha_l} + (2^{m_b}+1)\, {\alpha_l}^{\overline{m}+1} \sum_{i=0}^{m-\overline{m}-1} {\alpha_l}^{i} \Bigg) \tag{A.29}$$

$$= \frac{b_{0,0,0}}{2} \frac{1-{y_l}^{n+1}}{1-y_l} \Bigg(W_0 \frac{1-(2\alpha_l)^{\min(m,\overline{m})+1}}{1-2\alpha_l} + \frac{1-{\alpha_l}^{\min(m,\overline{m}+1)}}{1-\alpha_l} + \frac{(2^{m_b}+1)\, {\alpha_l}^{\overline{m}+1} \left(1-{\alpha_l}^{\max(0,m-\overline{m})}\right)}{1-\alpha_l} \Bigg). \tag{A.30}$$

It follows from (6.26) and (6.27)

$$b_{-1,h,j} = b_{-1,0,j} = \sum_{i=0}^{m} (1-\alpha_l) \cdot (1-P_{\text{noACK},l}) \cdot b_{i,0,j} \tag{A.31}$$

$$= (1-P_{\text{noACK},l}) \cdot (1-\alpha_l) \cdot b_{0,0,0} \cdot {y_l}^{j} \cdot \sum_{i=0}^{m} {\alpha_l}^{i} \tag{A.32}$$

$$= (1-P_{\text{noACK},l}) \cdot b_{0,0,0} \cdot {y_l}^{j} \cdot \left(1-{\alpha_l}^{m+1}\right) \tag{A.33}$$

and analog from (6.25) and (6.28)

$$b_{-2,h,j} = P_{\text{noACK},l} \cdot b_{0,0,0} \cdot {y_l}^{j} \cdot \left(1-{\alpha_l}^{m+1}\right) \tag{A.34}$$

$$= {y_l}^{j+1} \cdot b_{0,0,0}. \tag{A.35}$$

Inserting these into the second term of the normalization condition gives

$$\sum_{j=0}^{n} \left(\sum_{h=0}^{T_{succ}-1} b_{-1,h,j} + \sum_{h=0}^{T_{fail}-1} b_{-2,h,j} \right) \tag{A.36}$$

$$= \sum_{j=0}^{n} \Big(T_{succ} \cdot \Big((1-P_{\text{noACK},l}) \cdot b_{0,0,0} \cdot {y_l}^{j} \cdot \left(1-{\alpha_l}^{m+1}\right) \Big) + T_{fail} \cdot \left(P_{\text{noACK},l} \cdot b_{0,0,0} \cdot {y_l}^{j} \cdot \left(1-{\alpha_l}^{m+1}\right) \right) \Big) \tag{A.37}$$

$$= b_{0,0,0} \cdot \left(1-{\alpha_l}^{m+1}\right) \cdot \left(T_{succ} \cdot (1-P_{\text{noACK},l}) + T_{fail} \cdot P_{\text{noACK},l} \right) \cdot \sum_{j=0}^{n} {y_l}^{j} \tag{A.38}$$

$$= b_{0,0,0} \cdot \left(1-{\alpha_l}^{m+1}\right) \cdot \left(T_{succ} \cdot (1-P_{\text{noACK},l}) + T_{fail} \cdot P_{\text{noACK},l} \right) \cdot \frac{1-{y_l}^{n+1}}{1-y_l}. \tag{A.39}$$

The idle state probability is according to (6.19), (6.24), (6.30) and (6.31) given as

$$b_{idle} = (1 - q_l) \cdot b_{idle} + \sum_{j=0}^{n} \left(\alpha_l \cdot b_{m,0,j} + b_{-1,T_{succ},j} \right) + b_{-2,T_{fail},n} \tag{A.40}$$

$$b_{idle} - (1 - q_l) \cdot b_{idle} = \tag{A.41}$$
$$\sum_{j=0}^{n} \left({\alpha_l}^m \cdot {y_l}^j \cdot b_{0,0,0} + (1 - P_{\text{noACK},l}) \cdot b_{0,0,0} \cdot {y_l}^j \cdot \left(1 - {\alpha_l}^{m+1} \right) \right) + {y_l}^{n+1} \cdot b_{0,0,0}$$

$$q_l \cdot b_{idle} = b_{0,0,0} \cdot \left(\sum_{j=0}^{n} {y_l}^j \left({\alpha_l}^{m+1} + (1 - P_{\text{noACK},l}) \left(1 - {\alpha_l}^{m+1} \right) \right) + {y_l}^{n+1} \right) \tag{A.42}$$

$$b_{idle} = \frac{b_{0,0,0}}{q_l} \cdot \left(\frac{1 - {y_l}^{n+1}}{1 - y_l} \left({\alpha_l}^{m+1} + (1 - P_{\text{noACK},l}) \left(1 - {\alpha_l}^{m+1} \right) \right) + {y_l}^{n+1} \right). \tag{A.43}$$

Finally, inserting (A.30), (A.39) and (A.43) into the normalization condition (A.20) gives

$$\begin{aligned} {b_{0,0,0}}^{-1} = {} & \frac{1}{2} \frac{1 - {y_l}^{n+1}}{1 - y_l} \left(W_0 \frac{1 - (2\alpha_l)^{\min(m,\overline{m})+1}}{1 - 2\alpha_l} + \frac{1 - {\alpha_l}^{\min(m,\overline{m}+1)}}{1 - \alpha_l} \right. \\ & \left. + \frac{\left(2^{m_b} + 1\right) {\alpha_l}^{\overline{m}+1} \left(1 - {\alpha_l}^{\max(0,m-\overline{m})} \right)}{1 - \alpha_l} \right) \\ & + \left(1 - {\alpha_l}^{m+1} \right) \cdot \left(T_{succ} \cdot (1 - P_{\text{noACK},l}) + T_{fail} \cdot P_{\text{noACK},l} \right) \cdot \frac{1 - {y_l}^{n+1}}{1 - y_l} \\ & + \frac{1}{q_l} \cdot \left({y_l}^{n+1} + \frac{1 - {y_l}^{n+1}}{1 - y_l} \left({\alpha_l}^{m+1} + (1 - P_{\text{noACK},l}) \left(1 - {\alpha_l}^{m+1} \right) \right) \right). \end{aligned} \tag{A.44}$$

The probability τ_l is then calculated by summing over all $b_{i,0,j}$ with $0 \leq i \leq m$ and $0 \leq i \leq n$, giving

$$\tau_l = \sum_{i=0}^{m} \sum_{j=0}^{n} b_{i,0,j} \tag{A.45}$$

$$= \sum_{i=0}^{m} \sum_{j=0}^{n} {\alpha_l}^i b_{0,0,j} \tag{A.46}$$

$$= b_{0,0,0} \sum_{i=0}^{m} {\alpha_l}^i \sum_{j=0}^{n} {y_l}^j \tag{A.47}$$

$$= b_{0,0,0} \frac{1 - {\alpha_l}^{m+1}}{1 - \alpha_l} \frac{1 - {y_l}^{n+1}}{1 - y_l}. \tag{A.48}$$

A.2.2 Alternative Derivation of $Q(1, \mathbf{S})$

The expression in [MPF12] iterates over all combinations of one up to all neighboring links $l \in \mathbf{S}$. This can also described as the power set of $\mathbf{S}$ excluding the empty set

$$\mathcal{P}(\mathbf{S}) \setminus \emptyset = \{M | M \subseteq \mathbf{S} \wedge M \neq \emptyset\}. \tag{A.49}$$

With this, the expression can be written as

$$\begin{aligned}
Q(1,\mathbf{S}) &= \sum_{M\in\mathcal{P}(\mathbf{S})\setminus\emptyset} \left(\prod_{i\in M}\tau_i\right)\left(\prod_{i\in\mathbf{S}\setminus M}(1-\tau_i)\right)\left(1-\prod_{i\in M}\alpha_i\right) && \text{(A.50)}\\
&= \sum_{M\in\mathcal{P}(\mathbf{S})} \left(\prod_{i\in M}\tau_i\right)\left(\prod_{i\in\mathbf{S}\setminus M}(1-\tau_i)\right)\left(1-\prod_{i\in M}\alpha_i\right)\\
&\qquad - \left(\prod_{i\in\emptyset}\tau_i\right)\left(\prod_{i\in\mathbf{S}}(1-\tau_i)\right)\left(1-\prod_{i\in\emptyset}\alpha_i\right) && \text{(A.51)}\\
&= \sum_{M\in\mathcal{P}(\mathbf{S})} \left(\prod_{i\in M}\tau_i\right)\left(\prod_{i\in\mathbf{S}\setminus M}(1-\tau_i)\right)\left(1-\prod_{i\in M}\alpha_i\right) - 0 && \text{(A.52)}\\
&= \sum_{M\in\mathcal{P}(\mathbf{S})} \left(\prod_{i\in M}\tau_i\right)\left(\prod_{i\in\mathbf{S}\setminus M}(1-\tau_i)\right)\\
&\qquad - \sum_{M\in\mathcal{P}(\mathbf{S})} \left(\prod_{i\in M}\tau_i\right)\left(\prod_{i\in\mathbf{S}\setminus M}(1-\tau_i)\right)\prod_{i\in M}\alpha_i && \text{(A.53)}\\
&= \sum_{M\in\mathcal{P}(\mathbf{S})} \left(\prod_{i\in M}\tau_i\right)\left(\prod_{i\in\mathbf{S}\setminus M}(1-\tau_i)\right)\\
&\qquad - \sum_{M\in\mathcal{P}(\mathbf{S})} \left(\prod_{i\in M}\tau_i\alpha_i\right)\left(\prod_{i\in\mathbf{S}\setminus M}(1-\tau_i)\right). && \text{(A.54)}
\end{aligned}$$

Note that for $j \in \mathbf{S}$ holds

$$\mathcal{P}(\mathbf{S}) = \{M \cup \{j\} \mid M \in \mathcal{P}(\mathbf{S} \setminus \{j\})\} \cup \{M \mid M \in \mathcal{P}(\mathbf{S} \setminus \{j\})\}. \tag{A.55}$$

Therefore, the second term can be split up

$$\begin{aligned}
&\sum_{M\in\mathcal{P}(\mathbf{S})} \left(\prod_{i\in M}\tau_i\alpha_i\right)\left(\prod_{i\in\mathbf{S}\setminus M}(1-\tau_i)\right) && \text{(A.56)}\\
&\quad = \sum_{M\in\mathcal{P}(\mathbf{S}\setminus\{j\})} \left(\prod_{i\in(M\cup\{j\})}\tau_i\alpha_i\right)\left(\prod_{i\in(\mathbf{S})\setminus(M\cup\{j\})}(1-\tau_i)\right) && \text{(A.57)}
\end{aligned}$$

$$+ \sum_{M \in \mathcal{P}(\mathbf{S}\setminus\{j\})} \left(\prod_{i \in M} \tau_i \alpha_i \right) \left(\prod_{i \in \mathbf{S}\setminus M} (1-\tau_i) \right) \tag{A.58}$$

$$= \tau_j \alpha_j \sum_{M \in \mathcal{P}(\mathbf{S}\setminus\{j\})} \left(\prod_{i \in M} \tau_i \alpha_i \right) \left(\prod_{i \in (\mathbf{S}\setminus\{j\})\setminus M} (1-\tau_i) \right)$$

$$+ (1-\tau_n) \sum_{M \in \mathcal{P}(\mathbf{S}\setminus\{j\})} \left(\prod_{i \in M} \tau_i \alpha_i \right) \left(\prod_{i \in (\mathbf{S}\setminus\{j\})\setminus M} (1-\tau_i) \right) \tag{A.59}$$

$$= (\tau_j \alpha_j + (1-\tau_j)) \sum_{M \in \mathcal{P}(\mathbf{S}\setminus\{j\})} \left(\prod_{i \in M} \tau_i \alpha_i \right) \left(\prod_{i \in (\mathbf{S}\setminus\{j\})\setminus M} (1-\tau_i) \right). \tag{A.60}$$

This step can be recursively repeated for all nodes in $\mathbf{S}$, finally resulting in

$$\sum_{M \in \mathcal{P}(\mathbf{S})} \left(\prod_{i \in M} \tau_i \alpha_i \right) \left(\prod_{i \in \mathbf{S}\setminus M} (1-\tau_i) \right) = \prod_{l \in \mathbf{S}} (\tau_j \alpha_j + (1-\tau_j)). \tag{A.61}$$

The same calculation for the first term in (A.54) results in

$$\sum_{M \in \mathcal{P}(\mathbf{S})} \left(\prod_{i \in M} \tau_i \right) \left(\prod_{i \in \mathbf{S}\setminus M} (1-\tau_i) \right) = \prod_{j \in \mathbf{S}} (\tau_j + (1-\tau_j)) = 1. \tag{A.62}$$

Finally, the result is

$$Q(1, \mathbf{S}) = 1 - \prod_{j \in \mathbf{S}} (\tau_j \alpha_j + (1-\tau_j)). \tag{A.63}$$

A.3 Proofs for the Analytical TDMA Model

A.3.1 Expected Number of Packets Arriving at the MAC Queue

The expected value of $\boldsymbol{A}_{n,i}$ is calculated as

$$\begin{aligned} E[\boldsymbol{A}_{n,i}] &= \sum_{k=0}^{\infty} k \cdot P\left(\boldsymbol{A}_{n,i} = k\right) \\ &= \sum_{k=1}^{\infty} k \left((1-\beta_{n,i}) \cdot P\left(\boldsymbol{\Pi}_{n,i} = k\right) + \beta_{n,i} \cdot P\left(\boldsymbol{\Pi}_{n,i} = k-1\right)\right) \\ &= (1-\beta_{n,i}) \cdot \left(\sum_{k=1}^{\infty} k \cdot P\left(\boldsymbol{\Pi}_{n,i} = k\right)\right) + \beta_{n,i} \cdot \sum_{k=1}^{\infty} k \cdot P\left(\boldsymbol{\Pi}_{n,i} = k-1\right) \end{aligned} \tag{A.64}$$

so with

$$\begin{aligned} \sum_{k=1}^{\infty} k \cdot P\left(\boldsymbol{\Pi}_{n,i} = k\right) &= \sum_{k=1}^{\infty} k \cdot \frac{{\lambda_{n,i}}^k}{k!} e^{-\lambda_{n,i}} = \lambda_{n,i} \cdot e^{-\lambda_{n,i}} \sum_{k=1}^{\infty} \frac{{\lambda_{n,i}}^{k-1}}{(k-1)!} \\ &= \lambda_{n,i} \cdot e^{-\lambda_{n,i}} \sum_{j=0}^{\infty} \frac{{\lambda_{n,i}}^{j}}{j!} = \lambda_{n,i} \cdot e^{-\lambda_{n,i}} e^{\lambda_{n,i}} = \lambda_{n,i} \end{aligned} \tag{A.65}$$

and

$$\begin{aligned} \sum_{k=1}^{\infty} k \cdot P\left(\boldsymbol{\Pi}_{n,i} = k-1\right) &= \sum_{k=1}^{\infty} k \cdot \frac{{\lambda_{n,i}}^{k-1}}{(k-1)!} e^{-\lambda_{n,i}} \\ &= e^{-\lambda_{n,i}} \left(\left(\sum_{k=1}^{\infty} \frac{{\lambda_{n,i}}^{k-1}}{(k-1)!} \right) + \sum_{k=1}^{\infty} (k-1) \frac{{\lambda_{n,i}}^{k-1}}{(k-1)!} \right) \\ &= e^{-\lambda_{n,i}} \left(\left(\sum_{k=1}^{\infty} \frac{{\lambda_{n,i}}^{k-1}}{(k-1)!} \right) + \lambda_{n,i} \sum_{k=2}^{\infty} \frac{{\lambda_{n,i}}^{k-2}}{(k-2)!} \right) \\ &= e^{-\lambda_{n,i}} \left(\left(\sum_{j=0}^{\infty} \frac{{\lambda_{n,i}}^{j}}{j!} \right) + \lambda_{n,i} \sum_{j=0}^{\infty} \frac{{\lambda_{n,i}}^{j}}{j!} \right) \\ &= e^{-\lambda_{n,i}} \left(e^{\lambda_{n,i}} + \lambda_{n,i} \cdot e^{\lambda_{n,i}} \right) = 1 + \lambda_{n,i} \end{aligned} \tag{A.66}$$

we finally get

$$E[\boldsymbol{A}_{n,i}] = (1-\beta_{n,i})\, \lambda_{n,i} + \beta_{n,i}\, (\lambda_{n,i}+1)\,. \tag{A.67}$$

A.3.2 Stationary Distribution

The stationary distribution of the presented Markov chain for node v_n is denoted as

$$\vec{c} = \left(c_{n,\left\lfloor \frac{j}{l_S} \right\rfloor, j\%l_S} \right)_{j=0,\ldots,(K+1)\cdot l_S-1} \tag{A.68}$$

where all $0 \leq c_{n,q,i} \leq 1$ and $\vec{c} \cdot e = 1$ with

$$e = \begin{pmatrix} 1 & 1 & \ldots & 1 \end{pmatrix}^T, \tag{A.69}$$

that is the normalization criterion that the probabilities have to sum up to 1. The stationary distribution is calculated as the solution of $\vec{c}P = \vec{c}$, where P is the transition probability matrix

$$\begin{aligned} P &= \left[p_{j,k} \right]_{((K+1)\cdot l_S)\times((K+1)\cdot l_S)} \\ p_{j,k} &= P\left(\sigma^n_{\left\lfloor \frac{j}{l_S} \right\rfloor, j\%l_S} \rightarrow \sigma^n_{\left\lfloor \frac{k}{l_S} \right\rfloor, k\%l_S} \right). \end{aligned} \tag{A.70}$$

This can be rewritten as

$$\vec{c}(I - P) = 0 \Leftrightarrow (I - P)^T \vec{c}^T = 0, \tag{A.71}$$

with the identity matrix I. This is a homogeneous system of $(K+1) \cdot l_S$ linear equations and the same number of unknowns.

Irreducibility

Consider the example in Fig. A.1 with $l_S = 3$, $K = 1$, $g_{\text{up}} = 0$ and $\mu^{\text{RX}}_{n,0} > 0$. Since nothing is generated and every packet received in slot 0 is immediately sent out again, the state $\sigma^n_{1,2}$ is never visited when starting with $q = 0$. It is also possible to construct more complex examples where blocks of states exist that are linked together, but are not reachable from any state with $q = 0$. This property makes the Markov chain reducible and thus a unique $\vec{c}$ is not guaranteed by the following proof. Finding these states corresponds to finding the states that are not in the strongly connected set that contains $\sigma^n_{0,0}$.

Figure A.1: A schedule with unreachable state $\sigma^n_{1,2}$.

Since it is meaningless for the application to assign any $c_{n,q,i} > 0$ to those states, they need to be set to zero and the corresponding columns and rows have to be removed from P,

making the Markov chain and P irreducible, that is there is no permutation of the rows and columns of P resulting in

$$\begin{pmatrix} A & B \\ 0 & D \end{pmatrix}, \tag{A.72}$$

with the square matrices A and D and the matrix 0 with all elements zero.

Rank of $I - P$

In the following we prove that the matrix $Q = I - P$ has rank $n - 1$ if P is irreducible. The proof goes along the lines of [Cha75]. Since the outgoing transitions of a state have to sum up to one, that is

$$\forall \sigma^n_{q_1,i_1} \in \mathbf{\Sigma}_n : \sum_{\sigma^n_{q_2,i_2} \in \mathbf{\Sigma}_n} P\left(\sigma^n_{q_1,i_1} \to \sigma^n_{q_2,i_2}\right) = 1, \tag{A.73}$$

it holds that $Pe = 1$, thus $(I - P)e = 0$, so the matrix Q has at least one zero eigenvalue, is therefore singular and its rank is at most $n - 1$.

Assuming a rank of $n - 2$ or less, then there is at least one vector x with $Qx = 0$ that is orthogonal to e, i.e. $x^T e = 0$. Thus, all linear combinations of x and e are also in the null space of Q. In particular this holds for $d = x - m \cdot e$ where m is the minimum entry of x. Then at least one, but not all, elements of d are 0 and all others are larger than 0. Note that this would not hold for x parallel to e. However, this is not possible since $x^T e = 0$.

Since permutations do not change the rank, we assume without loss of generality the first $u > 0$ elements of d are positive and the remaining $n - u$ elements are zero. It holds

$$\begin{aligned} Qd = 0 \Leftrightarrow (I - P)d = Id - Pd = 0 \\ \Leftrightarrow Pd = Id \Leftrightarrow Pd = d. \end{aligned} \tag{A.74}$$

This can be partitioned as

$$Pd = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} d_1 \\ \vdots \\ d_u \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_u \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{A.75}$$

with C of dimension $(n-u) \times u$. Therefore, it holds

$$C \begin{pmatrix} d_1 \\ \vdots \\ d_u \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{A.76}$$

For $d_i > 0 \ \forall 1 \leq i \leq u$ this is only possible if all entries of C are zero or there are negative entries in C. The first one contradicts the irreducibility of P and the second one would require the existence of negative probabilities in P. Therefore, $I-P$ has rank $n-1$. Since the rank is maintained by transposition, the matrix $(I-P)^T$ has rank $n-1$, too.

Solution of $cP = c$

Since $(I-P)^T$ has rank $n-1$, the homogeneous system of linear equations $(I-P)^T c^T = 0$ has a solution space of dimension one, so since $c \cdot e = 1$, there is a unique stationary distribution. Furthermore, if we find any vector $y \neq 0$ with $(I-P)^T y^T = 0$, we can get c by normalization.

Though most methods for the numerical solution of systems of linear equations are tailored to the handling of regular matrices, many can be adapted for the singular case as presented in [Ste94]. For the given application, the implementation of the generalized minimal residual method (GMRES) in PETSc [BAA18] with SOR preconditioning and initial guess $0.5e$ turned out to work very well. For more background about why and when GMRES is applicable to Markov chains see Section 4.4.4 in [Ste94].

Bibliography

[802.11-2016] IEEE 802.11-2016 - IEEE Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2016. https://doi.org/10.1109/IEEESTD.2016.7786995

[802.11ah-2016] IEEE 802.11ah-2016 - IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation, 2017. https://doi.org/10.1109/IEEESTD.2017.7920364

[802.15.4-2015] IEEE 802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks, 2016. https://doi.org/10.1109/IEEESTD.2016.7460875

[802.15.4-2003] IEEE 802.15.4-2003 - IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2003. https://doi.org/10.1109/IEEESTD.2003.94389

[802.15.4e-2012] IEEE 802.15.4e-2012 - IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer, 2012. https://doi.org/10.1109/IEEESTD.2012.6185525

[ABF15] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne, *FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed* in Proceedings of the 2nd IEEE World Forum on Internet of Things (WF-IoT), Milan, Italy, Dec 2015. https://doi.org/10.1109/WF-IoT.2015.7389098

[AGS16] M. Al-Mahfoudh, G. Gopalakrishnan, and R. Stutsman, *Toward Rigorous Design of Domain-Specific Distributed Systems* in Proceedings of the 4th FME Workshop on Formal Methods in Software Engineering (FormaliSE), Austin, TX, USA, May 2016. https://doi.org/10.1145/2897667.2897674

[ALZ18] S. Anamalamudi, B. Liu, M. Zhang, A. Sangi, C. Perkins, and S. V. R. Anand, *Scheduling Function One (SF1): Hop-by-Hop Scheduling with RSVP-TE in 6TiSCH Networks*, Internet-Draft, Internet Engineering Task Force, April 2018. https://tools.ietf.org/html/draft-satish-6tisch-6top-sf1-04

[APM15] G. Alderisi, G. Patti, O. Mirabella, and L. L. Bello, *Simulative Assessments of the IEEE 802.15.4e DSME and TSCH in Realistic Process Automation Scenarios* in Proceedings of the 13th International Conference on Industrial Informatics (INDIN), Cambridge, UK, Jul 2015. https://doi.org/10.1109/INDIN.2015.7281863

[ASF17] A. Ali, G. A. Shah, M. O. Farooq, and U. Ghani, *Technologies and Challenges in Developing Machine-to-Machine Applications: A Survey*, Journal of Network and Computer Applications 83, April 2017. https://doi.org/10.1016/j.jnca.2017.02.002

[Atm09] Atmel, AT86RF231 - AVR Low Power 2.4 GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE, SP100, WirelessHART, and ISM Applications, Datasheet, 2009. http://ww1.microchip.com/downloads/en/DeviceDoc/doc8111.pdf

[Atm14] Atmel, ATmega256RFR2 - 8-bit AVR Microcontroller with Low Power 2.4 GHz Transceiver for ZigBee and IEEE 802.15.4, Datasheet, 2014. http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8393-MCU_Wireless-ATmega256RFR2-ATmega128RFR2-ATmega64RFR2_Datasheet.pdf

[AVP15] N. Accettura, E. Vogli, M. R. Palattella, L. A. Grieco, G. Boggia, and M. Dohler, *Decentralized Traffic Aware Scheduling in 6TiSCH Networks: Design and Experimental Evaluation*, IEEE Internet of Things Journal 2(6), December 2015. https://doi.org/10.1109/JIOT.2015.2476915

[AVT17] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, *Understanding the Limits of LoRaWAN*, IEEE Communications Magazine 55(9), September 2017. https://doi.org/10.1109/MCOM.2017.1600613

[BAA18] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. May, L. C. McInnes, R. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, *PETSc Users Manual*, Technical Report, ANL-95/11 - Revision 3.10, Argonne National Laboratory, September 2018. https://www.mcs.anl.gov/petsc/petsc-current/docs/manual.pdf

[BAR13] A. Bildea, O. Alphand, F. Rousseau, and A. Duda, *Link Quality Metrics in Large Scale Indoor Wireless Sensor Networks* in Proceedings of the 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), London, UK, September 2013. https://doi.org/10.1109/PIMRC.2013.6666451

[BAS06] F. J. R. Barboza, A. M. S. Andrade, F. A. Silva, and G. Lima, *Specification and Verification of the IEEE 802.11 Medium Access Control and an Analysis of its Applicability to Real-Time Systems* in Proceedings of the Brazilian Symposium on Formal Methods (SBMF 2006), Natal, Brazil, September 2006. https://doi.org/10.1016/j.entcs.2007.08.024

[BBD14] R. Boucebsi, F. Belala, and L. Derdouri, *Modeling Channel Allocation via BRS: Case of WMNs* in Proceedings of the 1st International Conference on Advanced Aspects of Software Engineering (ICAASE), Constantine, Algeria, November 2014. http://ceur-ws.org/Vol-1294/paper15.pdf

[BDH04] J. Beutel, M. Dyer, M. Hinz, L. Meier, and M. Ringwald, *Next-Generation Prototyping of Sensor Networks* in Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys), Baltimore, MD, USA, November 2004. https://doi.org/10.1145/1031495.1031541

[BGH16] E. Bres, R. van Glabbeek, and P. Höfner, *A Timed Process Algebra for Wireless Networks with an Application in Routing* in Programming Languages and Systems, Lecture Notes in Computer Science 9632, presented at the 25th European Symposium on Programming (ESOP), Eindhoven, The Netherlands, April 2016. https://doi.org/10.1007/978-3-662-49498-1_5

[BGM97] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient Management of Parallelism in Object Oriented Numerical Software Libraries* in Modern Software Tools in Scientific Computing, ed. by E. Arge, A. M. Bruaset, and H. P. Langtangen, ISBN 978-1-4612-1986-6, Birkhäuser, 1997. https://doi.org/10.1007/978-1-4612-1986-6_8

[Bia00] G. Bianchi, *Performance Analysis of the IEEE 802.11 Distributed Coordination Function*, IEEE Journal on Selected Areas in Communications 18(3), March 2000. https://doi.org/10.1109/49.840210

[BIS08] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, *The Hitchhiker's Guide to Successful Wireless Sensor Network Deployments* in Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys), Raleigh, NC, USA, November 2008. https://doi.org/10.1145/1460412.1460418

[BKK16] B. Beckert, M. Kirsten, V. Klebanov, and C. Schürmann, *Automatic Margin Computation for Risk-limiting Audits* in Electronic Voting, Lecture Notes in Computer Science 10141, presented at the 1st International Joint Conference for Electronic Voting (E-Vote-ID), Bregenz, Austria, October 2016. https://doi.org/10.1007/978-3-319-52240-1_2

[Blue5] Bluetooth SIG, Bluetooth Core Specification v5.0, 2016. https://www.bluetooth.com/specifications/bluetooth-core-specification

[BlueMesh] Bluetooth SIG, Bluetooth Specification Mesh Profile, 2017. https://www.bluetooth.com/specifications/mesh-specifications

[BNetzA18] Bundesnetzagentur, Vfg 5/2018 - Allgemeinzuteilung von Frequenzen zur Nutzung durch Funkanwendungen mit geringer Reichweite für nicht näher spezifizierte Anwendungen; Non-specific Short Range Devices (SRD), 2018.

[BRV16] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, *Do LoRa Low-Power Wide-Area Networks Scale?* in Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), Valletta, Malta, November 2016. https://doi.org/10.1145/2988287.2989163

[BS06] C. Bae and W. E. Stark, *Energy and Bandwidth Efficiency in Wireless Networks* in Proceedings of the 2006 International Conference on Communications, Circuits and Systems (ICCCAS), Guilin, China, June 2006. https://doi.org/10.1109/ICCCAS.2006.284882

[BS13] S. Brown and C. J. Sreenan, *Software Updating in Wireless Sensor Networks: A Survey and Lacunae*, Journal of Sensor and Actuator Networks 2(4), November 2013. https://doi.org/10.3390/jsan2040717

[BVD03] P. Bjorklund, P. Varbrand, and Di Yuan, *Resource Optimization of Spatial TDMA in Ad Hoc Radio Networks: A Column Generation Approach* in Proceedings of the 22th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), San Francisco, CA, USA, Mar 2003. https://doi.org/10.1109/INFCOM.2003.1208919

[CB08] N. Chee-Hock and S. Boon-Hee, *Queueing Modelling Fundamentals: With Applications in Communication Networks*, 2nd edn., ISBN 978-0-470-51957-8, John Wiley & Sons, 2008.

[CBR14] S. Capone, R. Brama, F. Ricciato, G. Boggia, and A. Malvasi, *Modeling and Simulation of Energy Efficient Enhancements for IEEE 802.15.4e DSME* in Proceedings of the 2014 Wireless Telecommunications Symposium (WTS), Washington, D.C., USA, April 2014. https://doi.org/10.1109/WTS.2014.6835017

[CCD11] M. Ceriotti, M. Corrà, L. D'Orazio, R. Doriguzzi, D. Facchin, S. T. Gună, G. P. Jesi, R. L. Cigno, L. Mottola, A. L. Murphy, M. Pescalli, G. P. Picco, D. Pregnolato, and C. Torghele, *Is There Light at the Ends of the Tunnel? Wireless Sensor Networks for Adaptive Lighting in Road Tunnels* in Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Chicago, IL, USA, April 2011. https://ieeexplore.ieee.org/document/5779037

[CFK09] I. Chatzigiannakis, S. Fischer, C. Koninis, G. Mylonas, and D. Pfisterer, *WISEBED: An Open Large-Scale Wireless Sensor Network Testbed* in Sensor Applications, Experimentation, and Logistics, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 29, presented at the 1st International Conference on Sensor Networks Applications, Experimentation and Logistics (SENSAPPEAL), Athens, Greece, September 2009. https://doi.org/10.1007/978-3-642-11870-8_6

[Cha75] I. M. Chakravarti, *On a Characterization of Irreducibility of a Non-Negative Matrix*, Linear Algebra and its Applications 10(2), April 1975. https://doi.org/10.1016/0024-3795(75)90002-6

[CLK16] Y. Chen, S. Lu, H. Kim, D. Blaauw, R. G. Dreslinski, and T. Mudge, *A Low Power Software-defined-radio Baseband Processor for the Internet of Things* in Proceedings of the 2016 IEEE International Symposium on High Performance Computer Architecture (HPCA), Barcelona, Spain, March 2016. https://doi.org/10.1109/HPCA.2016.7446052

[COP03] J. H. Cowie, A. T. Ogielski, B. Premore, E. A. Smith, and T. Underwood, *Impact of the 2003 Blackouts on Internet Communications*, Report, Renesys Corporation, November 2003.

[CPC14] A. di Cagno, M. Paoli, U. M. Colesanti, and A. Vitaletti, *REACTIVE: a Peaceful Coexistence between Deluge and Low Power Listening* in Proceedings of the 10th International Workshop on Wireless Network Measurements and Experimentation (WiNMeE), Hammamet, Tunisia, May 2014. https://doi.org/10.1109/WIOPT.2014.6850291

[CVV18] T. Chang, M. Vučinić, X. Vilajosana, S. Duquennoy, and D. Dujovne, *6TiSCH Minimal Scheduling Function (MSF)*, Internet-Draft, Internet Engineering Task Force, August 2018. https://tools.ietf.org/html/draft-ietf-6tisch-msf-00

[CWW16] T. Chang, T. Watteyne, Q. Wang, and X. Vilajosana, *LLSF: Low Latency Scheduling Function for 6TiSCH Networks* in Proceedings of the 2016 International Conference on Distributed Computing in Sensor Systems (DCOSS), Washington, D.C., USA, May 2016. https://doi.org/10.1109/DCOSS.2016.10

[DCL09] W. Dong, C. Chen, X. Liu, J. Bu, and Y. Liu, *Performance of Bulk Data Dissemination in Wireless Sensor Networks* in Distributed Computing in Sensor Systems, Lecture Notes in Computer Science 5516, presented at the 5th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), Marina del Rey, CA, USA, June 2009. https://doi.org/10.1007/978-3-642-02085-8_26

[DCV16] M. Domingo-Prieto, T. Chang, X. Vilajosana, and T. Watteyne, *Distributed PID-Based Scheduling for 6TiSCH Networks*, IEEE Communications Letters 20(5), May 2016. https://doi.org/10.1109/LCOMM.2016.2546880

[DGD17] S. Darroudi, C. Gomez, S. M. Darroudi, and C. Gomez, *Bluetooth Low Energy Mesh Networks: A Survey*, MDPI Sensors 17(7), June 2017. https://doi.org/10.3390/s17071467

[DGP17] D. Dujovne, L. Grieco, M. Palattella, and N. Accettura, *6TiSCH 6top Scheduling Function Zero (SF0)*, Internet-Draft, Internet Engineering Task Force, Jul 2017. https://tools.ietf.org/html/draft-ietf-6tisch-6top-sf0-05

[DGP18] D. Dujovne, L. Grieco, M. Palattella, and N. Accettura, *6TiSCH Experimental Scheduling Function (SFX)*, Internet-Draft, Internet Engineering Task Force, March 2018. https://tools.ietf.org/html/draft-ietf-6tisch-6top-sfx-01

[DGV04] A. Dunkels, B. Grönvall, and T. Voigt, *Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors* in Proceedings of the 29th International Conference on Local Computer Networks (LCN), Tampa, FL, USA, Nov 2004. https://doi.org/10.1109/LCN.2004.38

[DHJ06] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler, *Trio: Enabling Sustainable and Scalable Outdoor Wireless Sensor Network Deployments* in Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN), Nashville, TN, USA, April 2006. https://doi.org/10.1109/IPSN.2006.243880

[DHM14] T. Ducrocq, M. Hauspie, N. Mitton, and S. Pizzi, *On the Impact of Network Topology on Wireless Sensor Networks Performances: Illustration with Geographic Routing* in Proceedings of the 28th International Workshop on the Performance Analysis and Enhancement of Wireless Networks (PAEWN), Victoria, Canada, May 2014. https://doi.org/10.1109/WAINA.2014.118

[DNL15] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, *Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH* in Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys), Seoul, South Korea, November 2015. https://doi.org/10.1145/2809695.2809714

[DSZ12] G. Delzanno, A. Sangnier, and G. Zavattaro, *Verification of Ad Hoc Networks with Node and Communication Failures* in Formal Techniques for Distributed Systems, Lecture Notes in Computer Science 7273, presented at the Joint International Conference on Formal Techniques for Distributed Systems (FORTE/FMOODS), Stockholm, Sweden, June 2012. https://doi.org/10.1007/978-3-642-30793-5_15

[DVW18] S. Duquennoy, X. Vilajosana, and T. Watteyne, *6TiSCH Autonomous Scheduling Function (ASF)*, Internet-Draft, Internet Engineering Task Force, March 2018. https://tools.ietf.org/html/draft-duquennoy-6tisch-asf-01

[ETSI12] ETSI EN 300 220-1 V2.4.1, 2012. http://www.etsi.org/deliver/etsi_en/300200_300299/30022001/02.04.01_40/en_30022001v020401o.pdf

[EV06] S. Ergen and P. Varaiya, *PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks*, IEEE Transactions on Mobile Computing 5(7), July 2006. https://doi.org/10.1109/TMC.2006.100

[FB18] J. Finnegan and S. Brown, *A Comparative Survey of LPWA Networking*, presented at the Zhejiang University International Doctoral Students Conference 2017, arXiv:1802.04222 [cs], February 2018. http://arxiv.org/abs/1802.04222

[FBA17] A. J. Fahs, R. Bertolini, O. Alphand, F. Rousseau, K. Altisen, and S. Devismes, *Collision Prevention in Distributed 6TiSCH Networks* in Proceedings of the 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Rome, Italy, October 2017. https://doi.org/10.1109/WiMOB.2017.8115798

[FCC15] FCC, e-CFR: TITLE 47 - Telecommunication - Part 15 - Radio Frequency Devices, 2015.

[FEI09] G. Fiore, V. Ercoli, A. J. Isaksson, K. Landernäs, and M. D. D. Benedetto, *Multihop Multichannel Scheduling for Wireless Control in WirelessHART Networks* in Proceedings of the 2009 IEEE Conference on Emerging Technologies and Factory Automation (ETFA), Palma de Mallorca, Spain, September 2009. https://doi.org/10.1109/ETFA.2009.5347073

[FMF15] E. Frank, F. Mauthner, and S. Fischer, *Overheating Prevention and Stagnation Handling in Solar Process Heat Applications, IEA SHC Task 49, SolarPaces Annex IV*, Technical Report A.1.2, January 2015. http://task49.iea-shc.org/Data/Sites/7/frank_iea_shc_task49_overheatingstagnationreport_approved_v-2-3.pdf

[Fru11] M. Fruth, *Formal Methods for the Analysis of Wireless Network Protocols*, PhD Thesis, University of Oxford, Oxford, UK, 2011.

[FTZ16] X. Fafoutis, E. Tsimbalo, W. Zhao, H. Chen, E. Mellios, W. Harwin, R. Piechocki, and I. Craddock, *BLE or IEEE 802.15.4: Which Home IoT Communication Solution is more Energy-Efficient?*, EAI Endorsed Transactions on Internet of Things 2(5), December 2016. https://doi.org/10.4108/eai.1-12-2016.151713

[FWD14] S. Feo-Arenis, B. Westphal, D. Dietsch, M. Muñiz, and A. S. Andisha, *The Wireless Fire Alarm System: Ensuring Conformance to Industrial Standards through Formal Verification* in FM 2014: Formal Methods, Lecture Notes in Computer Science 8442, presented at the 19th International Symposium on Formal Methods (FM), Singapore, May 2014. https://doi.org/10.1007/978-3-319-06410-9_44

[GBA16] D. De Guglielmo, S. Brienza, and G. Anastasi, *IEEE 802.15.4e: A Survey*, Computer Communications 88, August 2016. https://doi.org/10.1016/j.comcom.2016.05.004

[GBS14] D. Garcia, R. Barber, and M. Salichs, *Design and Development of a Wireless Emergency Start and Stop System for Robots* in Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Vienna, Austria, September 2014. https://doi.org/10.5220/0004926602230230

[GD92] P. J. Gmytrasiewicz and E. H. Durfee, *Decision-theoretic Recursive Modeling and the Coordinated Attack Problem* in Proceedings of the 1st International Conference on Artificial Intelligence Planning Systems, College Park, MD, USA, June 1992. http://dl.acm.org/citation.cfm?id=139492.139503

[GJ79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, in A Series of Books in the Mathematical Sciences, ISBN 0-7167-1045-5, W. H. Freeman and Company, 1979.

[GK00] P. Gupta and P. Kumar, *The Capacity of Wireless Networks*, IEEE Transactions on Information Theory 46(2), March 2000. https://doi.org/10.1109/18.825799

[GK05] H.-O. Günther and K. H. Kim, *Container Terminals and Automated Transport Systems*, ISBN 978-3-540-26686-0, Springer-Verlag, 2005. https://doi.org/10.1007/b137951

[GNY04] J. Gronkvist, J. Nilsson, and D. Yuan, *Throughput of Optimal Spatial Reuse TDMA for Wireless Ad-hoc Networks* in Proceedings of the 59th IEEE Vehicular Technology Conference (VTC), Milan, Italy, May 2004. https://doi.org/10.1109/VETECS.2004.1390655

[GPB17] C. Gomez, J. Paradells, C. Bormann, and J. Crowcroft, *From 6LoWPAN to 6Lo: Expanding the Universe of IPv6-Supported Technologies for the Internet of Things*, IEEE Communications Magazine 55(12), December 2017. https://doi.org/10.1109/MCOM.2017.1600534

[GSA14] D. D. Guglielmo, A. Seghetti, G. Anastasi, and M. Conti, *A Performance Analysis of the Network Formation Process in IEEE 802.15.4e TSCH Wireless Sensor/Actuator Networks* in Proceedings of the 19th IEEE Symposium on Computers and Communications (ISCC), Madeira, Portugal, June 2014. https://doi.org/10.1109/ISCC.2014.6912607

[GVS18] M. Ghamari, E. Villeneuve, C. Soltanpur, J. Khangosstar, B. Janko, R. S. Sherratt, and W. Harwin, *Detailed Examination of a Packet Collision Model for Bluetooth Low Energy Advertising Mode*, IEEE Access 6, August 2018. https://doi.org/10.1109/ACCESS.2018.2866323

[Has15] T. Hasegawa, *Safety and Alarming Applications Using ISA100 Wireless*, presented at the ISA Process Control and Safety Symposium, 2015. http://www.isa100wci.org/en-US/Documents/Presentations/2015_ISA_PCS_Symposium_ID75(Yokogawa_Toshi_Hasegaw

[HC04] J. W. Hui and D. Culler, *The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale* in Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys), Baltimore, MD, USA, November 2004. https://doi.org/10.1145/1031495.1031506

[HGT17] R. Teles Hermeto, A. Gallais, and F. Theoleyre, *Scheduling for IEEE802.15.4-TSCH and Slow Channel Hopping MAC in Low Power Industrial Wireless Networks: A Survey*, Computer Communications 114, December 2017. https://doi.org/10.1016/j.comcom.2017.10.004

[HHR16] L. Hanschke, J. Heitmann, and C. Renner, *Challenges of WiFi-Enabled and Solar-Powered Sensors for Smart Ports* in Proceedings of the 4th International Workshop on Energy Harvesting and Energy Neutral Sensing Systems (ENSsys), Stanford, CA, USA, November 2016. https://doi.org/10.1145/2996884.2996887

[HM90] J. Y. Halpern and Y. Moses, *Knowledge and Common Knowledge in a Distributed Environment*, Journal of the ACM 37(3), July 1990. https://doi.org/10.1145/79147.79161

[HN14] K.-i. Hwang and S.-w. Nam, *Analysis and Enhancement of IEEE 802.15.4e DSME Beacon Scheduling Model*, Journal of Applied Mathematics 2014, May 2014. https://doi.org/10.1155/2014/934610

[HV12] K. Heurtefeux and F. Valois, *Is RSSI a Good Choice for Localization in Wireless Sensor Network?* in Proceedings of the 26th International Conference on Advanced Information Networking and Applications (AINA), Fukuoka, Japan, March 2012. https://doi.org/10.1109/AINA.2012.19

[HWW17] R.-H. Hwang, C.-C. Wang, and W.-B. Wang, *A Distributed Scheduling Algorithm for IEEE 802.15.4e Wireless Sensor Networks*, Computer Standards & Interfaces 52, May 2017. https://doi.org/10.1016/j.csi.2017.01.003

[IEC10] IEC 62591:2010 - Industrial Communication Networks - Wireless Communication Network and Communication Profiles - WirelessHART™, 2010.

[INET] *INET Framework for OMNeT++*, Software. https://inet.omnetpp.org/

[ISA11] ANSI/ISA-100.11a-2011 Wireless Systems for Industrial Automation: Process Control and Related Applications, 2011.

[JAG16] I. Juc, O. Alphand, R. Guizzetti, M. Favre, and A. Duda, *Energy Consumption and Performance of IEEE 802.15.4e TSCH and DSME* in Proceedings of the 2016 IEEE Wireless Communications and Networking Conference (WCNC), Doha, Qatar, April 2016. https://doi.org/10.1109/WCNC.2016.7565006

[JJM15] J. Jermyn, R. P. Jover, I. Murynets, M. Istomin, and S. Stolfo, *Scalability of Machine to Machine Systems and the Internet of Things on LTE Mobile Networks* in Proceedings of the 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Boston, MA, USA, June 2015. https://doi.org/10.1109/WoWMoM.2015.7158142

[JKK11] R. Jurdak, K. Klues, B. Kusy, C. Richter, K. Langendoen, and M. Brunig, *Opal: A Multiradio Platform for High Throughput Wireless Sensor Networks*, IEEE Embedded Systems Letters 3(4), December 2011. https://doi.org/10.1109/LES.2011.2177065

[JL12] W.-C. Jeong and J. Lee, *Performance Evaluation of IEEE 802.15.4e DSME MAC Protocol for Wireless Sensor Networks* in Proceedings of the 1st IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things (ETSIoT), Seoul, South Korea, June 2012. https://doi.org/10.1109/ETSIoT.2012.6311258

[Kau18] F. Kauer, *Per-Node Energy Consumption Measurements of the FIT/IoT-LAB*, June 2018. https://doi.org/10.5281/zenodo.1288644

[KDM09] T. Kuruganti, W. Dykas, W. Manges, T. Flowers, M. Hadley, P. Ewing, and T. King, *Wireless System Considerations When Implementing NERC Critical Infrastructure Protection Standards*, White Paper, Oak Ridge National Laboratory, Flowers Control Center Solutions, Pacific Northwest National Laboratory, February 2009.

[KJD07] G. J. Kolb, S. A. Jones, M. W. Donnelly, D. Gorman, R. Thomas, R. Davenport, and R. Lumia, *Heliostat Cost Reduction Study*, Technical Report SAND2007-3293, Sandia National Laboratories, June 2007. https://doi.org/10.2172/912923

[KK00] B. Karp and H. T. Kung, *GPSR: Greedy Perimeter Stateless Routing for Wireless Networks* in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom), Boston, MA, USA, August 2000. https://doi.org/10.1145/345910.345953

[KKL16] F. Kauer, M. Köstler, T. Lübkert, and V. Turau, *Formal Analysis and Verification of the IEEE 802.15.4 DSME Slot Allocation* in Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), Valletta, Malta, November 2016. https://doi.org/10.1145/2988287.2989148

[KKT18] F. Kauer, E. Kallias, and V. Turau, *A Dual-Radio Approach for Reliable Emergency Signaling in Critical Infrastructure Assets with Large Wireless Networks*, International Journal of Critical Infrastructure Protection 21, February 2018. https://doi.org/10.1016/j.ijcip.2018.02.002

[KKT19] F. Kauer, M. Köstler, and V. Turau, *openDSME - Reliable Time-Slotted Multi-Hop Communication for IEEE 802.15.4* in Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem, ed. by A. Virdis and M. Kirsche, ISBN 978-3-030-12841-8, Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-12842-5_15

[KMT17] F. Kauer, F. Meyer, and V. Turau, *A Holistic Solution for Reliable Over-the-Air Software Updates in Large Industrial Plants* in Proceedings of the 13th Workshop on Intelligent Solutions in Embedded Systems (WISES), Hamburg, Germany, June 2017. https://doi.org/10.1109/WISES.2017.7986928

[KP98] K. Khan and H. Peyravi, *Delay and Queue Size Analysis of TDMA with General Traffic* in Proceedings of the 6th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), Montreal, Canada, July 1998. https://doi.org/10.1109/MASCOT.1998.693698

[KRB11] S. Kubisch, M. Randt, R. Buck, A. Pfahl, and S. Unterschütz, *Wireless Heliostat and Control System for Large Self-Powered Heliostat Fields* in Proceedings of the 17th International Symposium on Concentrated Solar Power and Chemical Energy Technologies (SolarPACES), Granada, Spain, September 2011. https://elib.dlr.de/72385/

[KS14] M. Kirsche and M. Schnurbusch, *A New IEEE 802.15.4 Simulation Model for OMNeT++ / INET* in Proceedings of the 1st International OMNeT++ Community Summit, Hamburg, Germany, September 2014. http://arxiv.org/abs/1409.1177

[KT18a] F. Kauer and V. Turau, *An Analytical Model for Wireless Mesh Networks with Collision-Free TDMA and Finite Queues*, EURASIP Journal on Wireless Communications and Networking 2018:149, June 2018. https://doi.org/10.1186/s13638-018-1146-x

[KT18b] F. Kauer and V. Turau, *Constructing Customized Multi-hop Topologies in Dense Wireless Network Testbeds* in Ad-hoc, Mobile, and Wireless Networks, Lecture Notes in Computer Science 11104, presented at the 17th International Conference on Ad Hoc Networks and Wireless (AdHoc-Now), St Malo, France, September 2018. https://doi.org/10.1007/978-3-030-00247-3_28

[KW05] S. S. Kulkarni and L. Wang, *MNP: Multihop Network Reprogramming Service for Sensor Networks* in Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS), Columbus, OH, USA, June 2005. https://doi.org/10.1109/ICDCS.2005.50

[Lam99] L. Lamport, *Specifying Concurrent Systems with TLA+* in Calculational System Design, ed. by M. Broy and R. Steinbrüggen, IOS Press, 1999. http://lamport.azurewebsites.net/pubs/lamport-spec-tla-plus.pdf

[Lar14] L. Larson, *RF and Microwave Hardware Challenges for Future Radio Spectrum Access*, Proceedings of the IEEE 102(3), March 2014. https://doi.org/10.1109/JPROC.2014.2298231

[LBC01] J. Li, C. Blake, D. S. De Couto, H. I. Lee, and R. Morris, *Capacity of Ad Hoc Wireless Networks* in Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom), Rome, Italy, July 2001. https://doi.org/10.1145/381677.381684

[LBV06] K. Langendoen, A. Baggio, and O. Visser, *Murphy Loves Potatoes: Experiences from a Pilot Sensor Network Deployment in Precision Agriculture* in Proceedings of the 20th IEEE International Parallel & Distributed Processing Symposium (IPDPS), Rhodes Island, Greece, April 2006. https://doi.org/10.1109/IPDPS.2006.1639412

[LC16] Y.-S. Lee and S.-H. Chung, *An Efficient Distributed Scheduling Algorithm for Mobility Support in IEEE 802.15.4e DSME-Based Industrial Wireless Sensor Networks*, International Journal of Distributed Sensor Networks 12(2), February 2016. https://doi.org/10.1155/2016/9837625

[LHJ18] W. Li, X. Y. Hu, and T. Jiang, *Path Loss Models for IEEE 802.15.4 Vehicle-to-Infrastructure Communications in Rural Areas*, IEEE Internet of Things Journal Early Access, June 2018. https://doi.org/10.1109/JIOT.2018.2844879

[LHL13] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li, *Does Wireless Sensor Network Scale? A Measurement Study on GreenOrbs*, IEEE Transactions on Parallel and Distributed Systems 24(10), October 2013. https://doi.org/10.1109/TPDS.2012.216

[LJ12] J. Lee and W. C. Jeong, *Performance Analysis of IEEE 802.15.4e DSME MAC Protocol under WLAN Interference* in Proceedings of the 3rd International Conference on ICT Convergence (ICTC), Jeju, Korea, October 2012. https://doi.org/10.1109/ICTC.2012.6387133

[LLS03] Y. Liu, M. J. Lee, and T. N. Saadawi, *A Bluetooth Scatternet-route Structure for Multihop Ad Hoc Networks*, IEEE Journal on Selected Areas in Communications 21(2), February 2003. https://doi.org/10.1109/JSAC.2002.807338

[LLS13] X. Liu, X. Li, S. Su, Z. Fan, and G. Wang, *Enhanced Fast Association for 802.15.4e-2012 DSME MAC Protocol* in Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE), Hangzhou, China, March 2013. https://doi.org/10.2991/iccsee.2013.248

[LLV13] M. Li, Z. Li, and A. V. Vasilakos, *A Survey on Topology Control in Wireless Sensor Networks: Taxonomy, Comparative Study, and Open Issues*, Proceedings of the IEEE 101(12), December 2013. https://doi.org/10.1109/JPROC.2013.2257631

[LNW15] B. Li, L. Nie, C. Wu, H. Gonzalez, and C. Lu, *Incorporating Emergency Alarms in Reliable Wireless Process Control* in Proceedings of the 6th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), Seattle, USA, April 2015. https://doi.org/10.1145/2735960.2735983

[Lou03] R. Lougee-Heimer, *The Common Optimization INterface for Operations Research: Promoting Open-Source Software in the Operations Research Community*, IBM Journal of Research and Development 47(1), January 2003. https://doi.org/10.1147/rd.471.0057

[LPY97] K. G. Larsen, P. Pettersson, and W. Yi, U*ppaal in a Nutshell*, International Journal on Software Tools for Technology Transfer 1(1–2), December 1997. https://doi.org/10.1007/s100090050010

[LSW17] O. Liberg, M. Sundberg, E. Wang, J. Bergman, and J. Sachs, *Cellular Internet of Things*, ISBN 978-0-12-812458-1, Academic Press, 2017. https://doi.org/10.1016/B978-0-12-812458-1.00010-1

[Lüb15] T. Lübkert, *Joint Analysis of TDMA and Geographic Routing for a Large Scale Wireless Mesh Network*, Master's Thesis, Technische Universität Hamburg, Hamburg, Germany, July 2015. https://doi.org/10.15480/882.1664

[MAL15] W. Munawar, M. H. Alizai, O. Landsiedel, and K. Wehrle, *Modular Remote Reprogramming of Sensor Nodes*, International Journal of Sensor Networks 19(3/4), November 2015. https://doi.org/10.1504/IJSNET.2015.072868

[McC16] C. McCaffrey, *The Verification of a Distributed System*, Communications of the ACM 59(2), February 2016. https://doi.org/10.1145/2844108

[Mey18] F. Meyer, *Reliable Data Streams in the Presence of External Interference*, Master's Thesis, Technische Universität Hamburg, Hamburg, Germany, February 2018. https://doi.org/10.15480/882.1667

[MFS13] P. Di Marco, C. Fischione, F. Santucci, and K. Johansson, *Effects of Rayleigh-Lognormal Fading on IEEE 802.15.4 Networks* in Proceedings of the 2013 IEEE International Conference on Communications (ICC), Budapest, Hungary, June 2013. https://doi.org/10.1109/ICC.2013.6654756

[MHM18] A. Morgado, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, *A Survey of 5G Technologies: Regulatory, Standardization and Industrial Perspectives*, Digital Communications and Networks 4(2), April 2018. https://doi.org/10.1016/j.dcan.2017.09.010

[ML16] E. Municio and S. Latré, *Decentralized Broadcast-based Scheduling for Dense Multi-hop TSCH Networks* in Proceedings of the 11th Workshop on Mobility in the Evolving Internet Architecture (MobiArch), New York, USA, October 2016. https://doi.org/10.1145/2980137.2980143

[MMB07] S. Myers, S. Megerian, S. Banerjee, and M. Potkonjak, *Experimental Investigation of IEEE 802.15.4 Transmission Power Control and Interference Minimization* in Proceedings of the 4th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), San Diego, CA, USA, June 2007. https://doi.org/10.1109/SAHCN.2007.4292841

[MMG17] É. Morin, M. Maman, R. Guizzetti, and A. Duda, *Comparison of the Device Lifetime in Wireless Networks for the Internet of Things*, IEEE Access 5, April 2017. https://doi.org/10.1109/ACCESS.2017.2688279

[MMH12] X. Mao, X. Miao, Y. He, X. Y. Li, and Y. Liu, *CitySee: Urban CO2 monitoring with sensors* in Proceedings of the 31st IEEE International Conference on Computer Communications (INFOCOM), Orlando, FL, USA, March 2012. https://doi.org/10.1109/INFCOM.2012.6195530

[MPF12] P. Di Marco, P. Park, C. Fischione, and K. H. Johansson, *Analytical Modeling of Multi-hop IEEE 802.15.4 Networks*, IEEE Transactions on Vehicular Technology 61(7), September 2012. https://doi.org/10.1109/TVT.2012.2201221

[MPT13] K. Mikhaylov, N. Plevritakis, J. Tervonen, K. Mikhaylov, N. Plevritakis, and J. Tervonen, *Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI*, Journal of Sensor and Actuator Networks 2(3), August 2013. https://doi.org/10.3390/jsan2030589

[MSM06] J. Mišić, S. Shafi, and V. B. Mišić, *Performance of a Beacon Enabled IEEE 802.15.4 Cluster with Downlink and Uplink Traffic*, IEEE Transactions on Parallel and Distributed Systems 17(4), April 2006. https://doi.org/10.1109/TPDS.2006.54

[MT15] F. Meier and V. Turau, *An Analytical Model for Fast and Verifiable Assessment of Large Scale Wireless Mesh Networks* in Proceedings of the 11th International Conference on the Design of Reliable Communication Networks (DRCN), Kansas City, MO, USA, March 2015. https://doi.org/10.1109/DRCN.2015.7149011

[Mun17] K. Mun, *CBRS: New Shared Spectrum Enables Flexible Indoor and Outdoor Mobile Solutions and New Business Models*, White Paper, Mobile Experts, March 2017. https://cbrs.wirelessinnovation.org/assets/docs/mobile%20experts%20cbrs%20overview%20final.pdf

[MVV13] A. Morell, X. Vilajosana, J. L. Vicario, and T. Watteyne, *Label Switching over IEEE802.15.4e Networks*, Transactions on Emerging Telecommunications Technologies 24(5), August 2013. https://doi.org/10.1002/ett.2650

[NE124] User Association of Automation Technology in Process Industries (NAMUR), *Wireless Automation Requirements, NE124*, Recommendation, September 2010.

[NERC06] NERC Standard CIP-005-1 - Cyber Security - Electronic Security Perimeter(s), 2006.

[Nix12] M. Nixon, *A Comparison of WirelessHART™ and ISA100.11a*, White Paper, Emerson Process Management, September 2012. https://www.emerson.com/documents/automation/a-comparison-of-wirelesshart-isa100-11a-en-42598.pdf

[NRZ15] C. Newcombe, T. Rath, F. Zhang, B. Munteanu, M. Brooker, and M. Deardeuff, *How Amazon Web Services Uses Formal Methods*, Communications of the ACM 58(4), April 2015. https://doi.org/10.1145/2699417

[NSI15] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez, *IPv6 over BLUETOOTH(R) Low Energy*, RFC 7668, Internet Engineering Task Force, October 2015. https://doi.org/10.17487/RFC7668

[OBB13] T. O'donovan, J. Brown, F. Büsching, A. Cardoso, J. Cecílio, J. D. Ó, P. Furtado, P. Gil, A. Jugel, W.-B. Pöttner, U. Roedig, J. S. Silva, R. Silva, C. J. Sreenan, V. Vassiliou, T. Voigt, L. Wolf, and Z. Zinonos, *The GINSENG System for Wireless Monitoring and Control: Design and Deployment Experiences*, ACM Transactions on Sensor Networks 10(1), November 2013. https://doi.org/10.1145/2529975

[ODE06] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, *Cross-Level Sensor Network Simulation with COOJA* in Proceedings of the 31st International Conference on Local Computer Networks (LCN), Tampla, FL, USA, November 2006. https://doi.org/10.1109/LCN.2006.322172

[OGP17] M. Ojo, S. Giordano, G. Portaluri, D. Adami, and M. Pagano, *An Energy Efficient Centralized Scheduling Scheme in TSCH Networks* in Proceedings of the 2017 IEEE International Conference on Communications Workshops (ICC Workshops), Paris, France, May 2017. https://doi.org/10.1109/ICCW.2017.7962719

[PAB15] G. Patti, G. Alderisi, and L. L. Bello, *SchedWiFi: An Innovative Approach to Support Scheduled Traffic in Ad-hoc Industrial IEEE 802.11 Networks* in Proceedings of the 20th IEEE International Conference on Emerging Technologies Factory Automation (ETFA), Luxembourg, September 2015. https://doi.org/10.1109/ETFA.2015.7301460

[PAG13] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, *On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH*, IEEE Sensors Journal 13(10), October 2013. https://doi.org/10.1109/JSEN.2013.2266417

[PC09] S. Petersen and S. Carlsen, *Performance Evaluation of WirelessHART for Factory Automation* in Proceedings of the 14th IEEE International Conference on Emerging Technologies & Factory Automation (ETFA), Mallorca, Spain, September 2009. https://doi.org/10.1109/ETFA.2009.5346996

[PD08] K. S. J. Pister and L. Doherty, *TSMP: Time Synchronized Mesh Protocol* in Proceedings of the 2008 IASTED International Symposium on Distributed Sensor Networks (DSN), Orlando, FL, USA, November 2008.

[PGS16] G. Z. Papadopoulos, A. Gallais, G. Schreiner, and T. Noël, *Importance of Repeatable Setups for Reproducible Experimental Results in IoT* in Proceedings of the 13th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks (PE-WASUN), Valletta, Malta, November 2016. https://doi.org/10.1145/2989293.2989300

[PGS17] G. Z. Papadopoulos, A. Gallais, G. Schreiner, E. Jou, and T. Noël, *Thorough IoT Testbed Characterization: From Proof-of-Concept to Repeatable Experimentations*, Computer Networks 119, June 2017. https://doi.org/10.1016/j.comnet.2017.03.012

[PHC04] J. Polastre, J. Hill, and D. Culler, *Versatile Low Power Media Access for Wireless Sensor Networks* in Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys), Baltimore, MD, USA, November 2004. https://doi.org/10.1145/1031495.1031508

[PL10] M. Paavola and K. Leiviskä, *Wireless Sensor Networks in Industrial Automation* in Factory Automation, ed. by J. Silvestre-Blanes, ISBN 978-953-307-024-7, IntechOpen, 2010. https://doi.org/10.5772/9532

[PLB16] G. Patti, L. Leonardi, and L. L. Bello, *A Bluetooth Low Energy Real-time Protocol for Industrial Wireless Mesh Networks* in Proceedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society (IECON), Florence, Italy, October 2016. https://doi.org/10.1109/IECON.2016.7793093

[PMB09] U. Pešović, J. Mohorko, K. Benkič, and Ž. Čučej, *Effect of Hidden Nodes in IEEE 802.15.4/ZigBee Wireless Sensor Networks* in Proceedings of the 17th Telecommunications Forum (TELFOR), Serbia, Belgrade, November 2009. http://2009.telfor.rs/files/radovi/02_15.pdf

[PMB10] U. M. Pešović, J. J. Mohorko, K. Benkič, and Ž. F. Čučej, *Single-hop vs. Multi-hop–Energy Efficiency Analysis in Wireless Sensor Networks* in Proceedings of the 18th Telecommunications Forum (TELFOR), Belgrade, Serbia, November 2010. http://2010.telfor.rs/files/radovi/TELFOR2010_03_42.pdf

[PMR15] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, *On the Coverage of LPWANs: Range Evaluation and Channel Attenuation Model for LoRa Technology* in Proceedings of the 14th International Conference on ITS Telecommunications (ITST), Copenhagen, Denmark, December 2015. https://doi.org/10.1109/ITST.2015.7377400

[PRM06] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella, *Performance Study of IEEE 802.15.4 Using Measurements and Simulations* in Proceedings of the 2006 IEEE Wireless Communications and Networking Conference (WCNC), Las Vegas, NV, USA, April 2006. https://doi.org/10.1109/WCNC.2006.1683512

[PRM14] A. Pfahl, M. Randt, F. Meier, M. Zaschke, C. P. W. Geurts, and M. Buselmeier, *A Holistic Approach for Low Cost Heliostat Fields* in Proceedings of the 20th International Conference on Concentrated Solar Power and Chemical Energy Technologies (SolarPACES), Beijing, China, September 2014. https://doi.org/10.1016/j.egypro.2015.03.021

[PSB14] W.-B. Pöttner, H. Seidel, J. Brown, U. Roedig, and L. Wolf, *Constructing Schedules for Time-Critical Data Delivery in Wireless Sensor Networks*, ACM Transactions on Sensor Networks 10(3), April 2014. https://doi.org/10.1145/2494528

[PT92] P. Panangaden and K. Taylor, *Concurrent Common Knowledge: Defining Agreement for Asynchronous Systems*, Distributed Computing 6(2), September 1992. https://doi.org/10.1007/BF02252679

[PWW16] M. R. Palattella, T. Watteyne, Q. Wang, K. Muraoka, N. Accettura, D. Dujovne, L. A. Grieco, and T. Engel, *On-the-Fly Bandwidth Reservation for 6TiSCH Wireless Industrial Networks*, IEEE Sensors Journal 16(2), January 2016. https://doi.org/10.1109/JSEN.2015.2480886

[ROG06] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, *Energy-efficient, Collision-free Medium Access Control for Wireless Sensor Networks*, Wireless Networks 12(1), February 2006. https://doi.org/10.1007/s11276-006-6151-z

[RR00] R. Ramanathan and R. Rosales-Hain, *Topology Control of Multihop Wireless Networks using Transmit Power Adjustment* in Proceedings of the 19th IEEE Conference on Computer Communications (INFOCOM), Tel-Aviv, Israel, March 2000. https://doi.org/10.1109/INFCOM.2000.832213

[San05] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*, ACM Computing Surveys 37(2), June 2005. https://doi.org/10.1145/1089733.1089736

[SBH08] C. Sengul, M. Bakht, A. F. Harris, T. Abdelzaher, and R. Kravets, *On the Feasibility of High-Power Radios in Sensor Networks*, ACM SIGMOBILE Mobile Computing and Communications Review 12(1), January 2008. https://doi.org/10.1145/1374512.1374524

[Sch15] R. Schneiderman, *Modern Standardization: Case Studies at the Crossroads of Technology, Economics, and Politics*, ISBN 978-1-118-67859-6, John Wiley & Sons, 2015.

[SCN12] S. Sen, R. R. Choudhury, and S. Nelakuditi, *CSMA/CN: Carrier Sense Multiple Access With Collision Notification*, IEEE/ACM Transactions on Networking 20(2), April 2012. https://doi.org/10.1109/TNET.2011.2174461

[Seo14] D.-W. Seo, *Explicit Formulae for Characteristics of Finite-Capacity M/D/1 Queues*, ETRI Journal 36(4), August 2014. https://doi.org/10.4218/etrij.14.0113.0812

[SimPy] *SimPy - Discrete Event Simulation for Python*, Software. https://simpy.readthedocs.io

[Sjö17] D. Sjöström, *Unlicensed and Licensed Low-power Wide Area Networks*, Master's Thesis, KTH Royal Institute of Technology, Stockholm, Schweden, September 2017.

[Skl01] B. Sklar, *Digital Communications: Fundamentals and Applications, 2nd Edition*, 2nd edn., ISBN 0-13-084788-7, Prentice Hall, 2001.

[Smi11] J. MacGregor Smith, *Properties and Performance Modelling of Finite Buffer M/G/1/K Networks*, Computers & Operations Research 38(4), April 2011. https://doi.org/10.1016/j.cor.2010.08.014

[SML15] R. Soua, P. Minet, and E. Livolant, *Wave: a Distributed Scheduling Algorithm for Convergecast in IEEE 802.15.4e TSCH Networks (Extended Version)*, Research Report RR-8661, Inria, January 2015. https://hal.inria.fr/hal-01100420

[SML16] R. Soua, P. Minet, and E. Livolant, *Wave: a Distributed Scheduling Algorithm for Convergecast in IEEE 802.15.4e TSCH Networks*, Transactions on Emerging Telecommunications Technologies 27(4), April 2016. https://doi.org/10.1002/ett.2991

[SPW17] P. Sahoo, S. Pattanaik, and S.-L. Wu, *A Novel IEEE 802.15.4e DSME MAC for Wireless Sensor Networks*, MDPI Sensors 17(1), January 2017. https://doi.org/10.3390/s17010168

[SRH98] J. R. Szarka, T. Rempe, and M. Heckl, *Construction Equipment Lockout System with Emergency Shutdown*, Patent US6285860 B1, 1998.

[SRR16] I. Saha, S. Roy, and S. Ramesh, *Formal Verification of Fault-Tolerant Startup Algorithms for Time-Triggered Architectures: A Survey*, Proceedings of the IEEE 104(5), May 2016. https://doi.org/10.1109/JPROC.2016.2519247

[Ste94] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, ISBN 978-0-691-03699-1, Princeton University Press, 1994.

[STW15] G. Siegemund, V. Turau, and C. Weyer, *A Dynamic Topology Control Algorithm for Wireless Sensor Networks* in Ad-hoc, Mobile, and Wireless Networks, Lecture Notes in Computer Science 9143, presented at the 14th International Conference on Ad-Hoc Networks and Wireless (AdHoc-Now), Athens, Greece, June 2015. https://doi.org/10.1007/978-3-319-19662-6_1

[TA18] S. Tonyali and K. Akkaya, *A Scalable Protocol Stack for IEEE 802.11s-based Advanced Metering Infrastructure Networks* in Proceedings of the 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, January 2018. https://doi.org/10.1109/CCNC.2018.8319198

[TDH09] N. Tsiftes, A. Dunkels, Z. He, and T. Voigt, *Enabling Large-Scale Storage in Sensor Networks with the Coffee File System* in Proceedings of the 8th International Conference on Information Processing in Sensor Networks (IPSN), San Francisco, CA, USA, April 2009. https://ieeexplore.ieee.org/document/5211918

[Tex16] Texas Instruments, CC110L Value Line Transceiver, Datasheet, 2016. http://www.ti.com/lit/ds/symlink/cc110l.pdf

[Tex18] Texas Instruments, CC11xx Sensitivity versus Frequency Offset and Crystal Accuracy, SWRA122D, Design Note DN005, 2018. http://www.ti.com/lit/an/swra122d/swra122d.pdf

[TGD17] M. Talebi, J. F. Groote, and C. Dandelski, *Modelling and Verification of a Cluster-tree Formation Protocol Implementation for the IEEE 802.15.4 TSCH MAC Operation Mode* in Proceedings of the 2nd Workshop on Models for Formal Analysis of Real Systems (MARS), Uppsala, Sweden, April 2017. https://doi.org/10.4204/EPTCS.244.5

[Thu18] P. Thubert, *An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4*, Internet-Draft, Internet Engineering Task Force, April 2018. https://tools.ietf.org/html/draft-ietf-6tisch-architecture-14

[TJV08] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. V. Herwegen, and W. Vantomme, *The Industrial Indoor Channel: Large-scale and Temporal Fading at 900, 2400, and 5200 MHz*, IEEE Transactions on Wireless Communications 7(7), July 2008. https://doi.org/10.1109/TWC.2008.070143

[TP16] F. Theoleyre and G. Z. Papadopoulos, *Experimental Validation of a Distributed Self-Configured 6TiSCH with Traffic Isolation in Low Power Lossy Networks* in Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), Valletta, Malta, November 2016. https://doi.org/10.1145/2988287.2989133

[TPK17] P. Thubert, A. Pelov, and S. Krishnan, *Low-Power Wide-Area Networks at the IETF*, IEEE Communications Standards Magazine 1(1), March 2017. https://doi.org/10.1109/MCOMSTD.2017.1600002ST

[TW07] V. Turau and C. Weyer, *TDMA-Schemes for Tree-Routing in Data Intensive Wireless Sensor Networks* in Proceedings of the 2007 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), Pisa, Italy, October 2007. https://doi.org/10.1109/MOBHOC.2007.4428749

[TWH07] L. Tang, K.-C. Wang, Y. Huang, and F. Gu, *Channel Characterization and Link Quality Assessment of IEEE 802.15.4-Compliant Radio for Factory Environments*, IEEE Transactions on Industrial Informatics 3(2), May 2007. https://doi.org/10.1109/TII.2007.898414

[TWP11] A. Tinka, T. Watteyne, K. S. J. Pister, and A. M. Bayen, *A Decentralized Scheduling Algorithm for Time Synchronized Channel Hopping*, EAI Endorsed Transactions on Mobile Communications and Applications 11(1), September 2011. https://doi.org/10.4108/icst.trans.mca.2011.e5

[UCTE04] Union for the Coordination of the Transmission of Electricity (UCTE), *FINAL REPORT of the Investigation Committee on the 28 September 2003 Blackout in Italy*, Report, April 2004.

[Unt14] S. Unterschütz, *Methodologies and Protocols for Wireless Communication in Large-Scale, Dense Mesh Networks*, PhD Thesis, Hamburg University of Technology, Hamburg, Germany, ISBN 978-3-95404-707-9, Cuvillier Verlag, April 2014.

[UT12a] S. Unterschütz and V. Turau, *Fail-Safe Over-The-Air Programming and Error Recovery in Wireless Networks* in Proceedings of the 10th International Workshop on Intelligent Solutions in Embedded Systems (WISES), Klagenfurt, Austria, July 2012. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6273600

[UT12b] S. Unterschütz and V. Turau, *Reliable Signaling an Emergency Shutdown in Large-scale, Wireless Controlled Industrial Plants* in Proceedings of the 10th ACM International Symposium on Mobility Management and Wireless Access (MobiWac), Paphos, Cyprus Island, November 2012. https://doi.org/10.1145/2386995.2387004

[UWT12] Stefan Unterschütz, Andreas Weigel and Volker Turau, *Cross-Platform Protocol Development Based on OMNeT++* in Proceedings of the 5th International Workshop on OMNeT++, Desenzano del Garda, Italy, March 2012. https://dl.acm.org/citation.cfm?id=2263063

[Var01] A. Varga, *The OMNeT++ Discrete Event Simulation System* in Proceedings of the 15th European Simulation Multiconference (ESM), Prague, Czech Republic, June 2001. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.331.1728

[VBP17] C. Vallati, S. Brienza, M. Palmieri, and G. Anastasi, *Improving Network Formation in IEEE 802.15.4e DSME*, Computer Communications 114, December 2017. https://doi.org/10.1016/j.comcom.2017.09.016

[VPW17] X. Vilajosana, K. Pister, and T. Watteyne, *Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration*, RFC 8180, Internet Engineering Task Force, May 2017. https://doi.org/10.17487/RFC8180

[VRV15] L. Varga, G. Romaniello, M. Vučinić, M. Favre, A. Banciu, R. Guizzetti, C. Planat, P. Urard, M. Heusse, F. Rousseau, O. Alphand, É. Dublé, and A. Duda, *GreenNet: An Energy-Harvesting IP-Enabled Wireless Sensor Network*, IEEE Internet of Things Journal 2(5), October 2015. https://doi.org/10.1109/JIOT.2015.2425431

[WAV15] T. Watteyne, C. Adjih, and X. Vilajosana, *Lessons Learned from Large-scale Dense IEEE 802.15.4 Connectivity Traces* in Proceedings of the 11th International Conference on Automation Science and Engineering (CASE), Gothenburg, Sweden, August 2015. https://doi.org/10.1109/CoASE.2015.7294053

[WEC05] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, *Siphon: Overload Traffic Management Using Multi-Radio Virtual Sinks in Sensor Networks* in Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys), San Diego, California, USA, November 2005. https://doi.org/10.1145/1098918.1098931

[Wie03] J. Wiewiura, *Remote Emergency Power Shutoff and Alarm System*, Patent US7005997 B1, 2003.

[WLZ09] F. Wang, D. Li, and Y. Zhao, *Analysis and Compare of Slotted and Unslotted CSMA in IEEE 802.15.4* in Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing, Beijing, China, September 2009. https://doi.org/10.1109/WICOM.2009.5303580

[WPG15] T. Watteyne, M. R. Palattella, and L. A. Grieco, *Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement*, RFC 7554, Internet Engineering Task Force, May 2015. https://doi.org/10.17487/RFC7554

[WPL02] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, *Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement* in Proceedings of the 21th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), New York, USA, June 2002. https://doi.org/10.1109/INFCOM.2002.1019305

[WSW05] G. Werner-Allen, P. Swieskowski, and M. Welsh, *MoteLab: A Wireless Sensor Network Testbed* in Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN), Boise, ID, USA, April 2005. https://doi.org/10.1109/IPSN.2005.1440979

[WT15] A. Weigel and V. Turau, *Hardware-Assisted IEEE 802.15.4 Transmissions and Why to Avoid Them* in Internet and Distributed Computing Systems, Lecture Notes in Computer Science 9258, presented at the 8th International Conference on Internet and Distributed Computing Systems (IDCS), Windsor, UK, September 2015. https://doi.org/10.1007/978-3-319-23237-9_20

[WTB12] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, RFC 6550, Internet Engineering Task Force, March 2012. https://doi.org/10.17487/RFC6550

[WVK12] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, and K. Pister, *OpenWSN: A Standards-Based Low-Power Wireless Development Environment*, Transactions on Emerging Telecommunications Technologies 23(5), August 2012. https://doi.org/10.1002/ett.2558

[WVW18] Q. Wang, X. Vilajosana, and T. Watteyne, *6TiSCH Operation Sublayer Protocol (6P)*, Internet-Draft, Internet Engineering Task Force, June 2018. https://tools.ietf.org/html/draft-ietf-6tisch-6top-protocol-12

[WWP15] J. R. Wilcox, D. Woos, P. Panchekha, Z. Tatlock, X. Wang, M. D. Ernst, and T. Anderson, *Verdi: A Framework for Implementing and Formally Verifying Distributed Systems* in Proceedings of the 36th SIGPLAN Conference on Programming Language Design and Implementation (PLDI), Portland, OR, USA, June 2015. https://doi.org/10.1145/2737924.2737958

[XHS17] Z. Xu, T. Hu, and Q. Song, *Bulk Data Dissemination in Low Power Sensor Networks: Present and Future Directions*, MDPI Sensors 17(1), January 2017. https://doi.org/10.3390/s17010156

[YGK17] B. Yousefi, F. Ghassemi, and R. Khosravi, *Modeling and Efficient Verification of Wireless Ad hoc Networks*, Formal Aspects of Computing 29(6), November 2017. https://doi.org/10.1007/s00165-017-0429-z

[ZigBee12] Zigbee Alliance, ZigBee Specification, Document 053474r20, 2012. https://www.zigbee.org/download/standards-zigbee-specification/

[ZL07] J. Zheng and M. J. Lee, *A Comprehensive Performance Study of IEEE 802.15.4* in Sensor Network Operations, ed. by S. Phoha, T. LaPorta, and C. Griffin, ISBN 0-471-71976-5, John Wiley & Sons, 2007.

[ZSY17] A. Zolfaghari, M. E. Said, M. Youssef, G. Zhang, T. T. Liu, F. Cattivelli, Y. I. Syllaios, F. Khan, F. Q. Fang, J. Wang, K. J. Li, F. H. Liao, D. S. Jin, V. Roussel, D. Lee, and F. M. Hameed, *A Multi-mode WPAN (Bluetooth, BLE, IEEE 802.15.4) SoC for Low-power and IoT Applications* in Proceedings of the 2017 Symposia on VLSI Technology and Circuits, Kyoto, Japan, June 2017. https://doi.org/10.23919/VLSIC.2017.8008554

List of Symbols

$\mathcal{A}_n$	Total expected number of generated or received packets per slotframe.
$\boldsymbol{A}_{n,i}$	Random variable for the number of arriving, i.e. generated and received, packets in slot i at node n.
b_ζ	Probability of being in state θ_ζ.
BER_l	Bit error ratio.
$\mathcal{B}_n$	Relation of slots to blocked channels.
BO	Beacon order.
b_r	Number of properly received bits during synchronization.
b_s	Number of bits of the synchronization sequence.
BW_{chan}	Receiver channel filter bandwidth.
BW_{signal}	Bandwidth of the transmitted signal.
$\mathcal{C}$	Set of channels.
$\lvert\mathcal{C}\rvert$	Number of channels.
$\vec{c}$	Vector of the $c_{n,q,i}$.
$c_{act,t}$	Actual capacity.
$\mathcal{CA}_l$	Event that an acknowledgment on l collides.
$\mathcal{CB}_l$	Event that transmissions mutually collide.
$\mathcal{C}_n$	Relation of slots to channel.
$\mathbf{C}_n$	Children of node n.
$c_{n,q,i}$	Probability of being in state $\sigma^n_{q,i}$ in the stationary distribution.
$\mathcal{CP}_l$	Event that a packet on l collides.
$c_{req,t}$	Required capacity.
$\mathcal{CR}_l$	Event that transmissions collide again.
D	Duty cycle.
$D(v,w)$	Predicate that describes potential collisions between nodes.
D_{adv}	Inter-packet interval for transmitting Deluge advertisements.
D_{data}	Inter-packet interval for transmitting Deluge data packets.
$\boldsymbol{\mathcal{D}}_{i,l}$	The set of possibly disturbing links during slot i.
$\boldsymbol{D}_n$	Random variable for the queuing delay.
$\mathfrak{D}_{\mathrm{up},n}$	End-to-end packet delay.
f_c	Carrier frequency.
g_{down}	Packet generation rate in downstream.
g_{up}	Packet generation rate in upstream.
h	Modulation index.
I	Beacon interval.
I_{down}	Packet interval in downstream.
IFS	Inter-frame spacing as defined in the standard.
$I_{\max}$	Maximum beacon interval.

$I_{\min}$	Minimum beacon interval.
I_{up}	Packet interval in upstream.
K	Maximum queue length.
$\mathcal{K}_n$	Relation of slots to counterpart.
l	Number of bytes in a beacon.
$\mathbf{L}$	The set of active links.
$\mathcal{LA}_l$	Event that an acknowledgment on l gets lost.
$\mathbf{L}_{\text{down}}$	The set of active links in downstream direction.
$\mathbf{L}_i$	The set of active links during slot i.
l_M	Number of bytes in the message authentication code.
l_P	Number of bytes in the preamble.
$\mathcal{LP}_l$	Event that a packet on l gets lost.
L_{PSDU}	Number of Bytes in the PSDU, excluding preamble, SFD and PHY header.
l_S	Overall number of slots in the slotframe.
l_{sync}	Number of bytes in the sync word.
l_T	Number of bytes in the timestamp.
$\mathbf{L}_{\text{up}}$	The set of active links in upstream direction.
m	Maximum number of channel sensing attempts, excluding the first, unconditional one (*macMaxCsmaBackoffs*).
$\widetilde{m}$	Number of missing beacons that signal an emergency condition.
$\overline{m}$	Shortcut for $m_b - m_0$.
m_0	Initial backoff exponent.
m_b	Maximum backoff exponent.
MO	Multi-superframe order.
M_x	macDsmeGTSExpirationTime.
n	Maximum number of retransmission attempts (*macMaxFrameRetries*).
N	Total number of nodes in the network.
$\mathbf{N}_n$	Neighbors of node n.
N_p	Number of packets sent per node per run.
$N_{r,i}$	Number of packets received originating from node i.
N_s	Slots per multi-superframe.
$\mathcal{N}_{u,\hat{\lambda}}$	Set of neighbors of u with a loss smaller than or equal to $\hat{\lambda}$.
$\overline{p}$	Average number of incoming packets per multi-superframe.
$P_{\text{accept},n}$	Packet accepting probability.
P_{dist}	Minimum power that interferes with a reception.
PDR_i	End-to-end packet delivery ratio for node i.
$\text{PDR}_{\text{down},i}$	End-to-end packet delivery ratio for packets generated by v_0 and delivered to node i.
$\text{PDR}_{\text{up},i}$	End-to-end packet delivery ratio for packets generated by node i and delivered to v_0.
$\text{PER}_{b,l}$	Packet error ratio for a transmission of b bytes.
P_{int}	Probability that λ_t stays within the hysteresis interval.
p_n	Parent of node n.
P_N	Noise power.

$P_{\text{noACK},l}$	Probability that no acknowledgment arrives after sending a packet over l.
P_{rx}	Reception power.
p_t	Incoming packets per multi-superframe.
P_{tx}	Transmission power.
q_l	Packet sending probability.
$\boldsymbol{Q}_n$	Random variable for the number packets inserted into the queue.
$Q(t, \mathbf{S})$	Probability that at least one link in $\mathbf{S}$ starts a transmission within t time units.
r	Data rate.
R_l	Reliability of packet transmission on a link.
$R_{l,indep}$	Reliability of a link assuming constant collision probability over j.
$\mathcal{R}_n$	Reception slots.
$r_{n,i}$	Reception slot.
R_l^R	Reliability of request packet transmission on a link.
$\mathcal{R}_{\mathcal{R},l}$	Set of links with receivers in the interference range of the receiver.
$\mathcal{R}_{\mathcal{S},l}$	Set of links with senders in the interference range of the receiver.
S_b	Backoff unit time (used as basic time unit).
SD	Superframe duration.
SNR_l	Signal to noise ratio.
SO	Superframe order.
$\mathcal{S}_{\mathcal{R},l}$	Set of links with receivers in the interference range of the sender.
$\mathcal{S}_{\mathcal{S},l}$	Set of links with senders in the interference range of the sender.
t_{ack}	Time interval between packet transmission and acknowledgment.
T_{ack}	Duration of acknowledgment.
T_B	Transmission time of a beacon.
$T_{(i,j)}^{data}$	Duration until the data is successfully transmitted.
$T_{\text{data}}^{\text{Deluge}}$	Transmission duration in S_b of a Deluge data packet.
$T_{\text{req}}^{\text{Deluge}}$	Transmission duration in S_b of a Deluge request.
T_F	Time between generation and sending of a beacon.
T_{fail}	Total time for an unsuccessful transmission.
t_l	Current local time in milliseconds since reboot.
$t_{l,r}$	Local time at the beginning of a reception.
T_m	Multi-superframe duration.
T_M	Calculation time of the MIC.
$t_{m,ack}$	Time to wait for a lost acknowledgment (*macAckWaitDuration*).
$T_{p,j}^{mh}$	Duration until p pages are successfully transmitted from the sink.
$\mathcal{T}_n$	Transmission slots.
$\lvert\mathcal{T}_n\rvert$	Number of transmission slots in the schedule for node n.
$t_{n,i}$	Transmission slot.
T_p	Duration of packet sending.
$T(p, c)$	p is parent of c in the routing tree.
$T_{(i,j)}^{pg}$	Duration until a page is successfully transmitted.
$T_{(i,j)}^{req}$	Duration until a request is successfully transmitted.
t_s	Timestamp included in the beacon.

T_s	Duration of a slot.
T_{succ}	Total time for a successful transmission.
T_{shift}	Time shift between local and global time.
T_w	Timeout.
$\mathcal{U}$	Unusable slots.
v_0	Root node.
v_n	Node with index n.
W_i	Maximum backoff time span for the i^{th} channel sensing.
W_r	macResponseWaitTime.
$\boldsymbol{x}$	Vector to determine which nodes to select.
y_l	Shortcut for $P_{\text{noACK},l} \cdot \left(1 - {\alpha_l}^{m+1}\right)$.
α_l	Probability of sensing a busy channel.
$\alpha_{pkt,l}$	Probability of sensing a packet.
$\alpha_{ack,l}$	Probability of sensing an acknowledgment.
α_λ	Filter coefficient for packet predictor.
$\boldsymbol{\beta}_l$	Probability of sensing a free channel before the maximum backoff counter is reached, shortcut for $1 - {\alpha_l}^{m+1}$.
$\beta_{n,i}$	Packet probability in the Bernoulli part of the traffic model.
γ_n	Number of proper descendants of node n.
δ	The current depth.
Δ	Path loss margin.
δ_{best}	The maximum depth found.
Δf	Frequency deviation.
ϵ	Elected neighbor.
θ_s	State in the MAC layer Markov chain.
$\kappa(\delta)$	Minimum required number of nodes per level.
κ_s	State in the retransmission Markov chain.
$\hat{\lambda}$	The current path loss bound.
$\Lambda_{A,B}$	Loss between node A and B.
λ_l	Packet sending rate.
$\lambda_{n,i}$	Packet rate in the Poisson part of the traffic model.
λ_t	Predicted incoming packets per multi-superframe.
μ_l	Packet forwarding rate.
$\mu_{n,i}^{\text{RX}}$	Probability of receiving a packet in slot i.
$\mu_{n,i}^{\text{TX}}$	Probability of sending a packet in slot i.
$\boldsymbol{\Pi}_{n,i}$	Random variable for the number of arriving packets according to the Poisson part of the traffic model.
$\boldsymbol{\Sigma}_n$	The set of states for the queuing model.
$\sigma_{q,i}^n$	State in the queuing model.
τ_l	Probability of a channel sensing attempt on link l.
$\tau_{n,i}$	1 if $i \in \mathcal{T}_n$, 0 otherwise.
ϕ	Predicate denoting that the schedule is consistent.
ϕ_u	Divisor for schedule calculation.
ω	Number of available slots.

Curriculum Vitae

Personal Data

Name	Florian Steffen Kauer
Name at Birth	Meier
Date of Birth	May 22, 1988
Place of Birth	Hamburg, Germany

Education and Work

Since 11.2018	Embedded Systems Engineer *Eppendorf Instrumente GmbH, Germany*
11.2013 - 09.2018	Research and Teaching Assistant *Institute of Telematics, Hamburg University of Technology (TUHH), Germany*
10.2011 - 10.2013	Master Computer Science and Engineering *Hamburg University of Technology (TUHH), Germany*
10.2008 - 09.2011	Bachelor Computer Science and Engineering *Hamburg University of Technology (TUHH), Germany*
05.2008 - 10.2013	Internship and Working Student *Schäfter + Kirchhoff GmbH, Germany*
09.2007 - 05.2008	Civilian Service *Jugendfreizeithaus Buschweg, Norderstedt, Germany*
08.1998 - 06.2007	Secondary School *Gymnasium Grootmoor, Hamburg, Germany, Final degree: Abitur*
08.1994 - 07.1998	Primary School *Grundschule Hasenweg, Hamburg, Germany*