5-2018

Efficient Control Message Dissemination in Dense Wireless Lighting Networks

Conrad Dandelski
Cork Institute of Technology

Efficient Control Message Dissemination in Dense Wireless Lighting Networks

Conrad Dandelski, B.Eng.

Nimbus Centre for Embedded Systems Research

Electrical & Electronic Engineering, Cork Institute of Technology

Supervised by

Prof. Dr. Dirk Pesch & Prof. Dr.-Ing. Michael Kuhn & Dr.-Ing. Bernd-Ludwig Wenning

A thesis submitted for the degree of

Doctor of Philosophy

to Cork Institute of Technology, May 2018

Declaration

This thesis is entirely the candidates own work except where otherwise accredited and has not been submitted for an award at any other institution.

The research presented in this thesis was carried out in compliance with the CIT Code of Good Practice in Research.

Candidate: ____________________ **Date**: ____________

Principal Supervisor: ____________________ **Date**: ____________

Second Supervisor: ____________________ **Date**: ____________

Third Supervisor: ____________________ **Date**: ____________

GSP Chair: ____________________ **Date**: ____________

TO MY PARENTS, MARGIT AND ALBERT, THANKS FOR ALL THE SUPPORT DURING ALL THESE YEARS, I COULD NOT HAVE DONE THIS WITHOUT YOUR SUPPORT

FÜR MEINE ELTERN, MARGIT UND ALBERT, DANKE FÜR DIE GANZE UNTERSTÜTZUNG WÄHREND DEN GANZEN JAHREN. OHNE EUCH WÄRE ICH NICHT DORT WO ICH HEUTE BIN

Acknowledgements

I would like to thank my family for the support during my time in Ireland. Especially for the support I received from my parents, you always brought me down to earth, and my sisters Luisa and Johanna. I am thankful to my supervisors during my time at the Nimbus Centre Prof Dirk Pesch, Prof Michael Kuhn, and Dr Bernd-Ludwig Wenning for the suggestions on the work and thesis, the fruitful discussions and so much more. A special thank you to my colleagues at the Nimbus Centre for the great lunchtime conversations, Friday football sessions and much more: Sunny Jay, Martina Hayes, Bill Quinn, Dave O'Leary, David Rojas, Kritchai Witheephanich, Jean Michel Rubillon, John Swanton, Juan F Martinez, and 1mA (Juan Manuel Escano). The Wednesday football group and the UCC Olympic Handball team which made the life in Ireland much more fun. The work of the students, which I have supervised, has contributed to this thesis: Gwendoline Briere, Luc Denissen, Alex Wu Fan, Clement Festal, and Vicenc Pio Badia. Other huge thanks to my friends in Ireland or wherever they live now: Nial Marshall, Cian Crowley, Andi Albrecht, Paul Verling, Andrea Vesetin, and the apartment from Copley Court (especially Paul Coupel). And last, but not least, my friends back home in Germany who never missed an opportunity to cheer me up: Dennis Arras, Sebastian Bielesch, Marcel Chin, Sebastian Hartmann, Andreas Kühn, Andre Lannert, Annika Ripper, Miriam Schulz, Ina Wagner, Alexander Klenk, and Marcus Maxeiner. Thank you for all the wonderful moments we shared.

This research has emanated in part from research supported by research grants from Science Foundation Ireland (SFI) and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077 and in part supported by the DEWI project (`www.dewi-project.eu`), which has received funding from the ARTEMIS Joint Undertaking under Grant Agreement N^o 621353 and from Enterprise Ireland.

List of Publications

1. Mahmoud Talebi, Jan Friso Groote, **Conrad Dandelski**. Modelling and verification of a cluster-tree formation protocol implementation for the IEEE 802.15.4 TSCH MAC operation mode *2nd Workshop on Models for Formal Analysis of Real Systems (MARS 2017).* 2017.
2. **Conrad Dandelski**, Michael Kuhn. Data Dissemination in Dense Wireless Lighting Control Networks - Demonstration *CeBIT, Hannover* 2017.
3. **Conrad Dandelski**, Bernd-Ludwig Wenning, Michael Kuhn, Dirk Pesch. RLL - Reliable Low Latency Broadcast Data Dissemination in Dense Wireless Lighting Control Networks *21st IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2016).* 2016.
4. **Conrad Dandelski**, Bernd-Ludwig Wenning, Michael Kuhn, Dirk Pesch. Broadcast Storm Problem in Dense Wireless Lighting Control Networks *Twelfth International Symposium on Wireless Communication Systems (ISWCS'2015).* 2015.
5. **Conrad Dandelski**, Bernd-Ludwig Wenning, Daniel Viramontes Perez, Dirk Pesch, Jean-Paul M.G. Linnartz. Scalability of Dense Wireless Lighting Control Networks *IEEE Communication Magazine.* 2015.

Abstract

Modern lighting systems using LED light sources lead to dense lighting installations. The control of such systems using wireless Machine-to-Machine (M2M) where standard LED light sources are replaced by wirelessly controllable LED light sources create new problems which are investigated in this thesis. Current approaches for control message transmission in such networks are based on broadcasting messages among many luminaires. However, adequate communication performance – in particular, sufficiently low latency and synchronicity – is difficult to ensure in such networks, in particular, if the network is part of a wireless building management system and carries not only low-latency broadcast messages but also collects data from sensors. In this thesis, the problem of simultaneously controlling dense wireless lighting control networks with a high number of luminaires is addressed. Extensive computer simulation shows that current state-of-the-art protocols are not suitable for lighting control applications, especially if complex applications are required such as dimming or colour tuning. The novel D^3LC-Suite is proposed, which is specially designed for dense wireless lighting control networks. This suite includes three sub-protocols. First, a protocol to organize a network in form of a cluster tree named CIDER. To ensure that intra-cluster messages can be exchanged simultaneously, a weighted colouring algorithm is applied to reduce the inter cluster interference. To disseminate efficiently control messages a protocol is proposed named RLL. The D^3LC-Suite is evaluated and validated using different methods. A convergence analysis shows that CIDER is able to form a network in a matter of minutes. Simulation results of RLL indicate that this

protocol is well suitable for dense wireless lighting applications. In extensive experiments, it is shown that the D^3LC-Suite advances the current state-of-the-art in several aspects. The suite is able to deliver control messages across multiple hops meeting the requirements of lighting applications. Especially, it provides a deterministic latency, very promising packet loss ratios in low interference environments, and mechanisms for simultaneous message delivery which is important in terms of Quality of Experience (QoE).

Abbreviations

6LoWPAN IPv6 over Low-Power Wireless Personal Area Network

6TiSCH IPv6 over the TSCH mode of IEEE 802.15.4

ACK Acknowledgement

AMCA Asynchronous Multi-Channel Adaption

AP Access Point

APL Application

ARQ Automatic Repeat reQuest

ASN Absolute Slot Number

BLINK Radio Frequency Identification Blink

BS Base Station

CCA Clear Channel Assessment

CDF Cumulative Distribution Function

CE Consumer Electronic

CH Cluster Head

CIDER Clustering In Dense EnviRonments

CS Cluster Slave

CSMA/CA Carrier Sense Multiple Access/Collision Avoidance

D^{3}LC Data Dissemination in Dense Lighting Control

DARPA United States Defense Advanced Research Projects Agency

DeTAS Decentralized Traffic Aware Scheduling

DEWI Dependable Embedded Wireless Infrastructure

DODAG Destination Oriented Directed Acyclic Graph

DSATUR Degree of Saturation

DSDA Distributed Saturation Degree Algorithm

DSME Deterministic and Synchronous Multi-Channel Extension

DSN Distributed Sensor Network

DWEHC Distributed Weight-Based Energy-Efficient Hierarchical Clustering

DWLCN Dense Wireless Lighting Control Networks

ECPF Energy-Aware Distributed Dynamic Clustering Protocol Using Fuzzy logic

EDF Empirical Distribution Function

ECC Error Correction Codes

FIFO First-In First-Out

GERAN GSM EDGE Radio Access Network

GGPSR Geocast Greedy Perimeter Stateless Routing

GPRS General Packet Radio Service

HEED Hybrid Energy-Efficient Distributed clustering

HiPR High-Priority Rooms

HVAC Heating, Ventilation, and Air Conditioning

IETF Internet Engineering Task Force

IoT Internet of Things

IP Internet Protocol

IPv6 Internet Protocol Version 6

IR Infrared

LAN Local Area Networks

LEACH Low-Energy Adaptive Clustering Hierarchy

LLDN Low-Latency Deterministic Network

LON Local Operation Network

LoPR Low-Priority Rooms

LP Loss Probability

LPL Low Power Listening

LQIR Link Quality Indicator-Based Routing

LR-WPAN Low-Rate Wireless Personal Area Network

LSB Least Significant Bits

LWB Low-Power Wireless Bus

M2M Machine-to-Machine

MAC Medium Access Control

MANETs Mobile Adhoc NETworks

MTBF Mean Time Between Failures

NACK Negative ACK

NDN Named Data Networking

NWK Network

OSAS Open Service Architecture for Sensors

P2P Peer-to-Peer

PAN Personal Area Network

PCE Path Computation Element

PCTL Probabilistic Computation Tree Logic

PHY Physical

PLC Power Line Communication

PLR Packet Loss Rate

QoE Quality of Experience

QoS Quality of Service

RF Radio Frequency

RFID Radio Frequency Identification

RIT Receiver-Initiated Transmission

RLL Reliable Low-Latency Data Dissemination Protocol

RPL Routing Protocol for Low-Power and Lossy Networks

RSSI Received Signal Strength Indicator

SBA Scalable Broadcast Algorithm

SCADA Supervisory Control And Data Acquisition

SDI Saturation Degree Indicator

SEMS Smart Energy Management System

SNR Signal-to-Noise Ratio

SOSUS Sound Surveillance System

SPIN Sensor Protocols for Information via Negotiation

SSID Service Set Identifier

TASA Traffic-Aware Scheduling Algorithm

TCP Transmission Control Protocol

TDMA Time Division Multiple Access

TEEN Threshold-Sensitive Energy-Efficient Sensor Network

TSCH Time-Slotted Channel Hopping

UDP User Datagram Protocol

UTRAN Universal Terrestrial Radio Access Network

VANETs Vehicular Adhoc NETworks

VoIP Voice over IP

WCA Weighted Clustering Algorithm

WG Working Group

WLAN Wireless Local Area Network

WPAN Wireless Personal Area Network

WRAN Wireless Regional Area Network

WSN Wireless Sensor Network

WSAN Wireless Sensor and Actuator Networks

ZLL ZigBee Light Link

Table of contents

List of Figures

List of Tables

List of equations

Chapter 1

Introduction

M2M communications is a topic that has provided a lot of material for story writers. Take, for example, the movie “Minority Report” by Steven Spielberg, where police have sent intelligent and networked machines into a building, searching for any evidence of Tom Cruise [120]. These small and autonomous sensors communicated with each other in a fast-changing, wireless mesh network, showing how a basic technology can be transformed into applications. However, an M2M communication technology is much more than just an idea for a movie writer [126]. Over the past years, M2M has been the subject of many discussions. The aim of M2M is to connect objects, ranging from sensors to actuators, with robots or smart meters, with or without human interaction [133]. Most people are not aware that M2M communication is not a new technology. It has existed for more than a century and is well established in our daily lives. The evolution of communication systems goes back to the mid-1800s where it found its first application in telemetry. In 1845 the first transmission circuit was developed by the Russian Army in order to connect the Winter Palace of the Russian Tsar with the army headquarters [109]. This was the most basic form of a wired data transfer network. In fact, most of today’s known technologies have their origins in military applications [46]. During the Cold War, the Sound Surveillance System (SOSUS) was used to detect quiet Soviet submarines in strategic locations [27]. However, after the first system was established in Russia, in the mid-1990s, the technology moved quickly from using the telephone infrastructure to using weather balloons to aerospace telemetry [109]. In the 2000s, M2M quickly evolved to the so-called

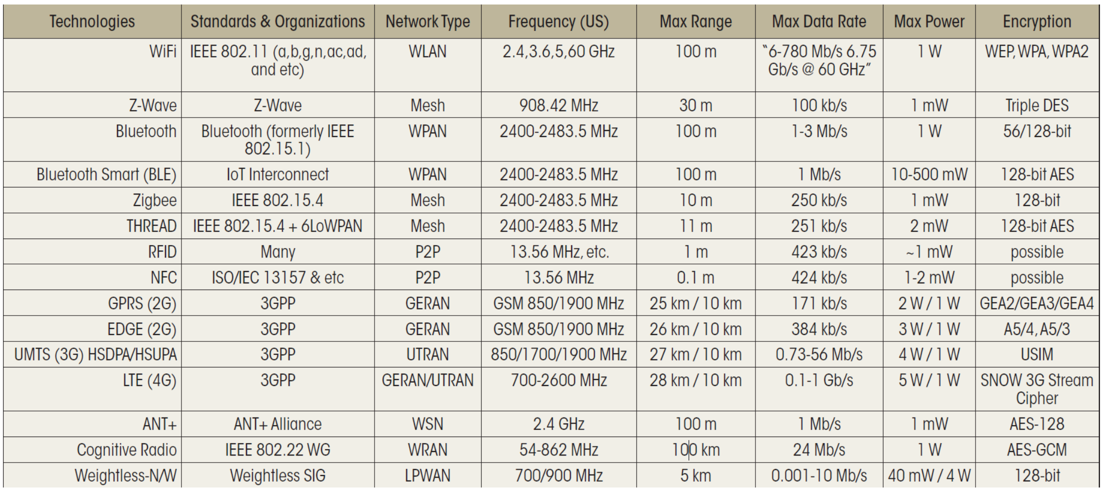
Internet of Things (IoT). The term was first mentioned by Kevin Ashton, who linked it with the idea of using Radio Frequency Identification (RFID) in a production facility [1]. RFID can be seen as the first technology that enabled things to connect to the Internet [123]. This technology received greater attention when in 2003, LG announced the deployment of RFIDs in refrigerators to keep track of its content and send notifications when items have run out. However, IoT is simply a term which is explained as connecting humans, computers, and things. Basically, nowadays, any device can be controlled through the IoT. This enabled controlling anything from anywhere in the world, e.g. a user could turn the heating on in his home in Ireland while boarding a plane in the U.S. Enabling this IoT combines many technologies such as Bluetooth, RFID, Wi-Fi, and telephone data services as well as embedded sensor and actuator nodes [58]. An example of how IoT finds its way into our daily life can be seen by the Amazon Dash Button [11]. The Amazon Dash Button is meant to order any product by a single click on a button. This means that one is able to link a button to a product, and as soon as one pushes the button, it places an order with Amazon. This is only one of many examples of how the IoT can be used. An core underlying aspect of IoT is the M2M communication. Without it, the IoT as we know it would have never become a reality [116]. Basically, M2M communication is necessary to connect the physical world around us; M2M provides communication ability between the physical and the cyber-world combined with additional intelligence M2M enables the autonomy of IoT devices. The devices are able to make decisions based on events, e.g. across a city. An example is traffic control where sensors and cameras are deployed across a city, feeding information into a central control unit running an intelligent software. The software can then, based on the information received, manipulate traffic lights and speed limits to optimize the traffic flow.

A key factor in the success of M2M is based on the availability of low cost and ubiquitous connectivity [130]. The use of Internet-connectable things, such as sensors, monitors, and actuators, has enabled the growth of interconnected, interoperable services that are capable of changing our everyday lives. It is important to take into account the large range of possible applications, the variety and range of existing applications, device functionalities, and other requirements as the key features that shape M2M communications and its future market. A typical M2M system can be divided into three major parts. The first is the local M2M network which is composed of a large number of devices, such as sensors, actuators, and smart meters. These devices are able to collect data from the deployed area, and based on smart decisions, they either forward the data to a gateway or not. The gateway is an intelligent device that is able to gather all data from the devices in the network and forwards them via another technology to a back-end server. The second part of an M2M system is the global communication infrastructure. A gateway is able to forward a message using different technologies such as telephone or cellular networks. The third part is the M2M application where the back-end server provides all gathered data to multiple clients [130]. While M2M covers a broad spectrum of different applications and systems, this dissertation focuses on local M2M networks only. An overview of the typical applications and their traffic demand for M2M systems can be found in Table 1.1. In designing systems for such applications, many things need to be considered, ranging from costs and power to long-term product life-cycle challenges and interference [36]. There is a need to collect the data, control the devices, and communicate in a

M2M application	Traffic demands
E-health [56]	Periodic
Information and navigation system [56]	Random
Surveillance and public safety [56]	On-demand
Home energy management systems (HEMS) [102]	Periodic
Smart grid [49]	Periodic
Elastic application (DTN) [76]	On-demand
Hard real-time applications (traditional telephony) [76]	Random
Delay-adaptive applications (e-health) [76]	On-demand
Rate-adaptive applications [76]	Random
Railway systems	Periodic
Regular monitoring [93]	Periodic
Emergency services [93]	Random

Table 1.1: Typical M2M application [130]

network based on connected devices, which can be costly in terms of power. Some devices only need to transmit very limited data and that on very low duty cycles. For those devices, the power budget is low compared to the devices that need to forward data or simply constantly monitor their surroundings [36]. In general, a higher data rate and longer transmission range increase the power consumption while there is always a trade-off between power efficiency and data rate. An overview of commonly used technologies to establish M2M communication is shown in Figure 1.1. The technologies are ranging from very short-range communication and low power consumption with NFC, to high power technologies with very long-range communication capabilities such as Cognitive Radios based on IEEE802.22 with a range of up to 100 km. While the most common ones for local M2M networks are *Z-Wave, WiFi, Bluetooth, and ZigBee.* Many alliances were formed to promote a set of technologies such as the m2mGlobal Alliance, Thread, Internet of Things Solution Alliance, AllSeen Alliance, and the Internet Protocol for Smart Objects Alliance [36]. One example of a technology often used in the area of home automation is ZigBee. The ZigBee technology is promoted by the ZigBee Alliance and find its application in many areas. An example of a very large-scale ZigBee network, combining M2M and the IoT directly, is the Aria MGM City Center Hotel in Las Vegas, Nevada, USA. Over 136,000 ZigBee



Technologies	Standards & Organizations	Network Type	Frequency (US)	Max Range	Max Data Rate	Max Power	Encryption
WiFi	IEEE 802.11 (a,b,g,n,ac,ad, and etc)	WLAN	2.4,3.6,5,60 GHz	100 m	"6-780 Mb/s 6.75 Gb/s @ 60 GHz"	1 W	WEP, WPA, WPA2
Z-Wave	Z-Wave	Mesh	908.42 MHz	30 m	100 kb/s	1 mW	Triple DES
Bluetooth	Bluetooth (formerly IEEE 802.15.1)	WPAN	2400-2483.5 MHz	100 m	1-3 Mb/s	1 W	56/128-bit
Bluetooth Smart (BLE)	IoT Interconnect	WPAN	2400-2483.5 MHz	100 m	1 Mb/s	10-500 mW	128-bit AES
Zigbee	IEEE 802.15.4	Mesh	2400-2483.5 MHz	10 m	250 kb/s	1 mW	128-bit
THREAD	IEEE 802.15.4 + 6LoWPAN	Mesh	2400-2483.5 MHz	11 m	251 kb/s	2 mW	128-bit AES
RFID	Many	P2P	13.56 MHz, etc.	1 m	423 kb/s	~1 mW	possible
NFC	ISO/IEC 13157 & etc	P2P	13.56 MHz	0.1 m	424 kb/s	1-2 mW	possible
GPRS (2G)	3GPP	GERAN	GSM 850/1900 MHz	25 km / 10 km	171 kb/s	2 W / 1 W	GEA2/GEA3/GEA4
EDGE (2G)	3GPP	GERAN	GSM 850/1900 MHz	26 km / 10 km	384 kb/s	3 W / 1 W	A5/4, A5/3
UMTS (3G) HSDPA/HSUPA	3GPP	UTRAN	850/1700/1900 MHz	27 km / 10 km	0.73-56 Mb/s	4 W / 1 W	USIM
LTE (4G)	3GPP	GERAN/UTRAN	700-2600 MHz	28 km / 10 km	0.1-1 Gb/s	5 W / 1 W	SNOW 3G Stream Cipher
ANT+	ANT+ Alliance	WSN	2.4 GHz	100 m	1 Mb/s	1 mW	AES-128
Cognitive Radio	IEEE 802.22 WG	WRAN	54-862 MHz	100 km	24 Mb/s	1 W	AES-GCM
Weightless-N/W	Weightless SIG	LPWAN	700/900 MHz	5 km	0.001-10 Mb/s	40 mW / 4 W	128-bit

Figure 1.1: Overview M2M technologies [36]

devices have been deployed throughout the hotel, distributed in over 4000 rooms. Each room has different sensors and actuators installed, such as temperature sensors, light switches, dimmers, etc. These sensors and actuators communicate with a control unit in each room and form a local ZigBee network. The control unit of each room is connected via fiber to a centralized server that controls the whole building. It is not only that the whole building is fully automated, but it is also designed to provide the highest level of comfort to the client. For example, when the client enters the room, a electronic bellhop greets the client by name, switches on the lights, plays some music on the stereo, and draws the curtains to reveal a ”spectacular view” [144]. This use case shows also how elaborate an installation can grow. However, such an infrastructure resembles a channel reuse rather than a dense wireless sensor network, in the strictest sense, because the 136,000 nodes are separated into 4000 networks with each network containing around 30 nodes. Where rooms far apart from each other can reuse channels again. In any case, future M2M systems will need to support thousands of devices.

1.1 Motivation

In the domain of wireless indoor networks, an application often studied is the collection of real-time or non-real-time data on room occupancy, the activities of people, and environmental conditions such as temperature, CO_2 levels, light, etc. With the deployment of a Wireless Sensor Network (WSN), problems arise, such as the self-organization of wireless networks, the energy efficient operation of sensor nodes, the scalability of networks, the real-time ability for time-critical applications, etc. The high costs associated with network maintenance operations (firmware update, battery replacement, etc.) is also one of the major reasons why very few real deployments can be found. Comparing the wireless solutions with the wired ones, the deployment phase is usually faster and less expensive with WSNs which allow the installation of more measurement points and thus enable the collection of much more information. However, this aspect critically depends on whether the network is designed in a way that allows an installation without elaborate technical configuration effects and which can be done routinely by relatively unskilled workers. In recent years, systems which can provide data such as the room occupancy, the activities of people, and environmental conditions such as temperature, CO_2 levels, and light seem to be very valuable since there is a variety of new services and applications that analyse such information in order to provide different solutions such as the scheduling of building maintenance and janitorial work, the improved use of meeting facilities, asset tracking, and savings of energy in lighting and heating. In realising these systems, large-scale M2M communication networks are necessary for providing the data to an end user or an automated management system in meeting facilities. For example, a user had booked a meeting room via Microsoft Outlook. When the time of the meeting came, the meeting had been cancelled, but the user forgot to free up the room in the system. The sensors indicating whether the room is occupied can now report back to the management system that no one has entered the room. Based on the rules defined, the management system can then free up the room for other bookings. Moreover, this infrastructure can easily be extended to not only collecting data but also to sending control commands to actuation devices such as an office with hundreds of lamps that are dimmed by a central system based on the data collected from luminosity sensors.

With the introduction of wireless-controlled LED light sources, wireless networks have become much denser. The current approaches for control message transmission are often based on broadcasting messages between many luminaires. Controlling wireless lighting systems using low-power wireless network control is attractive due to the reduced wiring requirements in such scenarios. However, existing building management systems are typically not designed for lighting applications where low latency data transmissions are key. This creates a need for not only low-latency dissemination protocols but also for protocols that self-organise the dense wireless lighting networks within an acceptable time frame and which reliably deliver the control commands to multiple lamps. For example, a user would expect an essentially immediate response, such as what one can expect in a wired network, and this translates into a delay of a few (200 - 300) hundred millisecond. In an installation with many light sources, a high variation in latency is perceived as a lack of synchronicity among the responses of the light sources as the latency in control message transmission leads to latency in switching lamps on/off, which is unacceptable for applications such as office environments but can, to some extent be tolerated in the illumination of parking areas. More importantly, it is necessary that the end-to-end packet loss rate of such control messages is very low. If an application contains a few hundred lamps, each controlled wirelessly, the occasional failure of single lamps responding needs to be avoided. End-to-end outage probabilities of more than one percent are undesirable, as this is then interpreted as *a few lamps are broken*. Hence, it is necessary to keep the outage probability at a minimum, with a guarantee that lamps will be in the same state again at the end of a control process.

In this dissertation, a suite of novel M2M protocols is discussed enabling to self-organise the network topology and providing time bounded message dissemination which enables wireless lighting control. The performance of state-of-the-art protocols is analysed using extensive computer simulations and evaluated based on key performance parameter for lighting control applications. Consequently, the D^3LC Suite is proposed to address problems such as an unpredictable system behaviour in terms of QoE. D^3LC is designed to operate on top of the IEEE 802.15.4 Time-Slotted Channel Hopping (TSCH) mode utilising the channel hopping mechanism. The protocol suite incorporates protocols for self organization of dense static network topologies, channel allocation to enable non-interfering simultaneous links, and to disseminate control messages with deterministic arrival times. To self-organise the network topology a weighted clustering algorithm is selected and adapted to the unique use case of lighting control. The channel allocation algorithm ensures that neighbouring clusters do not interfere with each other by applying a self-learning coloring algorithm. The message dissemination protocol ensures a high level of user experience by utilising the time-slotted data transmission mechanism of TSCH for synchronous reception of control messages at the endpoints. The proposed protocol suite is evaluated with several experiments in real-world environments, demonstrating the validity of D^3LC. By comparing D^3LC to very fast and reliable dissemination protocols the necessity of something like D^3LC is further proven.

1.2 Thesis contribution

In order to develop M2M protocols that enable the self-organisation of the network topology and provide time-based message dissemination, a number of requirements need to be defined in order to accommodate the QoE. The requirements that need to be fulfilled are listed below. The definition of those requirements was carried out as part of the authors involvement in the DEWI project [92].

- **Network Latency** For lighting control networks, it is essential to "almost instantly" disseminate control messages to a large number of devices,especially when a user expects a system response. A user expects that a lighting network immediately reacts to the input generated by a user pressing a light switch. Hence, the network latency must be low enough to give the user the impression of an instant reaction. For this, it is necessary to disseminate a new control message with less than 300 ms network latency [32].

- **Network Size and Network Density** Scalability plays an increasingly important role in embedded wireless networks, as in many domains, wireless solutions tend to have grown more complex in recent years. Hence, a network for lighting control must be able to support a large number of nodes where, in practice, networks of the size of 500 nodes are not unlikely. However, setting up experimental networks consisting of that many nodes involves immense costs. Therefore, newly designed protocols should be verified in simulation and tested in smaller experimental setups [47], as simulations allow to evaluate the scalability of a protocol while real experiments allow to demonstrate and verify the functionality and whether the approach fulfils the requirements. The size of a network is defined as the number of nodes connected within the same network, e.g. 500 nodes deployed in an area of size *X*. The network density is defined as deploying a number of nodes in a fixed area, e.g. deploying 500 nodes in an area of 100 m × 100 m. Increasing the network density means to increase the number of nodes deployed in the same fixed area of 100 m × 100 m.

- **Throughput** With some network protocols, it is possible to disseminate a single broadcast message with sufficiently low latency to a large number of receivers. However, in many protocols, nodes keep repeating the same message, sometimes even over several seconds. This leads to message loss due to collisions between old messages and newer messages that are supposed to override them. Some protocols are not able to transmit these messages at the same rate as they are created. Due to these effects, simple interactive lighting scenarios are not possible when a user defines the lighting level based on what he sees. For example, an interactive lighting scenario could be a dimming process where the user turns a dimming knob until the light level reaches the user's desired setting. During this process a number of messages are created and sent out to all or a subset of lamps. Now, if only a few lamps receive these messages and others do not receive them, the overall light level in a room will be varying. It is a natural habit of a user to continue turning a control knob, thus triggering even more messages, which, in turn, saturates the network. Hence, a dissemination protocol for wireless lighting control networks needs to be able to provide a certain throughput to deliver messages throughout a network at a certain rate.

- **Packet Loss Rate (PLR)** Typically, transmitting a message only once, the failure probability can be large. To reduce the failure probability, different mechanisms can be applied, such as Error Correction Codes (ECC) or Automatic Repeat reQuest (ARQ) schemes. While the former increase the message transmission time, the latter affect the real-time performance. However, the end-to-end probability of failing to deliver a control message to all lamps needs to be low. For example, if an office floor contains a few hundred lamps, each controlled via a wireless link, any failure to respond by single lamps needs to be avoided. These link failures of individual lamps are seen as "a few lamps are broken". Hence, it might be acceptable to miss single messages during a burst of messages, but it needs to be ensured that all lamps end up in the same state.

In this thesis, only the network latency, network density, and PLR are addressed, since these parameters are the most important ones which need to be considered while designing M2M protocols for dense wireless lighting networks. Latency is especially crucial since the latency relates to the responsiveness of the lighting network where the user expects to see something happening in a fraction of a second. Addressing the PLR metric is of similar importance, as not receiving a control message affects the switching/dimming of the light and might suggest to the user that a light is broken. These three parameters are addressed in the various proposed protocols within the D^3LC-Suite.

Other important criteria often used for the characterization and analysis of wireless networks are network load, reliability, and throughput. However, throughout this thesis, only the criteria PLR and delay are considered for the following reasons: The network load is typically important as it can be linked to the energy consumption of the nodes. If the network load is high all the time, nodes use more of their available energy resources due to processing incoming messages. Since dense wireless lighting networks are usually mains powered energy efficiency was not a main concern and therefore reducing the network load was not considered as a main evaluation criteria. However, as the network load analysis in Section 6.5 shows, the load is also reduced as a byproduct of the cluster-tree structure established by CIDER and the channel access scheduling through RLL. Reliability can be defined in different ways, e.g. defining the Mean Time Between Failures (MTBF) or as the probability that the device will work for a specific period of time. In the context of a lighting network reliability is defined in terms of a system reacting as expected by the user. This means:

a. description All lamps are connected to the network and will receive messages. This is investigated in Section 6.1 where it is proven that the clustering algorithm will converge and no deadlocks can occur.

b. description All lamps will receive messages and switch to the appropriate colour and light level. This aspect will be considered by the PLR.

The reliability will therefore be inherently considered by means of convergence and PLR. Throughput was not considered as main evaluation criteria since data rates in lighting systems are typically low and are not directly related to the QoE. Focussing on the user experience, PLR and latency are therefore the most important points and will therefore be used as performance criteria in this thesis.

Details on each contribution are provided below.

- **CIDER** With state-of-the-art protocols it was found that some protocols suffer from sending redundant messages where many nodes forward messages even though all nodes have already received the message. The CIDER protocol focuses on the self-organisation of the network topology in order to address this problem. CIDER aims at reducing the number of forwarders to a minimum while still covering the whole network by using a weight based clustering approach. CIDER self-organises the network topology into a cluster-tree topology. For lighting networks it is important that a Cluster Head (CH) covers as many nodes as possible and also that additional CHs are only elected if necessary to reduce the number of hops needed to cover the network. Another aspect which was taken into account while designing CIDER is the setup time. A network organisation protocol for dense wireless lighting networks has to be efficient in terms of setup time. Even though the installation phase of a lighting network with several hundred lamps can easily take several days, it is important that the setup time is in the order of minutes when the network is turned on. This applies not only for the initial setup, but even more so in case of a network reorganisation due to node failures, or in case of a change in propagation conditions that requires a change in topology to ensure reliable operation. CIDER is implemented within the Contiki-OS and is analysed as part of the D^3LC-Suite in extensive experiments where the results are presented in Chapter 6.

- **Channel Allocation** With the current channel access protocols, such as the IEEE 802.15.4 CSMA/CA protocol, transmissions are limited to a single channel only. One aspect of the D^3LC-Suite is to provide a mechanism to enable simultaneous transmissions. For this, D^3LC exploits the channel hopping feature of the IEEE 802.15.4 TSCH mode where multiple links can be assigned within the same timeslot. TSCH has the capability to change the channel immediately, however, for this it is necessary to assign a channel offset with which nodes can calculate which channel they should use. To assign a unique channel offset for each link, a channel allocation protocol is necessary. For this a colouring algorithm is applied which utilises the information already collected with CIDER. The colouring algorithm then assigns a unique channel offset for each cluster which enables simultaneous links using different channels.

- **RLL** State-of-the-art protocols used in lighting control lack the capability to provide synchronous transmission. With standard protocols such as Flooding it is not possible to ensure when a message is delivered. With RLL these problems are addressed. RLL provides a forwarding strategy which enables all lamps in a dense wireless lighting network to receive a control message at the same time. This feature is crucial in respect to the user experience, since this is the expected behaviour of a lighting installation and known from conventional lighting installations. RLL ensures that new control messages are first exchanged within all CHs and then all CHs forward a control message to all nodes in a lighting network. To enable this feature, a specific slotframe is designed which TSCH can follow and each node knows when to transmit at any point in time during a control process. RLL is evaluated in both, simulations and real-world experiments, and compared to an existing state-of-the-art protocol called LWB known for being the fastest flooding protocol currently available.

Together these three protocols, combined in the D^3LC-Suite, provide the high level of QoE and Quality of Service (QoS) needed to enable dense wireless lighting control.

1.3 Thesis outline

The remainder of this thesis is structured as follows:

Chapter 2 discusses the current state-of-the-art of wireless lighting control networks. This provides an overview of communication protocols and networking approaches and how they are used in different areas, including commercial systems typically used for lighting control. Subsequently, a motivation is derived for the approaches designed and evaluated in this thesis.

Chapter 3 provides an overview of the tools that were used in the simulations and experiments. Furthermore, it describes the simulation models used to evaluate the proposed protocols as well as the locations where the experimental evaluation was performed.

Chapter 4 analyses the performance of a dense wireless lighting control network with state-of-the-art communication protocols affected by a broadcast storm caused by a single control device. This analysis contains models for the CSMA/CA mechanism and MAC operations specified by the IEEE 802.15.4 non-beacon-enabled mode. The chapter compares lighting networks of various densities, transmission power, and various MAC protocol parameters. The network performance is analysed with respect to QoS parameters such as packet loss rate and latency, and it is shown how this would affect a lighting network in terms of user satisfaction.

Chapter 5 presents the novel D^3LC-Suite which incorporates a clustering protocol, referred to as Clustering In Dense EnviRonments (CIDER), a data dissemination protocol named Reliable Low-Latency Data Dissemination Protocol (RLL), and a channel-offset allocation protocol that applies a greedy colouring algorithm. The D^3LC-Suite increases the scalability of wireless lighting networks by applying a weighted clustering method designed for static and mains-powered Wireless Sensor and Actuator Networks (WSAN)s, as well as disseminating lighting control messages, providing for a deterministic arrival at the destinations. Finally, it allocates channels to enable simultaneous transmissions without any interference from each other. This chapter also discusses the state-of-the-art of clustering protocols, TSCH scheduling approaches, and data dissemination in WSNs.

Chapter 6 presents the evaluation and analysis of the proposed protocols. The performance of the D^3LC-Suite is evaluated based on the metrics and requirements defined. Also, the proposed protocols are compared with the state-of-the-art protocols to demonstrate their effectiveness and relevance.

Chapter 7 summarises all of the contributions. It discusses the findings and conclusions which can be drawn from the evaluation and presents the future research directions opened up by this work.

Chapter 2

State-of-the-art in wireless lighting control networks

This chapter discusses and analyses the specific challenges and requirements for lighting control networks from a data dissemination perspective and provides an overview of the approaches found in the literature and thus, deriving a motivation for the D^3LC-Suite proposed in this thesis.

In the past, the control of luminaires was based on the classical way where all light bulbs were connected to different power circuits and controlled by one or more switches [9]. For example, for controlling multiple zones, where a zone is defined as a local group of lamps controlled in the same way, each zone has the same control wire and one wire that runs from the zone to a central point which is connected to one or more control units. Here, a control unit could be a light switch or a dimmer. In this kind of installation, it is hard to change the control of lamps or simply rearrange the zones. For every change, a re-wiring is necessary. Therefore, conventional lighting systems lack the flexibility to easily change in their configuration. An early approach to solve this problem is presented by [10,128], using a Local Operation Network (LON). A LON uses LONTALK as a communication protocol, which is based on BUS communication. All lamps are physically connected via twisted pair cables, which form subnets. A LON allows forming up to 254 subnets with 127 nodes each. Basically, each lamp is assigned to its own address and then mapped onto subnets. This approach provides the flexibility to simply re-arrange the current configuration or add new devices. On the other hand, this approach adds additional cables to an installation as besides using a power cable, an additional twisted pair cable is needed to connect the lamps with each other and to the control hub. To avoid connecting a second cable to lamps, Power Line Communication (PLC) was investigated as a possible solution, as PLC also uses the power cable as the communication medium [20]. PLC in an early stage was considered unusable because the power line itself is subject to a lot of noise and interference, which resulted in high signal losses. With advances made in signal modulation, these effects on the signal were minimised [117]. This made PLC a possible solution for controlling lights [90]. PLC uses Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) or Time Division Multiple Access (TDMA) Medium Access Control (MAC) protocols, similar to the wireless solution where the same limitations in terms of channel access or possible collisions occur. As PLC is a wired solution, and this work is focused on wireless technologies for lighting control, PLC is not considered a comparable solution for lighting control and is only listed here for completeness. With the emergence of wireless technologies, the wireless medium has attracted an interest in lighting control. Wireless lighting control was the research focus in many areas over the past years, such as *Commercial Buildings* [21, 105, 107, 137, 141], *Home Environments* [15, 35, 62, 88, 112, 121], or in *Public Spaces* [22, 50, 64, 108, 131] focusing on similar research goals:

- increasing the energy efficiency of the lighting system: [15, 22, 50, 105, 107, 131, 137, 141]
- reducing the installation costs / costs of retrofitting of a lighting control system: [64]
- fulfilling user requirements in terms of optimal illumination of workspaces and other environments: [21, 35, 107, 132, 137, 141]

Based on these different fields, this literature review discusses the references listed above, based on the three application domains *Work* (Section 2.1), *Home* (Section 2.2) and *Cities* (Section 2.3). However, the review focuses on how the wireless medium is used in these areas, especially how wireless M2M communication is realized for lighting control.

2.1 Wireless lighting in commercial buildings

In 2005, almost half of the electricity produced for lighting applications was consumed by commercial buildings [72]. Therefore, increasing the energy efficiency of commercial lighting applications by using WSAN has been the main focus of research over recent years. In [141], the authors developed an intelligent lighting control system that both saves energy and provides for an individual control by users in open-plan offices. The focus of this research is an optimisation algorithm which assumes that each luminaire is equipped with a wireless transceiver and can be dimmed or turned on or off individually. In evaluating the proposed algorithm, the authors developed a prototype of a wireless lighting system which includes actuation modules as well as a base server for decision making. The actuation modules were designed to self-configure the network and minimise the retrofitting cost. The module is simply installed between the power line and the ballast. The platform is programmed to listen to incoming messages and to change its actuation state accordingly. Also, the module periodically reports its status back to the base server. All modules form a multi-hop tree network which is periodically updated to avoid bad links and packet loss. As an operating system, *TinyOS* [86] is used. The experimental setup represents an office room with dimensions of 9.1 m x 5.8 m x 4.3 m (L x W x H). The room contains 12 lamps, for which two scenarios were considered, which can basically be interpreted as two different traffic models. For both scenarios, it was assumed that the occupants only control the lighting level of their workspace. A room with only a few occupants results in low network traffic, and a scenario with more occupants in the room results in slightly higher traffic since the occupants might change the light more often, depending on their co-workers' own preferences. This early work established a first insight on how wireless sensor networks can be used to control the lighting in an office environment. Also, it shows that the control of individual lamps in a room with multiple occupants is not a challenging matter. Unfortunately, the authors had only analysed their lighting control algorithm and had assumed that the wireless network will work without any problems and, therefore, did not give any indication as to the networking performance itself. In a second publication by the same authors [137], the system was improved by adding wireless photo-sensors to the network. Together with the already existing wireless actuators, these form a larger, multi-hop network. The control of individual lamps is still managed by a central computer, which collects the data from the photo-sensors deployed. Based on the control algorithm developed by the authors, the lamps are dimmed individually or turned off to gain the most energy savings. Still, the authors provided no analysis in terms of the wireless network performance but did offer a comment that issues with the wireless network were not the focus of the research. This statement suggests that even on this small scale, there were problems related to the wireless network implementation.

In [105] and [21], the authors developed a sensor-driven wireless lighting control system, where each lamp has a co-located presence, light sensor, and a wireless transceiver. The aim of this research is an increase in energy savings by applying an intelligent control algorithm. The system works in a way that an open-plan office is equipped with luminaires which are organised in a so-called aggregation window. The light level itself is dimmed automatically, depending on the user's location in the room. For determining the user's location, the authors use an indoor positioning system. In this system, the lamps function as an anchor point with their locations known to the system. An anchor point transmits a message every T seconds where a receiving user device can then determine its position within the room based on the Received Signal Strength Indicator (RSSI) and a localisation algorithm. Information such as the MAC IDs of the anchor nodes, an aggregation window, the mapping of radios to the aggregation window, and the position of the anchor nodes are known to every user device. Besides the wireless traffic generated by the lamps and the user devices, the aforementioned sensors generate measurements for every T as well. These measurements are transmitted to a central device. The central device then applies a control algorithm to dim the lamps accordingly. For their evaluation, they use an office room with 24 luminaires, presence and light sensors, and a wireless transceiver. The authors state that T should be small enough to guarantee an adequately fast response to occupancy and daylight changes. In the earlier publications, the authors of both publications gave no indication of the performance of the wireless network and only focussed on the lighting performance itself. However, based on the information gathered from this publication, a first impression can be derived on the overall number of messages generated in a smart lighting use case where light levels are set individually by one central device using multiple sensors. It is assumed that the majority traffic is of unicast traffic, e.g. of sensor data to the central device and of dim settings to the lamps. Also, broadcast messages are necessary for determining the location of a user in a room, which adds an additional contention to the network.

In [107], for the first time, the authors have provided an indication of the network performance and how this influences the lighting performance. The authors use a similar setup as described in [105], where the sensors and lamps form a mesh network. Again, each lamp has a co-located light and occupancy sensor where the light sensor detects the average illuminance from the daylight and artificial light, and an occupancy sensor simply detects when a person is located in its area of operation. The measurements of the light and the occupancy sensors are then periodically transmitted to a central controller and then utilised in a control algorithm. The controller calculates the data and then transmits a dim level to the corresponding lamp. For wireless communication, the authors use the ZigBee protocol stack using the mesh topology to avoid link failures. As is typical for ZigBee networks, it consists of so-called end devices, such as sensors and relay devices. If sensors want to communicate with other sensors, they need to use relay devices. The authors use 80 devices in total for their evaluation whereof 13 devices act as bridge nodes for relaying messages from the controller to lamps and vice versa. The frame size of a ZigBee message was set to 400 bits. To capture the unreliability of the wireless medium, the authors consider a Loss Probability (LP) parameter to represent the packet loss due to the wireless medium rather than a packet loss due to collisions, e.g. interference. Also, the authors define the ZigBee retry limit as RL, which represents the maximum number of times that a node will try to re-transmit a message before giving up. This parameter was set to different values during the simulation with Matlab. For the simulation, the authors used the *TrueTime* toolbox. TrueTime enables a co-simulation of controllers in real-time kernels with network transmissions. The transmit power of each device is set to 20 dBm, and the receiver signal threshold is set to -20 dBm which results in a maximum of three hops needed to reach any device. However, as the 20 dBm transmit power in the ISM band, it is the maximum transmit power legally permitted for 2.4 GHz WiFi networks [94]; in the narrower channels of IEEE802.15.4, 20 dBm exceeds the legal limit due to the limitations of the spectral density. Therefore, the presented results have to be seen quite critically. To determine a route to any device, the shortest path routing based on Euclidean distance is used. The sensor sampling time is set to $T_s = 2\ s$, which can be interpreted as a new message every two seconds. With 80 devices in the network, this results in a maximum of 40 unicast messages per second. Each node in the network is implemented to either send or receive a message at any given time. During their analysis, the authors investigated the effects of packet losses on the settling time of the controller. First, they compare the proposed wireless controller to a wired reference controller, both using $RL = 0$ and $RL = 3$, while $LP = 0\%$ With no retransmissions, the proposed controller already performs quite close to the performance of the wired solution. Using three retransmissions, the proposed controller shows similar behaviour to that of the wired solution. The authors state that a certain redundancy is needed to obtain a performance close enough to that of a wired solution where no packet losses are assumed. However, the number of retries for each message should not be set to too high as this might result in a higher contention and additional packet losses. In a second comparison, the authors compare a wireless reference controller to a wired solution as well as the proposed wireless controller to the wired solution. The parameters for this setup are $RL = 3$ and $LP = 1\%$. Using these parameters, the wireless reference controller already shows an increased settling time, while the proposed wireless controller still shows the same behaviour as the wired solution. In later simulations, the LP parameter is increased, which had an impact on the settling time for both the wireless reference and the proposed wireless controllers. This paper gives a first idea on how the network performance can affect lighting control, as it is stated that a higher packet loss results in a higher settling time of the control algorithm and, therefore, affects the system reaction time. However, the network was evaluated in simulation using Matlab with only fixed packet loss rates. It remains open how such a system would perform in a real-life scenario where many other parameters might affect the network performance, such as WiFi interference.

Summary This section gives an overview of how wireless lighting control is currently used in a commercial building. In commercial buildings, the main focus of using wireless lighting control is on saving energy while keeping the light level in a building/room at a level satisfying to the user. This section also shows that the research done in this area focuses on developing control algorithms and strategies with only a few indications of how the wireless network performs. Most researchers assume that the wireless system used performs as expected, and only one publication indicates that the performance of the wireless network directly influences the lighting itself. Also, the current research shows that traffic patterns in lighting networks are currently of the nature of unicast traffic rather than of broadcast traffic since the lamps are dimmed individually by a central device over which the user often has no control. This shows that a commercial building where a user has direct access to the wireless lighting system is still an unexplored area.

2.2 Wireless lighting in home environments

The lighting control system in the area of a home environment is usually part of a home automation system which is often referred to as a "Smart Home". A smart home system consists of several subsystems using both battery-powered and mains-connected devices. These subsystems are responsible for a certain area, such as security, Heating, Ventilation, and Air Conditioning (HVAC), lighting control, and more, that are incorporated into the smart home system. A smart home system brings in a number of advantages compared to a conventional wired solution. With lighting control, one can easily control the lights from any switch in the house, but they can also turn on if certain conditions in the house are fulfilled. For example, sensors and timers can be added to the lighting system, and based on pre-defined rules, the lights can turn on or off. This also brings in the possibility to add security features allowing one to turn on the lights from anywhere in the world and simulate a presence at home while abroad. While in the early years, home automation wired solutions such as the X10 [55] or EIB/KNX [112] were commonly used, the wireless solutions based on these wired standards are proposed. In [112], the authors propose a tunnelling mechanism for EIB/KNX over an IEEE802.15.4 network by simply adding the EIB/KNX content as payload to an IEEE802.15.4 packet transmission. While with EIB/KNX, the data exchange is organized into groups, such a group exchange, or better known as multicast messages, are not available with IEEE802.15.4; the authors then implemented a broadcast mechanism, called *Simple Flooding*, where each receiving node retransmits the message exactly once. Each of these messages includes the EIB/KNX (group) address, and if a receiving node does not belong to this group, the node will drop the message. The aim of the publication is to offer a wireless solution to EIB/KNX, and since this is a widely used building automation standard, the proposed method would easily extend to the wired solution.

Bhardwaj et al. present a power management system for smart lighting, which enables the collaboration of different Consumer Electronic (CE) devices [15]. The aim of the work is to show the capability of different systems that inter-operate using various networking solutions. The lighting system is restricted to an energy plan and maintains a target power consumption level by automatically adjusting the power consumed by the luminaires in the building. For the evaluation, the authors assume two types of rooms, High-Priority Rooms (HiPR) and Low-Priority Rooms (LoPR), of which the first one is allowed to use energy based on the demand, while the latter one is obliged to use the power that is left over from the HiPR. This assumes that the maximum power consumed cannot exceed the total available power itself. A service located in a LoPR is stated as the lighting of a parking lot, any non-functional (e.g., decorative, artistic) lighting, and of entertainment services (e.g., television, music). These equip one room as a LoPR using the Open Service Architecture for Sensors (OSAS) [17]. OSAS has been developed as a framework for programming networks of very low capacity nodes. It uses an underlying publish-subscribe communication style and an event-based programming. In HiPRs, a different system is used, called Smart-M3 [127]. The information in the Smart-M3 architecture is represented using ontology models.

In Smart M3, ontology is used for representing semantic information and reasoning about it. As for communication, the authors state that IEEE 802.15.4, ZigBee, or IP can be used. In their evaluation, ZigBee and IP are used. ZigBee is used for the communication between the sensors while IP is used for communication between the rooms. For the ZigBee system, the authors use a traffic rate of one message per second to update the system with new parameters. However, unfortunately, the authors do not provide any analysis of how the network performs, but they rather evaluate the system based on power consumption.

The authors of [88] present a smart home control system, combining two technologies, WSN and PLC. The work aims at reducing the impact of wireless interference on smart home control networks as well as reducing the energy consumption of a smart home. Their system considers three rooms, consisting of different home appliance devices, each equipped with a PLC transceiver. Additionally, each room includes a WSN that has sensors deployed all over the room and is connected to a WSN coordinator. The coordinator is equipped with a PLC transceiver. Hence, the coordinator receives the data from the sensor nodes and then forwards these via the PLC. Besides the integration of these two different systems, part of the work was designing a smart lighting-control algorithm and then evaluating it. The communication between the WSNs is realized by using the ZigBee protocol stack. In their evaluation, the authors also consider the effects of WiFi interference. They compare a pure WSN where everything is connected only using ZigBee with their proposed system. Their findings are that WiFi has much less impact on the communication within the network when PLC and ZigBee are used than when only ZigBee is used in a network. Next, the authors evaluated the performance of the smart lighting-control algorithm. For this, the authors built a prototype, composed of the ZigBee WSN, a PLC network, and a gateway device. For the ZigBee network, a TI CC2530 ZigBee network processor is used. To control the light and minimise the energy consumption of the lighting system, each sensor transmits the current illumination level to its coordinator every 3 seconds. The coordinator then forwards the value received to a central control unit. The evaluation shows that the proposed systems achieve a packet failure rate of 0.8% and an average end-to-end delay of 75 ms in a real-world environment. Unfortunately, only three sensor nodes were used in the evaluation;

hence, the overall traffic was rather small. Nevertheless, this work provides a first idea of how traffic can be perceived in a home environment.

In [35], the authors propose a network architecture using Named Data Networking (NDN) as a smart lighting solution. With NDN, routing uses name-based packet delivery which enables great flexibility in naming, security, caching, and inherent multicast support. NDN works without dependencies on other protocols or the various middle-ware used in IoT networks. The proposed system architecture consists of several IoT devices, such as a wireless home router connecting all the IoT devices, a Raspberry Pi used as the IoT platform, and several smart light bulbs. The smart light bulbs are realized utilising a wireless actuator that simply turns the conventional light bulbs on or off. In addition, various wireless presence and luminosity sensors are deployed. Unfortunately, the authors do not state the number of devices that they used, or in which intervals traffic is generated. However, the authors do analyse the network performance based on end-to-end delay, showing that 85% of the received messages are received within 142 ms. After deducting the internet transmission overhead, 85% are received within 72 ms.

Suh et al. present an intelligent home control system which is able to assign tasks to several home appliances [121]. The control system consists of a wireless sensor network with an actuation functionality. The proposed system is able to collect physical information and utilize this to control the consumer home devices. They named their network the *Active Sensor Network*. The active sensor network considers two types of devices, generic sensors, and actuators. The generic sensors can either be used to scan the environment periodically for temperature, humidity, and light, or for non-periodic events, such as gas leaks, presence detection, and windows status changes; whereas an actuator is used to directly control the user devices or home appliances, e.g. turning a TV on or off. However, the actuator is simply designed as a wireless switch that is placed near the user device to turn it on or off. To make it more complex, the authors equipped some of the actuators with Infrared (IR) communication interfaces that also control TVs. For this, the authors developed their own smart nodes supporting the IEEE802.15.4 PHY-layer. The MAC layer is similar to the B-MAC protocol [110]. For forwarding messages between different devices within the network, the authors propose the so-called Link Quality Indicator-Based Routing (LQIR) protocol. The LQIR protocol is a tree-based routing protocol, forwarding messages towards the sink by using the nodes with the best LQI values among its neighbours. The LQIR updates the LQI values whenever data are sent or received. The smart node supports several attachable sensors or actuators. Unfortunately, the authors do not mention how many such devices were deployed in total, only that a gas sensor is located next to a gas valve, along with an actuator to close the valve in case of a leakage. Several sensors are deployed to detect human motion. For intruder detection, a window is equipped with a REED switch to detect a sudden state change. The home control system is connected to a wireless actuator that controls the curtains and is connected to the lighting system to turn the lights on or off. No indication is given of how the wireless sensor network performs in terms of packet delivery ratio or end-to-end delays of message transmissions, nor is there any indication how often messages are generated and transmitted.

The authors of [62] propose a ZigBee-based intelligent house lighting-control system. The proposed system includes a fuzzy logic-based control algorithm to lower the energy consumption and to attain certain user comfort levels. The comfort level is represented as the recommended Lux levels for various areas, e.g., in a living room, 200 to 600 Lux are recommended. The fuzzy logic is to adjust the light levels based on the indoor brightness that the system determines as the illumination intensity required for current conditions. This system also takes the outdoor illumination into account, and if the outside light level is high enough, the system will automatically open the blinds to improve the energy consumption when turning off the lights. The system consists of three zones, each containing 2 light bulbs which are connected to a central control unit. The ZigBee network is used to obtain the outdoor brightness, as well as the indoor brightness, and forwards the data collected to the central control unit. Based on the data obtained, the control unit decides to open the blinds and/or to turn the lights on or off. This work only presents that, generally, a wireless sensor network is suitable for lighting control networks. However, the authors do not mention how the state of a single luminaire is affected by the performance of the wireless sensor network, e.g., by packet loss, latency, or network contention; for instance, what happens if, due to a high packet loss, multiple luminaires receive different control commands?

Summary This section provides an overview of how lighting control is used in a home environment. In a home environment, the main focus is similar to that of commercial spaces, which is to focus on saving energy as well as keeping the light level in a room at a user comfort level. The focus is also rather on developing the control algorithms than on improving the wireless communication for such a purpose. All systems presented here assume that the performance of the wireless system is as expected and that they operate without any problem. It only can be assumed how the traffic patterns in a home environment are defined, but it seems to be similar to those in the commercial environment. This shows that a home environment where multiple lamps need to be controlled simultaneously with minimal delays is a rather unexplored area.

2.3 Wireless lighting in public spaces

Illumination in public spaces has many use cases, ranging from the illumination of a public market, lights at the side of a road, to controlling the lights in a tunnel. This section provides an overview of how wireless lighting is used in these areas, especially of the wireless communication part of such systems.

The authors of [22] investigate the possibility of deploying a WSN in a road tunnel to autonomously control the illumination within the tunnel. In their work, the authors propose a hardware/software architecture for such an environment while exploiting the WSN component and analysing its performance. A WSN was deployed to control the lighting and to measure the light levels along the tunnel walls. The light intensity measured is then reported back to a central controller. The controller then adjusts the lights according to the values measured and as per the applicable laws. The environment used for experiments is a two-way, two-lane tunnel, 260 m-long. A total number of 40 wireless nodes were deployed to measure the illumination as well as two gateway nodes to forward the measured data to the controller via an Ethernet connection. A total of 40 lamps existed in the tunnel, of which the authors were allowed to replace 16 with controllable lamps. The remaining 24 lamps are controlled with a timer based on the old system for ensuring safety during the experiments. TinyOS with Low Power Listening (LPL) MAC is used for the wireless communication. Experiments were conducted over a period of 7 months, during which each node reported the light level back to the controller every 30 seconds. Furthermore, once a minute, a node reported back additional information such as battery voltage, temperature, and routing parameters. This reporting schedule resulted in network traffic flow of two unicast messages per second. An average loss rate of between 0.1% and 0.2% was achieved throughout all the experiments. For the proposed system, it was required that the data observed is transmitted in a timely manner due to the closed-loop system. This means that their proposed control algorithm runs once every 30 seconds on data collected during the last 30 seconds. Therefore, if a node is not able to deliver a message in a timely manner, the PLC executes a cycle without a sample of that node. If two samples are more than 60 seconds apart, the PLC will miss samples of multiple intervals. The results show that 96.5% of all samples are delivered within the 30 seconds interval, considering node sleep intervals of 500 ms. The delivery time can be improved if the sleep interval is reduced to 100 ms or 250 ms, hence, a node is able to transmit/forward a packet more often. This work provides the first deep analysis on how a WSN can be used in combination with a wired installation by providing environmental data. However, the authors use a similar approach compared to work presented above. Sensors are used to monitor the environment, and the sampled data is transmitted to a central control unit which controls the light using a wired installation.

In [50], the authors exploit the advantages of a WSN applied to street lighting applications. Their idea includes using the lighting network also for Smart Metering applications or adding distributed sensors such as air quality sensors. A system is proposed where several wireless nodes form a cell in which each node reports back to a sink node. Each cell is connected via an Ethernet connection to a central Supervisory Control And Data Acquisition (SCADA) system. The wireless communication used to create a WSN is based on the IEEE802.15.4 standard for the MAC and PHY layers. With IEEE802.15.4, it is possible to choose different frequency bands, i.e. 2.4 GHz or sub-GHz. In the system proposed, the 2.4 GHz frequency band was chosen, which establishes a communication range of up to 100 m. To keep their system as "general" as possible, the non-beacon-enabled mode of the 802.15.4 standard is adopted for power saving reasons. Additionally, the Receiver-Initiated Transmission (RIT) technique defined by the IEEE802.15.4e standard is adopted. With RIT, the node divides its time into an active and in an inactive period. During the inactive period, a node turns off its radio and puts itself into a sleep mode to save energy. When returning to the active part, it transmits a "DATA REQUEST" to inform other nodes in the network about readiness to receive data. If a node has data to transmit, it wakes up and waits until the specific counterpart sends its "DATA REQUEST" message. For the data routing, the Geocast Greedy Perimeter Stateless Routing (GGPSR) protocol is used. Due to the position of the street lamps being known, the authors state that it is not necessary to equip each node with an additional GPS module, moreover, the location is defined during the network initialization phase. The data forwarding to the sink is separated into two operational modes of GGPSR. First, a node tries to forward the message to the node closest to the destination, this is called the "Greedy Mode". If there is no node closer to the target it will use the second mode, called the "Perimeter Mode". This mode is able to overcome a gap in the network where the next node closest to the previous hop node is selected. This node then tries to forward the message using the GGPSR protocol again. The validation of the system is done by using NS-2. The environment implemented is based on the geographic coordinates of the real public installation in Mairiporã City in São Paulo state. Using these arrangements, the authors achieved a node degree ranging from 18 nodes up to 34 nodes by setting the transmission range to 100 m. One evaluation point is the delivery time of a created message from a wireless node. Depending on the hop count, the delivery time is in the order of seconds for a 4-hop distance and up to 100 seconds for a 19-hop distance. This shows that the number of hops needed to cover a network/link has a direct influence on the delivery time. However, the duty cycling of the RIT mode adds additional delivery time to the overall performance.

Pinto et al. also exploit the advantages of equipping street lighting systems with wireless sensors [108]. Similar to the work of [50], Pinto uses the lighting infrastructure to monitor the environment. Also, the authors added a fault detection to single luminaires (broken lamps), as well as a fault detection to the energy grid, e.g., peak voltages which are indications of grid problems. However, the street lighting control system aims at a better energy efficiency by applying a hard-coded dimming curve. To monitor the environment, each light post is equipped with a so-called smart module which is connected wirelessly to a control module. The control module is then connected to a server module via an Ethernet connection, collecting the data from all control modules. The wireless connection is realized using IEEE802.15.4 in combination with ZigBee to form a mesh network. The evaluation of the proposed system focuses only on the dimming operation of the different lamp posts and the detection of events such as "overvoltage". These events are reported back to the central server module. The authors provide no indication of how often data is reported to the server module nor if any problems had occurred in the wireless communication.

In [64], the authors present a system using IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) to replace an existing PLC installation of a Smart Energy Management System (SEMS). The current SEMS consists of the several PLC light controllers that are deployed in every street lamp. These PLC controllers are connected to a centralised controller using the power line. Multiple street lamps are connected to this centralised controller which is located in a switch box, covering all the lamps in its range. All the central controllers report back to a server, using a General Packet Radio Service (GPRS) connection. The authors chose 6LoWPAN as communication stack due to that its operation is similar to that of the TCP/IP protocol stack. For their system, the Contiki-OS was used since it includes 6LoWPAN as well as uIP. To test the proposed system, the authors replaced the PLC controllers of five street lamps with wireless nodes, and at one side of the street, they placed a border router that acts as a gateway to the outside world. A system was simulated in which a laptop connected to the border router, periodically transmitting packets to the nodes. A receiving node replies with an ACK. The system was evaluated based on the data rate, reliability, and packet latency; the average data rate for all tests was stated as 44 kbps. It is assumed that the sub-GHz operation mode of IEEE802.15.4 was used as the authors had indicated a high transmission range and a low data rate. For all experiments, an average reliability of 95% was achieved, and the average latency was less than 1 second. These results were stated as being acceptable for a street lighting application. The paper provides a first insight into the requirements for a street lighting application where higher packet loss rates and latencies are acceptable.

Viani et al. investigated the performance of a WSN-based solution which converts a standard street lighting system into a smart lighting system [131]. The authors define the requirements for such a system as follows. First, the system must be easily operable in combination with the existing lamps without additional wiring. Second, the system must guarantee a high robustness and reliability since the street lamps are essential to citizens. Third, the proposed system architecture must be able to work as a backbone network to create a smart city interface. The authors designed a wireless node, operating on the 2.4 GHz band with a 3 dBm transmit power. Part of the system is a firmware developed based on the IEEE802.15.4 standard. The authors also investigated the ZigBee communication stack that was available, but due to the electromagnetic environment, ZigBee was not appropriate for the street lighting application. Unfortunately, no additional information about the communication stack was provided. However, each wireless node handles turning a lamp on and off as well as the dimming operation. Additionally, the wireless node measures the power consumption of the street light. To provide security in the case that the wireless system fails, the node can detect a failure when it appears and switch the lamp into an “offline” mode where it follows a predefined illumination scheme. The proposed system was tested in two different locations with a total of 737 nodes connected to 12 different gateway nodes, with a maximum of 125 nodes connected to a single gateway. However, the experimental validation focused only on the system operation, i.e. only the control loop operation was monitored and how much energy could be saved by using an adaptive dimming algorithm. Nothing about the performance of the WSN itself was mentioned, nor was there any information provided about any problems, except that the ZigBee communication stack is not suitable for environments with a high electromagnetic impact.

Summary This section provides an overview of how lighting control is used in public spaces, such as for street lights. Different from the office and home environment, the focus of wireless lighting in public spaces relates to how the maintenance work can be minimized, i.e. how the wireless infrastructure can be used to detect a broken lamp rather than having a maintenance worker check every lamp periodically; the reduction of maintenance costs is hence the main concern. Apart from this, the wireless lighting was often used for optimizing the illumination. This was done, for example, in a road tunnel, to better tune the dim level of different lamps with the result of reducing the energy costs. However, similar to the previous environments, it was always assumed that the wireless infrastructure performs without problems. Therefore, in the public spaces, the wireless communication has not been evaluated by any of the papers reviewed. It was mentioned that with ZigBee, problems occur in special electromagnetic environments. Also, as different from the previous systems, the wireless networks here often contained several hundreds of nodes, which gave an indication of how many lamps a system must support.

2.4 Commercial systems

In [57], the authors reviewed the most common wireless communication solutions used for smart applications, which are *ZigBee*, *6LoWPAN*, *Z-Wave*, and *INSTEON*.

ZigBee . Zigbee is a Low-Rate Wireless Personal Area Network (LR-WPAN) technology promoted by the ZigBee Alliance. The ZigBee stack combines the IEEE802.15.4 Physical (PHY) layer and a MAC layer with a Network (NWK) layer and an Application (APL) layer, defined by the ZigBee Alliance. With ZigBee, it is possible to operate mesh, tree, and star topology networks at 868 MHz, 915 MHz, and 2.4 GHz frequencies. ZigBee offers two MAC layer modes, the so-called beacon-enabled and the non-beacon mode. The three types of device roles with ZigBee are coordinator, router, and end device, where the end device is in most cases a device with constraints such as limited energy and/or computing capabilities. The ZigBee alliance offers pre-defined application profiles for different areas such as *Home Automation*, *Building Automation*, *Light Link*, *Health Care*, and many more, specifically defined for these areas.

ZigBee Light Link . ZigBee Light Link (ZLL) is defined by the ZigBee Alliance to provide a global standard for interoperable and easy-to-use consumer lighting products [75]. ZLL defines two different categories of devices, which are light devices and control devices. The lighting categories includes light on/off, dimmable light, colour light, extended colour light, and colour temperature light while the control devices include light switches, occupancy sensors, remote controls, smartphones, or PCs/laptops. To form a network with ZLL, no coordinator is needed; instead, a method called Touchlink is used. In pure ZigBee, a device can join a network by using beacons which are sent by a coordinator. However, with ZLL, the join process is initiated by a control device which is capable of assigning addresses, for instance, a remote control. To join the network, the light device needs to be physically close to the control device, which is determined by the signal strength received, and each device in the network must be added using this procedure. ZLL defines the control of the on/off state, changing the colour, saturation brightness, and colour temperature. Furthermore, it is possible to define groups using ZLL, which can be controlled simultaneously. ZLL aims at the home consumer market where there are only a few devices within a network.

Z-Wave . Z-Wave is a wireless protocol stack developed by ZenSys, and its MAC and PHY layers are defined by the ITU-T Recommendation G.9959. The PHY operates in the 850 and 950 MHz frequency bands, depending on the country in which Z-Wave is used. With Z-Wave, it is possible to achieve up to 100 kB/s when using CSMA/CA. The MAC is designed to support so-called soft real-time applications that are not time-critical in nature [74]. Z-Wave itself claims to be the world market leader in wireless control [143]. The areas targeted by Z-Wave applications are the residential and light commercial sectors. The two types of devices are defined to be control and slave devices. A control device can either poll data from a slave device or send control commands directly, e.g., polling sensor data or transmitting a command to switch a light on or off. The Z-Wave stack supports source routing up to 4 hops, which is stated to be sufficient for most home applications.

6LoWPAN . 6LoWPAN is a protocol was defined by the Internet Engineering Task Force (IETF) to enable the use of IPv6 on top of the IEEE802.15.4 protocol [98]. Enabling IPv6 for low-power networks presents the challenge of mapping the immense payload of up to 1280 bytes to the maximum of 127 bytes under IEEE802.15.4. Another challenge is to compress the 40-byte header of an IPv6 frame to a 2-byte header suitable for IEEE802.15.4. 6LoWPAN is an adaptation-layer protocol that operates between the IEEE802.15.4 and the Transmission Control Protocol (TCP)/User Datagram Protocol (UDP). For routing, the IETF Working Group (WG) developed the Routing Protocol for Low-Power and Lossy Networks (RPL) which is included with 6LoWPAN.

INSTEON . INSTEON is a communication protocol aimed at home automation [34]. INSTEON enables a mesh network, combining Radio Frequency (RF) communication and power-line communication. A device can support either RF or power-line, or both. The wireless communication operates in the 904 MHz band with a data rate of up to 38.4 kb/s. Communication can be initiated by any device in the network, and any device can be in the role of a sender, receiver, or relayer. Similar to Z-Wave, there is maximum number of four hops with INSTEON. Each message is relayed by a receiving device, unless it is the destination device. Channel access follows a TDMA approach where nodes are only permitted to transmit during certain time slots, and nodes within the same range do not transmit different messages at the same time. The length of a timeslot is defined by the number of power line zero crossings. Only devices that are not attached to the power line are allowed to transmit asynchronously.

Summary This section provides an overview of smart home communication protocols available. All cover similar areas, but apply different technologies, e.g., some use TDMA for channel access instead of CSMA. ZigBee provides a solution that covers a wide range of use cases, also lighting using the ZigBee Light Link. It is easy to use, e.g., when adding new devices to the network. Nevertheless, it has been developed for small-size networks, i.e. it would be quite time-consuming to add hundreds of lamps to a network using the ZLL approach. 6LoWPAN is a communication protocol that connects smart devices directly to the Internet. For this, it translates between IPv6 packets and IEEE802.15.4 packets. Z-Wave works similar to ZigBee, but with a lower bit rate, and aims at smaller control applications. INSTEON combines the two technologies of low-power wireless networks and power-line communication. INSTEON defines a communication stack that permits mains-connected devices to directly interact with devices that use only powerline. In any case, the systems available are usually designed to support only smaller networks. Therefore, it remains open how the available systems perform in dense wireless lighting-control networks.

2.5 Conclusion

This chapter provides an overview of how M2M is used in lighting control networks. The focus of the discussion was how the authors realized the different wireless lighting control networks, i.e. which data dissemination methods were used and which problems they might have to tackle. For this, three areas are investigated, the commercial building environment, the home environment, and the public spaces environment. Several publications show that the wireless technology is suitable for lighting applications on a small scale. However, for all environments, it has been found that the proposed systems focus mostly on applying smart control algorithms to increase the energy efficiency of the lighting system. It was mostly assumed that the wireless technology used works without any problems, and insight into the performance of the wireless network is only rarely provided. Of all the papers reviewed, only one paper provided some insight into how the wireless network performance affects a control loop and indirectly the lighting application. Two important facts could be retrieved from the papers reviewed. First, the traffic in all areas is based on unicast traffic where the lamps are individually controlled. Second, in most cases, the user is not assumed to have direct access to the lighting system. Even in the home environment, it was assumed that rather than using a light switch, the light is switched on or off based on presence detection sensors. The network size in the home and commercial environments was usually in the order of tens of nodes. In public spaces, the test beds were often deployed across a city with hundreds of nodes.

Based on the above-mentioned points, the current state of the art leaves open a number of questions.

1. The effects of certain network parameters, such as latency or packet loss, on a wireless lighting control network are not known. The network latency represents the responsiveness of the dense wireless lighting network where a user expects an immediate system response. The PLR is linked to link failures which need to be avoided. Since in dense wireless lighting networks a failure of individual lamps due to not receiving messages relate to lamps not turning on. An evaluation based on these parameters is provided in Chapter 6 and is summarized in Chapter 7.

2. Does adding luminaires to an existing installation affect the network performance? It is not known if increasing a network density by adding additional lamps to a given area has effects on latency and PLR. The effect of an increasing network density is discussed in Chapter 6

3. Do these effects influence the user perception? A user has a certain expectation on how a lighting system should behave based on conventional lighting. With conventional lighting lamps respond immediately and at the same time. How state-of-the-art dissemination protocols affect the user perception is shown in Chapter 4 and how D^3LC performs in terms of user perception is presented in Chapter 6.

The conclusion of these three questions is drawn in Chapter 7.

Chapter 3

Methods and tools

This chapter provides an overview of the tools that were used to build the simulations and physical experiments. Furthermore, it describes the simulation models used to evaluate the proposed protocols as well as the locations where the experimental evaluation was performed. For evaluating of an early idea, in general, there are three methods available: Mathematical analysis, measurements in an experimental setup, and computer simulations. The results presented in this dissertation are based on computer simulation and physical experiments only. Mathematical analysis was not found applicable since it is often too abstract to obtain an in-depth view of the system, which is important for understanding the behaviour of a wireless lighting-control network. The remainder of this chapter is organised as follows: Section 3.1 provides a detailed overview of the simulation environment, including the simulation engine and the simulation model. The experimental environment is discussed in detail in Section 3.2. All source code created as part of this dissertation is publicly available on `https://github.com/conrad2210`.

3.1 Simulation environment

A simulation is used when gathering results from physical experiments seems impractical, e.g., when setting up an experiment might incur high costs for hardware, or when testing a new approach to an experiment would be too time-consuming. In such cases, using a simulation approach allows a user to investigate the system behaviour under different conditions. Also, simulations are used as a design tool before building the real system. Finding the best design is often faster and easier in a simulation than building and running the experiment using the real system. However, before running a simulation, a careful selection of suitable methods and tools has to be made.

3.1.1 Computer Simulation Engine

There are various simulation tools available on the market. In terms of wireless sensor network simulators, the most common ones are: OMNeT++ [129], ns-2 [96], ns-3 [113], and OPNET [23]. These simulation tools are all well-established within the research community and have been proven reliable. Often, the choice of using a certain simulation tool is based on personal preference. OMNeT++ is the simulation tool of choice for this dissertation; the reasons are listed below. Using OMNeT++ offers a number of advantages compared to the other options.

ns-2 [96] ns-2 is an open-source network simulator using C++. With ns-2, it is possible to simulate all layers of the OSI model. ns-2 supports models for satellite communication as well as models for WiFi communication. Models for WSNs are available and also include IEEE802.15.4 communication models. ns-2 is a discrete event simulator in which each additional message or event has a direct impact on the processing time of the simulation. Due to the large user community, support is available in online forums. The official support and further development of ns-2 stopped in 2011, as it has been replaced by ns-3; however, it is still used by many researchers. The main reason not to use ns-2 was that many state-of-the-art models were missing.

ns-3 [113] ns-2 was the most commonly used network simulator in academic research, but a drawback that each network simulator brings with it is that the models are too abstract, and the implemented code cannot be reused in real implementations. The main focus in developing ns-3 was to improve the realism of the models used. ns-3 is based on C++, as is ns-2. The architecture of ns-3 is similar to that of Linux kernel drivers and application interfaces. This brings with it the advantage that the code can be reused on real hardware. However, when ns-2 was replaced by ns-3, the previously developed ns-2 models, in particular for wireless communication, were no longer supported. With this limitation in mind, using ns-3 would have added significant work in redeveloping old ns-2 models for the ns-3 simulation tool before even being able to develop own models. This has made ns-3 not an attractive option.

OPNET [23] OPNET is a discrete event simulator focusing on communication networks, ranging from simple Local Area Networks (LAN) to complex global satellite networks. OPNET comes with a detailed library of different communication models, which allows researchers to modify these or to develop new ones. In 2012, OPNET became part of Riverbed, a company specialised in network optimisation tools, and it was then renamed to *Riverbed Modeller*. OPNET is a commercial tool which requires licence fees. Nonetheless, the terms and conditions attached to the license led to the exclusion of OPNET as a suitable option.

OMNeT++ [129] OMNeT++ is a discrete-event simulation tool, based on C++ libraries and including an integrated development environment in the form of an *Eclipse IDE* and a graphical runtime environment [129]. OMNeT++ uses simple building blocks to form modules which are then linked to a network. The modules are designed in a nested way where multiple modules can form a parent module, e.g., the network layer can consist of multiple routing protocols in which an algorithm decides which routing protocol is to be used. In this way, it is not only easy to represent each layer of a device inside a single module, but also outside of a sensor-node module, everything is represented by modules such as the channel model. Apart from its easy-to-use environment and modelling mechanism, OMNeT++ offers a large variety of open-source network models for WSNs, such as Castalia [18], INETMANET [12], or MiXiM [82].

3.1.2 Simulation Model

The simulation model used is built in the same way, where each layer of the model can create its own event, e.g., by creating a new message at the application layer. As stated in Section 3.1.1, many different libraries are available for

OMNeT++; the three most common ones are discussed here in more detail. All three provide models for the IEEE802.15.4 standard. The Castalia library focuses on a very accurate model of the physical layer of the IEEE802.15.4 standard and also includes the often-neglected features such as clock drift, sensor bias, sensor energy consumption, and CPU energy consumption [18]. The MiXiM framework provides detailed models of the wireless channel, wireless connectivity, mobility models, models for obstacles, and models for the medium access control [82]. The MiXiM framework has been discontinued, which led to its exclusion as an underlying simulation model for this work. The INETMANET framework is an extension of the standard library INET of OMNeT++, which adds the routing protocols for Mobile Adhoc NETworks (MANETs) as well as the MAC protocols for wireless sensor networks [12]. While INETMANET also provides simple channel models with standard fading and path loss models, as well as many examples of already implemented WSN models, the focus of the INETMANET is on providing a comprehensive base for the further research of network layer protocols. In this research, the INETMANET framework was used to develop the protocols proposed.

Use Case The use case for this research is based on a simple dimming application which was developed in cooperation with Philips Lighting within the Dependable Embedded Wireless Infrastructure (DEWI) project [92], and the details of the use case were suggested by Philips Lighting based on their own requirements. A dimming application was chosen because it is the utmost case compared to a simple on/off application where only a single message is generated. When using a single message, the network is considerably less loaded than with a sequence of messages. Based on this assumption, if the developed protocols perform well under the more complex application, they will also work with the simpler application. Thus, the use case scenario is as follows. A room occupant uses a dimming device in an open space and turns the knob for two seconds. During these two seconds, in a first stage, the dimming device samples the current state every 200 ms and then generates a control message; this is later referred to as the state-of-the-art traffic rate. In a second stage, the sample rate of the dimming device will be increased to 100 ms for achieving a higher user satisfaction level.

Physical environment The open space is represented by an open-plan office with dimensions of (80 x 20 x 3.3) m (L x W x H). In this open plan office, up to 400 nodes are simulated. The nodes themselves are deployed in a uniform grid, see Figure 3.1. The lamps are deployed at a ceiling height of 3.3 m. As the lamps are mounted on the ceiling, it is assumed that the rays will be reflected by the ground. Therefore, a Two-Ray Ground reflection model is used as the channel model. For the log-distance path loss part, a path loss coefficient of $\alpha = 3$ is used since a furnished office environment is assumed [91]. The path loss for a two-ray ground reflection model is calculated using Equation 3.1 [111], with d as the distance between transmitter and receiver, G as the combined antenna gain of transmitter and receiver, and h_t, h_r as the height of the transmitter and receiver, respectively.

$$PL = 40log_{10}\left(d\right) - 10log_{10}\left(G \cdot h_t^2 \cdot h_r^2\right) \tag{3.1}$$

However, the Two-ray ground reflection model is only used when the distance of transmitter and receiver is greater than the cross-over distance, otherwise, a log-distance path loss model is used, see Equation 3.3. The cross-over distance is calculated by Equation 3.2 [111].

$$d_c = \frac{4\pi h_t h_r}{\lambda} \tag{3.2}$$

With the antenna height at 3.3 m ceiling level and at 2.4 GHz frequency, d_c results in $d_c > 1$ km, which means that the log-distance path loss model is always used. If the distance between the transmitter and receiver is smaller than d_c, the log-distance path loss model is used, see Equation 3.3 [111], $\overline{PL}\left(d_0\right)$ is the free space path loss and α is the path loss coefficient.

$$PL = \overline{PL}\left(d_0\right) + \alpha \cdot 10\log_{10}\left(d\right) \tag{3.3}$$

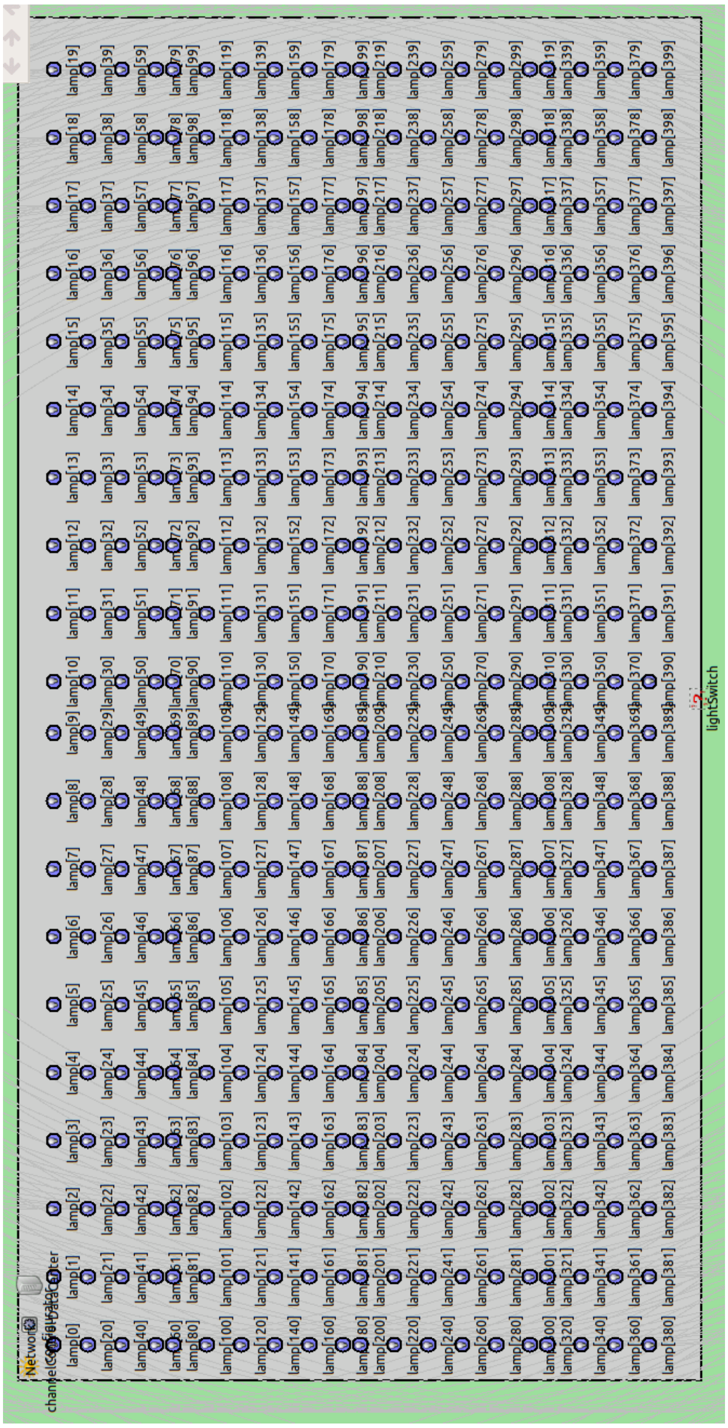


Figure 3.1: Simulation model: Network overview

The free-space path loss at distance d_0 is defined using Equation 3.4 [111]

$$PL\left(d_0\right) = -10\log_{10}\left[\frac{\lambda^2}{\left(4\pi\right)^2 \cdot {d_0}^2}\right] \tag{3.4}$$

Nodes The simulated nodes represent a lamp where each lamp is equipped with a wireless transceiver. The OMNeT++ node model is shown in Figure 3.2. The model is simplified, and only the required layers are implemented; therefore, only models of the application and network layers are implemented, as well as a wireless module which includes a packet queue and the IEEE802.15.4 MAC and PHY layers. The application layer represents the use case described in 3.1.2. In a first approach, the network layer implements a flooding application which is described in more detail in Chapter 4. Later, this simple network layer is replaced by the RLL which is developed as part of this thesis and is described in more detail in Section 5.4. Using the flooding approach at the network layer, the MAC layer is based on the IEEE802.15.4 non-beacon enabled mode and is described in detail in Chapter 4. For the RLL, a different operating mode of IEEE802.15.4 is used, which is called TSCH and is described in Section 5.1. The PHY layer model

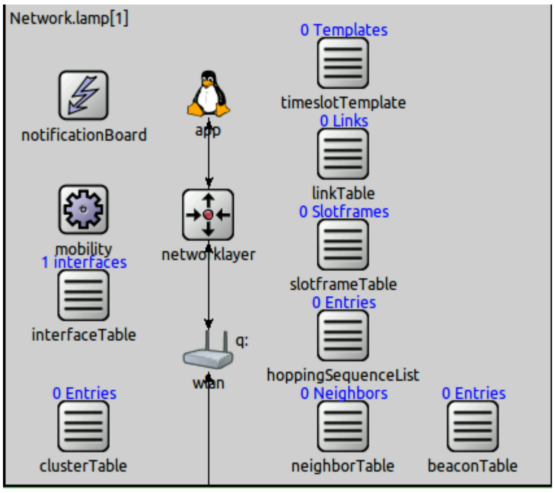


Figure 3.2: Simulation model: Wireless node detailed view

represents the operation of a 2.4 GHz IEEE802.15.4 radio transceiver provided by the INETMANET framework and developed by Chen et. al [28].

3.2 Experimental environment

In this section, the experimental setup is discussed. First, the hardware used to conduct the experiments is described as well as the operating system used for protocol development. Second, the physical environment of where the experiments were conducted is described.

3.2.1 Hardware and Software

Re-Mote The hardware platform used is a commercial off-the-shelf node from Zolertia, called *Re-Mote* (see Figure 3.3) [146], which was released in 2015. The Re-Mote is based on the Texas Instruments CC2538 system-on-chip, which includes an ARM Cortex M3 MCU with up to 32 MHz clock speed and a 2.4 GHz IEEE 802.15.4 radio. It has 32kB of RAM and 512 kB of flash memory, which is sufficient to accommodate the proposed protocol stack. Besides the already mentioned 2.4 GHz radio, the CC2538 includes a second sub-GHz radio for long-range communication. The hardware platform provides support for many open-source WSN software platforms, such as the Contiki OS [31] or RIOT [114]. The Re-Mote is designed for low-power operation, which means that when in sleep mode it is possible that the Re-Mote consumes as little as 150 nA, which

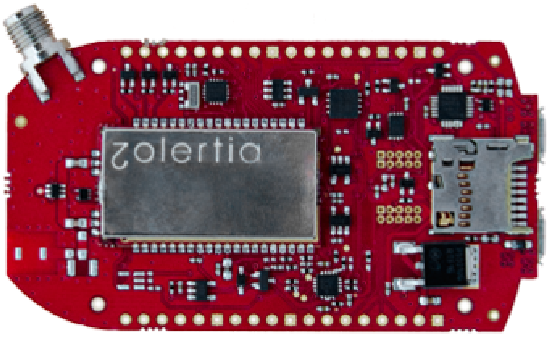
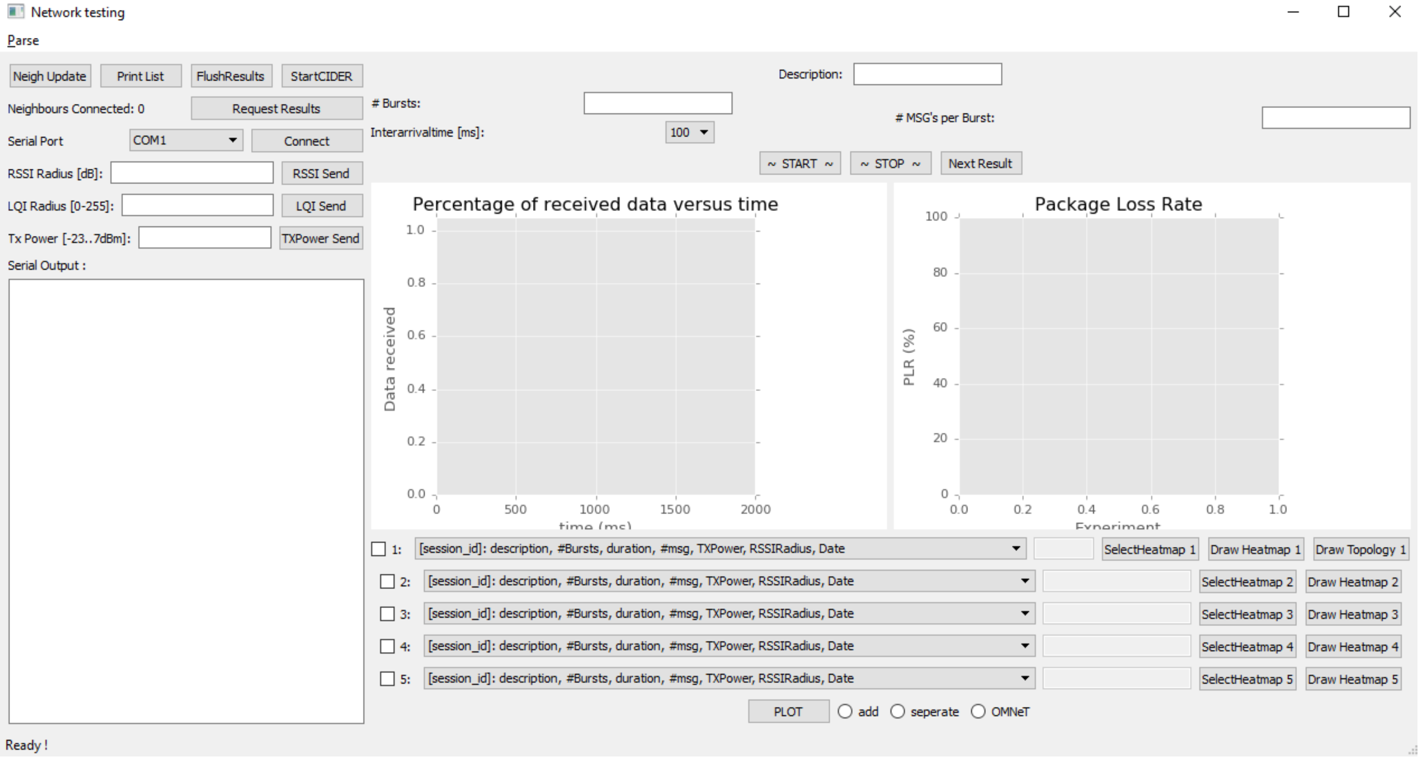


Figure 3.3: Off-the-shelf Re-Mote platform

can theoretically extend the lifetime of a node running an application for up to 10 years without recharging. However, this dissertation studies the wireless lighting networks with mains-powered devices where devices are not powered by a battery, and hence, energy efficiency and long battery lifetime are not a major concern in designing the protocols proposed in this thesis. While a battery is not used in a real-world deployment, for experiments, using a hardware platform attached to an internal battery is an advantage. Due to this, it is possible to easily change the experiment's location without the need to provide the necessary infrastructure such as cabling and power sockets. The Re-Mote comes with an internal battery charging circuit which allows a LiPo battery to be attached and to charge it directly via a USB connection. Also, external connectors are available using the I^2C interface, e.g. an I^2C-driven RGB-LED to demonstrate the protocol operation. Overall, the Re-Mote platform is a state-of-the-art platform in terms of boards developed for low-power wireless sensor networks.

Contiki-OS [43] For the implementing the proposed protocols, the Contiki OS is used (called Contiki for the remainder of this thesis). Contiki is a lightweight open-source operating system designed for resource-constrained wireless sensor devices. Contiki uses the C-programming language and is available for various hardware platforms and can easily be ported to even more hardware platforms. It includes features such as a multitasking kernel, pre-emptive multi-threading, proto-threads [42], and has many communication protocols already implemented. Scheduling with Contiki is event-driven when a FIFO strategy is used [14]. The models are executed in a manner that they are all within the same context, although partial multi-threading is supported. A key feature of Contiki is the Rime stack [44]. Rime is a layered communication stack aimed at simplifying the implementation of sensor network protocols. Each communication layer is designed to make code reuse simple. Using Rime allows simply implementing one's own protocols without the need of complex protocols such as IPv4/6. Rime can be combined with several different MAC layers or application layers; however, with Contiki, there are numerous open source protocols available. With the release of Contiki 3.0 and commit number *5b72869*, the TSCH mode of the IEEE802.15.4 standard was made available. The Rime stack is the basis for the implementation; however, the protocols proposed in this thesis do not rely on the Rime-Stack itself. The Rime stack simply provides an easy way to use the infrastructure to develop new protocols.

Evaluation Software To run the experiments, an evaluation software (Figure 3.4) was developed using *Python* and the PyQt4 library (`https://wiki.python.org/moin/PyQt`), a graphical user interface library. The evaluation software is used to define the various parameters needed for an experiment, such as transmit power, cluster radius, number of messages, and so on. The software is also used to set up a new network and gather topology information as well as to gather the results automatically at the end of an experiment and store them in a database. The results can be processed by the software and plotted to graphs directly using the matplotlib library [66].



Neighbours Connected: 0

Serial Port COM1

Interarrivaltime [ms]: 100

□ 1: [session_id]: description, #Bursts, duration, #msg, TXPower, RSSIRadius, Date

□ 2: [session_id]: description, #Bursts, duration, #msg, TXPower, RSSIRadius, Date

□ 3: [session_id]: description, #Bursts, duration, #msg, TXPower, RSSIRadius, Date

□ 4: [session_id]: description, #Bursts, duration, #msg, TXPower, RSSIRadius, Date

□ 5: [session_id]: description, #Bursts, duration, #msg, TXPower, RSSIRadius, Date

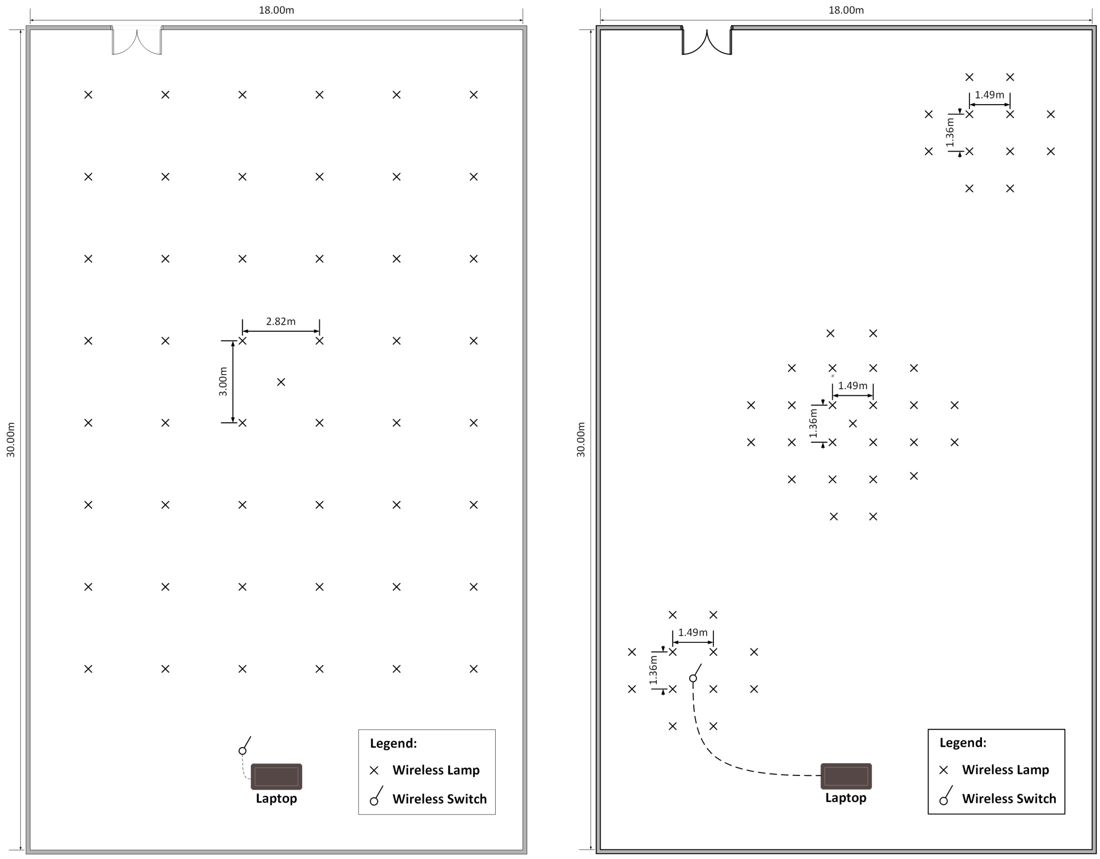
Figure 3.4: Evaluation software

3.2.2 Physical Environments

The experiments were carried out in three different environments, e.g., a sports hall, a laboratory, and an apartment, and also using different arrangements of wireless nodes (grid and grouped). Each location has its own characteristics, mainly varying in their node density and WiFi interference levels; however, the three different locations and arrangements are used to demonstrate that the proposed protocols operate as expected regardless of the environment. Moreover, these three environments were chosen since they represent typical office environments. The sports hall represents a large open indoor space, the laboratory represents an open furnished indoor environment, and the apartment represents a residential area including furniture and multiple rooms. In all environments, a total number of 50 nodes are used, including the light switch node.

3.2.2.1 Sports Hall

The sports hall is utilised to evaluate the proposed protocols in a large-scale environment. In the sports hall, two different node arrangements are used. First, a grid arrangement is used (see Figure 3.5a).



(a) Experimental setup: Sports Hall, Grid

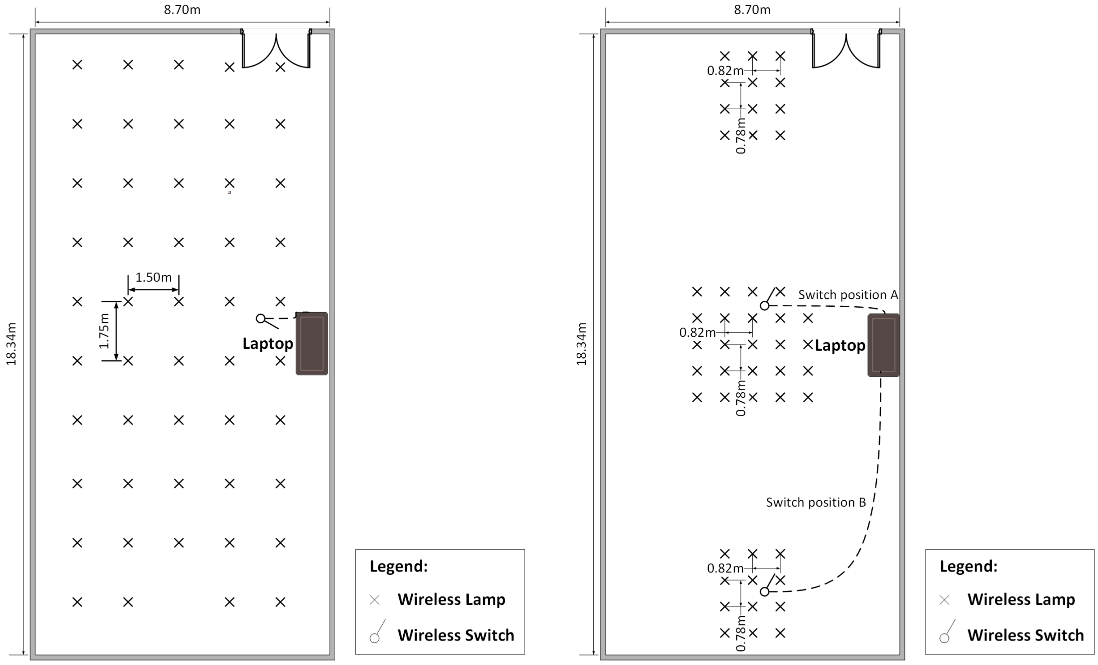
(b) Experimental setup: Sports Hall, Grouped

Figure 3.5: Experimental setup: Sports Hall

Here, all nodes are deployed at equal distances to the adjacent nodes. The vertical distance separating them is 3m and the horizontal distance is 2.82 m, covering an area of 296.1 m^2, which results in a node density of approx. 0.17 $\frac{nodes}{m^2}$. The location of the light switch is at the lower centre end, connected to a laptop running the evaluation software. The second arrangement is a grouped arrangement (see Figure 3.5b). As the name already indicates, the nodes are deployed in groups. Three groups are used, one with 25 nodes located in the centre of the sports hall, one group with 12 nodes located at the lower left side of the sports hall and one group with 12 nodes located at the top right side of the sports hall. The nodes within a group are again deployed at equal distances to the adjacent ones, at a vertical distance of 1.36 m and at a horizontal distance of 1.49 m to each other. The light switch location is within the lower left group and is connected to a laptop running the evaluation software. In the sports hall, the interference level was very low and had no influence on the radio transmissions; therefore, the experiments conducted in the sports hall are used as reference results.

3.2.2.2 Laboratory

The laboratory environment is used to evaluate the proposed protocols in a medium-scale environment. As in the sports hall, for the laboratory, two arrangements are used. First, a grid arrangement is of a higher node density than before (see Figure 3.6a), because the node density has an effect on the overall system behaviour, such as latency, packet loss rate, and user perception, which is shown later in Chapter 4.

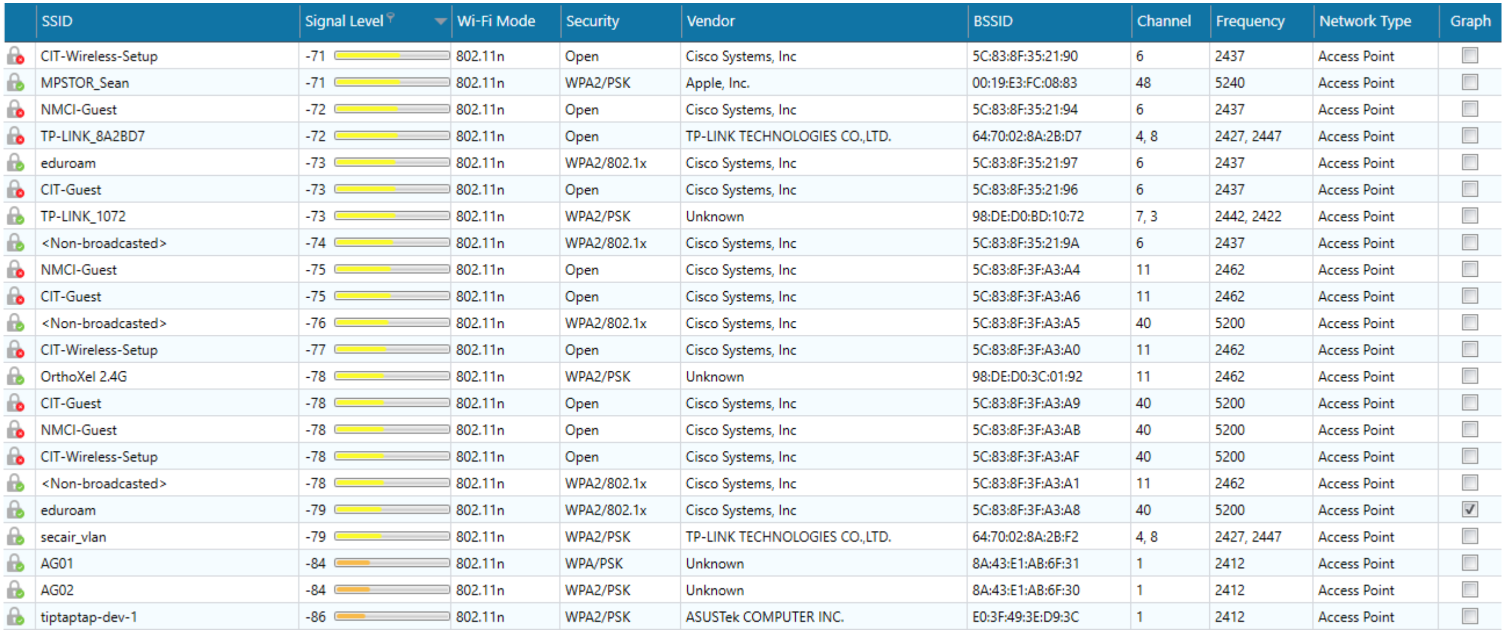


(a) Experimental setup: Laboratory, Grid

(b) Experimental setup: Laboratory, Grouped

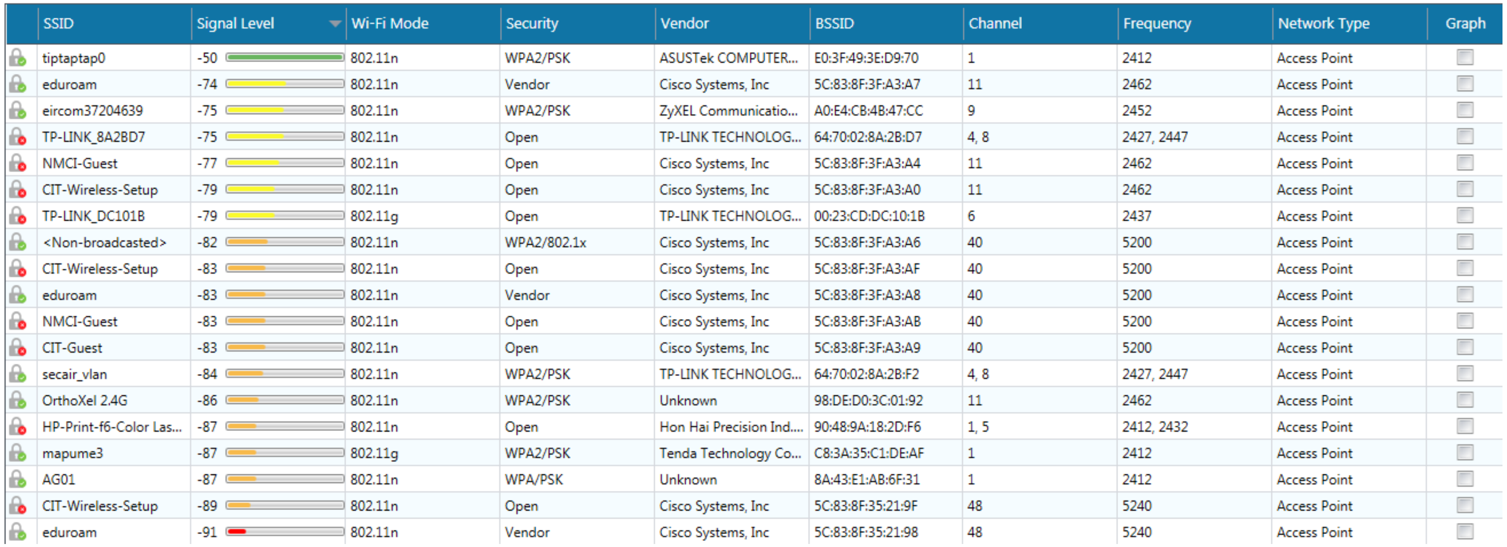
Figure 3.6: Experimental setup: Laboratory

All nodes are deployed at equal distances to the adjacent nodes. The vertical distance separating them is 1.75m and the horizontal distance is 1.5 m, covering an area of 94.5 m^2, which results in a node density of approx. 0.53 $\frac{nodes}{m^2}$. The light switch location is at the centre right, connected to a laptop running the evaluation software. The second arrangement is also a grouped arrangement (see Figure 3.5b). Three groups are used, one with 25 nodes located in the centre of the laboratory site, one group with 12 nodes located at the bottom centre, and one group with 12 nodes located at the top centre. The nodes within a group are again deployed at equal distances to the adjacent ones, at a vertical distance of 0.82 m and a horizontal distance of 0.78 m, which increases the density compared to the sports hall setup. Two physical locations are possible for the light switch, position A is within the 25-nodes group at the centre, and position B is within the bottom centre group. The purpose of changing the light switch is to investigate whether the location of the light switch influences the network performance. It is possible to control the interference in the laboratory to some extent since this room is also used for other researchers' experiments. By turning off all WiFi Access Point (AP) in the room and those in its close neighbourhood, a low interference level is reached. Figure 3.7 shows that even after turning off all APs in the room and the ones directly neighbouring, still more than 20 Service Set Identifier (SSID) are found. Out of the 20 SSIDs, many share the same physical device, e.g. the *CIT-Wireless-Setup* and *NMCI-GUEST* are part of the campus network and are hosted by the same AP. These APs cannot be turned off, and due to their low signals level, this setup is referred to as *Experiments with a low interference level.* The signal level of these APs is lower than -70 dBm, measured at the light switch location. A medium level of interference is reached when only one WiFi router is turned on in the approx. centre of the room. Figure 3.8 lists all APs found in this setup. The list is similar to the low interference setup but is extended by one AP in the centre of the room with an SSID named *tiptaptap0* and of a signal level of -50 dBm. This AP is used by another research group for the regular internet access of their devices. When all WiFi routers in the room and those in the close neighbourhood are turned on, a high interference level is reached. Figure 3.9 shows that now, a total 9 APs are turned on in the lab. These APs are used for the various experiments of other research groups using the laboratory. Their signal levels range from -37 dBm to -70 dBm within the laboratory. By using the different interference levels in a somehow controlled way, the robustness against WiFi interference can be evaluated.



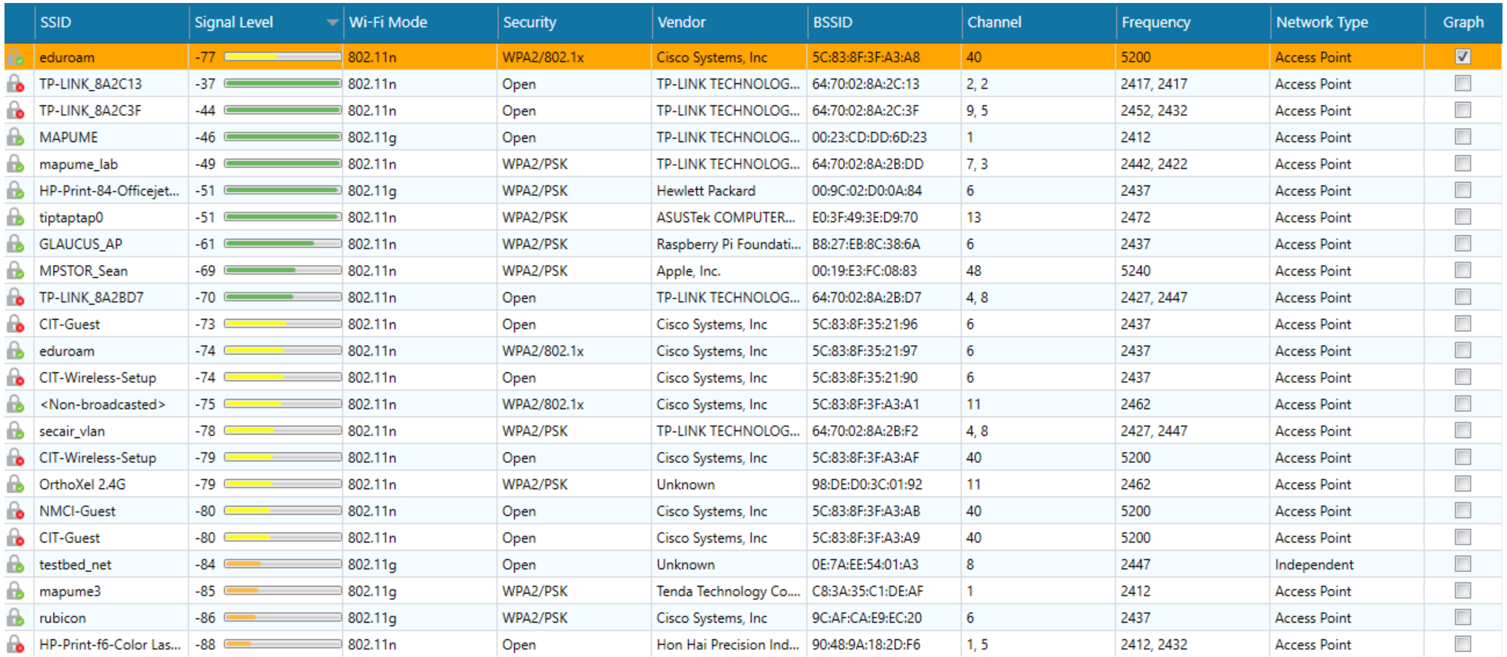
SSID	Signal Level	Wi-Fi Mode	Security	Vendor	BSSID	Channel	Frequency	Network Type	Graph
CIT-Wireless-Setup	-71	802.11n	Open	Cisco Systems, Inc	5C:83:8F:35:21:90	6	2437	Access Point	☐
MPSTOR_Sean	-71	802.11n	WPA2/PSK	Apple, Inc.	00:19:E3:FC:08:83	48	5240	Access Point	☐
NMCI-Guest	-72	802.11n	Open	Cisco Systems, Inc	5C:83:8F:35:21:94	6	2437	Access Point	☐
TP-LINK_8A2BD7	-72	802.11n	Open	TP-LINK TECHNOLOGIES CO.,LTD.	64:70:02:8A:2B:D7	4, 8	2427, 2447	Access Point	☐
eduroam	-73	802.11n	WPA2/802.1x	Cisco Systems, Inc	5C:83:8F:35:21:97	6	2437	Access Point	☐
CIT-Guest	-73	802.11n	Open	Cisco Systems, Inc	5C:83:8F:35:21:96	6	2437	Access Point	☐
TP-LINK_1072	-73	802.11n	WPA2/PSK	Unknown	98:DE:D0:BD:10:72	7, 3	2442, 2422	Access Point	☐
<Non-broadcasted>	-74	802.11n	WPA2/802.1x	Cisco Systems, Inc	5C:83:8F:35:21:9A	6	2437	Access Point	☐
NMCI-Guest	-75	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:A4	11	2462	Access Point	☐
CIT-Guest	-75	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:A6	11	2462	Access Point	☐
<Non-broadcasted>	-76	802.11n	WPA2/802.1x	Cisco Systems, Inc	5C:83:8F:3F:A3:A5	40	5200	Access Point	☐
CIT-Wireless-Setup	-77	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:A0	11	2462	Access Point	☐
OrthoXel 2.4G	-78	802.11n	WPA2/PSK	Unknown	98:DE:D0:3C:01:92	11	2462	Access Point	☐
CIT-Guest	-78	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:A9	40	5200	Access Point	☐
NMCI-Guest	-78	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:AB	40	5200	Access Point	☐
CIT-Wireless-Setup	-78	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:AF	40	5200	Access Point	☐
<Non-broadcasted>	-78	802.11n	WPA2/802.1x	Cisco Systems, Inc	5C:83:8F:3F:A3:A1	11	2462	Access Point	☐
eduroam	-79	802.11n	WPA2/802.1x	Cisco Systems, Inc	5C:83:8F:3F:A3:A8	40	5200	Access Point	☑
secair_vlan	-79	802.11n	WPA2/PSK	TP-LINK TECHNOLOGIES CO.,LTD.	64:70:02:8A:2B:F2	4, 8	2427, 2447	Access Point	☐
AG01	-84	802.11n	WPA/PSK	Unknown	8A:43:E1:AB:6F:31	1	2412	Access Point	☐
AG02	-84	802.11n	WPA2/PSK	Unknown	8A:43:E1:AB:6F:30	1	2412	Access Point	☐
tiptaptap-dev-1	-86	802.11n	WPA2/PSK	ASUSTek COMPUTER INC.	E0:3F:49:3E:D9:3C	1	2412	Access Point	☐

Figure 3.7: Laboratory interference level: Low



SSID	Signal Level	Wi-Fi Mode	Security	Vendor	BSSID	Channel	Frequency	Network Type	Graph
tiptaptap0	-50	802.11n	WPA2/PSK	ASUSTek COMPUTER...	E0:3F:49:3E:D9:70	1	2412	Access Point	□
eduroam	-74	802.11n	Vendor	Cisco Systems, Inc	5C:83:8F:3F:A3:A7	11	2462	Access Point	□
eircom37204639	-75	802.11n	WPA2/PSK	ZyXEL Communicatio...	A0:E4:CB:4B:47:CC	9	2452	Access Point	□
TP-LINK_8A2BD7	-75	802.11n	Open	TP-LINK TECHNOLOG...	64:70:02:8A:2B:D7	4, 8	2427, 2447	Access Point	□
NMCI-Guest	-77	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:A4	11	2462	Access Point	□
CIT-Wireless-Setup	-79	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:A0	11	2462	Access Point	□
TP-LINK_DC101B	-79	802.11g	Open	TP-LINK TECHNOLOG...	00:23:CD:DC:10:1B	6	2437	Access Point	□
<Non-broadcasted>	-82	802.11n	WPA2/802.1x	Cisco Systems, Inc	5C:83:8F:3F:A3:A6	40	5200	Access Point	□
CIT-Wireless-Setup	-83	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:AF	40	5200	Access Point	□
eduroam	-83	802.11n	Vendor	Cisco Systems, Inc	5C:83:8F:3F:A3:A8	40	5200	Access Point	□
NMCI-Guest	-83	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:AB	40	5200	Access Point	□
CIT-Guest	-83	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:A9	40	5200	Access Point	□
secair_vlan	-84	802.11n	WPA2/PSK	TP-LINK TECHNOLOG...	64:70:02:8A:2B:F2	4, 8	2427, 2447	Access Point	□
OrthoXel 2.4G	-86	802.11n	WPA2/PSK	Unknown	98:DE:D0:3C:01:92	11	2462	Access Point	□
HP-Print-f6-Color Las...	-87	802.11n	Open	Hon Hai Precision Ind....	90:48:9A:18:2D:F6	1, 5	2412, 2432	Access Point	□
mapume3	-87	802.11g	WPA2/PSK	Tenda Technology Co...	C8:3A:35:C1:DE:AF	1	2412	Access Point	□
AG01	-87	802.11n	WPA/PSK	Unknown	8A:43:E1:AB:6F:31	1	2412	Access Point	□
CIT-Wireless-Setup	-89	802.11n	Open	Cisco Systems, Inc	5C:83:8F:35:21:9F	48	5240	Access Point	□
eduroam	-91	802.11n	Vendor	Cisco Systems, Inc	5C:83:8F:35:21:98	48	5240	Access Point	□

Figure 3.8: Laboratory interference level: Medium



SSID	Signal Level	Wi-Fi Mode	Security	Vendor	BSSID	Channel	Frequency	Network Type	Graph
eduroam	-77	802.11n	WPA2/802.1x	Cisco Systems, Inc	5C:83:8F:3F:A3:A8	40	5200	Access Point	☑
TP-LINK_8A2C13	-37	802.11n	Open	TP-LINK TECHNOLOG...	64:70:02:8A:2C:13	2, 2	2417, 2417	Access Point	□
TP-LINK_8A2C3F	-44	802.11n	Open	TP-LINK TECHNOLOG...	64:70:02:8A:2C:3F	9, 5	2452, 2432	Access Point	□
MAPUME	-46	802.11g	Open	TP-LINK TECHNOLOG...	00:23:CD:DD:6D:23	1	2412	Access Point	□
mapume_lab	-49	802.11n	WPA2/PSK	TP-LINK TECHNOLOG...	64:70:02:8A:2B:DD	7, 3	2442, 2422	Access Point	□
HP-Print-84-Officejet...	-51	802.11g	WPA2/PSK	Hewlett Packard	00:9C:02:D0:0A:84	6	2437	Access Point	□
tiptaptap0	-51	802.11n	WPA2/PSK	ASUSTek COMPUTER...	E0:3F:49:3E:D9:70	13	2472	Access Point	□
GLAUCUS_AP	-61	802.11n	WPA2/PSK	Raspberry Pi Foundati...	B8:27:EB:8C:38:6A	6	2437	Access Point	□
MPSTOR_Sean	-69	802.11n	WPA2/PSK	Apple, Inc.	00:19:E3:FC:08:83	48	5240	Access Point	□
TP-LINK_8A2BD7	-70	802.11n	Open	TP-LINK TECHNOLOG...	64:70:02:8A:2B:D7	4, 8	2427, 2447	Access Point	□
CIT-Guest	-73	802.11n	Open	Cisco Systems, Inc	5C:83:8F:35:21:96	6	2437	Access Point	□
eduroam	-74	802.11n	WPA2/802.1x	Cisco Systems, Inc	5C:83:8F:35:21:97	6	2437	Access Point	□
CIT-Wireless-Setup	-74	802.11n	Open	Cisco Systems, Inc	5C:83:8F:35:21:90	6	2437	Access Point	□
<Non-broadcasted>	-75	802.11n	WPA2/802.1x	Cisco Systems, Inc	5C:83:8F:3F:A3:A1	11	2462	Access Point	□
secair_vlan	-78	802.11n	WPA2/PSK	TP-LINK TECHNOLOG...	64:70:02:8A:2B:F2	4, 8	2427, 2447	Access Point	□
CIT-Wireless-Setup	-79	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:AF	40	5200	Access Point	□
OrthoXel 2.4G	-79	802.11n	WPA2/PSK	Unknown	98:DE:D0:3C:01:92	11	2462	Access Point	□
NMCI-Guest	-80	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:AB	40	5200	Access Point	□
CIT-Guest	-80	802.11n	Open	Cisco Systems, Inc	5C:83:8F:3F:A3:A9	40	5200	Access Point	□
testbed_net	-84	802.11g	Open	Unknown	0E:7A:EE:54:01:A3	8	2447	Independent	□
mapume3	-85	802.11g	WPA2/PSK	Tenda Technology Co....	C8:3A:35:C1:DE:AF	1	2412	Access Point	□
rubicon	-86	802.11g	WPA2/PSK	Cisco Systems, Inc	9C:AF:CA:E9:EC:20	6	2437	Access Point	□
HP-Print-f6-Color Las...	-88	802.11n	Open	Hon Hai Precision Ind...	90:48:9A:18:2D:F6	1, 5	2412, 2432	Access Point	□

Figure 3.9: Laboratory interference level: High

3.2.2.3 Apartment

The third location is an occupied apartment space. The apartment is used to evaluate the protocol operation in a dense environment. The nodes are deployed all over the apartment at equal distances to their adjacent nodes, similar to what is seen in Figure 3.10. This deployment relates to a residential smart-home deployment, where a high number of luminaires deployed in an apartment are controlled by a single device, e.g. a ZigBee gateway which is connected to a smartphone. While the laboratory is a controlled environment in terms of interference, the apartment is not, which means that it is as close as possible to a real-world setup.

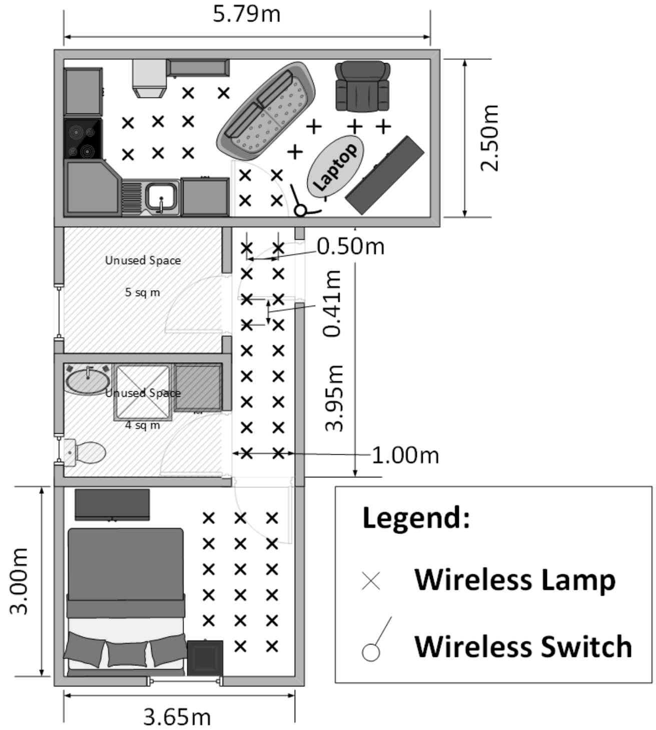
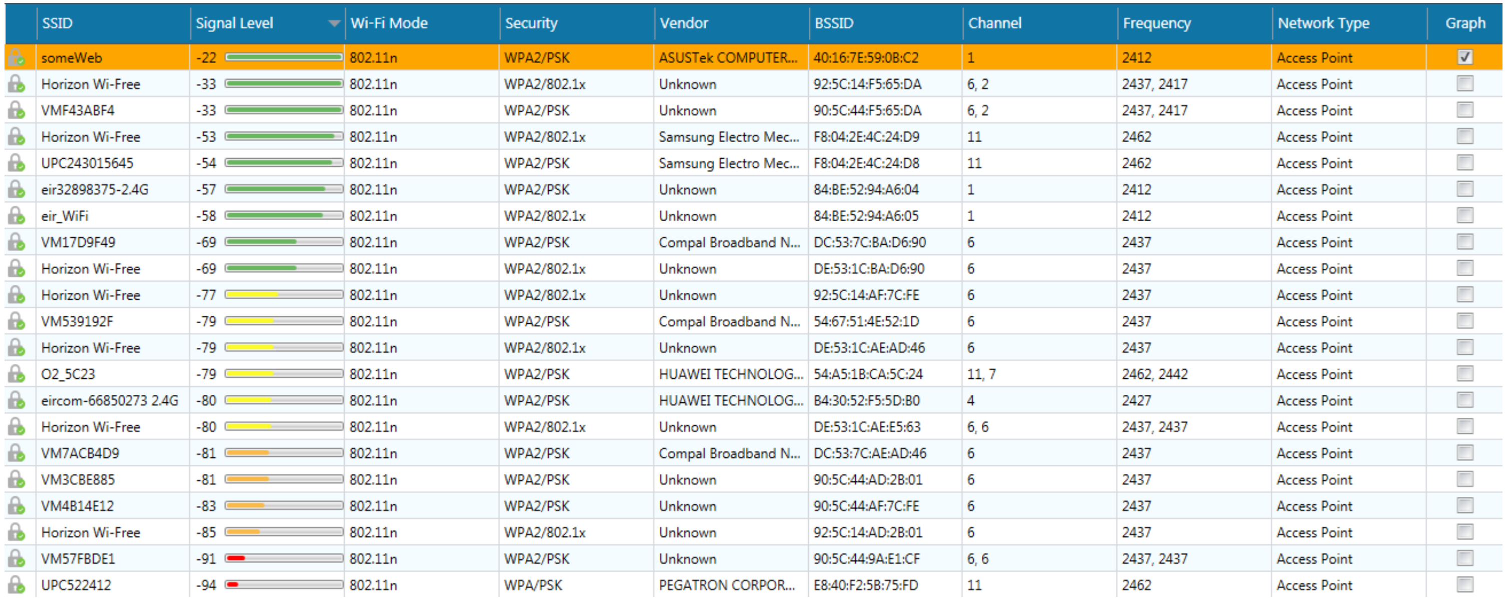


Figure 3.10: Experimental setup: Apartment, Grid

The vertical distances between the nodes are 0.41 m, and the horizontal distance are 0.5 m, covering an area of 22.9 m^2, which results in a node density of approx. 2.18 $\frac{nodes}{m^2}$. The light switch location is at the centre right, connected to a laptop running the evaluation software. The apartment is located in an apartment complex with multiple neighbouring apartments, and as a consequence, there is no control of the neighbouring WiFi networks. Therefore, the interference level in the apartment is very high (see Figure 3.11).



SSID	Signal Level	Wi-Fi Mode	Security	Vendor	BSSID	Channel	Frequency	Network Type	Graph
someWeb	-22	802.11n	WPA2/PSK	ASUSTek COMPUTER...	40:16:7E:59:0B:C2	1	2412	Access Point	☑
Horizon Wi-Free	-33	802.11n	WPA2/802.1x	Unknown	92:5C:14:F5:65:DA	6, 2	2437, 2417	Access Point	□
VMF43ABF4	-33	802.11n	WPA2/PSK	Unknown	90:5C:44:F5:65:DA	6, 2	2437, 2417	Access Point	□
Horizon Wi-Free	-53	802.11n	WPA2/802.1x	Samsung Electro Mec...	F8:04:2E:4C:24:D9	11	2462	Access Point	□
UPC243015645	-54	802.11n	WPA2/PSK	Samsung Electro Mec...	F8:04:2E:4C:24:D8	11	2462	Access Point	□
eir32898375-2.4G	-57	802.11n	WPA2/PSK	Unknown	84:BE:52:94:A6:04	1	2412	Access Point	□
eir_WiFi	-58	802.11n	WPA2/802.1x	Unknown	84:BE:52:94:A6:05	1	2412	Access Point	□
VM17D9F49	-69	802.11n	WPA2/PSK	Compal Broadband N...	DC:53:7C:BA:D6:90	6	2437	Access Point	□
Horizon Wi-Free	-69	802.11n	WPA2/802.1x	Unknown	DE:53:1C:BA:D6:90	6	2437	Access Point	□
Horizon Wi-Free	-77	802.11n	WPA2/802.1x	Unknown	92:5C:14:AF:7C:FE	6	2437	Access Point	□
VM539192F	-79	802.11n	WPA2/PSK	Compal Broadband N...	54:67:51:4E:52:1D	6	2437	Access Point	□
Horizon Wi-Free	-79	802.11n	WPA2/802.1x	Unknown	DE:53:1C:AE:AD:46	6	2437	Access Point	□
O2_5C23	-79	802.11n	WPA2/PSK	HUAWEI TECHNOLOG...	54:A5:1B:CA:5C:24	11, 7	2462, 2442	Access Point	□
eircom-66850273 2.4G	-80	802.11n	WPA2/PSK	HUAWEI TECHNOLOG...	B4:30:52:F5:5D:B0	4	2427	Access Point	□
Horizon Wi-Free	-80	802.11n	WPA2/802.1x	Unknown	DE:53:1C:AE:E5:63	6, 6	2437, 2437	Access Point	□
VM7ACB4D9	-81	802.11n	WPA2/PSK	Compal Broadband N...	DC:53:7C:AE:AD:46	6	2437	Access Point	□
VM3CBE885	-81	802.11n	WPA2/PSK	Unknown	90:5C:44:AD:2B:01	6	2437	Access Point	□
VM4B14E12	-83	802.11n	WPA2/PSK	Unknown	90:5C:44:AF:7C:FE	6	2437	Access Point	□
Horizon Wi-Free	-85	802.11n	WPA2/802.1x	Unknown	92:5C:14:AD:2B:01	6	2437	Access Point	□
VM57FBDE1	-91	802.11n	WPA2/PSK	Unknown	90:5C:44:9A:E1:CF	6, 6	2437, 2437	Access Point	□
UPC522412	-94	802.11n	WPA/PSK	PEGATRON CORPOR...	E8:40:F2:5B:75:FD	11	2462	Access Point	□

Figure 3.11: Apartment interference level

3.3 Evaluation criteria

To evaluate the proposed protocols and to be able to compare them with the state-of-the-art protocols, first, the evaluation criteria need to be defined. The protocols analysed and evaluated within this thesis are based on two key criteria, network latency and PLR. This section provides an overview of the criteria. The requirements listed in Section **??** are closely linked to the evaluation criteria listed in this section. While the network latency and the reliability are the most important requirements to achieve a high level of QoE, the evaluation criteria are based on these two.

3.3.1 Network latency

Latency is defined as the delay occurring between an input into a system and the desired outcome. Latency might be seen differently, depending on the area, and also might have different effects, depending on the system. Most importantly, latency has an effect on how usable a system is perceived by a user. In terms of communication systems, latency can be demonstrated as live transmissions from different points around the world as the transmission from the transmitter to a ground station to a satellite, and from the satellite to a receiver. Each of these transmissions takes time, which combined, is the latency [138].

The network latency is a definition of how much time it takes for a data packet to be transmitted from the source to the destination. In the best case, the latency is close to zero. However, the network latency includes different parts, the first of which is the propagation latency. This is the time it takes for a signal to travel from one point to another point at the speed of light. This part can be neglected for lighting control networks, as the source and destinations are very close to each other. The propagation latency is approx. $3.3 \times 10^{-9} \frac{s}{m}$. The second is the transmission latency, which is highly dependent on the transmission medium and the packet size. For instance, a 100-byte packet takes much more transmission time in a network operating using the IEEE802.15.4 standard than in a network using the IEEE802.11. For the lighting networks discussed in this thesis, the IEEE802.15.4 standard is used as underlying communication, operating in the

2.4 GHz ISM band. Hence, the transmission delay for a full-size packet of 127 bytes is approx. 4 ms and needs to be considered for the overall latency as this is a latency occurring per hop. This means if a message takes 10 hops to cover the network, it is already 40 ms latency. The third is the processing time of the system. Every device receiving a message takes time to process a message until it can either be forward to the next destination or the information can be retrieved from a message received. The last part of the network latency is the time that a message remains in a buffer. Before a message can be transmitted, it has to be added to a buffer; each message remains in a buffer for a certain amount of time, depending on the communication protocol used.

For this thesis, the latency is considered to be the network latency where the time measured is how long it takes between generating a control command and receiving this command in the application layer of the end devices. The network latency represents the average latency of every single device for each received message. Messages that are not received are not considered for the network latency. The latency includes the processing time per node, which itself can affect the latency as shown later. The latency should not exceed the in Section **??** defined latency.

3.3.2 Packet Loss Rate

The packet loss is closely associated with QoS. The packet loss rate that is acceptable depends on the type of data being sent. For example, for Voice over IP (VoIP), one or two missing packets per second does not affecting the quality of a conversation [95], but a higher loss rate of 5% or more will definitely affect the transmission. However, for lighting control networks, a low PLR is a key factor. If only a few lamps do not receive a single message, a user may easily think that a lamp is broken when it does not turn on while others do. The PLR can be affected by many factors, e.g. by network contention. If a node is supposed to forward incoming data, and this data is arriving faster than the node is able to forward the data, the queue will eventually be full, and packets will be dropped. Besides this cause for packet loss, a correct transmission can be influenced by other devices transmitting at the same time, called interference. When there is interference, a receiver is not able to decode the incoming packet correctly.

In this thesis, the PLR for a single device is defined as the percentage of packets lost during a transmission. This includes only the lighting commands – the messages generated to form and maintain the network topology and acknowledgements are not considered. The overall PLR is thus the average of each single PLR value.

Chapter 4

Broadcast storm problem in dense wireless lighting control networks

Chapter 2 showed that current studies of wireless lighting control networks, especially those designed for indoor environments, consist only of few wireless devices. The traffic patterns of the work reviewed were based on unicast traffic with one central device controlling individual lamps, either to reach an acceptable comfort level or to increase the energy savings in a building, or both. This chapter analyses the performance of a wireless lighting control network, using state-of-the-art communication protocols such as flooding. The analysis is based on extensive computer simulations, using the tools described in Chapter 3, simulating dense networks consisting of up to 400 lamps. The performance is analysed based on key performance parameters such as latency and PLR. Furthermore, these parameters a brought into the context of QoE. The remainder of this chapter is as follows, Section 4.1 discusses selected broadcast protocols. Section 4.2 presents a detailed overview of the simulation model used and its configuration is given. Then, the following sections discuss the simulation results from different aspects, and finally, the chapter is concluded in Section 4.9.

4.1 The state-of-the-art in broadcasting by flooding

Broadcast systems could be implemented simply as a single powerful transmitter that reaches all receivers simultaneously in a single hop. If such powerful transmissions would be permitted and the frequencies could be reused in all buildings worldwide, it could support arbitrary high densities since an arbitrary number of receiving nodes can be added. Unfortunately, WSNs are limited to certain frequencies, e.g. the 2.4 GHz ISM radio band, but also their maximum transmit power is limited. To reach every member of a network, multi-hop communication is often necessary. In terms of lighting control, a common approach is broadcasting control messages. The easiest method to reach all nodes is called "Classic Flooding" (flooding in the remainder of this thesis). With flooding, any node that receives a message verifies whether it is a new message, or a message already received. If it is a new message, the node will send it to its neighbours, hence, every single node retransmits each flooded message exactly once. While in sparse networks, flooding may fail to reach all nodes [101], in dense networks it results in excessive redundancy, contention, and collisions [101], known as *Broadcast Storms*. Therefore, flooding mechanisms must trade-off between minimizing redundancy and maintaining a good latency and reachability (the ratio of nodes receiving a broadcast message to the total number of nodes that are directly or indirectly reachable from a source node) [119].

Reducing the effects of a broadcast storm has drawn the attention of many researchers in recent years. Ni et al [101] introduce five basic mechanisms to mitigate the problem: 1. the probabilistic scheme, 2. the counter-based scheme, 3. the distance-based scheme, 4. the location-based scheme, and 5. the cluster-based scheme. With the probabilistic scheme, only a certain number of nodes retransmit the initial message. For each network, it is necessary to pre-set a threshold. A receiving node then generates a random number and compares it to the threshold. If the number exceeds the threshold it will not retransmit the message, otherwise, it will retransmit the message. This reduces the traffic in the network in both sparse networks and dense networks, but it also creates new problems, such as the lack of guaranteeing coverage over the whole network. The counter-based scheme works similar to the probabilistic scheme. It is necessary to pre-set a maximum counter value for the network. The authors state that due to a busy medium, backoff procedures, or other queued messages, it is possible for a single node to overhear a transmission of the same message before starting its own transmission. Whenever a node overhears a transmission, it increases its internal counter, and if the counter value exceeds the threshold at the beginning of the transmission, the transmission will be dropped. The counter-based scheme reduces the wireless sensor network traffic but still lacks the guarantees to cover the whole network. The decision whether to retransmit in a distance-based scheme is based on the distance to the transmitter. The distance is calculated based on signal strength. In general, signal strength is a relatively error-prone indicator for distance because it is influenced by many factors. The location-based method requires the use of a GPS module for acquiring the exact location where each transmitting/retransmitting node adds its location to the message. A receiving node delays its retransmission by a random amount of time. While the node is in the wait period, it overhears other transmissions. Whenever a node receives a new message, a node updates its additional coverage area based on the messages already received, and, therefore, also the locations of the other nodes. If the additional coverage area is below a pre-set threshold, the node discards the retransmission of the message. A location-based method using GPS works for outdoor applications; however, for indoor applications, the localisation systems are often not accurate enough as in lighting applications the luminaires are very close together. Besides that, it is not guaranteed that a node with a high additional coverage area actually covers more additional nodes than a node with a low additional coverage area. The fifth method is the cluster-based method, introduced by the authors of [101]. With this method, the network divides itself into clusters, containing three types of nodes: cluster slaves, cluster heads, and gateway nodes. Cluster slave nodes only receive broadcast messages or start new broadcasts. Cluster heads always forward messages or start new broadcasts. Cluster heads cover the whole cluster but are not able to reach the nodes of other clusters. Gateway nodes are nodes within reach of multiple cluster heads. A receiving gateway node uses one of the aforementioned methods to forward a message. Clustering appears to be a well-suited solution but generates new problems, e.g. selecting best cluster heads, which is a not a trivial problem. Based on these five basic approaches, many researchers have developed protocols to mitigate broadcast storms in wireless networks.

The authors of [106] introduce the Scalable Broadcast Algorithm (SBA). SBA is based on the two-hop neighbourhood information as well as on a coverage-based approach. At the beginning, the nodes exchange "Hello" messages, where each node adds its neighbour's information to the message. With this method, every node is aware of its two-hop neighbourhood. After the topology information is exchanged, a node is either a transmitter of a broadcast or a re-transmitter of a broadcast message. Based on the neighbourhood information, a receiving node can determine the nodes already covered. If a node's neighbours are already covered, it omits the retransmission of a message. Simulation results show that SBA reduces the communication cost up to 60% compared to flooding. The protocol was evaluated in a network consisting up to 120 wireless nodes; unfortunately, the information on how the nodes were distributed is missing, especially that of the density of the network. Due to additional traffic at the beginning, it is likely that in a dense network, many messages are lost due to collisions; therefore, it remains open how this algorithm would work in a dense network.

Kim et al [41] present improved methods of the distance- and counter-based schemes presented in [101], by combining both methods into a hybrid approach. The aim of the improved method is to combine the advantages of the schemes to increase the reachability and reduce the redundancy without the need of equipping the nodes with additional hardware such as GPS devices. The authors use first the initial distance-based approach where each node receiving the message rebroadcasts the message if they are located at distance x from the transmitter node. Now, to avoid that all nodes located at distance x from the transmitter node rebroadcast the message, they include a counter value. To determine whether a node should rebroadcast or not, the authors defined a bounding algorithm containing four steps:

Step 1: When a node receives a message, it sets the distance of d_{min} to the distance to the broadcasting node. If the node receives the message for the first time, it initializes the counter, otherwise, it will incrementally increase the counter by one.

Step 2: In the next step, the node checks first whether d_{min} exceeds the distance threshold. If yes, the node proceeds to Step 4. If the counter value is below the counter value threshold, the node will proceed to Step 3. If the counter value exceeds the threshold, the node continues to Step 4.

Step 3: In this state, a node sets a timer and waits for a random period of time. If the node should receive the same message again while waiting, the node cancels the timer and returns to Step 2. If the timer expires, the node will transmit the message.

Step 4: In this state, the node cancels the retransmission of a received message due to being too close to the initial node or due to overhearing the message too often. Also, the node ignores all future receptions of the same message.

This approach reduces the number of rebroadcasts when compared to the initial distance-based approach as well as the counter-based scheme. Also, the reachability is increased compared to previous protocols. However, with the proposed hybrid protocol, it is not guaranteed that all nodes receive the message since the initial message might only be received from the number of nodes that fulfil the requirement of being a certain distance away from the initial node. In this case, it is possible that a region in the network will not receive the message. Additionally, if the rebroadcast is delayed by all of the forwarding nodes, the network coverage time and thus, the network latency increases. With this in mind, this protocol seems not to be suitable for lighting networks.

A reliable broadcast mechanism for ZigBee networks is proposed by the authors of [122]. The authors assume a ZigBee network, including one *ZigBee Coordinator* device (ZC), and multiple *ZigBee Router* (ZR) and *ZigBee End Devices* (ZED), and furthermore, the authors use a tree topology with the coordinator device as the root node. The paper proposes the *ZARB*-algorithm (ZigBee Acknowledgement-Based Reliable Broadcast). As the name already implies, the broadcast mechanism is based on sending acknowledgements. For this, the authors distinguish between active acknowledgements and passive acknowledgements, where an active acknowledgement is one sent directly to its parent node, and a passive acknowledgement is one identified by a parent when receiving a rebroadcast message from its child node. The authors also use an ACK-based self-pruning algorithm to reduce the redundancy in the network. A node can decide to discard a rebroadcast if when can ensure that all child nodes have already received the broadcast message. As in the tree topology, two ZED and ZR devices can physically be neighbours but be logically assigned to different parent nodes. Therefore, it can happen that nodes receive the broadcast message from a different parent node than their own but still acknowledge the reception to their own parent node. However, when using the self-pruning algorithm, a ZR node receiving a broadcast message will delay the retransmission. During the wait time, the node decides whether to rebroadcast or not. If the node receives ACK-messages from all of its child nodes, it will simply send an ACK-message to its parent node; otherwise, the node will rebroadcast the message at the end of the delay. The delay is calculated using a pre-set constant delay, the tree degree of the node, and a random time, using equation 4.1

$$t_{jitter} = \frac{T_{const}}{d_{tree}} + t_{random} \tag{4.1}$$

In this way, a ZR device closer to the end of the tree will wait shorter than a node closer to the root device before rebroadcasting a message. The authors tested their algorithm in a simulation consisting of 50 to 300 nodes. The results show that the number of rebroadcasting nodes is reduced by approx. 50% compared to the standard ZigBee broadcasting mechanism. Also, the authors attain a 99% packet delivery ratio in a 300-node network. Unfortunately, the results give no indication about the latency covering the whole network as well as regarding the number of acknowledgements sent during the broadcast. Even if the number of rebroadcasts is reduced by up to 50%, the number of acknowledgements transmitted can easily exceed that number. In lighting networks where networks with several hundred nodes are common, a high contention by acknowledgements can be expected.

Ferrari et al. propose the *Glossy* protocol, a protocol for very fast network flooding and time synchronisation in WSNs [51]. The authors of Glossy use a novel approach in which they consider the interference to be an advantage rather than something to be avoided. Glossy enables simultaneous transmissions of the same packets, which interfere constructively. Constructive interference is simply defined as when a receiver detects the superposition of a signal generated by multiple transmitters. To make interference constructive in wireless sensor networks based on IEEE802.15.4, two signals have to fulfil a temporal displacement requirement. The authors ran extensive experiments to define the upper bound of the temporal displacement. The results show that any IEEE802.15.4 transmitter interferes constructively when the temporal displacement is smaller than 0.5 μs. To achieve this using Glossy, nodes turn their radio on, listen for any incoming packet and forward packets immediately if any received. The key with Glossy is ensuring that all receivers turn on their radio at the same time as well as forwarding incoming messages at the same time. Therefore, the authors have minimised the software execution time and decoupled the application's operation. To minimize the software delay with Glossy, the authors use buffered packet transmissions and receptions to keep the number of software instructions to a minimum. Simply, by issuing interrupts, the Rx buffer informs the Tx buffer when a packet is fully received and ready to transmit. Decoupling the application's operations is possible with Glossy since each node knows the interval between two Glossy phases. Thus, it is possible to stop the application process right before a Glossy flood and resume it right after a flood; as a result, the Glossy period never interferes with any other operation running on a single node, which makes the behaviour of a node during a flood highly deterministic. With Glossy, it is possible to synchronise the network as each packet includes a relay counter, and each node can estimate the slot length of a flood. In this way, each node can calculate a common reference time by using the two parameters.

However, the authors evaluate Glossy by running extensive experiments on real sensor nodes in two different setups, a setup in a controlled environment and setups on real testbeds around the globe. In the controlled environment, the authors evaluate the impact of temporal displacement, a number of concurrent transmitters, and the accuracy of time synchronisation with Glossy. First, they evaluate the impact of the temporal displacement where the results clearly show that if the temporal displacement exceeds 0.5 μs, the reliability drops below 40%. This again indicates that a synchronous operation of the nodes is the key with Glossy. Secondly, the authors used up to 10 nodes to evaluate the impact of concurrent transmitters. The results show that the number of concurrent transmitters has no impact on the reliability of Glossy; for every setup, Glossy achieves a reliability of more than 98%. Finally, the authors demonstrate the accuracy of time synchronisation with Glossy. For this, they use five nodes, one node to initiate a flood, and four receiving nodes. At the first reception, the nodes calculate the reference time and schedule the next Glossy period. Whenever a Glossy period is due, each node activates an external pin. The authors connected these pins to oscilloscopes and measured the time difference between the activation at the initiator and at the receivers. The results show that with an increasing distance to the initiator, the synchronisation error rises to up to 0.4 μs, with a standard deviation of 4.8 μs for nodes 8 hops away. This indicates that Glossy achieves an accurate synchronisation in a multi-hop network. In addition to these evaluations, the authors investigate the impact of network characteristics on Glossy as well as the number of maximum transmissions of a Glossy flood in a real network. The authors use three different testbeds with the number of nodes in a network ranging from 39 up to 94 nodes. The results show that Glossy is not affected by a changing node density. For all experiments, Glossy achieves a reliability of higher than 99% in all experiments. Also, the results show that the Glossy performance is highly dependent on the network size. The authors state that the latency and radio-on time increase linearly with the number of hops, but still average 2.4 ms for latency and 10.2 ms for a radio-on time in an 8-hop network. Finally, the authors show that increasing the maximum number of retransmissions increases the flooding reliability. For one retransmission, the reliability is already above 99% while with three retransmissions, the reliability has increased to above 99.9%.

After the first analyses, Glossy seems like the perfect solution for any application with low latency requirements. However, after a thorough analysis, Glossy has a few drawbacks, especially for wireless lighting control networks. With the very low network latency as well as the very high-reliability, Glossy fulfils the most important requirement for wireless lighting. Nonetheless, the decoupling of the application (stopping the application) from its operation represents the biggest issue in adapting Glossy to a lighting use case. For example, in a lighting use case, a change of state needs to be detected by the application at any time from any externally connected switch. As Glossy stops the operation of the application during a Glossy flood, such a detection is not guaranteed. This issue could be solved by using a multi-core hardware platform that runs two independent processes; however, as the hardware for sensor networks are low in cost, using a multi-core platform would increase the cost of such devices. Another drawback is that Glossy is hardware-dependent, which means that porting to newer hardware platforms is a difficult task even though the authors have provided the information in [51]. At the time of this thesis, Glossy was only available for two hardware platforms: The common TelosB hardware platform and a low-cost platform by OLIMEX. But, Glossy shows great potential in its operation, and it is used as the base protocol for other approaches. One promising approach is the *Low-power Wireless Bus* which is discussed in Section 5.4.1.

Summary Broadcasting is a common mechanism in wireless communication. It is usually used to get information from one device to many others. In larger deployments, multi-hop communication is needed. The literature shows a simple mechanism, called *Flooding*, where each node that receives a new message forwards this message exactly once. However, this mechanism was proven error-prone in many publications, and also, using different approaches, many solutions mitigated the problems created by flooding. All proposed protocols have in common reducing the number of forwarder nodes; however, in WSNs, broadcast protocols need to adapt to how and when they should propagate their information. Simple approaches to mitigating a broadcast storm, like a counter-based or probabilistic scheme, reduce the communication redundancy in WSNs, still providing that all nodes are reached. To achieve a better performance by further reducing the redundancy, additional information is needed, such as of the one-hop neighbourhood. Another approach is to use the broadcast storm as an advantage to enable constructive interference, like in the Glossy protocol. Using this, it is possible to achieve a very high reliability and a low latency, with the drawback that specific hardware is necessary and only periodic traffic is supported, which makes Glossy unsuitable for lighting applications. This section provided an overview of common broadcasting techniques. However, the effects of a broadcast storm on a wireless lighting control network are unknown, and the following sections investigate how a flooding-based broadcast storm affects such networks.

4.2 Simulation configuration

To investigate the lighting network behaviour in terms of latency, PLR, and user experience, the simulation model described in Section 3.1.2 is used. The network layer flooding is implemented in a first instance to analyse the effects of the aforementioned broadcast storm on a lighting control network. Flooding is used because it is easy to implement and still commonly used, e.g. the multi-hop broadcasting mechanism of ZigBee is based on flooding. In a second instance, a probabilistic forwarding scheme is implemented to analyse whether the effects of a broadcast storm are improved, and the requirements of a lighting control network can be met. In probabilistic forwarding, after receiving a new message, each node decides whether the message should be forwarded or not. To make this decision, a node generates a random number between 0 and 1 upon reception. If this number is greater than a pre-defined threshold, the message is forwarded, otherwise, the message is dropped. The authors of [101] suggest that for dense networks, a small forwarding probability is sufficient to cover the network; thus, the threshold is set to 0.65 which results in a forwarding probability of approx. 35%. The MAC-layer is based on the IEEE802.15.4 non-beacon enabled mode which implements a CSMA/CA protocol. It is possible to modify several protocol parameters such as the backoff retry parameter and the transmit power level. The backoff retry parameter of the CSMA/CA algorithm is set to 5 by default, and thus, it is investigated whether for lighting control networks, a different backoff retry parameter than the default parameter achieves better results. The backoff retry parameter defines how often a node is allowed to access the channel before declaring a channel access failure and dropping the message. The transmit power levels chosen are similar to real-world transceivers, such as the Texas Instruments CC2420 radio, and these can range from -25 dBm up to 0 dBm, while 0 dBm is the default value. A summary of possible simulation parameters is shown in Table 4.1.

The exact configuration can be found in Appendix E.

These settings are possible and can vary for different configurations. Each of the following sections defines the exact settings used. As described in Section 3.1, the simulated networks contain up to 400 nodes, and the simulated environment is based on the Postgrad room located in the Nimbus Centre.

Parameter	Setting
Application	Dimming
Traffic	Message Burst Duration: 2 seconds New Message: 100 ms/200 ms (default)
Network Density	20 Nodes up to 400 Nodes in an area of (80 x 20 x 3.3) m
Backoff Retry	1 to 5 plus Random Backoff
Transmit Power	-25 dBm to 0 dBm
Number of simulation runs	20 per configuration
Confidence level of Akaroa	0.95

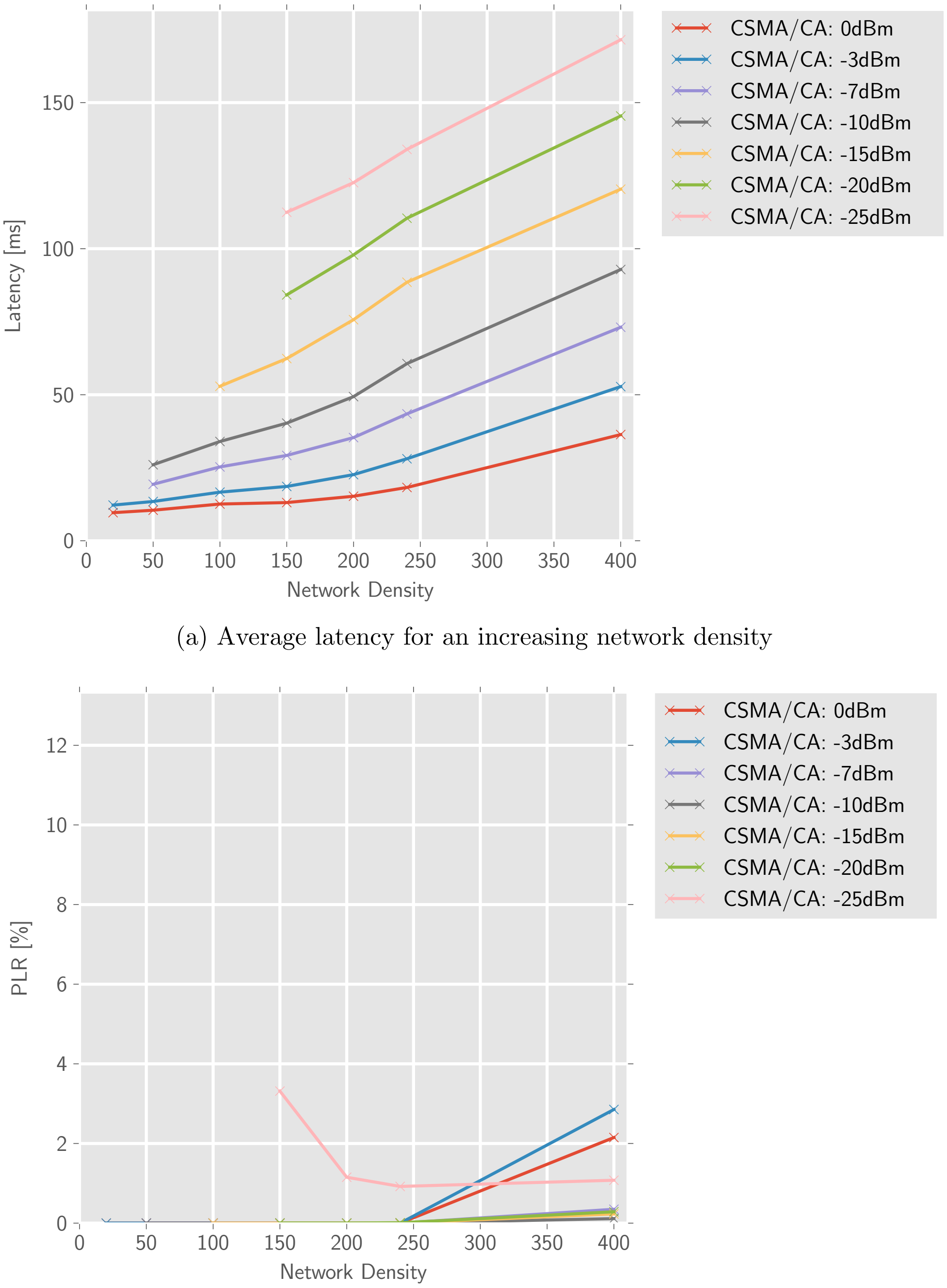
Table 4.1: Simulation configuration

4.3 Effects of an increasing network density

This section investigates the effects of an increasing network density in combination with flooding on the overall network performance. The number of nodes ranges from 20 to 400 nodes. By increasing the number of nodes in a network, the density increases while the room size remains the same. Transmit powers ranging from -25 dBm to 0 dBm and the default backoff retry parameter are used. Note that when lowering the transmit power level, networks with 20, 50, or 100 nodes are not dense enough to be fully connected; hence, it was necessary to increase the number of nodes when a lower transmit power is used. The traffic is as described in Table 4.1, modelled with a new command every 200 ms. Figure 4.1 shows the average latency and Packet Loss Rate (PLR) for this simulation configuration. The PLR is defined using Equation 4.2. n is the total number of nodes in the network, $numTx$ is the number of messages sent by the light switch, and $numRx_i$ is the number of messages received at node i, $i \in \mathbb{N}\,[1..n]$.

$$PLR = \frac{\sum_{i=1}^{n}\left(100\% - \frac{100\% \cdot numRx_i}{numTx}\right)}{n} \tag{4.2}$$

At first, it can be seen that the network density has an impact on both the average latency and the PLR. For each increase in the network density, the latency also increases. At a 0 dBm transmit power level, the latency increases from approx. 10 ms for a network containing 20 nodes, to approx. 35 ms for a network containing 400 nodes. The PLR for the same level of transmit power shows that only a dense network results in a slightly higher packet loss; in up to 200 nodes, no increase in the PLR can be seen. Only when the network density increases to 400 nodes, the PLR is slightly higher than 2%. This shows that when using a transmit power level of 0 dBm, the latency fulfils the requirements of a latency below 200 ms; however, the PLR is approx. 0% for sparse networks while for denser networks, an increase to above 2% can be seen. The low latency and low PLR of 0 dBm can be explained by the high transmission range. This means that the light switch itself is able to reach most of the lamps in the network with one hop. Also, the 200 ms between two new commands are long enough for the network to reach an idle state again. The network is in an idle state during the transmission of data messages, except for the messages necessary to maintain the network, e.g. each node has forwarded the initial message while flooding the network at some point in time, and the network returns from a potential high traffic load to an idle state. When the network reaches the idle state, possible retransmissions of old commands do not collide with new commands, which results in a lower PLR. Only when the network becomes dense, nodes have to wait longer until the received command is retransmitted because the channel is busy more often, which results in an increased probability of collision between the new and old commands. Not



(a) Average latency for an increasing network density

(b) Average PLR for an increasing network density

Figure 4.1: Network performance for an increasing network density

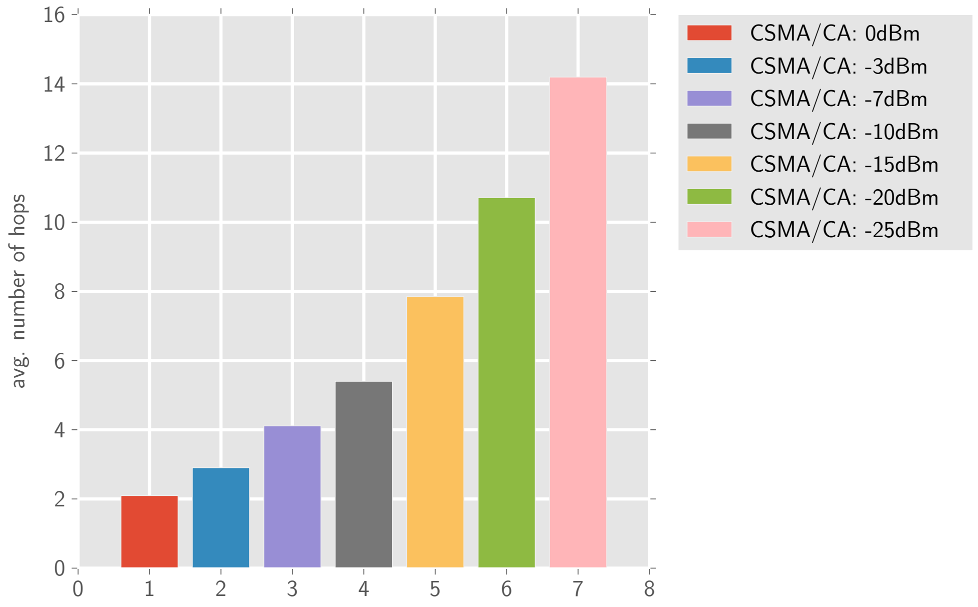


Figure 4.2: Average hop count; network density: 400 nodes

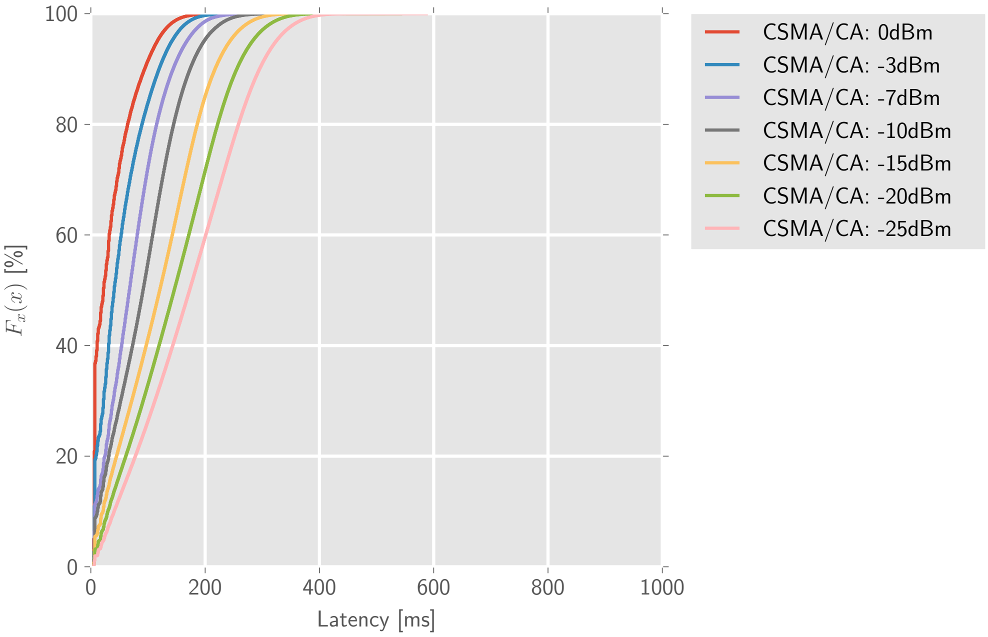
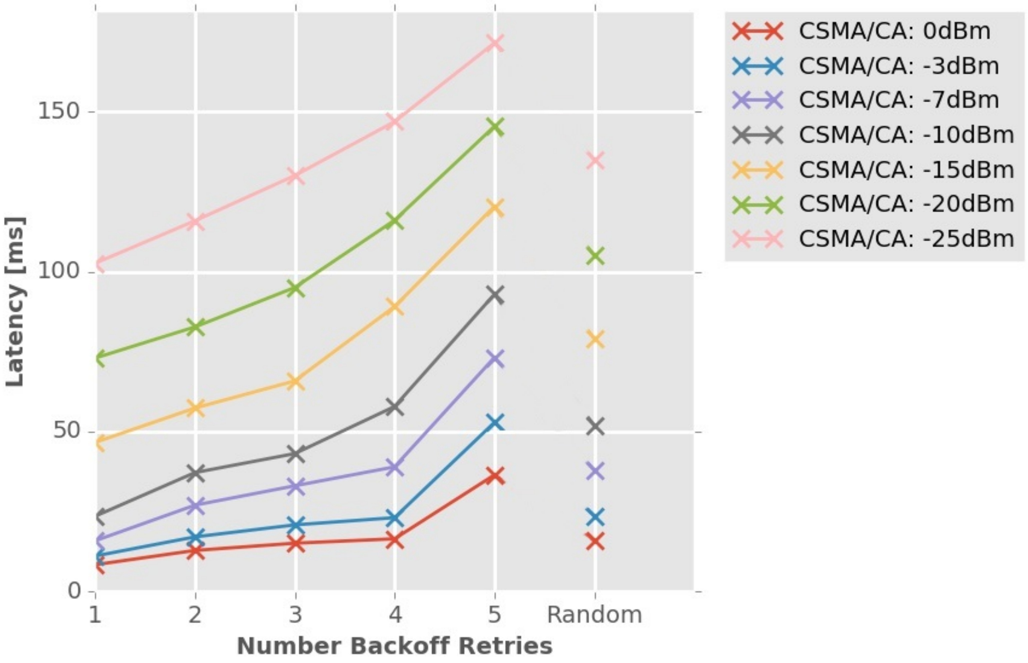


Figure 4.3: Empirical CDF; network density: 400 nodes

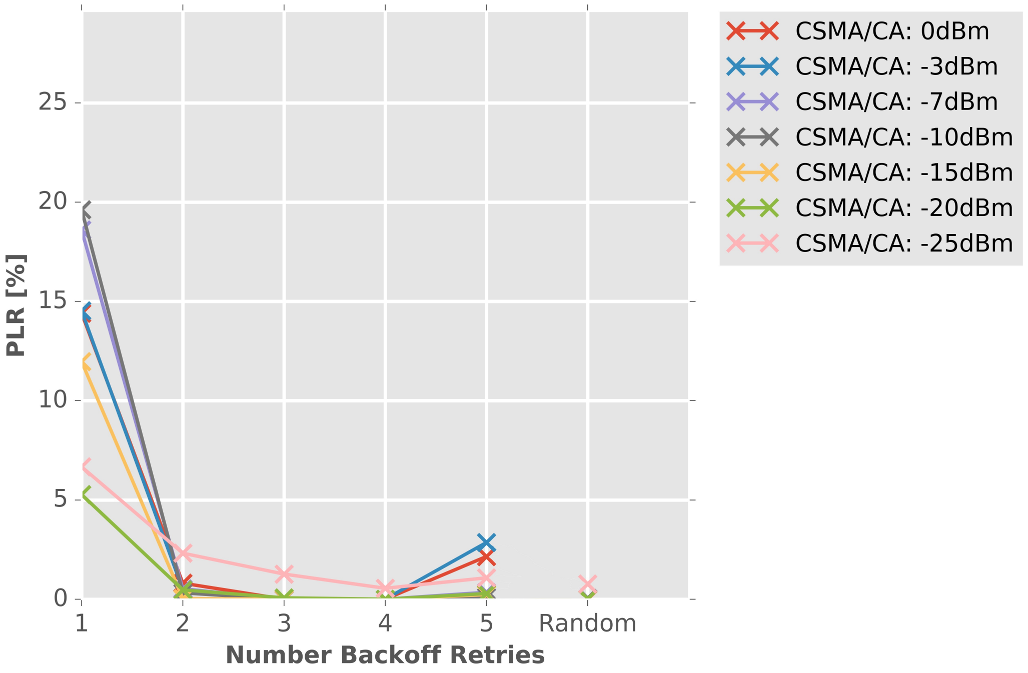
only the network density has an impact on the network performance but also the transmit power. It can be seen that lowering the transmit power level to -3 dBm increases the latency for both sparse networks and dense networks. This can be observed for each lower level transmit power, too. When using -3 dBm of transmit power, the latency for a network of 400 nodes is approx. 50 ms, which increases rapidly to 180 ms when a transmit power level of -25 dBm is used for a network of the same density. Lowering the transmit power level directly results in lowering the transmission range; therefore, the number of hops needed to cover the network increases, e.g. for 0 dBm of transmit power, only an average of 2 hops are needed to cover the network, while for -3 dBm, already 3 hops are needed. This increases to an average hop count of 14 when a transmit power level of -25 dBm is used, which reduces the number of nodes that are covered by one transmission. While at 0 dBm, at least half of the network was covered by one transmission, due to the larger transmission range, at -25 dBm, it is only a fraction of this coverage. This results in a higher latency. Lowering the transmit power level affects both latency and PLR. While lowering the transmit power level indicates a negative effect on the latency, it also reduces the number of concurrent transmissions which results in a lower PLR. At -3 dBm, an increase can be seen in a dense network, but for any transmit power level below -3 dBm, the PLR decreases to below 1%. Transmissions at -25 dBm represent a special case. Here, the nodes are barely in transmission range to one another, and this leads to a higher PLR for 150 and 200 nodes, which supports the initial statement that flooding may fail to reach all the nodes in sparse networks. However, the phenomenon of a decreasing PLR at a lower transmit power level is due to a change in the relationship between the power of the interference and the Clear Channel Assessment (CCA) threshold. The CCA fails when the sensed signal on the channel is at most 10 dB greater than the receiver sensitivity of the radio. If the CSMA performs the CCA in a setup of low transmit power, the power detected on the channel will more often than not be below the CCA threshold, which means that the CSMA will attempt to transmit instead of resuming the backoff. This increases the number of transmissions and leads to a lower number of messages being lost by having reached the backoff retry limit. In any case, the latency results presented as the mean of all latencies gives only a vague idea of when a node actually receives a message. Therefore, Figure 4.3 shows the latency result as an empirical Cumulative Distribution Function (CDF). This permits a deeper look into when a node actually receives a message. At 0 dBm, it takes approx. 40 ms on average to receive a message as shown in Figure 4.1, but in Figure 4.3 after only 40 ms, approx. 60% of the nodes had received the message. It requires another approx. 120 ms to reach the remaining 40%. When lowering the transmit power level, this effect worsens. While at -25 dBm, a message is received after 180 ms on average, the CDF shows that only approx. 50% actually receive the message within 180 ms. For a 99% to receive the message, it takes approx. 380 ms. This indicates that at higher transmit power levels, flooding might fulfil the requirements of delivering 99% of the messages within 200 ms; however, when the transmit power is below -3 dBm, the flooding fails.

4.4 Effects of varying back-off exponents

After discussing the impact of network density, as well as that of the transmit power on the network performance, this chapter discusses the effects of using different backoff retry parameters other than the default one. The latency and PLR results are shown in Figure 4.4. Of the backoff retry parameter, five different values are investigated, i.e. 1 to 5, plus one special case where each device during start-up randomly generates a backoff retry parameter value between 1 and 5, which is called *Random Backoff* for the remainder of this work. The distribution of the random backoff is based on a discrete uniform distribution. At a discrete uniform distribution, each value is equally likely to be observed with a probability of $p = \frac{1}{n}$, with n as the total number of possible values. In the case of the random backoff, p is equal to 0.2. For this analysis the node density is set to 400 nodes, and transmit power levels of -25 dBm to 0 dBm are used. The traffic is as described in Table 4.1 with a new command every 200 ms. It can be seen that the latency increases with an increasing backoff retry parameter. This is due to the increased contention and increasing probability that the channel is busy, e.g. with only one retry where a node tries only once to access the channel. Regarding this, the first node that tries to access the channel is the most likely to be successful, while among 400 nodes with 0 dBm of transmit power, several nodes try to access the



(a) Average latency with varying backoff parameter



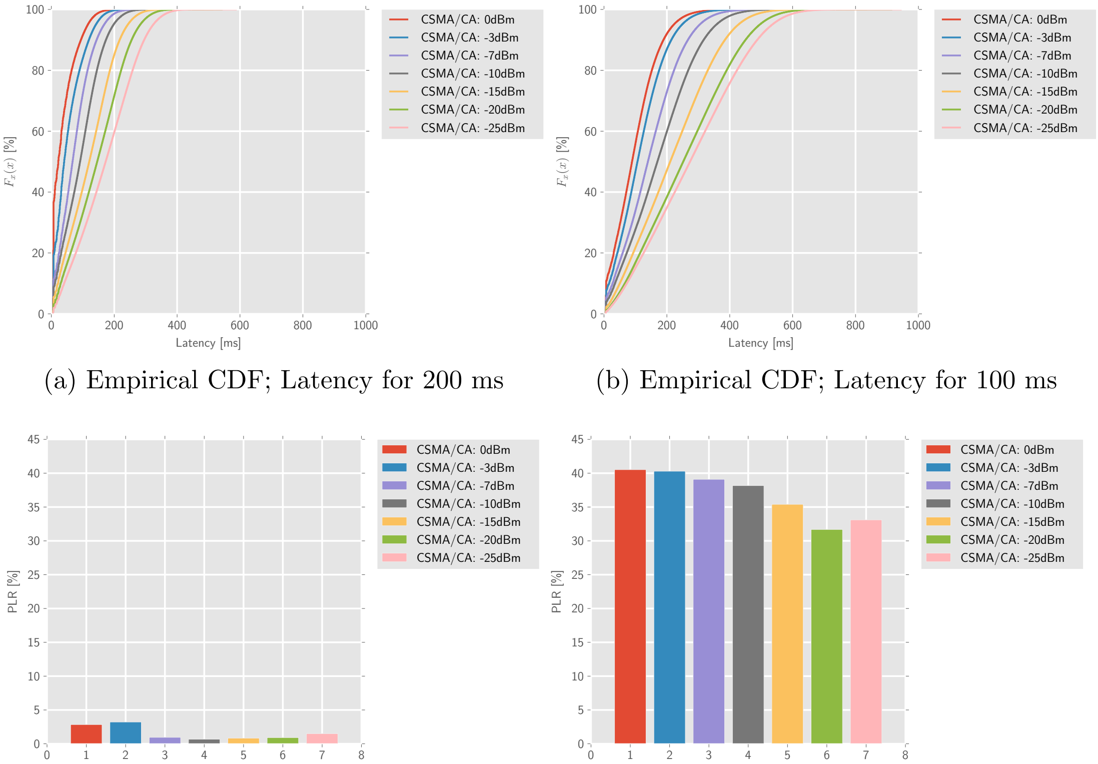
(b) Average PLR with varying backoff parameter

Figure 4.4: Network performance with varying backoff parameters

channel at the same time after receiving the initial message from the light switch. Therefore, multiple nodes would start a transmission which results in a higher packet loss rate at a higher transmission power. For -25 dBm of transmit power, the transmission range is smaller, hence, fewer nodes receive a message at the same time, which is directly related to fewer nodes trying to access the channel at the same time afterwards. This results in the PLR decreasing at decreasing transmission power. However, this explanation is only valid for one backoff retry. If the backoff retry is greater than one, nodes try to access the channel over a longer period of time, therefore, the transmissions occur more successfully. Hence, the PLR is lower, but the latency increases on average. Another special case is to set a random backoff retry parameter during an initialisation period. As the selection of the backoff retry parameter is discretely uniform, distributed over all nodes, a backoff retry parameter of an average of three should result. Using a random backoff parameter results in a latency and PLR that are both very similar to the results of using a backoff retry parameter of three. Only a small deviation can be observed; therefore, using a random backoff parameter has no positive effect on the network performance.

4.5 Effects of a shorter message inter-arrival ti-me

The impact of a shorter message inter-arrival time is discussed in this section. Lowering the message inter-arrival time is necessary for complying with wired digital dimmer units such as the *MDT Dimming Actuator* [97]. Such devices dim the light within 5 seconds, using approx. 50 commands. This results in a message inter-arrival time of 100 ms. Another reason for lowering the message inter-arrival is that with a higher number of messages, the overall process is smoother. Therefore, instead of generating 10 messages during the 2-second burst duration, 20 messages will now be generated. This means that a new control command will be transmitted every 100 ms while a burst is active. The simulation configuration is as follows, the network density is set to 400 nodes, the transmit power varies from -25 dBm up to 0 dBm, and the backoff retry parameter is set to 5. The results in Section 4.4 showed that 3 backoff retries or the random backoff mechanism achieve the best results; however, it is investigated here how the network performs with the default parameters, therefore the backoff retries remain at 5. Also, the results in Section 4.3 showed how an increasing network density affects the network performance. Therefore, the decision was made that it is sufficient to simulate all the remaining configurations of the highest density of 400 nodes only. For presenting the latency, an Empirical Distribution Function (EDF) is used. Using an EDF has the advantage of seeing how many lamps (in %) have received a message at time X or earlier. The results of the latency and PLR are shown in Figure 4.5 for a message inter-arrival time both 200 ms (Figures 4.5a and 4.5c) and 100 ms (Figures 4.5b and 4.5d). At 0dBm to -10 dBm of transmit power, the latency is below 200 ms for 95% of the messages received. As stated before, if the transmit power decreases, then the latency increases. For the transmit power from 0 dBm to -10 dBm, the latency is below 200 ms for more than 95% of the messages if the message inter-arrival time is 200 ms. While at a -15 dBm transmit power, 85% of the messages are received latest after 200 ms, and at -25 dBm, the percentage of messages received in time drops to 60%. Anyway, the result for a 200 ms message inter-arrival time shows that when using a higher transmit power, the requirements in terms of latency are fulfilled and the PLR barely exceeds the threshold of 1%. Now, the traffic is doubled during the 2 seconds burst, and this results in a drastic impact on both latency and PLR. At 0 dBm, the percentage of messages received in less than 200 ms decreases to 95%, and the PLR increases to over 40%. This effect can be seen for all transmit powers, while the worst impact can be seen at -25 dBm, where only 35% of the messages received have a latency of below 200ms while the PLR is still above 30%. With a 100 ms message, the inter-arrival time nodes have less time to access the channel for actually successfully transmitting the message. This means if an old message is still not sent when a new message is generated, the new message will end up in a queue until the old message is transmitted or dropped due to a used First-In First-Out (FIFO) queue. In some cases, this results in several new commands being stashed up in the queue before they can be sent. Additionally, this effect increases the probability of a failed CCA and with it, the probability of exceeding the backoff retry limit.



(a) Empirical CDF; Latency for 200 ms

(b) Empirical CDF; Latency for 100 ms

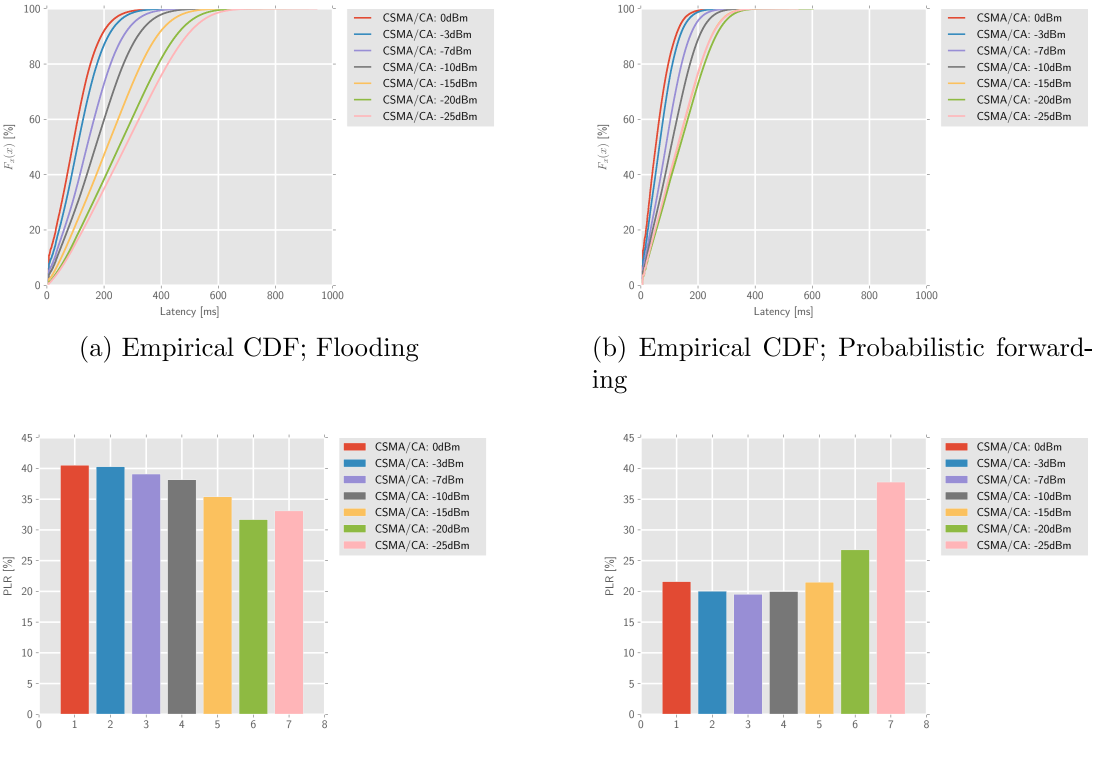
(c) Average PLR for 200 ms

(d) Average PLR for 100 ms

Figure 4.5: Network performance with varying message inter-arrival time

4.6 Flooding vs. probabilistic forwarding

The previous sections demonstrated that flooding in combination with a lower traffic can fulfil the requirements for wireless lighting control; however, as it is the goal to increase the traffic for a better user experience, the flooding failed. To test whether another easy-to-implement flooding-based mechanism is able to perform as expected with the high traffic rate, a probabilistic forwarding protocol was implemented. The results are reported in Figure 4.6. As stated in Section 4.2 approx. 35% of the receiving nodes forward a new message. If a node drops a message due to the failure to fulfil the requirement of forwarding a message, all duplicates will be dropped too. A duplicate is when a node receives the same message again from another forwarding node. The simulation configuration is as before, simulated of a 400-node network, the transmit power levels range from 0



(a) Empirical CDF; Flooding

(b) Empirical CDF; Probabilistic forwarding

(c) Average PLR; Flooding

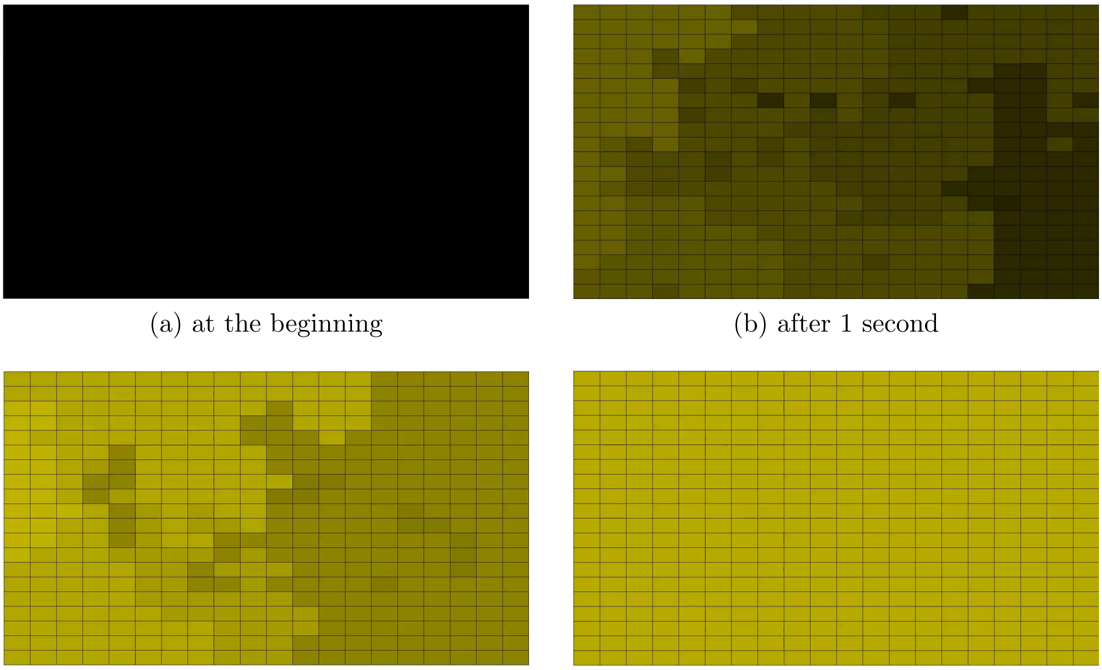
(d) Average PLR; Probabilistic forwarding

Figure 4.6: Network performance of flooding compared to probabilistic forwarding

dBm to -25 dBm, and the 100 ms traffic rate is used. The number of backoff retries remains unchanged at a maximum of 5. At first sight, probabilistic forwarding has a positive effect on both latency and PLR. At 0 dBm of transmit power, the latency is below 200 ms for 99% of the received messages, which is a 4% improvement. In terms of PLR, a significant improvement of almost 20% can be seen; however, the PLR is still 22% for 0 dBm of transmit power. This effect can be seen for all transmit powers, except for -25 dBm. At -25 dBm of transmit power, the latency improves even more while with plain flooding, approx. 35% of the messages are received in 200 ms or less; with probabilistic forwarding, this improves to almost 80%. While the latency improves significantly, the PLR worsens by a few percent. This is due to the low transmission level of -25 dBm. Using probabilistic forwarding, only a few nodes forward the received message. If the number of forwarding nodes is then reduced further, the probability of a message getting lost increases, especially in the areas that are further away from the initiating node. In general, probabilistic forwarding outperforms flooding in terms of latency and PLR as long as the transmit power is greater than -25 dBm; however, none of the two mechanisms is able to fulfil the requirements for a wireless lighting control network.

4.7 Effects of a broadcast storm on the user perception

As the previous results are only related to the network performance itself, this section focuses on how a user would experience such a network performance; in other words, what impact does flooding or probabilistic forwarding have on the lighting level itself. To visualize the effects of a broadcast storm, a single burst of 20 commands was simulated, and each node recorded exactly when each command was received for the first time. Simulated here was a dimming application with a user turning up the light level from 0% to 100%. Theoretically, each command increases the light level by 5%; however, each command also includes the current light level. By this, it is possible that a node misses a few messages but still receives the final command changing to the correct light level. Both flooding


(a) at the beginning

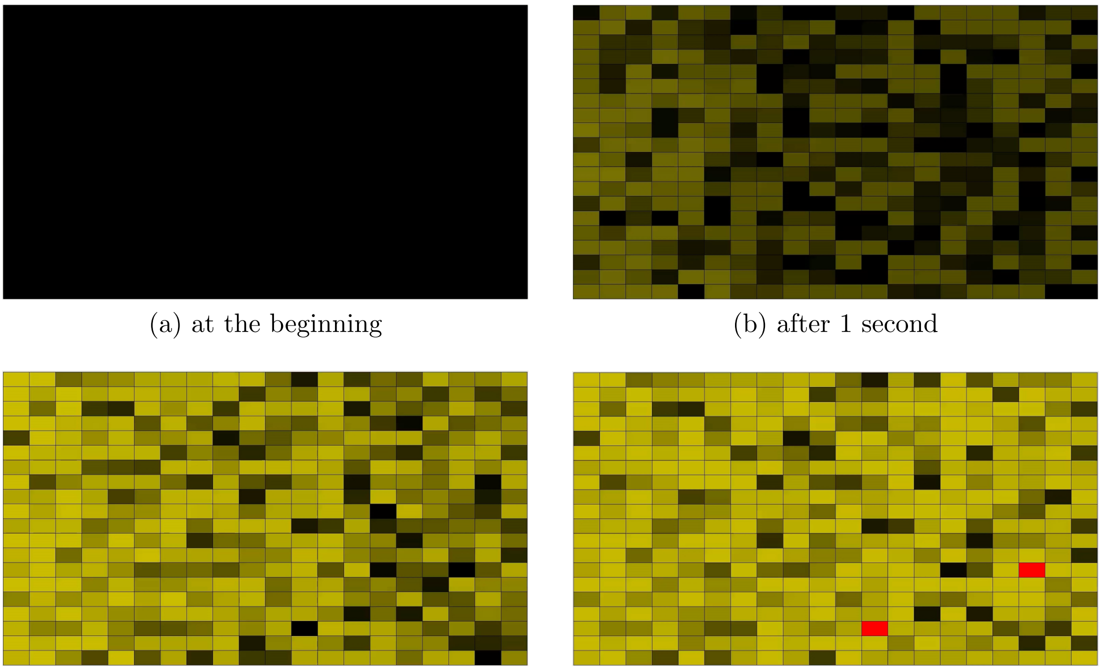
(b) after 1 second

(c) after 2 seconds

(d) at the end

Figure 4.7: Message dissemination with flooding for a dimming application [33]

and probabilistic forwarding are visualised. The effects of flooding on a lighting network are shown visually in Figure 4.7 and for probabilistic forwarding, in Figure 4.8. Here, a black colour indicates a 0% light level (off), and yellow indicates a 100% light level. A red square is used to indicate a node that never received a single message during the 2s burst. For both, the network density is set to 400 nodes, the transmit power is at -25dBm, the backoff retry parameter is set to 5, and the light switch is located at the left centre of the room. In the flooding, it is visible how the light control messages travel through the room. After one second (Figure 4.7b), so far, five new commands were generated by the light switch and were sent out; however, it is visible that each of these messages is still propagating throughout the network. This is indicated by the five different light levels visible at that time. After two seconds (Figure 4.7c), the last command is generated by the light switch, but at that time, several messages are still propagating throughout the network. The earlier results indicated that with the flooding in this setup, there was approx. 33% PLR; however, the visualisation



(a) at the beginning

(b) after 1 second

(c) after 2 seconds

(d) at the end

Figure 4.8: Message dissemination with probabilistic forwarding for a dimming application [33]

shows that although there are messages lost during the dimming process, the final message is received by all lamps, resulting in the same light level all over (Figure 4.7d). Using a probabilistic forwarding where fewer nodes forward a message, it is visible that this has a high impact on the system behaviour. Already after one second (Figure 4.8b), the lamps are all in different states. This behaviour is visible throughout the whole dimming process, and at the end, too, all lamps are still in different states. Notably, the visualisation shows that two lamps have never received a single message during the entire dimming process; these are marked red in Figure 4.8d. In lighting systems, if a lamp remains dark after switching on the light, a user assumes that the lamp is broken, which is unacceptable.

4.8 Method to ease a broadcast storm

In this section it is discussed whether a simple method that deletes existing messages from a node's message queue when newer messages arrive, can prevent a broadcast storm or can at least improve message delay and delivery performance. The discussion is based on the assumption that in a dimming process new control commands are generated every 100ms, as discussed during the previous sections. Before it is possible to discuss if this would improve the performance, it is necessary to take a deeper look on what actually happens when a message is transmitted using a flooding mechanism on top of the IEEE802.15.4 non-beacon CSMA/CA mode. It is assumed that a message is created by the application layer and handed over to the MAC layer to actually transmit the message to its destination. The MAC layer which has a packet ready for transmission first backs off for a random number of backoff slots, chosen uniformly between 0 and $2^{BE} - 1$, before sensing the channel, where the parameter BE is the backoff exponent which is initially set to 3. A backoff slot is defined by 20 symbols which results in a time of 320 μs [68]. Based on these numbers, a node always waits between 0 and $(2^3 - 1) \times 320$ μs before transmitting a message, resulting in max. 2.24 ms delay. After this backoff time a node tries to access the channel. If the channel was found idle the message will be sent, if the channel was found busy a node repeats the backoff process up to five times before declaring a channel access failure and deleting a message from the queue. In addition to trying to access the channel five times, BE is incremented each time as long as it is smaller than maxBE. By default maxBE is set to 5. For each of the transmission attempts, a message remains in the queue for the following maximum durations:

- **1. Backoff retry** $(2^3 - 1) \times 320\ \mu s = 2.24\ ms$
- **2. Backoff retry** $(2^4 - 1) \times 320\ \mu s = 4.8\ ms$
- **3. Backoff retry** $(2^5 - 1) \times 320\ \mu s = 9.92\ ms$
- **4. Backoff retry** $(2^5 - 1) \times 320\ \mu s = 9.92\ ms$
- **5. Backoff retry** $(2^5 - 1) \times 320\ \mu s = 9.92\ ms$

This results in a total maximum of 36.8 ms that a message remains in the queue before it is deleted due to channel access failures. Now, comparing the 36.8 ms with the traffic rate of a new message every 100 ms shows that whenever the application in the dimming process creates a new message the old message is either already sent or deleted due to the channel access failure. Hence, the application can not delete old messages which were created based on the dimming process. Therefore, no improvement of Flooding nor Probabilistic Forwarding is expected.

4.9 Conclusion

This chapter has analysed and discussed the effects of a broadcast storm on a wireless lighting control network from different aspects. First, the network performance with an increasing network density was presented. Flooding might perform very well in sparse networks, however, when the network reaches a certain density, the PLR increases above the threshold for lighting networks, even though the latency remains acceptable. At the same time, simple methods were used to try and improve the network performance without changing the overall protocol algorithm. For instance, lowering the transmit power has had a negative effect on the network performance for both latency and PLR. When using another backoff retry parameter, lower than the default, an improvement can be seen when using retry parameters greater than 1 and smaller than 5. Using another retry parameter affects the PLR, and only a slightly higher impact on the latency was observed. Besides analysing the network performance with a state-of-the-art traffic model, the performance was also investigated using a higher traffic rate. This showed that an unmodified flooding is not able to deliver good performance; therefore, a modified version of flooding was used, called probabilistic forwarding. This improved the network performance in terms of latency and PLR; however, also probabilistic forwarding was not able to meet the requirements of a lighting control network. The last contribution in this chapter is the visualisation of a dimming process that uses flooding and probabilistic forwarding. It was shown that with flooding, a consistent light level is reached at the end of the dimming process, even though there are many packets lost during the process. However, the side effects of flooding were visible, i.e. that dimming the light had happened in waves, which were clearly visible. On the other hand, probabilistic forwarding, which had performed better when purely based on network performance, shows that it is not at all suitable for a lighting application since a consistent light level could not be reached, and some lamps do not receive any messages at all. In summary, the results definitely show that state-of-the-art communication protocols do not fulfil the requirements of a lighting use case where high reliability and low latency are required.

Chapter 5

The D^3LC Suite

Chapter 4 showed that the control of a large number of luminaires in a dense environment is challenging and that the state-of-the-art protocols fail to fulfil the requirements of a wireless lighting control network. These challenges have been addressed by the D^3LC Suite. The D^3LC Suite incorporates a clustering protocol, referred to as CIDER (Section 5.2), a data dissemination protocol, called RLL (Section 5.4), and a channel offset allocation protocol that applies a greedy colouring algorithm (Section 5.3). The D^3LC Suite is designed for Dense Wireless Lighting Control Networks (DWLCN). The D^3LC Suite is a collection of OSI-Layer-3 protocols (Network Layer) and works on top of the IEEE802.15.4 TSCH mode (see Figure 5.1). The rationale for using the IEEE802.15.4 TSCH mode instead of the CSMA MAC protocol used earlier is discussed in detail in Chapter 5.1. The D^3LC Suite increases the scalability of wireless lighting networks by using a weighted clustering method designed for static and mains-powered WSAN, as well as disseminating lighting control messages by deterministic arrival at the destinations. Finally, it allocates channels to enable a simultaneous transmission without mutual interference.

The key performance attributes of the D^3LC Suite are as follows:

- Low-latency data dissemination in dense networks
- Provides a high message delivery rate in low-interference environments
- Enables a deterministic data reception

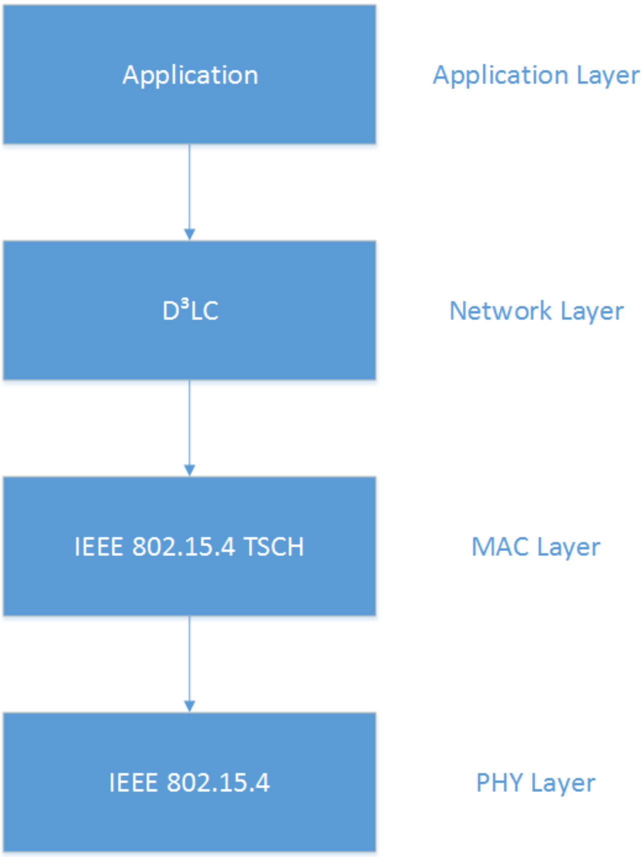


Figure 5.1: Overview D^3LC Suite

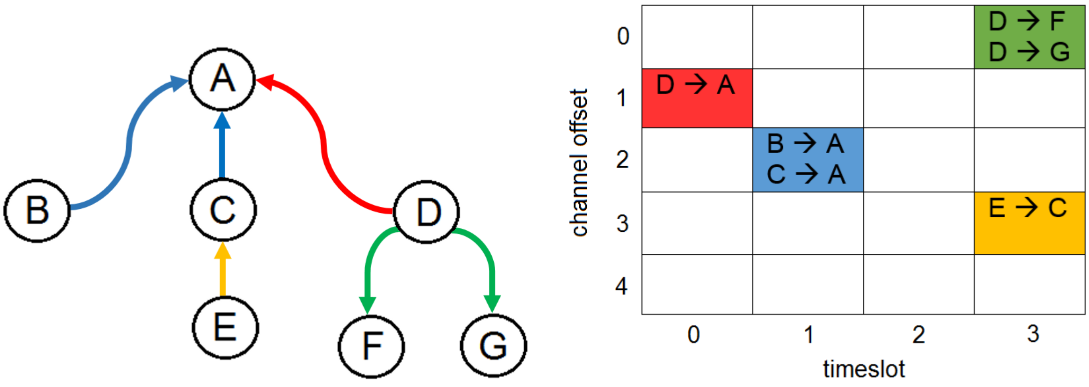
- Significant decrease in redundant message transmissions
- Minimising inter-cluster interference
- Increase in the scalability of wireless sensor networks

The following section describes the TSCH-Mode. The remainder of this chapter focuses on the technical description of the D^3LC Suite.

5.1 IEEE802.15.4 - Overview of Time-Slotted Channel Hopping

In 2012, the IEEE802.15.4e standard [69] was released, which is an amendment of the IEEE802.15.4:2011 standard [68], It introduced the new MAC operation modes: Radio Frequency Identification Blink (BLINK), Asynchronous Multi-Channel Adaption (AMCA), Deterministic and Synchronous Multi-Channel Extension (DSME), Low-Latency Deterministic Network (LLDN), and TSCH. This extension was subsequently incorporated into the IEEE802.15.4 standard in 2015 [70]. The new operating modes target problems such as timeliness, reliability, robustness, scalability, and flexibility. TSCH was selected as the underlying MAC protocol for a number of reasons. A big advantage when compared to CSMA is that with TSCH, all 16 channels of the IEEE802.15.4 standard can be used, while CSMA is bound to one channel only. This enables creating multiple transmissions simultaneously using different channels. Furthermore, due to the TDMA approach of TSCH, all nodes follow a communication schedule, and this can be used to create a controlled message flow throughout the network with no interference from other nodes in the network. However, in this thesis TSCH as a TDMA protocol is not investigated in combination with flooding. Combining an unmodified flooding protocol with TDMA would raise similar problems. Assuming that nodes receive a message in one slot, and try to forward in the next one following. If multiple nodes receive a message, they all try to forward at the same time, which leads to the same problems as with a CSMA protocol. While the study of flooding in combination with TDMA is not the objective of this work, a TDMA flooding protocol is compared with the proposed protocol of this work.

As the protocols introduced in this chapter operate on top of the TSCH mode, a brief overview of the TSCH operation is provided in this section as it is defined by the standard. TSCH is a TDMA-based approach where a schedule regulates the channel access. Based on the schedule, it is possible to control the message flow. Nonetheless, in general, a schedule can be created in a distributed manner, or by a central device. IEEE802.15.4 does not provide a mechanism that creates a schedule or organises the network, the standard only provides the rules for how



channel offset	0	1	2	3
0				D → F D → G
1	D → A			
2		B → A C → A		
3				E → C
4				

Figure 5.2: Simple TSCH schedule [103]

the MAC layer should operate. A simple schedule is shown in Figure 5.2. In this network node, *A* acts as timekeeper, to which all nodes are synchronised, but not necessarily directly. When the nodes are synchronised, they follow a slotframe which repeats over time (Figure 5.3). Each timeslot allows a node to transmit a maximum-sized packet. One of the changes of the TSCH mode, compared to the IEEE802.15.4:2011 standard, is the channel-hopping operation, where the nodes are able to utilise up to 16 channels. Due to the capability to change the channel immediately, it is possible that multiple nodes communicate with different destinations within the same timeslot without interfering with each other. E.g. in Timeslot 3 of Figure 5.2, Nodes F and G are listening for data from Node D. In Timeslot 1, Node A listens for data from node B or C; however, here it is possible that both nodes are attempting to transmit. This might result in a collision. Here, a backoff mechanism similar to the one of CSMA/CA is applied. A node waits for a number of timeslots before trying to transmit again. The channel selection is based on Equation 5.1 where f is the resulting channel frequency, F is a lookup table that contains all of the available channels in the network, Absolute Slot Number (ASN) is the absolute slot number (total number of timeslots elapsed since the start of the network), and *NumberChannels* is the number of channels available in the network.

$$f = F[(ASN + ChannelOffset)\ mod\ NumberChannels] \tag{5.1}$$

The channel offset is set by the schedule to be always the same value for one link, e.g. in Figure 5.2, for timeslot 0, link D to A, the channel offset is 1. The time-slotted approach of TSCH increases the potential throughput by reducing the collision probability amongst neighbouring nodes. By the channel-hopping operation, the interference of neighbouring nodes is reduced, and the use of multiple channels allows more nodes to transmit at the same time. Hence, TSCH improves the overall communication reliability. Since it is possible to select the channels on which TSCH is communicating it is possible to only select channels which do not overlap with active WiFi networks to increase the robustness against WiFi interference.

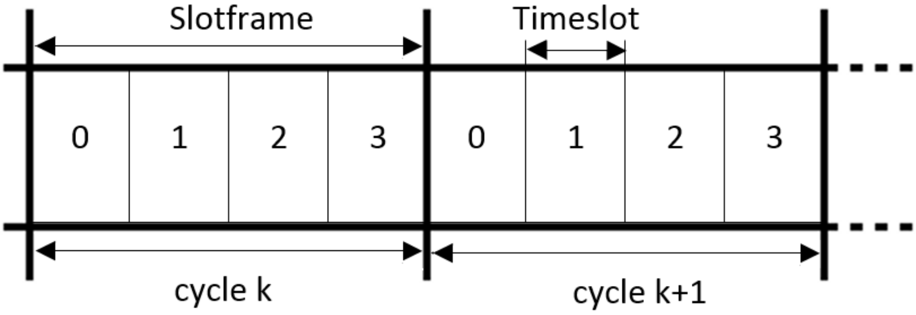


Figure 5.3: Overview of slotframe [103]

5.1.1 Slotframe structure for D^3LC

IEEE802.15.4 does not provide any mechanism for creating this slotframe or organise the network; the standard only provides the rules for how the MAC layer should operate. Therefore, as part of this work, mechanisms were devised to define the slotframe on each node in a distributed way with respect to the state-of-the-art approaches which are discussed in Section 5.3.1. In general, a slotframe contains at least one timeslot for synchronisation, a so-called beacon slot, and multiple data slots. By definition, the maximum number of timeslots allocated to one slotframe is 65536. During the beacon slot, the timekeeper periodically transmits a beacon message. The beacon message includes information to keep the network synchronised based on the standard definition [70]. The standard also defines the options on how a data slot can be used. These options are: transmit, receive, shared, or timekeeping [70]. When a data slot is defined as a transmit slot, a node will check whether data is available to transmit, otherwise, the node goes back to sleep. With the receive option set, a node will listen for incoming data. The timekeeping option is used to indicate that data received during this timeslot contains synchronisation information. These three options are used for dedicated operations, i.e. either transmitting or receiving. With the shared option set to a link, a node will first check whether it has data to transmit, otherwise the node listens for incoming data. For D^3LC, two different slotframe structures are used, one for the network formation process (CIDER and Colouring) and one for the data dissemination process (RLL). The basic structure is the same for both, only the usage of the different timeslots will change for RLL. In this section, the basic structure and the usage for CIDER and Colouring are explained. The slotframe follows the recommendation of the standard and consists of one beacon slot and several data slots, using Equation 5.2.

$$SF = \underbrace{1}_{\text{Beacon Slots}} + \underbrace{10 \times R}_{\text{Data Slots}};\ R \in \mathbb{N}\{1,2,..,10\} \tag{5.2}$$

To keep the network well synchronised and to minimise the possibility of missing a beacon, the maximum slotframe length is limited to one beacon slot plus 100 data slots for D^3LC. For experiments conducted in this work, 50 data slots are used, therefore R is set to five. Fifty slots were selected to keep the network well synchronised, due to the long runtime of the experiment and to avoid that an experiment fails. When using 100 data slots, it was found in early experiments that a synchronisation loss is more likely than when using 50 data slots. For CIDER and Colouring, the data slots are used with the shared option. In this way, it is possible that all nodes always listen for incoming data; however, as all nodes have also the chance to transmit in each timeslot, mechanisms are needed to minimise the possible collisions. For this, each node delays a transmission based on its 16-bit link address by applying Equation 5.4. Each device has a unique link address in the range of $0x1$ to $0xFFFF$, the 12 Least Significant Bits (LSB) result in the delay. The link address is provided by the operating system Contiki-OS. The delay is multiplied by a time constant which results in T_{Delay}, which is in the range of 1 ms to 4095 ms.

$$T_{Delay} = \left(\text{Link Address } \& \ \underbrace{0xFFF}_{\text{12 LSB'}}\right) \times \underbrace{1\,ms}_{\text{Time constant}} \tag{5.3}$$

$$ts_{Delay} = \frac{T_{Delay}}{10\,ms} \tag{5.4}$$

$$ts_{send} = \text{ASN} + ts_{Delay} \tag{5.5}$$

With Contiki, it is possible to assign packets to specific timeslots; for this, it is necessary to map the calculated delay on a delay expressed in timeslots. Therefore, Equation 5.4 is used, where 10 ms is the duration of one timeslot. ts_{Delay} combined with the ASN results in ts_{send} which defines the timeslot in which the message will be sent. Using this mechanism, the possibility of a message collision of two transmitting nodes is minimised. The slotframe structure introduced here is only used for CIDER and Colouring. While the slotframe length remains the same for RLL, the timeslot operation changes and is explained in Section 5.4.2.2.

5.2 CIDER – Clustering In Dense EnviRonments

Wireless lighting control networks are a special type of wireless networks where wireless nodes are connected to mains power. Hence, energy consumption and preservation are not the main concern when designing communication protocols. In a typical office environment, the lamps are often deployed in a regular grid where the distances between individual lamps are the same. Office lighting installations are deployed with densities ranging from 0.17 $\frac{lamps}{m^2}$ to 0.63 $\frac{lamps}{m^2}$ [89] but can easily reach higher densities in apartments. Several requirements must be fulfilled to form an efficient network topology for wireless lighting control networks. As traffic is based on broadcast transmissions which easily result in broadcast storms, the number of forwarding nodes must be reduced to a minimum. Forwarding nodes should always cover as many nodes as possible, and new forwarding nodes should only be added when needed. Based on these preconditions, a cluster-tree topology is used where the cluster heads (CH) act as forwarding nodes, and the top tier CH has the highest coverage area. To form such a topology, many clustering approaches are possible; however, usually, clustering protocols are aimed at *energy preservation or load balancing* and also use these parameters for selecting the CHs. In lighting networks, such parameters have to be seen as critical because the nodes will not run out of power as they are mains powered. Therefore, this section introduces a weight-based clustering protocol for dense environments, named CIDER. CIDER is designed for indoor lighting networks and forms an efficient network topology based on a cluster tree. The remainder of this section is as follows, first, Section 5.2.1 gives an overview of the state-of-the-art clustering protocols. In Section 5.2.2, the general operation of CIDER is described followed by a detailed description of the three CIDER phases in Sections 5.2.3 to 5.2.5. Section B shows how messages should be designed using CIDER.

5.2.1 State-of-the-art of clustering in WSNs

Many algorithms have been proposed that form networks into clusters to increase the network performance, where the focus was on energy-efficient operation to extend the network lifetime. Forming clusters is a non-trivial problem where the main problem is to elect the right cluster head (CH) in a self-organizing fashion. The methods used to elect CHs can be divided into three groups: *preset, random*, and *attribute-based* [6]. While presetting the CH is not an option for a self-organising network, this section will give a general overview of the most common random and attribute-based methods for electing the CH and forming a network topology. In [61], Heinzelman et al.. introduce a Low-Energy Adaptive Clustering Hierarchy (LEACH). LEACH is a very simple algorithm for clustering, which requires only a low overhead to form clusters. The CH election in LEACH is random. Each node generates a random number between 0 and 1 and compares it to Equation 5.6:

$$T(n) = \begin{cases} \frac{p}{1-p \times \left(r \bmod \frac{1}{p}\right)} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

where G is the set of nodes that have not been a CH in the last $\frac{1}{\mathrm{p}}$ rounds, p the desirable percentage of CHs, and r the current round. Each node in the set G compares its number with *T(n)*. If a node's number is smaller, it declares itself as CH and broadcasts its new state to all nodes within transmission range. This algorithm ensures that every node becomes a CH at least once in a predetermined epoch. The length of an epoch depends on the number of nodes in the network and on the number of clusters ($epoch = \frac{\mathrm{n}}{\mathrm{k}}$, where n is the number of nodes in the network and k the number of clusters in the network). However, besides the advantages mentioned earlier, LEACH has some drawbacks, e.g., the communication between the CH and the Base Station (BS) is only single-hop. That means that the CHs further away from the BS consume more energy to deliver the message. Also, LEACH requires that all nodes are in transmission range of the base station since multi-hop communication is not supported by LEACH. Because the CH election is random, the CH distribution in the network is not necessarily ideal as the energy of the CH is not considered, and nodes with a low energy might get elected as CH and might die during their round. The authors state that LEACH can be easily extended to a hierarchical clustering algorithm, but they do not describe what is required for such an extension. Another approach for electing the CH is Hybrid Energy-Efficient Distributed clustering (HEED) [142]. HEED is a hybrid clustering algorithm which takes the residual energy and the communication cost, such as AMRP (minimum power required by a node to communicate with its CH) or node degree, into account for electing the CH. In HEED, the CH election is divided into three phases: initialization, main processing, and finalization. First, each node computes a probability based on:

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}} \tag{5.7}$$

where C_{prob} is an initial percentage of CH's in the network and $E_{residual}$ and E_{max} the current and the initial energy, respectively. If the computed probability is 1, the node elects itself as a final CH; otherwise, the node only becomes a CH if there is not another low-cost CH in the neighbourhood. All non-CH nodes then select the CH with the lowest communication cost, calculated either by AMRP or based on the node degree. The distribution of the CH's across the networks is properly done, and CH to CH communication is possible. LEACH and HEED are the two fundamental clustering methods for WSNs, and many clustering algorithms are based on these two, e.g. [7,39,59,63,77,124]. The authors of [7] introduce the Threshold-Sensitive Energy-Efficient Sensor Network (TEEN) protocol. TEEN operates using LEACH as the underlying clustering method; however, TEEN improves LEACH in terms of energy efficiency as the energy consumption of a wireless node highly depends on how often a node transmits or receives data. Hence, the authors introduce thresholds for when a node should transmit data back to the base station. Two thresholds are used, a *hard threshold* H_T and a *soft threshold* S_T. The nodes sense the environment continuously. A node reports the value of the sensed data the first time the data reaches H_T. For instance, if the temperature rises over $H_T = 10C$ for first time, the temperature sensed is reported. This value is stored by a node. Whenever the value changes by at least S_T but the value remains greater than H_T, then a node reports the changed value. Using these simple rules, the network lifetime is improved immensely, e.g., using an initial energy of 2 J, half of the nodes died after 400 s with LEACH, while with TEEN half of the nodes lasted for approx. 1000 s.

Heinzelmann et al. propose as a LEACH extension, LEACH-C [59]. While LEACH is a distributed clustering protocol and there are no guarantees about CH placement as well as the number of CH nodes, aims at a better distribution of CH nodes throughout the network by using a centralised CH selection algorithm. Using LEACH-C, all nodes transmit their location and energy level to the base station during the setup phase. The authors recommend using a GPS module to determine the location; however, it is stated that also other methods can be used and should be selected based on the use case that LEACH-C will be used for. The base station then selects the CHs from a set of nodes, in which the remaining energy is above a pre-set energy threshold, by using the simulated annealing algorithm. In applying this algorithm, the energy usage of the non-CH nodes is minimised. As soon as the BS has selected all the CHs needed to cover the network, a broadcast message is transmitted, including all CH IDs. This approach results in an improvement of 40% in terms of data delivered per unit energy compared to LEACH.

Kang et al. propose an improved LEACH protocol that aims at an even distribution of CH within the network [77]. The basic approach follows LEACH, and it is assumed that LEACH uses the aforementioned hierarchy approach. By this, the CHs closer to the BS are placed at a higher level in the hierarchy. For this, the authors divide the network into areas based on the distance to the base station. The basic approach follows LEACH where each node elects itself as a CH if the initial condition of LEACH is fulfilled. During the setup phase, each CH sends an advertisement message to all its neighbours after it declares itself a CH. If other CHs in the same region receive this message, they will return into ordinary nodes and select a CH later; however, it is not further described how this “Competition Mode” works, only that CHs transmit the advertisement message, and some CHs give up their candidacy. The proposed method increases the total system lifetime by more than 1000 rounds, while a ‘round’ is defined based on the network size and the number of clusters. Also, the number of CHs is reduced; with LEACH approx. 19 CHs were needed to cover the network, while with the new method, only 13 CHs are required.

In [39], the Distributed Weight-Based Energy-Efficient Hierarchical Clustering

(DWEHC) protocol is proposed to improve the energy efficiency and network lifetime. DWEHC aims at creating well-balanced clusters in the network. DWEHC is a so-called a *weight-based* clustering approach where a node becomes a CH when it has the highest weight in its neighbourhood. Each node calculates its weight using Equation 5.8, where $w(s)$ represents the weight for node s, $E_{residual}\left(s\right)$ and $E_{initial}\left(s\right)$ is the residual energy and initial energy of node s, respectively. R is the predefined cluster range and d is the distance between node s and a neighbouring node u. $N_{a,c}\left(s\right)$ is the set of neighbouring nodes of node s.

$$w\left(s\right) = \left[\sum_{u \,\in\, N_{a,c}(s)} \frac{(R-d)}{6R}\right] \times \frac{E_{residual}\left(s\right)}{E_{initial}\left(s\right)} \tag{5.8}$$

A node will update its neighbourhood after a calculation. If a node has been a CH during the previous 20 rounds, it will not become a CH. All nodes not becoming a CH will then select the node that has the highest weight in the neighbourhood as the CH. In DWEHC, the overall cluster distribution has improved compared to HEED. With a cluster range of 100 m, DWEHC reduces the number of active clusters from 203 (with HEED) to 125; however, this changes when the cluster range increases. If a cluster range is 350 m, HEED forms 27 clusters while DWEHC forms 45. This is due to a limitation of DWEHC in terms of the total number of cluster slaves one CH can support. It is not stated to how many nodes this is limited. The authors achieve a better load balancing with DWEHC as only CHs transmit the data to the base station, this mechanism also increases the network lifetime by approx. 24% compared to HEED.

Another HEED extension is proposed by Huang et al. [63], called *Extended HEED*. The authors apply a so-called CORE algorithm proposed by the authors of [118]. When using this, a set of nodes is selected from which CHs are elected. While the original HEED approach considers all nodes for the CH election process, Extended HEED uses only this subset, ensuring that only nodes which are very cost efficient are considered. Simulation results show that the new approach requires less CHs; hence, the network lifetime is improved. While HEED did not perform well in sparse networks, Extended Heed solves this problem by a better CH selection process.

Taheri et al. propose an Energy-Aware Distributed Dynamic Clustering Protocol Using Fuzzy logic (ECPF) [124]. The key features of ECPF are a distributed CH election which only relies on local information to avoid extra communication costs with the base station. CHs are elected in two steps, first, a set of tentative CHs are selected based on a non-probabilistic election phase, followed by using fuzzy logic to determine the fitness of a node for becoming a final CH. A major difference to other protocols is that the clustering process is not round-based; the ECPF clusters are formed on demand or when a CH depletes a given fraction of its energy resources. ECPF is based on HEED; hence, the steady-state-phase follows the same procedure. In the beginning, neighbourhood information is exchanged using broadcast messages. Based on the information gathered, each node can calculate its cost by using fuzzy logic; the fuzzy logic algorithm takes into account the node degree and the centrality of a node. Then each node calculates a wait time based on its remaining energy. After the waiting period is over, a node declares itself as a tentative CH by broadcasting its decision, including the cost calculated. In this way, only nodes with a high energy might become a CH because the wait time is shorter for nodes with a high energy remaining. Also, a node still waiting for the timer to expire, which receives such a message from its neighbours, will no longer declare itself as a tentative CH. In the next step, a tentative CH with the highest cost amongst its neighbouring tentative CHs will declare itself as a final CH and broadcasts this decision. All other tentative CHs receiving this message will give up their candidacy. In this way, the protocol ensures that there are no redundant CHs. During the final step, all nodes that are not a final CH select as their parent the final CH with the least cost. Any node that did not receive any message from a final CH will declare itself as a final CH. ECPF is compared with both LEACH and HEED; however, the results show that ECPF outperforms both LEACH and HEED. Not only is ECPF more energy-efficient, but also the network lifetime is immensely improved, and the number of CH elections are reduced to a minimum since the election process is only triggered when a CH runs below a certain energy level.

The work of Chatterjee et al. [26] introduces the Weighted Clustering Algorithm (WCA). WCA is an on-demand clustering protocol. The aim of WCA is to keep the topology stable as long as possible to reduce the computation and communication costs. WCA effectively combines multiple parameters with the weighting factors to form a stable network topology. However, as a basis for their algorithm, the authors consider the battery power of a node and the mobility of a node following a pre-defined ideal node degree. The ideal node degree is used to limit the number of nodes that a CH can handle. In this way, in accessing the channel using TDMA, unnecessary delays are avoided. The battery power is used to ensure that only nodes with a high remaining energy are used as CH's. An important factor for WCA is the mobility, where only nodes with low velocity should be elected as CH to avoid too frequent CH changes. To keep the topology as long as possible, the algorithm is not periodic and is only triggered when the relative distances between CHs change. The clustering procedure is structured in eight steps, first, each node needs to discover its neighbourhood (defined as neighbourhood degree d_v) and then compute the difference between the pre-defined ideal degree δ and d_v, stored as Δ_v. Third, each node calculates the overall distance to all its neighbours, defined as D_v. In Step 4, a node calculates its movement within a time frame to determine its mobility M_v. Next, each node computes the cumulative time P_v during which a node acts as a CH. P_v implies how much battery power has been consumed. In Step 6, a combined weight W_v is calculated and represents the suitability of a node to serve as CH. W_v is defined in Equation 5.9.

$$W_v \; = \; w_1\Delta_v \; + \; w_2D_v \; + \; w_3M_v \; + \; w_4P_v \tag{5.9}$$

During Step 7, the node with the smallest W_v value will be CH, and all nodes in its neighbourhood are no longer participating in the clustering process. All nodes not assigned to a CH or selected as a CH will repeat steps 2-7 until the whole network is covered. The first component of Equation 5.9, $w_1\Delta_v$ is used as an indication of an efficient MAC functioning. It is desirable for a CH to handle only up to a certain number of nodes in its cluster. The part including D_v is related to the energy consumption. D_v represents the overall distance to all neighbours; hence, it is assumed with a higher value of D_v, a higher energy is needed for communication with the neighbourhood. The third component, $w_3 M_v$, represents the mobility of a node, where a node moving slower or not at all is more suitable to be CH. The last component, P_v, is the total time that a node has acted as a CH. The authors assume that all nodes have the same battery power at the beginning. However, if a node acts as a CH, the battery drainage will be higher compared to a node being a CS. Therefore, P_v is used as an indication of how much battery power is left in a node's battery. The authors define the weighting factors as follows: $w_1 = 0.7$; $w_2 = 0.2$; $w_3 = 0.05$; $w_4 = 0.05$. The authors state that $w_{1..4}$ are chosen arbitrarily so that $w_1 + w_2 + w_3 + w_4 = 1$. It is also stated that the weighting factors should be tuned for each use case individually. WCA was simulated with a network size of up to 60 nodes and was compared to different heuristic clustering methods such as *Lowest-ID*, *Highest-degree*, or *Node-Weight* clustering protocols. WCA is designed as a distributed clustering algorithm which is able to adapt dynamically to the changing environment of a mobile adhoc network but could be easily adapted to other areas such as lighting control.

Apart from the mentioned probability and attribute-based clustering algorithms, other clustering proposals have been inspired by nature, e.g. the *Bee-Colony-inspired Backbone selection algorithm* (BEES) [4]. In BEES, the network is formed into a *"beehive"*-like structure. To form a network with BEES, it is required that the BS is located in the centre of the network and is equipped with an omnidirectional and a directional antenna; further, all other nodes must be equipped with GPS-modules. First, the BS divides its surrounding area into six hexagonal sectors, the so-called beehives. A sector is assigned from the BS using the directional antenna; a signal is sent in six directions, each 60 degrees apart. Using the exact position of each node, these sectors are again divided into rows, in which the first row consists of one hexagon, the second row of two hexagons, and so on. After this, a coordinate system is applied to identify each hexagon in the network. The next step is to select the CHs for each sector. The BS selects the first six CHs in the closest sectors; however, the requirement is that the CHs must be located closest to the centre of the sector in which the node is located. After the first six CHs are selected, these six CHs select the CHs for the next rows. This procedure continues until all sectors are assigned with a CH. BEES is a novel protocol and was proposed to show how clustering could use ideas from nature to solve problems such as localization, data aggregation, or CH election. However, since BEES requires two types of antennas (directional and omnidirectional) as well as GPS modules to determine the exact position, it is unsuitable for lighting applications.

Summary This section shows that clustering can be used to address many different problems, such as increasing the network lifetime by rotating the role of nodes acting as a CH. Each protocol is tailored to a specific use case, using a different approach or an already existing protocol as a foundation. However, all protocols presented here aim at the same goal, to improve the network lifetime and to keep nodes alive for as long as possible. Clustering protocols are usually designed for battery-powered WSNs; therefore, it is important to conserve as much energy as possible. Besides the LEACH protocol, each protocol includes an energy part to decide about CHs. Wireless Lighting Control Networks do not share this characteristic since wireless transceivers are included in the light bulbs themselves, hence, a steady power connection is always available. Due to this limitation of current clustering protocols being designed to increase the energy efficiency of individual nodes and hence, increasing the network lifetime, these protocols can not be used for lighting applications without modification. For lighting applications parameters such as the battery level are not available in mains-powered lighting networks. Therefore, it is necessary to investigate which values can be used from existing protocols to design a clustering protocol suitable for lighting applications. A set of parameters could be the node degree or the link quality. However, in this thesis a novel protocol is proposed as it was felt that this would be the best solution rather than modifying an existing protocol. The WCA protocol [26] is used as an inspiration since one of the weight metrics represents the node degree and can be reused while designing CIDER.

5.2.2 CIDER Overview

This section introduces the CIDER protocol. CIDER is specially designed to solve the network formation process for a hierarchical cluster topology, a so-called cluster tree. In Chapter 4, the simulation results show that for data dissemination in lighting networks, it is crucial that only a minimum number of nodes forward a message. Using a large number of forwarding nodes, the latency and the PLR are affected. Therefore, it is required that a clustering protocol for wireless lighting elects CHs only when needed, e.g. when the CHs already assigned are not able to cover the whole network. To accomplish this, CIDER uses a weight-based clustering approach since this makes it possible to incorporate many different parameters into one utility function. The weight-based approach was chosen as this one does not rely on values such as the remaining battery level and weight-based approach can be fine tuned for different applications by defining different sets of weighing parameters. CIDER is a distributed protocol that requires no prior knowledge. All information needed to calculate the utility is exchanged during the first phase of CIDER. The message exchange within CIDER is based on broadcast transmission. To minimise collisions, each node delays the broadcast by a certain number of time slots based on the 12-LSB of their link address. Hence, a broadcast is delayed for 0 to 409 time slots and, therefore, capable of supporting a large number of nodes. The cluster formation design is based on three phases: the Discovery Phase, the Setup Phase, and the Steady-State Phase (see Figure 5.4). The CIDER state machine consists of nine states, where three states are allocated to the discovery phase, and six states are allocated to the setup phase. The discovery phase with its three states is used to discover the neighbourhood of a single node and to gain an overview of which node is suitable for becoming a CH. During the setup phase with its six states, the actual topology is formed using the information gathered during the discovery phase; basically, the CHs are elected, and the different clusters are formed. The design process of CIDER has shown that it is necessary to use nine states to avoid possible deadlocks and other problems which could cause a critical error. At the end of CIDER, each node is either in the CH state or in a cluster slave (CS) state, which are also allocated to the steady state phase. Only the first CH is elected based on a competition between all nodes, while additional CHs are selected by prior CHs when needed. In general, while nodes exchange messages to form clusters, each node keeps track of messages received by updating its neighbour table. In this way, each node is aware of the state of a neighbouring node. For each state,

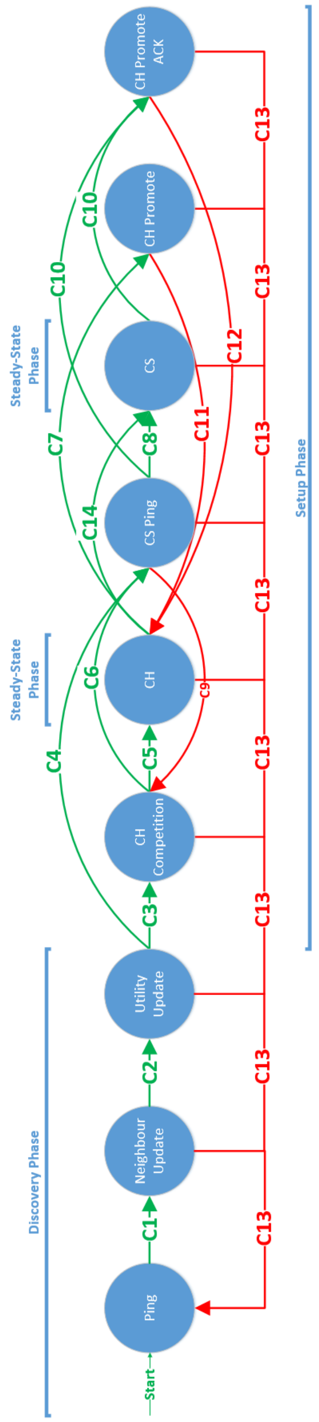


Figure 5.4: CIDER state machine

a node transmits a corresponding message. To increase reliability, each message is transmitted twice. The following rule applies to most states: when a message is sent twice, and a node detects that the neighbour nodes are all in the same state by receiving their messages, a node can proceed to the next state. Where this rule does not apply, a note is given in the respective section as follows.

5.2.3 First Phase: Discovery Phase

The first phase is the discovery phase. Here all information needed to determine becoming a CH or CS are exchanged. In the discovery phase, three states are needed (see Table 5.1).

State	Transition	Links to
PING	C1	Neighbour update
Neighbour update	C2	Utility update
	C13	PING
Utility update	C3	CH competition
	C4	CS PING
	C13	PING

Table 5.1: First-phase states

5.2.3.1 PING

The first state is the *PING* state. In the PING state, each node broadcasts a message containing their current transmit power and link address, the so-called the PING message. The PING message is needed to discover all neighbours in transmission range. Each receiving node either adds or updates an entry in their neighbour table. All information needed for CIDER is stored in the neighbour table. To increase the reliability of a possible successful transmission, each node transmits a message twice. As soon as a node has transmitted the PING message twice, the transition condition *C1* is fulfilled, and the node jumps into the next state, which is the neighbour update state.

5.2.3.2 Neighbour update

The neighbour update state is needed to gain an overview of how many neighbours one's neighbours has. In typical office lighting applications, lamps are distributed in a room at equal distances to each other, and based on this assumption, the node degree is expected to be highest closer to the centre of the network. Therefore, the overview of the neighbours is used to determine if a node is closer to the centre of a network or closer to the border of a network. A Neighbour Update message contains the node degree and the cluster degree of a node. The node degree represents all nodes within the transmission range of a node, while the cluster degree represents all nodes in a specified transmission range smaller than the maximum transmission range. The cluster degree is used to ensure a well-connected cluster. After receiving a message, a node updates the corresponding entry in the neighbour table or adds a new entry if it has not received a message from that particular node before. The transition condition *C2* is fulfilled if a node has transmitted the neighbour update message at least twice and received from each neighbour the neighbour update message. Then a node jumps into the next state, which is the *utility update* state. To ensure that a single node does not block the network by not transmitting its updates (this is possible due to synchronisation errors), other nodes proceed with the clustering if an update is not received within ten cycles. A cycle is defined as the number of own updates transmitted. This mechanism is used for all states, except when stated otherwise.

5.2.3.3 Utility Update

The *Utility update* state is needed to calculate each node's utility in becoming a CH. The utility function is defined by Equation 5.10. The function is based on the approach presented in [25].

$$U = w_1 * d_n + w_2 * d_c + w_3 * \overline{RSSI} \tag{5.10}$$

As stated earlier, CIDER is a weight-based clustering protocol, where Equation 5.10 is used to calculate the utility of each node. The equation requires the information of the node degree (d_n), the cluster degree (d_c), and the average link quality to all neighbours ($\overline{RSSI}$). Each parameter is multiplied by a weighting parameter ($w_1 ... w_3$), which indicates the importance of each factor. The values for $w_1 ... w_3$ can be chosen arbitrarily as no study currently exists on how to choose the optimal values. Several publications state that the values should be adjusted for each configuration [25, 26, 30, 67, 135]. For CIDER, it is most important that a CH is able to reach a large number of neighbouring nodes, a so-called high coverage area. Besides a high coverage area, it is important that a CH has a large number of cluster members(d_c) because it is possible that a node is able to reach a large number of nodes, but these are located at the border of the coverage area. Thus, we reduce the cluster area by setting an RSSI threshold to form only the well-connected clusters. In this way, we ensure that a CH does not only have a high coverage area, but also that the nodes in this area are well connected. When nodes are located closer to the centre, it is most likely that they have a similar if not even the same utility. To reduce the number of tentative CHs, the average RSSI is used. The RSSI is one indicator of how good the link between two nodes is. Typically, when the distance between two nodes increases, the RSSI decreases. Hence, a node located closer to the centre will have a different average RSSI compared to a node closer to the border, as the node in the centre has more nodes in the immediate neighbourhood and, therefore, a lower average RSSI. The weighting parameters in this case are defined as: $w_1 = 0.5$, $w_2 = 0.3$, and $w_3 = 0.2$. The parameters are selected in order of the importance of the three terms, as the experiments showed that these parameters work well in the environments utilised for forming the network topology. In future, different combinations of these parameters need to be investigated to confirm the optimal selection or to provide a set of parameters for different arrangements. The utility itself ranges from 0 to 1; for this, it is necessary to feature scale all parameters. Feature scaling is a method used to standardize the range of independent variables. The simplest method is rescaling the range of features to scale the range in [0,1], and this method is used to feature scale the utility. To feature scale the parameters, the lower and upper bounds need to be defined. The lower bound is zero for the neighbour and cluster degrees. The upper bound for the neighbour degree and cluster degree is the largest neighbour degree in the network, which is gathered by the exchange of the neighbour update message. The lower bound for the average RSSI is the radio-sensitivity of the radio chip, which is in the case of the CC2538 ([73]) -97 dBm. The upper bound is set to -20 dBm because the path loss using the free space path loss model is already approx. 15 dB at a distance of 10 cm. However, before calculating the utility, each node has to fulfil two requirements, i.e. if a node has no neighbours, the utility is set to zero, and if the node has neighbours but no possible cluster members, the utility is also set to zero. After calculation, each node sends an update message that includes the node's utility to become a CH. After receiving a message, a node updates the corresponding entry in the neighbour table or adds a new entry in the case it has not received a message from that particular node before. The utility update state has three follow up states, the CH competition state, the CS ping state, and the ping state. To get to the CH competition state, condition *C3* has to be fulfilled, i.e. a node has to have the highest utility, or which is equal to the highest in the neighbourhood. To get to the CS ping state, condition *C4* has to be fulfilled, i.e. the node's utility has to be lower than the highest one in the neighbourhood. If a node should receive a reset signal at any time during the first phase, the node will be reset and goes back to the initial ping state.

5.2.4 Second phase: Setup phase

All nodes have exchanged network information during Phase one; therefore, each node is aware of the neighbour's state and the suitability for becoming a CH. In selecting a cluster, the setup phase is used, which is entirely focused on creating a minimum number of clusters. Six states are needed for the setup phase, see Table 5.2. At the beginning of the setup phase, a node is either in the *CH competition* state or in the *CS ping* state. If a node receives a reset signal in any state during the second phase, it will reset its current state and revert back to its initial state PING (Transition C13).

5.2.4.1 CH Competition

The CH competition state is needed in the case there is more than one node suitable for being a CH; however, a node suitable to become a CH always enters

State	**Transition**	**Links to**
CH Competition	C5	CH
	C6	CS PING
	C13	PING
CS PING	C8	CS
	C9	CH Competition
	C10	CH Promote Ack
	C13	PING
CH	C7	CH Promote
	C13	PING
	C14	CS
CS	C13	PING
CH Promote	C11	CH
	C13	PING
CH Promote Ack	C12	CH
	C13	PING

Table 5.2: Second-phase states

this state regardless of whether there are other nodes suitable available or not. The competition mechanism is inspired by the mechanism proposed in [63]. In general, when there is more than one node in the CH competition state, they will compete to become a CH. This means that a node has the highest utility of its neighbours, or the utility is equal to the highest utility of the neighbours. In this case, the node transmits a CH competition message. A node that is not in the CH competition state and receives this message ignores it, but if the node is in the CH competition state, immediately sends a CH competition message as well if it has not yet send one. However, as soon as a node in the CH competition state receives this message, it sets a random timer, which expires after 1 to 5 seconds. During development of CIDER, different timing parameters were tested, and the range of 1 to 5 seconds worked best for the configurations in this thesis. To confirm whether the range from 1 to 5 seconds is best for all possible arrangements, further investigations are necessary. After the timer expires, the node will jump to the CH state, transition *C5*, and then announce itself as a CH immediately by sending the CH message. If a node receives a CH message while in a CH competition state, it immediately gives up its candidacy for becoming a CH and jumps to the CS Ping state (Transition C6). On the other hand, if a node in CH competition state does not receive a CH competition or CH message from any other node during the time it takes to transmit the CH competition message twice, the node assumes that there is no other node suitable for being CH at this time and declares itself a CH and jumps to the CH state.

5.2.4.2 CS Ping

The CS Ping state needs to be advertised if nodes are not the first choice to become a CH but are still not assigned to a cluster. A node in a CS ping state will broadcast a CS ping message every 5 seconds to inform the neighbourhood about its state. Again, different timing parameters were tested, and repeating the CS ping message every 5 seconds worked best. With a shorter repetition time, the channel contention became unnecessarily high, and with a longer time, a CH assumed too often that the clustering process is finished due to a reception of a CS ping message being missed. Included in the message is the node's utility for becoming a CH. Every node receiving the CS ping message will update or add an entry in its neighbour table. However, a node in CS Ping state can either become a CH or a CS but not by choice. A node will be always be selected as either one by a prior CH. That means, when a node receives a CH message, it checks whether its link address is included in the message. If so, it will change its state to CS (Transition C8). There are two ways to become a CH from CS Ping state: when a node will receive a CH promote message from a prior CH, then it will change its state to CH Promote Ack (Transition C10); or when a node detects there are no CH's in its neighbourhood, and the node's utility is the highest in its neighbourhood, then the node will change its state to CH Competition (Transition C9). Before a node in the CS Ping state can select itself to be a CH, it has to make certain that there is not a single CH in the neighbourhood;

therefore, a node waits until a send counter exceeds a pre-set number. For this thesis the send counter is set to 20, which means that a node in the CS Ping state will try to become a CH after 100 seconds at the earliest, but only when no CH advertisement from any other node is received. The limit of the send counter was tested with different values, and for the experiments in this thesis, 20 worked the best.

5.2.4.3 CH

The CH node is needed to advertise existing clusters as well as selecting cluster members and selecting new CH's. After a node declares itself a CH, it broadcasts an advertisement message periodically. This message has several bits of information. First, a list of all cluster members is included. To create the list, a CH scans its neighbour table for potential members. A node will be selected to be a member when certain requirements are fulfilled. The first is that a potential node is not already assigned to another CH, and the second is that a potential node needs to be within the cluster range, which is determined based on the last RSSI received from that particular node. If these requirements are fulfilled, the link address of the selected node will be added to the list. A CH message can include up to 50 link addresses due to the CH message format defined in Table B.8. Therefore, the number of members in one cluster is limited to 50. For the experiments conducted within this thesis, 50 nodes in total were used; however, if CIDER is used in larger networks of more than 50 nodes, a mechanism is necessary to enable larger cluster sizes. To increase the cluster size, the list of cluster members in the CH message could be implemented as a kind of ring wherein a first message, the first 50 members are broadcast, and in a second message, the next 50 members. If all members were on the list, the first 50 nodes are added to the message again. In this way, the cluster size can be easily increased. Besides the list of all members, the CH includes the information of the colour and tier used for channel allocation. However, besides advertising the cluster itself, a CH is also responsible for electing new CH's in case there are any nodes remaining outside its coverage area. Nodes outside the coverage area can simply be detected when a CH receives a CS ping message, but the transmitting node is too far to become a member of this CH. In such a case, the CH selects the most suitable node from its neighbour table, which is not assigned as a member of its cluster or any other cluster. That is, a node must be in the CS Ping state and must have the highest utility of all nodes not assigned to a cluster. When a node is selected as a new CH, the promoting CH node jumps to the *CH Promote* state (Transition C7).

At the end of the second phase, the Tier 0 CH is responsible for detection if the clustering process is finished. For this, the CH transmits a *CIDER Complete* message. After transmitting the Complete message, the initial CH sets a wait timer. The waiting period is chosen arbitrarily and should be adapted for each scenario. For CIDER used for lighting networks, the waiting period is defined by Equation 5.11, with NUM_{CH} as the number of CHs in the neighbour table. If it is a larger network with multiple CHs, this results in a longer wait period. This ensures that the processing time for all CH that receive a *CIDER complete* message is large enough.

$$t_{wait} = 1s + (10 * NUM_{CH}) * 1s \tag{5.11}$$

However, a receiving node who is in any state except the CH state will drop the CIDER complete message. A node in a CH state receiving the Complete message will either respond by forwarding the Complete message or reply with a *CIDER incomplete* message. For either one, the CH will check its neighbour table for nodes not assigned to any CH. If there are nodes not assigned to any CH, it will reply with the incomplete message; and if all nodes are assigned to CHs, it will forward the complete message. In the same way, a CH checks whether it has any child nodes assigned. If there are no child nodes, the CH will demote itself to the CS state (Transition C14) with the prior CH as a parent. If the initial CH receives an Incomplete message, the wait timer will be cancelled. The Complete process will again be called after a waiting period of 10 s. This time is chosen arbitrarily and should be adapted for each scenario. As this process is used to detect whether there are no uncovered nodes left, the value of this timer is negligible. However, in detecting whether the complete process can be started, the Tier 0 CH checks its neighbour table before transmitting the CH message itself. The CH checks whether all nodes in its neighbour table are either in the state CH or CS; if this condition is true, the CH will start the Complete process, otherwise, the CH will wait for the next CH message period. When the wait timer expires, CIDER is complete, and the protocol goes into a steady state.

5.2.4.4 CS

The CS state is needed for the nodes that got assigned to a CH. The CS state is one of two final states in CIDER. A node that changes its state to CS will advertise its state exactly twice by transmitting a CS message. To avoid collisions with other nodes, the node will delay its transmission as defined in Section 5.2. After advertising its state, the node only listens to incoming messages from other nodes and updates its neighbour table, respectively. Also, if a node in the CS state does not receive any CH message from its parent node for a defined period, it will reset its state to the initial state PING (transition C13).

5.2.4.5 CH Promote

The CH Promote state is needed to elect new CHs to the network for increasing the overall coverage area of CIDER. A node is selected by a prior CH. The prior CH selects the node while in the CH state and then jumps to the CH Promote state. In the Promote state, a CH node broadcasts a message that includes the link address of the selected node. The receiving node checks the message for whether the included link address matches its own link address. If it is the same link address, it will change its state to *CH Promote ack*. If the included link address does not match, the node will drop the message and continue the current state operation. The node that transmitted the CH promote message will transmit the message exactly once and then change its state back to the CH state, waiting to receive an acknowledgement from the selected node.

5.2.4.6 CH Promote ACK

The CH Promote Ack state is needed to make sure that a node is only elected once because it is possible that two CH nodes decide to elect the same node as a new CH at the same time. The receiving node can only be assigned as a CH child to one parent CH. If multiple Promote messages are received, the node only acknowledges the reception to one CH. However, the node that received the Promote message (usually in a CS Ping state earlier), will change its state to *CH Promote ACK*. First, the node checks if the link address matches the link address included in the message. If the address is a match, the node will generate an acknowledgement and transmit it immediately. If the address is different, the node will drop the message and continue to operate in its current state. Based on the information included in the promote message, a node can determine its tier. After transmitting the acknowledgement, the node updates its state to CH and will act as CH.

5.2.5 Third phase: Steady-state phase

The third and final phase of CIDER is the *steady-state phase*. In steady state, there are only two states active, either *CH* or *CS* (see Table 5.3).

State	**Transition**	**Links to**
CS	C13	PING
CH	C13	PING

Table 5.3: Third-phase states

5.2.5.1 CS

The CS state is one of the two steady states of CIDER. If a node is in the CS state, it only listens to CH messages of its parent. If a node does not receive any CH message from its parent node for a defined period, it will reset its state to the initial state (Transition C13) and re-initiate its clustering process.

5.2.5.2 CH

The CH state is the second state of the steady-state phase of CIDER. A node in the CH state will periodically advertise its cluster. Also, a node in the CH state will accept in its cluster new nodes joining the network as long as the maximum cluster capacity is not exceeded. If a CH detects a new node joining the network, it will check its capacity and will assign it to its cluster if possible. With the next CH message, the link address of the new node will be added to the list in the message. In case a CH cannot add the new node to its cluster, there is the possibility that another CH is also within distance of the new node. Therefore, the CH will wait for a period, and if after waiting, if the new node is still not assigned to a cluster, the CH will reset its own state to the initial state (Transition C13) and re-initiate its clustering process. By doing this, the whole cluster will not receive any CH messages. Hence, all child nodes of the specific CH will reset their state and the single cluster will be formed again. This behaviour prevents the whole network from collapsing due to just adding new nodes to the network.

5.3 Channel offset allocation

Generally, allocating a channel offset with TSCH can be done using either a centralized approach or a decentralized approach; however, the standard does not define any mechanisms for doing this. Therefore, this section solves the problem of distributed channel offset allocation by applying a greedy colouring algorithm. By doing this, each cluster can use its own channel offset for intra-cluster communication and a shared channel offset for inter-cluster communication. The usage of the different channel offsets within D^3LC is described in detail in the next section. This section focuses only on the selection of the channel offsets. The remainder of this section is as follows: first, an overview of the state-of-the-art TSCH scheduling methods are given in Section 5.3.1. Section 5.3.2 presents the applied colouring algorithm and its original purpose, and in Section 5.3.3, it is described how this algorithm is modified and implemented using Contiki. Finally, the frame format used in the messages of this protocol is presented in Section C.

5.3.1 State-of-the-art on TSCH scheduling approaches

For many years, a research area has been the channel allocation problem that exist within wireless networks operating in the 2.4 GHz ISM band. Many approaches exist for different networks, e.g., for networks complying with the IEEE802.11 standard [37, 71, 79, 81, 85, 99, 115], the IEEE802.15.4 for ZigBee networks [8, 24, 139], and the IEEE 802.15.4 TSCH standard [5, 40, 45, 53, 103, 104, 136]. This section reviews only the channel allocation methods focusing on TSCH networks. TSCH differs from traditional MAC protocols for WSNs due to its organised and scheduled links. Each device in a TSCH network follows a schedule that tells a device what to do at a specific point in time: transmit, receive, or sleep. Regarding this, only a little research has been done so far. A common approach in the area of IoT is to use a protocol standard called IPv6 over the TSCH mode of IEEE 802.15.4 (6TiSCH), which was standardized by the IETF in 2015 [53]. 6TiSCH combines the Internet Protocol Version 6 (IPv6) protocol with the TSCH method. Palattella et al. dedicated a book chapter to 6TiSCH [104]. The goal of the 6TiSCH Working Group is to develop a standard approach to manage this schedule and match it to the traffic needs of the network [136]. 6TiSCH considers three different modes in building such a schedule: a minimalistic schedule, a centralized schedule, and a distributed schedule. The minimalistic schedule is a static schedule, usually preconfigured, which can also be learnt by a node joining the network; however, a minimalistic schedule does not take advantage of the full TSCH capabilities. A minimalistic schedule can be used as a fall-back mode. The centralized schedule includes a schedule entity, called Path Computation Element (PCE), which is connected to the Internet. The PCE continuously tracks the network state as well as the network traffic from the devices in the network. The schedule is then altered according to the network needs. In the distributed scheduling mode, all nodes agree on a general schedule, but also each neighbouring pair can negotiate links. For all three methods, the channel offset is chosen arbitrarily; the only requirement is that for a certain link, the combination of the timeslot and channel offset is not already used. This is easy for a static global approach as well as for the centralized approach where a central device assigns links. With the distributed scheduling approach, it is not possible to avoid overlaps in the neighbourhood as nodes negotiate only the links with its neighbours but do not have knowledge of their schedule.

In [103], a Traffic-Aware Scheduling Algorithm (TASA) is proposed. A TASA is a centralised scheduling algorithm where the authors define the scheduling problem as a graph colouring problem. For the centralised approach, a tree topology is needed where the Personal Area Network (PAN) coordinator is aware of the topology, defined as graph G. To assign the links between the nodes, the PAN coordinator defines sets of duplex-conflict-free links, which are sets of links where no parent-child or child-parent conflict appears, e.g., a parent receives messages from multiple child nodes. Next, in the same manner, the coordinator creates a set of interference-conflict-free links which can use the same channel offset in the same slot. A set of duplex-conflict-free links consists of multiple interference-conflict-free links, each of which has to be assigned using a different channel offset. These sets are built for each slot, and a colouring algorithm is then applied to assign a different colour for each link in the duplex-conflict-free link set. Hereby, a heuristic method is applied to assign different colours to the interfering links in one set. However, in the case where there are fewer colours are available than links, only some of the links of the duplex-conflict-free link set will be assigned for one timeslot. The remaining links will be added in the procedure for the next timeslot. This method shows that solving the channel offset allocation process using graph-colouring algorithms is feasible; however, as this approach requires the central device to be aware of all links and connections in the network, it will result in a high overhead in dense lighting networks with a large number of devices.

Accettura et al. propose a decentralized scheduling method using the same requirements of TASA, named Decentralized Traffic Aware Scheduling (DeTAS) [5]. This decentralized approach uses RPL to form a Destination Oriented Directed Acyclic Graph (DODAG). Using RPL, each node is assigned to a DODAG rank which is related to the minimum hop distance to the DODAG root node. At the same time, the root node acts as the PAN coordinator of the TSCH network and is assigned the DODAG rank of 0. DeTAS supports multiple sink nodes which all are assigned the DODAG rank of 1. A sink node forms its own graph which is directly linked to a micro schedule; hence, each sub-graph represents a micro schedule. Any micro schedule is built in a way that no collisions occur in the network. Each micro schedule uses the 3-channel offset, e.g. sub-graph 1 uses the channel offsets $0, 1,$ and 2, sub-graph 2 use the channel offsets $3, 4,$ and 5, and so on. A micro schedule defines its channel offsets using the following rule: $3(k-1),$ $3(k-1)+1$ and $3(k-1)+2$, while k defines to which subset the schedule belongs, also called partitioning. In this way, a sink node is able to assign links with its 1-hop neighbour using one of the three available channel offsets. The evaluation only investigates the queue's occupation as an indication of how much traffic can be handled in this schedule creation method. Unfortunately, no indication is given of any possible overlap between any two competing sub-graphs.

In [45], Orchestra is introduced. Using Orchestra, nodes are able to compute their own local schedule autonomously. Each node maintains multiple schedules, each for a different traffic pattern, e.g. one schedule for networking information from RPL. With Orchestra, no extra overhead is needed, nor does it require a central or distributed scheduler. An Orchestra schedule relies only on the network stack information already available such as the RPL information. A typical schedule always consists of following elements:

1. one slot for beacon messages for synchronisation, repeating every X timeslots
2. one slot for exchanging RPL information, repeating every Y timeslots
3. one slot from every node to its RPL parent, repeating every Z timeslots
4. N slots from each node to each of its child nodes, repeating every Z' timeslots

Each slot selects its time and channel offset as a function of the sender or receiver identifiers which are usually the MAC address or network node ID. Using these principles, each node using Orchestra is able to create its own schedule to communicate with its parent node and child nodes. However, as the work is very detailed as to how different slots are used, e.g. parent-child communication or general broadcast slots, no definition is given of the actual time for a node's operation nor the channel offset that should be used. There is only a general note that a node selects the time and offset as a function of the identifier. That is, one slot operation is called *Receiver-Based Shared Orchestra Slot* (RBS) where two neighbours are assigned to each other for communication. Here, the timeslot and channel offset are derived from the receiver's unique ID, not given by any explicit function.

The work of Domingo-Prieto et al. presents a distributed PID-based scheduling algorithm [40]. A PID controller is a control loop feedback mechanism used for stabilizing control loops; however, the work defines the TSCH scheduling problem to be a closed loop-control problem. The PID controller is used to optimise the schedule for fulfilling a node's traffic demand. The scheduling mechanism works in a distributed manner and on top of the 6TiSCH protocol. Based on the traffic demand, the PID controller calculates the slots required during the next slotframe. The result is then sent to the 6top layer [134] (part of the 6TiSCH protocol) which takes care of the negotiation with the neighbours. This is one example of applications demanding the resources of the available slotframe, and lower protocols actually take care of the negotiation, in this case, the 6TiSCH (6top) protocol.

Summary This section gave an overview of how scheduling algorithms are applied within a TSCH network. In general, three different approaches are possible: centralized, distributed, and autonomously. All three methods target the same goal: trying to create a schedule that meets the traffic demands of the network in the most efficient way. In terms of avoiding overlaps in the concurrent transmission or assigning two links to the same channel offsets, a common approach is using a graph colouring algorithm. The algorithms used for the various protocols are either centralized protocols, requiring an overall network knowledge, or distributed protocols with an additional high message overhead. For D^3LC, a decentralised algorithm is the best fit because the data already available from CIDER can be reused and can be simply exchanged between the nodes, rather than forwarding all needed information to a central device. A candidate algorithm is Degree of Saturation (DSATUR) which is explained in more detail in Section 5.3.2 [19]. DSATUR was selected as it is a suitable approach in particular as the distributed approach works well in a distributed sensor network. The protocol was then analysed in simulation as part of the DEWI project by a colleague with the result that it was found useful. Based on this result DSATUR was integrated into the D^3LC-Suite with minor changes described in this chapter.

5.3.2 The Colouring Algorithm

This section introduces the graph colouring algorithm applied to solve the channel offset problem within D^3LC. The selection of a suitable colouring scheme for dense wireless lighting networks was carried out by Dr. Kritchai Witheephanich as part of the DEWI project. Dr. Witheephanich rationale to select graph colouring were as follows: Graph colouring [19] has been extensively considered in the literature as a task for channel allocation in telecommunication spanning cellular networks [78, 100], wireless LANs [3, 16] and wireless sensor networks (WSNs) [54,80,140]. Importantly, applying centralised channel allocation algorithms is not an option for being efficiently implemented in sensor networks and in particular in dense networks due to

- existing hardware platforms for WSNs are unavoidably limited in energy, network bandwidth, computational capacity and memory.

- a centralised algorithm may not be possible to be implemented on a single node that requires information of the whole network topology to apply the algorithm by establishing direct communication between the base station and all individual nodes, [83].

However, the previous section showed that for solving the scheduling problem, a common approach is using graph colouring. Also, graph colouring algorithms are trivial and easily implementable on resource-constrained devices such as wireless sensor nodes. A centralised algorithm for node colouring can be distributed and solved in either a heuristic (not optimal, lower computation cost) or an optimisation (definitely optimal) method. When using the CIDER protocol to form the network topology, a lot of network information is available, which can be used to colour the topology. The requirements for selecting a graph colouring algorithm are as follows:

1. The network must be coloured in a way that each cluster is coloured differently

2. Two clusters in interference range of each other cannot share the same colour; however, two clusters not in interference range are allowed to share the same colour

3. By colouring the network, only a minimum of colours should be used

To fulfil these requirements, a distributed graph-colouring algorithm was implemented on the nodes, ensuring that no neighbouring clusters use the same channel at the same time and using the information already available with a minimal overhead. The algorithm used in this work is a distributed algorithm based on the centralized saturation degree (DSATUR)-based graph-colouring algorithm [19]. The topology formed using CIDER is represented by a general and directed graph $G = (V, E)$, where V represents a set of n vertices (sensor nodes) and E a set of m edges (interference relations). The objective of the problem is to determine a node colouring scheme with a minimum number of colours, which is subject to no two adjacent nodes assigned to the same colour being connected. At the same time, the node colour represents the channel offset used for TSCH itself. Before discussing the distributed algorithm, the centralized DSATUR is briefly explained. This algorithm requires each node $v \in V$ to have full global information about the network. The algorithm is as follows:

1. Arrange the nodes in decreasing order of their degree, $d(v)$
2. Colour a node of maximal degree, $max(d(v))$, with the Colour 1.
3. Choose an uncoloured node of maximal saturation degree, $max(sd(v))$. If there is a tie, choose any one of the nodes of a maximal degree in the uncoloured sub-graph.
4. Colour the node selected using the colour with the lowest possible number.
5. When all nodes are coloured, then STOP, else go to Step 3.

As requiring a global knowledge is impractical, if not impossible, a distributed algorithm called Distributed Saturation Degree Algorithm (DSDA) [29], which is derived from DSATUR, is used to solve the channel allocation problem in a fully distributed manner. The main idea of DSDA is to let each node select its own channel within a TDMA schedule by only using the information collected from its neighbourhood. For DSDA, it is necessary to add additional variables, the *saturation degree* of *vertex* v as $sd(v)$, representing the number of different colours assigned to the neighbours of v. Also, as a manner of calculating the saturation degree in a distributed way, the Saturation Degree Indicator (SDI) is introduced. The SDI is used to calculate the priorities of each node in a node's neighbourhood for colouring and is defined using Equation 5.12. $SDI_v\,(u)$ is the SDI of node u, calculated by node v. $\Delta(v)$ represents the maximal node degree of node v, $sd\,(u)$ is the saturation degree of node u, $d\,(u)$ is the node degree of node u, and $\mathrm{rand}(u) \in (0,1)$ is a random number of node u used for tie-breaking.

$$SDI_v\,(u) \;=\; (\Delta(v) \;+\; 1) \times sd\,(u) \;+\; d\,(u) \;+\; \mathrm{rand}\,(u) \tag{5.12}$$

For a successful DSDA operation, the following assumptions are adopted:

1. Nodes are distinguished by a unique identifier (ID)

2. After initialisation, each node knows the IDs and degrees of its neighbours ($\Delta(v)$).

To distinguish the nodes, two separate lists are used for their states as well as their colours. LC(v) represents a list of all coloured neighbours of node v, including their colour. A list of all uncoloured neighbours of node v is represented by LUC(v). This list also includes information such as saturation degrees, degrees, and random values. Using these two lists, it is possible to extract the information of $sd(v)$ from LC(v) and $\mathrm{SDI}_v\,(u)$ from $LUC(v)$. The general DSDA operation is divided into two rounds, $subrnd1$ and $subrnd2$, while the algorithm operates individually on each node regardless of the overall network progress. During $subrnd1$, each node calculates the $\mathrm{SDI}_v\,(u)$ for itself and for all uncoloured neighbours. If any node detects that it is an *Extremum Node*, this node will be coloured during the current round. A node is an *Extremum Node* if a node's SDI value is the highest among its neighbours. This node will select the minimum colour of the available colours not previously selected by its neighbours. The remaining colours can be extracted from the LC(v). The node will change its state to *coloured* and broadcasts a so-called *release* message containing the selected colour.

During $subrnd2$, an uncoloured node receiving the *release* message will update its neighbourhood information and broadcast a so-called *update* message. This message includes updated values of sd(v), $d(v)$ and rand(v). Each node receiving the update message will update the LUC(v). In case a node did not receive a release message, the node will broadcast an update message every T rounds with unchanged values where T is a predefined constant. Each node runs this algorithm until it has selected a colour. If a node receives the update message and has not yet selected a colour, it will continue with $subrnd1$.

The performance analyses show that any two neighbouring nodes will not select the same colour, as a node only selects a colour if it is an *Extremum Node*. DSDA uses a minimum number of colours, which, in the worst case, is $\Delta(v)\ +\ 1$ if all nodes are within range of each other.

5.3.3 Channel-offset allocation implementation

This section presents the implementation of the DSDA colouring algorithm in D^3LC. The colouring algorithm aims at assigning different channel offsets for each cluster; hence, for the remainder of this section, a *node* represents a CH. All nodes in a CS state are not considered in any calculation during the colouring phase. At the same time, the node colour represents the channel offset used for TSCH itself. A channel offset can range from 0 to 15 but can also be an arbitrary sequence of values between 0 and 15, e.g. when only five channel offsets are used, one sequence can be $[0, 4, 5, 8, 15]$.

5.3.3.1 Variable Definition

First, a set of variables is needed for DSDA, and their definition is given in Table 5.4. It is necessary to bring the variables defined by DSDA into the context of D^3LC. Therefore, $\Delta(u)$ represents the maximum degree of any node in the neighbourhood. Node u has two neighbouring nodes, v and w, with $d(v) = 3$ and $d(w) = 1$, thus, $\Delta(u) = 3$. $\mathrm{sd}(u)$ is the saturation degree which is defined as the number of different colours used by a node's neighbours. The $d(u)$ variable represents the neighbourhood size of node u, and here, only CH's are considered. If a node has five neighbouring CHs and 50 CSs, the neighbourhood size is five.

DSDA Variable	Definition by DSDA
$\Delta(u)$	Max. degree of node u
$\mathrm{sd}(u)$	Saturation degree of node u
$d(u)$	Neighbourhood size of node u
$\mathrm{rand}(u)$	Random number of node u
$\mathrm{SDI}_v\,(u)$	Saturation Degree Indicator of node u calculated by node v
$\mathrm{LC}(u)$	List of all coloured neighbours of node u
$\mathrm{LUC}(u)$	List of all uncoloured neighbours of node u

Table 5.4: Values of the Identifier Field

At the beginning, each node generates a random number between 0 and 1, represented by $\mathrm{rand}(u)$. Equation 5.12 is used to calculate the $\mathrm{SDI}_v(u)$. The two lists, $\mathrm{LC}(u)$ and $\mathrm{LUC}(u)$, are not separately managed; furthermore, both lists are integrated into the already existing neighbour table, hence, all information needed is stored within one list.

5.3.3.2 Protocol Operation

To implement DSDA, two states are needed, one for updating the neighbours with new information and one to select a colour if a node is an *Extremum Node*. This results in the state machine shown in Figure 5.5, consisting of the state *Update* and *Release*. After CIDER completion, each node will start the colouring process by entering into the *Update* state for the first time. DSDA assumes that each node is already initialized; therefore, each node entering the update state for the first time will broadcast an *Update* message. The message exchange follows the same rules as for the CIDER protocol, i.e. each node will delay a transmission to avoid collisions. The update message includes information of $\mathrm{sd}(u)$, $d(u)$, and $\mathrm{rand}(u)$. To ensure that all nodes receive the update messages, each node sets a *Wait Timer* to the maximum wait period possible using the same wait mechanism defined for the CIDER protocol. This timer will be restarted after expiration repeatedly until the colouring process is completed. Each node receiving the update message will update the neighbour table accordingly. After the timer has expired, each node calculates the SDI for each of its neighbours, including its own. If a node is an *Extremum Node*, it will enter the *Release* state (Transition *C1*). The node in the *Release* state will select the lowest available colour not already selected by a neighbouring node and update its neighbourhood

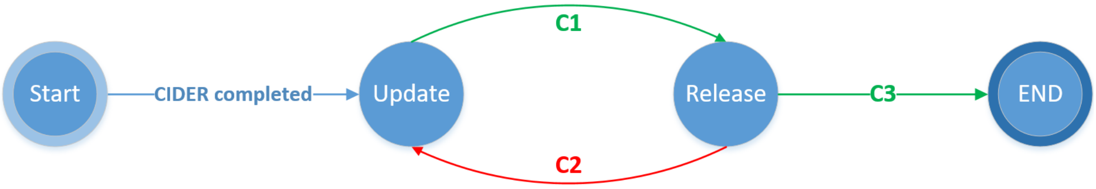


Figure 5.5: DSDA state machine

by broadcasting a *Release* message. A node receiving a Release message will update its neighbour table accordingly. Nodes in the *Update* state will broadcast an *Update* message containing the updated values of sd(u), $d(u)$, and rand(u). This process will continue until each node is coloured; however, because none of the nodes is aware of all nodes in the network, a mechanism to detect the completeness of the colouring process is necessary. For this, the same mechanism is used as for CIDER. The Tier 0 CH will periodically check that all its neighbour nodes are coloured, i.e. whenever the *Wait Timer* expires, the Tier 0 CH will check all its neighbours whether they are coloured or not, and then start the Complete process. This requires that the Tier 0 CH has already selected a colour. The Complete process follows the same principles, first, the Tier 0 CH broadcasts a *Complete* message. Upon its reception, each receiving node will check if a colour has been selected and whether the colour is unique in its neighbourhood. If yes, the node will respond with a *Complete* message; otherwise, this node will respond with an *Uncompleted* message and continue with the colouring process. If it was detected that the colour is not unique, which is possible due to certain circumstances such as with the exact same SDI value of two neighbouring nodes, both nodes will reset their colour and restart the colouring process by entering the *Update* state (Transition C2). Once all the nodes have chosen their colour using this algorithm, the stack switches over to the RLL operation.

During the colouring process, only the CH nodes are active; however, all CS nodes are listening to the incoming messages. If a CS node detects a release message from its parent, the node will store the colour. In this way, every node knows the colour that was selected by its parent node during the colouring process.

5.4 RLL - Reliable low-latency data dissemination in dense wireless lighting-control networks

This section introduces the *Reliable Low Latency* (RLL) data dissemination protocol. Lighting is a use case with critical QoS and QoE requirements where users expect an immediate visual response from the system within milliseconds. The RLL protocol operates on top of the IEEE802.15.4 TSCH mode, requiring an underlying cluster-tree topology. The design goal of RLL is to develop a dissemination protocol that provides synchronous message delivery to all nodes in a network within a certain time frame. This is an essential feature of RLL since this ensures that lamps receive a control command at the same time. RLL exploits the cluster tree by applying forwarding rules between different CHs. The remainder of this section is as follows: Section 5.4.1 provides an overview of the state-of-the-art data dissemination protocols, Section 5.4.2 describes the general operation of the RLL protocol, while Section D presents the frame format used for RLL.

5.4.1 State-of-the-art of data dissemination in WSN

This section presents a short overview of data dissemination protocols. The lifetime of a deployed Wireless Sensor Network can be up to several years. In most deployments, it is unavoidable to fix software bugs, reconfigure network settings, update the software, or send control commands. Manually collecting all nodes might be an option in small deployments with just a few nodes, but in dense networks with up to several hundred to several thousand nodes, the collecting, reconfiguring, and redeploying of them takes a long time and in some applications, may even be impossible due to their location [145] or their type of installation. These applications are typical for data dissemination protocols where the traffic appears very infrequent and the reliability of the protocol was the main goal and the dissemination time was not one of the main concerns. Disseminating the data via multiple hops is a commonly used technique. However, the common goal of data dissemination protocols is that all nodes in a network are in the same state after a certain time. A simple dissemination protocol is known as *Flooding*. Flooding is rigorously discussed in Section4.1 and its advantages and disadvantages are laid-out in that section.

A popular protocol is called Trickle [87]. Trickle is a multi-hop broadcast algorithm that was originally developed for code propagation in wireless sensor networks. Trickle uses a polite gossip method in which nodes periodically update their neighbours about their current code status. If it has already received a recent update from one of its neighbours, a node remains quiet. Different from flooding, trickle follows a policy to avoid network floods. Trickle controls the send rate within a network and ensures that each node receives a small "trickle" of packets to stay up to date. Trickle can operate in dense networks, disseminate new code in the order of seconds and is able to keep a network up to date with only a few messages per hour. However, with Trickle, a node always sends its code as a broadcast message. A node can react to the receipt of a broadcast message in two ways. The first is for when all nodes are up to date and no update is needed, the second is for when a receiving node detects the need for an update. The detection is possible when a node overhears a message with a newer code; it updates its status accordingly. Or, a node detects that another node broadcasts an old code; that node will then be updated. The authors evaluate Trickle based on a number of parameters. The first is the number of transmissions within a Trickle interval in different sized networks. It shows that with an increasing number of nodes in the network, more transmissions are needed to update the network. The average number of receptions per transmission; this shows that when networks become denser, the number of receptions increases since the number of transmissions increases. Therefore, the authors found that it becomes more likely that two messages collide, which then require a retransmission. By changing the Trickle interval, it is possible to influence the latency until all nodes are updated. With a Trickle interval of 60 seconds, all nodes had received the update within 15 seconds. When increasing the Trickle interval to 300 seconds, the nodes are only fully updated after approx. 60 seconds. Due to this, the time to consistency in Trickle is in the order of seconds to minutes, which is not suitable for real-time applications.

In [60], the authors present Sensor Protocols for Information via Negotiation (SPIN) which is a collection of protocols to overcome the reliability problem of non-negotiation protocols, such as Flooding. The focus is on the efficient dissemination of observed sensor data to all nodes in a network where all nodes in the network are treated like sink nodes. The approach of SPIN is that nodes negotiate with each other for which node is best suited to forward a message. This approach ensures that only useful data is forwarded. However, to do so, a node needs to be able to identify the data that it would like to transmit. Hence, the authors defined a set of descriptors, called meta-data, to identify the type of data. A negotiation process is started when a node observes an event and transmits an advertisement message describing the data. If the node will not transmit a data request message, a receiving node checks if it has already received that data item or has even requested it earlier. The negotiation is completed when the initial nodes transmit the actual data. The authors evaluated SPIN by using the *ns* simulation tool in a simulated network consisting of 25 nodes where each node had an average neighbour degree of 4.7. The results show that SPIN outperforms flooding in terms of throughput and energy consumption. Unfortunately, the authors provide no indication of the data dissemination latency. However, SPIN seems to work fine in small deployments, but when the network size and density increase, SPIN might create a huge overhead due to its negotiation scheme.

Chlipala et al. present a reliable data dissemination protocol, called Deluge [65]. To achieve a certain level of robustness the authors adopt an epidemic approach. The protocol itself is a Negative ACK (NACK)-based protocol which relies on periodic updates from its neighbourhood. A general communication period is where nodes update their neighbourhood about its current state by transmitting advertisements. Based on these advertisements, a node can determine whether it requires an update of its data, similar to the Trickle approach. Based on the received advertisements, nodes can select the best node from which they should receive the updated data. The authors state that several heuristics can be applied to select the most suitable node. After the best node is selected, a data request is sent, and after a short waiting period, the selected node will answer with the requested data. After the requester has received the last packet, it will broadcast an advertisement before other potential new requests are sent.

There are no explicit NACKs sent during the whole process. The authors define the NACK procedure as more of an ongoing reminder of potential missing data by periodically transmitting advertisements with the data available. Deluge was evaluated by simulating different network topologies and network densities. The density is defined as the number of nodes in a certain area. The authors used 25 nodes, of which one node is always the initiating node. To change the density, the authors simulated different inter-node spacing values. The results show that the time to completion is in the order of seconds when the nodes are located close to each other. If the node spacing is high the completion time can be as high as 1000 s. Based on these results, Deluge is not suitable for a lighting application due to the high latency, even with a low number of nodes. It can be expected that with an increasing number in the nodes' density, the completion time increases.

Arumugam et al. present Infuse, a TDMA-based reprogramming service for wireless sensor networks [13]. Infuse can operate on top of any TDMA-based protocol. With Infuse, all nodes need neighbourhood information, and especially, a node needs to know the successor nodes to the east and south direction. The direction is determined by using a GPS module. Infuse gathers this information during the initialization process. The data dissemination process when using Infuse is as follows. In an ideal case, the initiator is located at the coordinates (0,0). The dissemination process is started by transmitting a *Start-Download* message. A receiving node prepares itself to receive an update and forwards the *Start-Download* message in the next TDMA slot. After the first message, the initiator starts transmitting fragments of the update. Whenever a node receives this fragment it stores the data in the appropriate location and, in turn, forwards the fragment to the next slot. Besides the dissemination process, Infuse provides features that make the code propagation robust, called the backpressure mechanism. Implicit acknowledgements are used with the backpressure mechanism where a device keeps track of the received fragments. If a device detects a missing fragment, it will request all fragments, starting from the missing one. Infuse was evaluated in networks up to 100 nodes using different data sizes. The reprogramming latency with Infuse is in the order of minutes, increasing with an increase in data size. Also, the reprogramming time is independent of the network size. Due to the very high latency, Infuse is unsuitable for wireless lighting networks.

In [52] the authors present the *Low-Power Wireless Bus* (LWB) protocol which uses Glossy as the underlying communication protocol but adds additional functionality to it. Glossy was discussed in Section 4.1. LWB is designed to turn the wireless network into an infrastructure similar to a shared bus by using only Glossy's fast network floods. LWB keeps all nodes time-synchronized and applies a centrally computed communication schedule to ensure fair access for all nodes to the channel. Thus, LWB supports one-to-many, many-to-many, and many-to-one communication. The communication schedule of LWB is similar to a TDMA schedule where the host node creates a global schedule and a node knows exactly when it has access to the channel; this is called a communication round. The period of the round can vary and is based on the current demand. Each round consists of a number of non-overlapping slots in which at most, one node initiates a flood. All other nodes receive and relay the flood by using the Glossy mechanism. Each round has different slots, e.g. the first slot is always used by the host for distributing the communication schedule. The second slot is a so-called *contention slot* through which every node can access the bus. This slot is used by nodes to inform the host of their traffic needs so that the host is able to adapt the schedule for the next round. After the *contention slot*, the assigned data slots follow. An application can interact with LWB in two ways. First, it provides an interface for an event-based traffic where a message is simply placed in an outgoing queue. Second, it is possible to define a periodic stream of packets, which is defined by a packet interval and a starting time. LWB provides additional features, such as the handling of node failures and disconnection. For this, LWB uses a counter-based scheme. If a host does not receive any packet within a certain time, the data slots for this node are removed. A second feature to prevent a network from collapsing is the host failure detection. By deploying the scheduler on all nodes, it is made possible that any node can take over the host role if the current host fails. LWB is evaluated in different scenarios following different traffic patterns, e.g. many-to-one communication with light, heavy, and fluctuating traffic. In their experiments, the authors used different test beds with the number of nodes ranging from 26 up to 260. In all scenarios, LWB shows that it is able to adapt to changing conditions. LWB achieves a very high packet delivery rate of almost 100%. Based on the protocol operation described and the results, LWB is definitely a candidate solution for lighting control applications. Hence, our proposed solution will be compared with LWB.

5.4.2 Operation of RLL

RLL is designed in a way that a synchronous message delivery to all members of the cluster tree is accomplished. To achieve this, RLL applies a number of forwarding rules, utilizing the Tier information of CIDER. However, a message can be issued by any device in a network. Hence, RLL can not only be used for lighting application, but also for any other applications requiring QoS or QoE. When using RLL, it is guaranteed that only the new commands are forwarded; the commands that have already been added to the queue, waiting for their transmission, are dropped from the queue. First, Section 5.4.2.1 shows which forwarding rules are applied with RLL, and in Section 5.4.2.2, it is explained how the slotframe structure changes compared to CIDER.

5.4.2.1 Dissemination Procedure

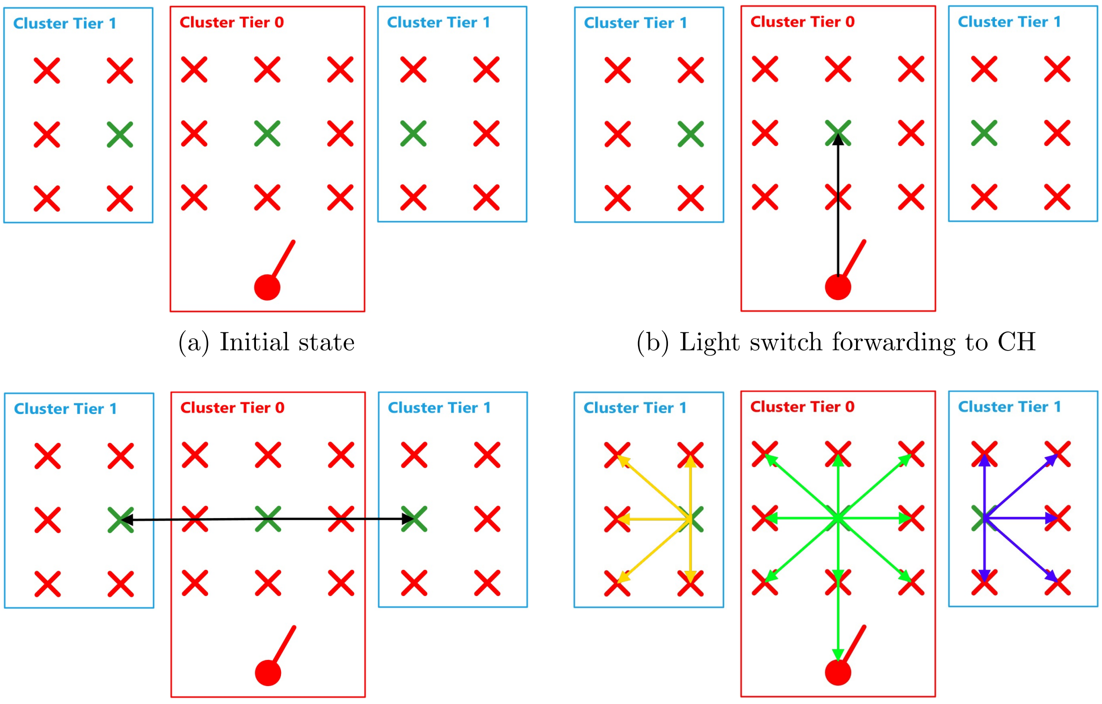
This section explains the dissemination procedure of RLL. RLL exploits the topology formed by the colouring algorithm. The general dissemination process in RLL is that a light switch generates a control message upon interaction by a user. The light switch forwards this control message to its CH. The CH then forwards the message to its parent and child CHs. To avoid redundant transmissions, the receiving CH (parent or child) forwards the message only in one direction, either to its child CHs or to its parent CH. To determine the direction of a message, the CH can retrieve this information from its neighbour table and thus forwards the message in the right direction. Once all CH's have received the message, they will all simultaneously forward the message to their CSs. Anyhow, this procedure is described in Algorithm 1, and the procedure is also described below. A light switch creates a new control message with a unique control message ID. Based on this ID, the message queue is able to differentiate between new and old messages. If there is an old message in the queue (which means an old and obsolete command), the queue deletes the old message and adds the new command to the queue. This ensures that only the most recent message is disseminated. After the relevance test, the CS forwards the message to its CH. After the CH has received the message, it checks the tier from which the message originated. Based on this information, the CH knows to whom it has to forward the message, e.g. if the message source is a lower tier, the CH forwards the message only to the higher tier and to its own CSs. On the other hand, if the message source is a higher tier, the node forwards the message to other existing higher tier CHs, the lower tier CHs, and its own CSs. The other receiving CHs then follow the same procedure until the message is disseminated throughout the whole network and each associated node has received the message.

Figure 5.6 shows the dissemination procedure based on an example. A network with several wireless nodes is assumed (marked as an “X”). The nodes are separated in three clusters, with the largest cluster at tier 0, marked with a red frame, and two equal-sized clusters at tier 1, each with a light blue frame. A CS node is coloured red, a CH is coloured green, and the light switch is also coloured red. The light switch in this example is connected to the CH of Tier 0. First, the light switch generates a new control message and forwards it to its coordinator (see Figure 5.6b). The CH then forwards the message to its child CH’s (see Figure 5.6c). Once all CHs have received the control message, they forward the message simultaneously to all their CSs on different channels (see Figure 5.6d). In doing this, each cluster uses a different channel, each indicated by a different colour.

Algorithm 1 Data forwarding algorithm

```
if CS then
    Check MSG source
    if From Application Layer then
        Forward to CH
    else
        return
else
    forward MSG:
    switch MSG source do
        case from application layer
            if CH higher tier exist then
                forward to CH higher tier
            if CH lower tier exist then
                forward to CH lower tier
        case lower tier
            if CH higher tier exist then
                forward to CH higher tier
        case higher tier
            if other CH higher tier exist then
                forward to CH higher tier
            if CH lower tier exist then
                forward to CH lower tier

    if CS exist then
        forward MSG to CS
```



(a) Initial state

(b) Light switch forwarding to CH

(c) Exchange between CHs

(d) Simultaneous dissemination to all CSs

Figure 5.6: Message dissemination with RLL

5.4.2.2 Slotframe Structure

This section discusses the slotframe structure of RLL. The slotframe design is the key feature of RLL. To this end, RLL utilizes the information gathered during the clustering and colouring processes and is tightly linked with CIDER to take full advantage of the clustering approach. As different from the previous slotframe structures used for CIDER and Colouring (see Section 5.1.1), RLL uses three different link options.

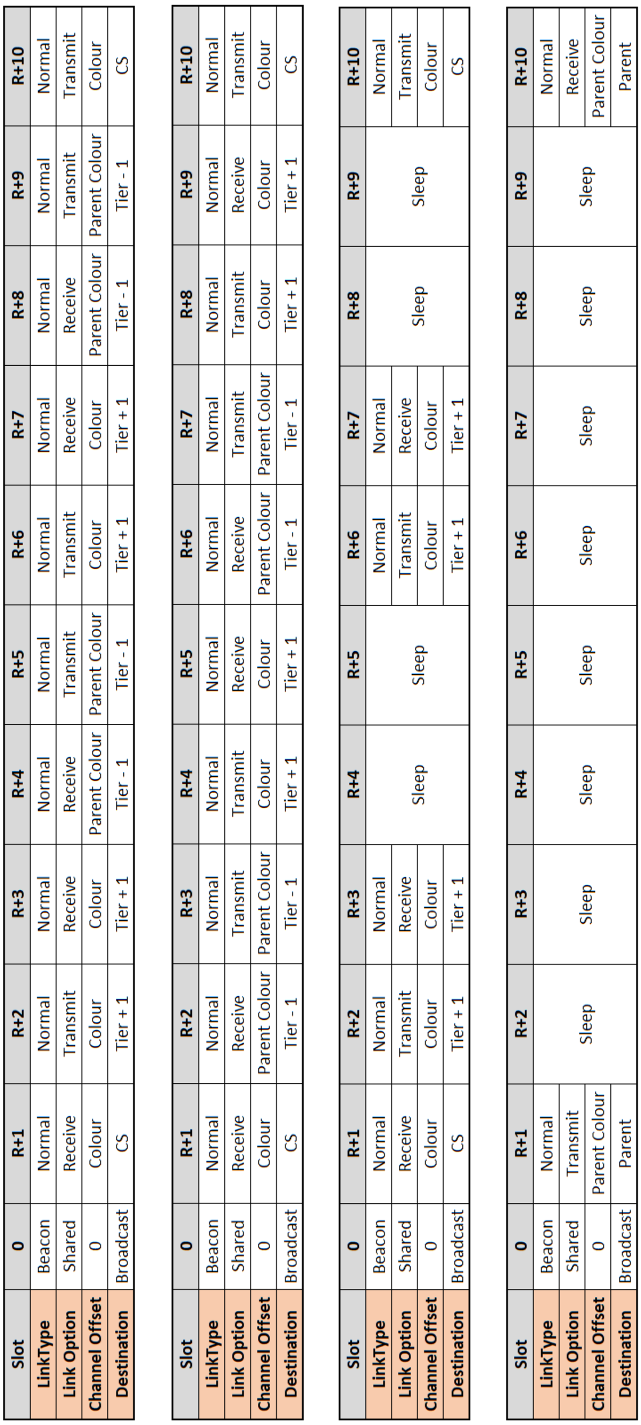
1. **Shared**, which is used for transmitting and receiving the *Beacon* message in timeslot 0. Except for this timeslot, the Shared option is not used.

2. **Receive**, when this option is set, the node will turn on its radio and listen to the channel for incoming messages

3. **Transmit**, using this option, a node checks its queue for data that is available to transmit, if so, the message will be sent, otherwise, the node turns

off its radio.

As described in Section 5.1.1, RLL also uses a total number of timeslots within the slotframe calculated by Equation 5.2. Hence, for RLL, the minimum number of timeslots is 11, and the maximum is 101. One timeslot is always used for transmitting and receiving the beacon message, and $10 \times R$ timeslots are used for exchanging data messages, of which eight are used to exchange messages between clusters and two for exchanging them within a cluster.



Slot	0	R+1	R+2	R+3	R+4	R+5	R+6	R+7	R+8	R+9	R+10
LinkType	Beacon	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal
Link Option	Shared	Receive	Transmit	Receive	Receive	Transmit	Transmit	Receive	Receive	Transmit	Transmit
Channel Offset	0	Colour	Colour	Colour	Parent Colour	Parent Colour	Colour	Colour	Parent Colour	Parent Colour	Colour
Destination	Broadcast	CS	Tier + 1	Tier + 1	Tier - 1	Tier - 1	Tier + 1	Tier + 1	Tier - 1	Tier - 1	CS

(a) CH: Even tier

Slot	0	R+1	R+2	R+3	R+4	R+5	R+6	R+7	R+8	R+9	R+10
LinkType	Beacon	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal	Normal
Link Option	Shared	Receive	Receive	Transmit	Transmit	Receive	Receive	Transmit	Transmit	Receive	Transmit
Channel Offset	0	Colour	Parent Colour	Parent Colour	Colour	Colour	Parent Colour	Parent Colour	Colour	Colour	Colour
Destination	Broadcast	CS	Tier - 1	Tier - 1	Tier + 1	Tier + 1	Tier - 1	Tier - 1	Tier + 1	Tier + 1	CS

(b) CH: Odd tier

Slot	0	R+1	R+2	R+3	R+4	R+5	R+6	R+7	R+8	R+9	R+10
LinkType	Beacon	Normal	Normal	Normal	Sleep	Sleep	Normal	Normal	Sleep	Sleep	Normal
Link Option	Shared	Receive	Transmit	Receive			Transmit	Receive			Transmit
Channel Offset	0	Colour	Colour	Colour			Colour	Colour			Colour
Destination	Broadcast	CS	Tier + 1	Tier + 1			Tier + 1	Tier + 1			CS

(c) CH: Tier 0

Slot	0	R+1	R+2	R+3	R+4	R+5	R+6	R+7	R+8	R+9	R+10
LinkType	Beacon	Normal	Sleep	Sleep	Sleep	Sleep	Sleep	Sleep	Sleep	Sleep	Normal
Link Option	Shared	Transmit									Receive
Channel Offset	0	Parent Colour									Parent Colour
Destination	Broadcast	Parent									Parent

(d) CS

Figure 5.7: RLL slotframe structure

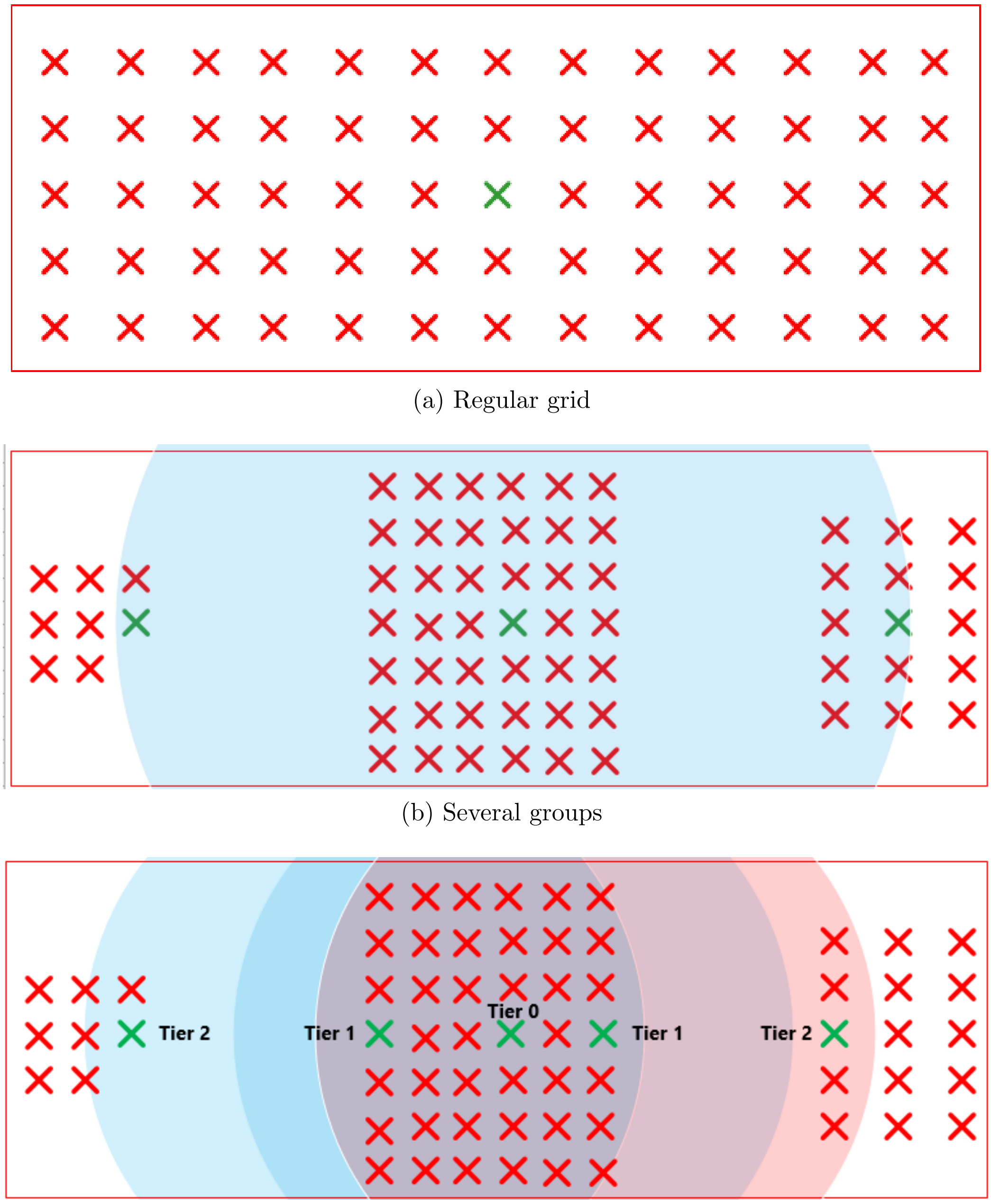
Based on this information, four different slotframes are needed. One for CHs assigned to an even tier (see Figure 5.7a), one for CH's assigned to an odd tier (see Figure 5.7b), one special slotframe for the CH on Tier 0 (see Figure 5.7c), and one for all CSs (see Figure 5.7d). The slotframe for the Tier-0 CH is the same as the one for an even tier, except that there are no parent CHs. Hence, the Tier-0 CH will remain in sleep mode during Timeslots 4, 5, 8, and 9. The CSs only communicate with their parent nodes; therefore, the CS nodes are only active during Timeslots 0, 1, and 10. The slotframes are designed in a way that they repeat after ten timeslots have elapsed. The length of the slotframe only determines how often the beacon is transmitted, which means that with a shorter slotframe length, the beacon is transmitted more often, and therefore, the network is synchronized better. This is useful in harsh environments with a lot of interference; while in networks where the interference is negligible, a longer slotframe can be set since it is then less likely that nodes miss a beacon. With such a slotframe structure, it is possible to cover 4 hops within 10 data slots. Hence, based on the 200 ms requirement, it is possible theoretically to cover 8 hops. These 8 hops include the transmission from the light switch to its parent, followed by a maximum of 6 transmissions between the CHs, and the final transmission to all CSs. If 8 hops need to be covered, the first set of CSs will receive the message after 9 timeslots, and the second set after 19 timeslots. This is dependent on the topology and on which tier the light switch is located.

As mentioned in the beginning of this section, RLL utilises the information gathered by CIDER and the Colouring algorithm. First, using CIDER, each node is assigned to a CH. Therefore, each node is able to determine whether is located on an odd tier or an even tier and can set its slotframe accordingly. Furthermore, a CH knows exactly from which tier a message has originated and can forward a message during the correct timeslot. Second, the assigned colour represents the channel offset needed for TSCH. The colouring algorithm accomplishes assigning the neighbouring clusters to a different colour.

5.5 Scalability of D^3LC

Since CIDER currently only supports 50 CSs per CH, as stated in Section 5.2.4.3, in this section it is discussed how scalable CIDER is in different scenarios. The aim of CIDER is to form clusters as large as possible with the first CH placed as centrally as possible in terms of physical location in relation to the network area and the radio propagation conditions. This means that the protocol forms clusters based on the radio propagation conditions and places the first CH in the centre of the propagation area rather than the geographical area. In Figure 5.8 a few examples are presented.

The examples are only a small selection of possible network topologies with Figure 5.8a representing a regular grid deployment of a wireless lighting network where lamps are distributed with equal distances between each other. The assumption here is that the CH (marked in green) is located in the centre of the room due to the operation of CIDER. This one CH is able to cover the whole network without the need of selecting another CH, as long as the number of CS do not exceed the maximum possible 50 members as stated in Section 5.2.4.3. This is considered the best case scenario for a deployment using CIDER. However, since it is usually not the case to have a best case scenario, next it is discussed what happens if one CH is not able to cover the whole network, or even not being connected to some parts of the network. The second example presented in Figure 5.8b represents a grouped setup which for example can be found in office buildings covering different office rooms. Here the first CH is still located in the centre of the area since the largest amount of nodes is located there. This CH is connected to most of the network. Since the CH cannot cover the whole network the CH would due to the CIDER process elect additional CHs to cover the whole network. In the last example in Figure 5.8c it is assumed that the CH is only connected to the group located in the centre of the area, and can not see any nodes in the left group and the right group. At the beginning, the first CH will assume that the network only consists of the nodes it can see directly, basically the group in the centre of the area. However, during the CIDER finalisation process this CH would be notified by other nodes which are located closer to the edge of this group that there are still nodes not covered. The CH then would



(a) Regular grid

(b) Several groups

(c) Shorter transmission range

Figure 5.8: Example of different network topologies

elect additional CHs closer to the edge of the room (marked in green and labelled with *Tier 1*). The Tier 1 CHs are then able to cover partly the groups on the left and right of the area where they elect additional CHs (marked green and labelled *Tier 2*). This process continues until the whole network is covered. Of course, if for example one of the groups is completely disconnected there is no possibility to cover this group. This is not a problem which arises only with CIDER, this is a general problem which happens with wireless communication technologies.

The scalability of D^3LC can not only be defined on the basis on how CIDER handles different types of topologies. Furthermore, it is important to look into how many CHs can be supported with CIDER or in relation of the overall D^3LC-Suite. With the design of RLL a slotframe structure is used to support up to 8 hops. This means that if a light switch is connected to a Tier 3 CH, the forwarding rules of RLL in combination with the slotframe structure be forwarded up to the Tier 0 CH and then back down to any other CH up to Tier 3. How many CHs are distributed on all stages (Tier 0 to Tier 3) is not limited by CIDER itself. More importantly the number of child CHs are limited by the channel offset reuse of the colouring algorithm. The aim of the colouring algorithm is to assign a unique channel offset to each CH, where there are a total of 16 channel offsets available. Now, assuming that there are 16 CH in a very dense network where CHs are in communication range of each other. The colouring algorithm would assign unique channel offset to each of the CHs. Adding an additional 17th CH would mean to reuse an already used channel offset and two CHs would then communicate on the same channel. Therefore, in very dense environments where each CH is in communication range of each other the number of total CHs is limited to 16 which results in a total of 800 nodes. However, if the environment is less dense and not all CHs are in communication range with each other, more than 16 CHs can be supported. The limiting factor then is how many hops are needed to cover the network. If the number of hops is larger than 8, the dissemination with RLL takes more time since the nodes outside the 8 hops would only receive the control message with a delay of an additional slotframe duration.

In general it can be said that CIDER is a scalable approach, since CIDER is designed in a way to detect uncovered nodes and continue the clustering process until each and every node is either a CH or CS. Additionally if more nodes are added to the network, e.g. in a location where a CH is close by, but the CH reached already the limit of 50 members and no other CH is nearby to cover the new additional node, the clustering process would simply restart triggered by the CH detecting this problem.

5.6 Tuning for other applications

The implementation of D^3LC discussed in the previous sections is tuned to fit the use case of dense wireless lighting networks. However, D^3LC can not only be used for dense wireless lighting applications, since the parameters can be tuned. In this section the most important parameters are listed which can be tuned and how they affect the performance of D^3LC.

- **Weighting parameters** Currently $w_1,\ w_2,\ w_3$ are set to 0.5, 0.3, and 0.2. By tuning these parameters the CH distribution and the cluster size are affected. For example, by giving more importance to w_2 the influence of the cluster degree changes. Another example is to change parameter w_3 which indicates how well a cluster will be connected. If w_3 is increased the connectivity within a cluster is also improved.

- **Timing parameters** The current timing parameters are set to values fitting the dense wireless lighting application where usually a large number of nodes are in a network. To improve the setup phase of CIDER these timing parameters can be changed, e.g. if the network is rather small, the time-out to detect if CIDER has finished can be smaller as well, on the other hand if the network is larger this time-out should be increased.

5.7 Conclusion

This chapter introduces the novel D^3LC Suite, a protocol suite for control message dissemination in large and dense wireless lighting control networks, as well as for other dense wireless networks requiring time bounded message broad- or multi-casting to many nodes. D^3LC includes three protocols. The CIDER protocol forms a cluster tree topology in dense static wireless lighting networks. To ensure that the clusters formed do not interfere with each other, a colouring algorithm is applied, which assigns a colour to each cluster. The colour is then translated into a channel offset which ensures that each cluster communicates on a different channel. Last, the novel RLL protocol has been introduced. RLL is designed to reduce the redundant message exchange that was found in state-of-the-art dissemination protocols such as flooding. D^3LC is designed to provide a low-latency data dissemination with a deterministic data reception to provide a high QoE. The next chapter evaluates the D^3LC based on the performance parameter defined in Chapter 3.

Chapter 6

Evaluation and validation

This chapter analyses the performance of a dense wireless lighting control network using D^3LC as underlying communication stack. Section 6.1 provides a convergence analysis of CIDER, showing that CIDER is capable of successfully forming a cluster tree. In studying its feasibility, the RLL protocol was analysed using the OMNeT++ simulation engine. The results are presented in Section 6.2. In Section 6.3, CIDER is analysed in the different environments described in Chapter 3. CIDER's performance is compared with LWB in Section 6.4. Finally, this Chapter is concluded with Section 6.6. The evaluation of CIDER is based on the criteria defined in Section 3.3.

6.1 Convergence analysis of CIDER

The convergence of CIDER was analysed as part of the DEWI project with the help of Mahmoud Talebi, a colleague from TU Eindhoven, using methods described in [125]. While the methods and algorithms used were defined by Mr. Talebi, the protocol operation and use case were contributed by the author. This section presents the results of this joint work. They are presented for the sake of completeness and as an independent verification of the CIDER protocol operation. The verification of CIDER consists of rigorous tests, (mathematically) showing that in any dense network, for every possible sequence of events, CIDER is capable of successfully forming a cluster tree in the network. The aim of the analyses was to investigate whether the following property holds for every set of possible events:

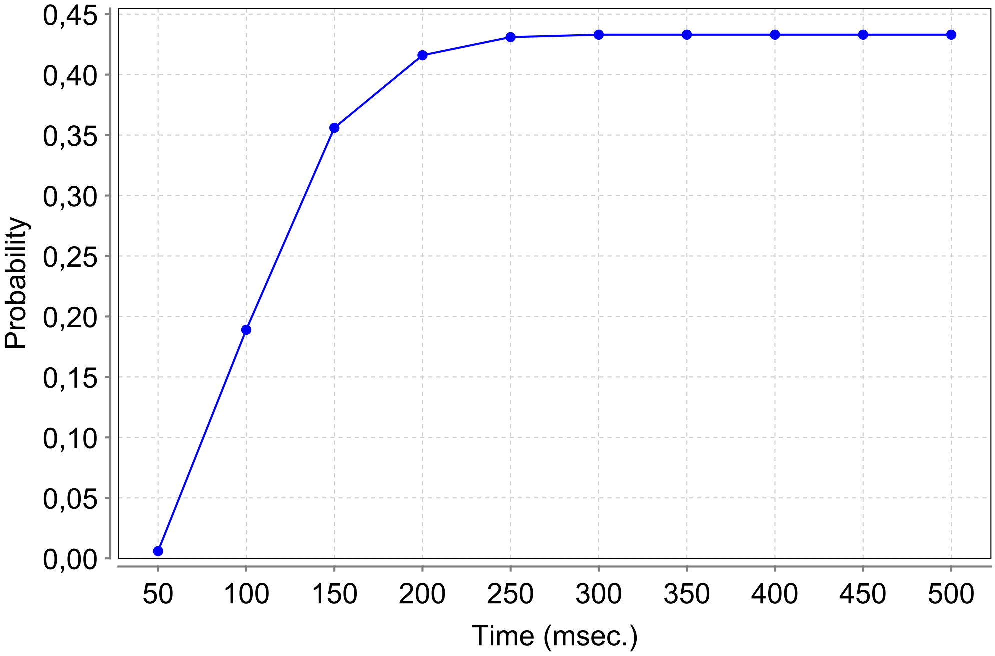


Figure 6.1: Probability of becoming a cluster slave

P1: What is the probability of all nodes being either a cluster head or a cluster slave by time t? To show that CIDER converges, it is necessary to prove that by increasing the time t, it is possible to get *P1* arbitrarily close to 1. However, due to the complexity of the protocol, instead, an intermediate mathematical model in which given a certain probability p, we can find a time t at which $P1 \geq p$ was derived. For this purpose, the PRISM model checker [2, 84] was used (for the remainder, called PRISM). PRISM is a tool for formally modelling and analysing systems, which is capable of verifying systems of various types for their correct operation, in particular, those which exhibit a timed and probabilistic behaviour. For verifying CIDER using PRISM, Markov Decision Processes (MDPs) were chosen as the primary means to specify the behaviour of nodes in the network. Since PRISM does not provide mechanisms for modelling phenomena such as message broadcasting and the collisions that occur in wireless communication, these parts were added manually to the specification. Subsequently, the models are quantified using the properties expressed in Probabilistic Computation Tree

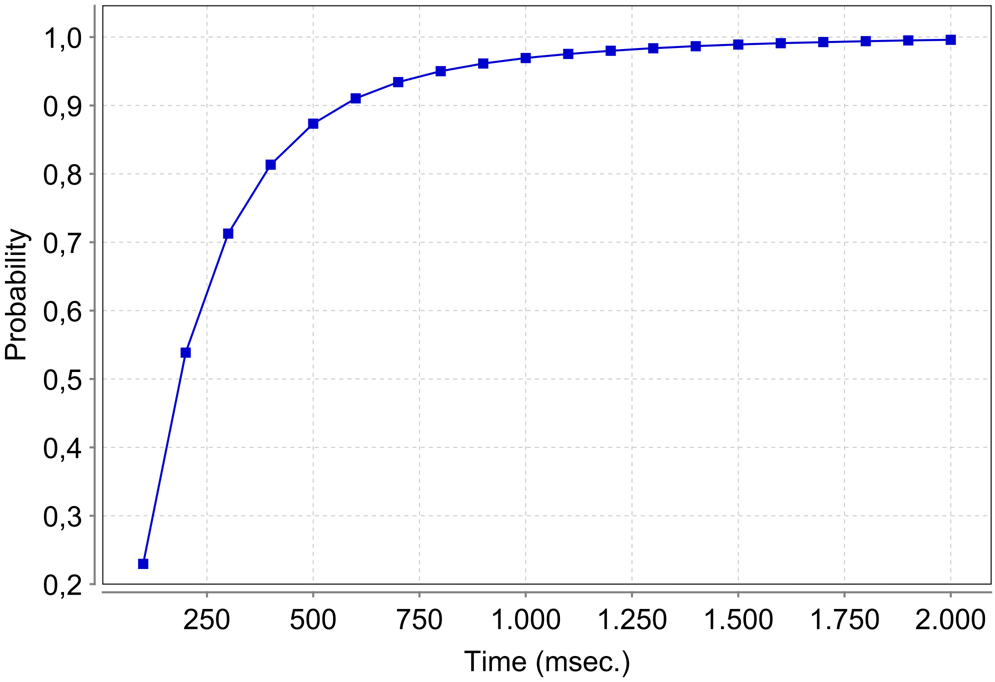


Figure 6.2: Probability of the successful termination of CH competition with a unique CH

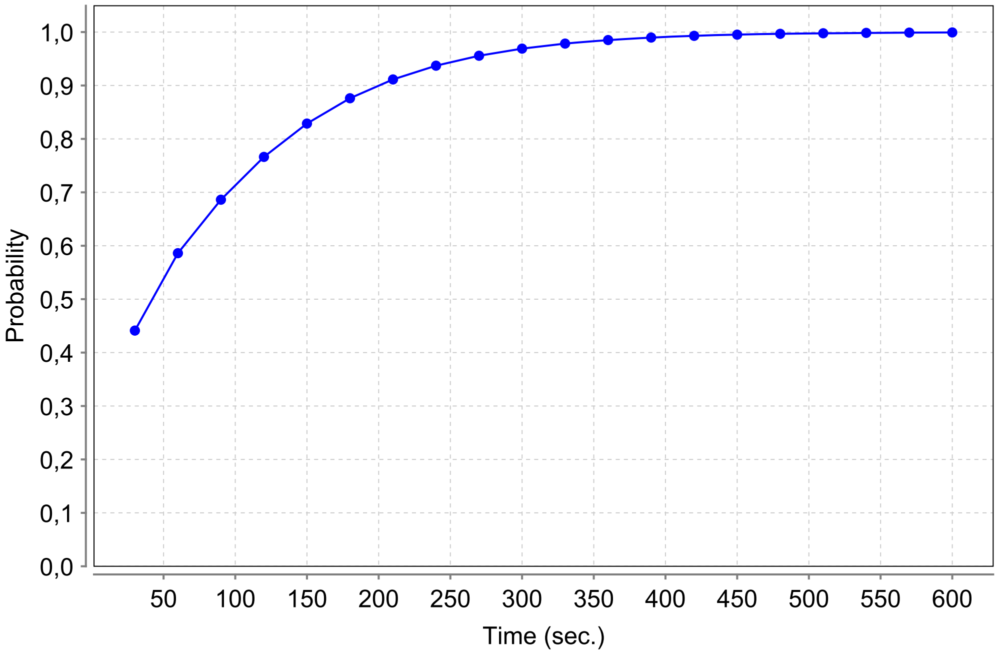


Figure 6.3: Probability of the successful completion of CIDER

Logic (PCTL) formulas. Due to the complexity of CIDER, the modelling was broken down into three parts which map to the different CIDER stages. These parts are:

1. The discovery phase of CIDER.

2. The cluster head competition of the setup phase of CIDER.

3. The completion process of the setup phase of CIDER.

For quantifying the probability of *P1*, the following quantifications are performed on each of the three models above:

1. the probability of a node becoming a cluster slave after the termination of the utility update message exchange of the discovery phase is shown in Figure 6.1.

2. The above probability is then used in the model for the cluster head competition in the setup phase. The probability of all the nodes either being a

cluster head or a cluster slave at the end of the competition (the *successful termination*) is determined here. Additionally, the probability of ending up with a single cluster head in the network (*uniqueness*) was determined. The results of the verification for this study are shown in Figure 6.2.

3. Finally, assuming that there is one Tier-0 cluster head, the probability of CIDER completion was independently quantified until time t (*successful completion*). These are presented in Figure 6.3.

For each stage, the expected time (i.e. number of steps) until successful termination was determined, using the calculation of *reachability rewards* in MDP models. The total expected time to CIDER completion for a network with 4 cluster heads (1 in Tier 0 and three in Tier 1) with an arbitrary number of cluster slaves was determined as 1 minute and 17 seconds. In addition, according to the same model, with a probability of more than 0.975, the CIDER completion occurs in less than 5 minutes. In lighting networks with installation times of up to several days in large deployments, 5 minutes for an initialisation of the network is an excellent time especially, since the initialisation happens only once at the end of the installation phase and remains unchanged for a longer period. However, to ensure that always the best topology for the current system is used, the system can be programmed so that a reinitialisation happens overnight.

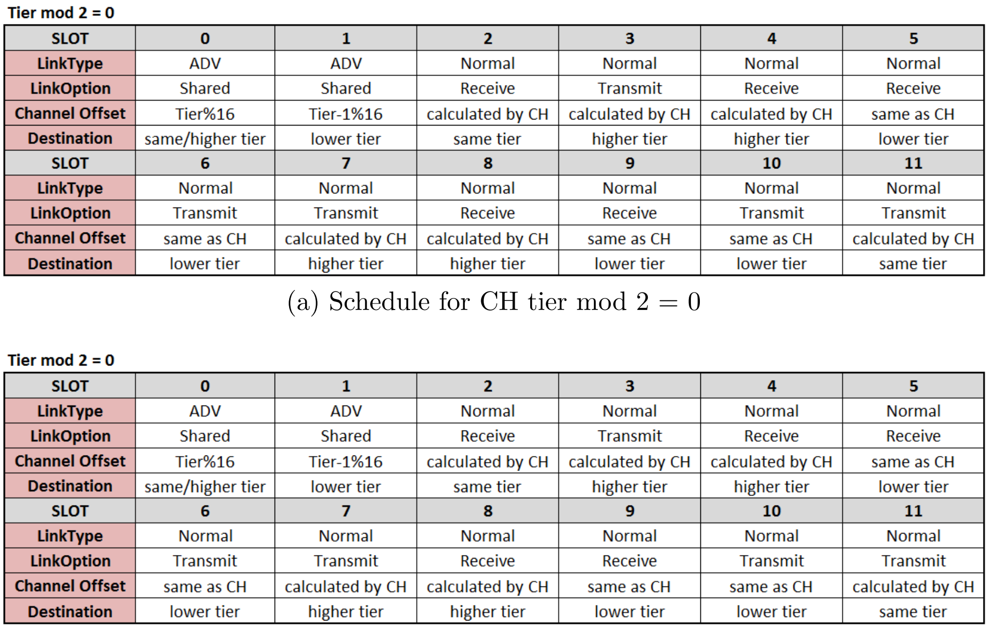
6.2 Simulation-based evaluation of RLL

The simulation model described in Section 3.1.2 is used to investigate the network performance with RLL. This section investigates the performance of RLL only as a feasibility study. The simulation configuration follows a similar configuration as defined in Section 4.2. The application is the same as simulating a user turning a dimming knob for two seconds, generating a new command every 100 ms; hence, a total of 20 messages per burst. Per simulation run, a total of 1000 bursts were simulated. To investigate the performance of different network densities, the simulation includes configurations for 56, 100, 200, and 400 nodes. As per Chapter 4, the transmit power is usually set to the maximum value available; therefore, the transmit power is fixed at 0 dBm. The simulation parameters can be found in Table 6.1. The exact configuration can be found in Appendix F. In the simulation, the focus was only on RLL, while a simple clustering protocol was used to set up the topology. The rationale why only RLL was evaluated in simulation is based on the fact that CIDER and colouring are network configuration protocols where it is important that they converge in any given situation. The convergence was already discussed in Section 6.1. Implementing CIDER and colouring in the simulation model creates no new knowledge, except that it would have been another proof that they converge. To simplify the simulation a more simple clustering protocol is implemented toe evaluate RLL in simulation and is described below. The clustering is done in a way that the CHs are separated by at least a certain distance while keeping them in communication range of each other. The topology built with this approach forms a cluster tree. This approach worked fine in a simulation, however, the simulation showed that the initialisation by the simple approach was in the order of hours, up to a day. Therefore, this simple approach could not be used in experiments but was sufficient for a simulation. One node, determined at the beginning, is the Tier-0 CH. The starting node changes for each simulation run so that a new topology is formed each time. The CHs on Tier 1 become the child CHs of the initial CH at Tier 0, and those CHs are away at least by the predefined distance. All nodes within this distance become cluster slaves (CSs) of that certain tier, e.g. a CS of a CH at Tier 0 is also located on Tier 0. The same method continues for the next tiers, e.g. a CH on Tier 2 becomes a child CH of a CH on Tier 1 until the cluster tree topology is formed.

This simulation represents an early approach to RLL, and a slightly different slotframe structure and channel allocation mechanism were used. The slotframe structure is shown in Figure 6.4. The main difference from the one defined in Section 5.1.1 is that two beacon slots are used for the synchronisation, one slot for each tier. Also, the length of one slotframe was fixed at two beacon slots and 10 data slots. However, this was later found to be obsolete since the TSCH implementation of Contiki provides a mechanism for synchronising, using only one beacon slot, by simply attaching the timing information to the messages when the coordinator is out of reach. The channel allocation mechanism for the early simulation was that a CH provides the channels that should be used for

Parameter	Setting
Application	Dimming
Traffic	Message Burst Duration: 2 seconds New Message: 100 ms
Network Density	56 Nodes up to 400 Nodes in an area of (80 x 20 x 3.3) m
Transmit Power	0 dBm
Number of simulation runs	20 per configuration
Confidence level of Akaroa	0.95

Table 6.1: Simulation configuration for RLL



Tier mod 2 = 0

SLOT	0	1	2	3	4	5
LinkType	ADV	ADV	Normal	Normal	Normal	Normal
LinkOption	Shared	Shared	Receive	Transmit	Receive	Receive
Channel Offset	Tier%16	Tier-1%16	calculated by CH	calculated by CH	calculated by CH	same as CH
Destination	same/higher tier	lower tier	same tier	higher tier	higher tier	lower tier
SLOT	6	7	8	9	10	11
LinkType	Normal	Normal	Normal	Normal	Normal	Normal
LinkOption	Transmit	Transmit	Receive	Receive	Transmit	Transmit
Channel Offset	same as CH	calculated by CH	calculated by CH	same as CH	same as CH	calculated by CH
Destination	lower tier	higher tier	higher tier	lower tier	lower tier	same tier

(a) Schedule for CH tier mod 2 = 0

Tier mod 2 = 0

SLOT	0	1	2	3	4	5
LinkType	ADV	ADV	Normal	Normal	Normal	Normal
LinkOption	Shared	Shared	Receive	Transmit	Receive	Receive
Channel Offset	Tier%16	Tier-1%16	calculated by CH	calculated by CH	calculated by CH	same as CH
Destination	same/higher tier	lower tier	same tier	higher tier	higher tier	lower tier
SLOT	6	7	8	9	10	11
LinkType	Normal	Normal	Normal	Normal	Normal	Normal
LinkOption	Transmit	Transmit	Receive	Receive	Transmit	Transmit
Channel Offset	same as CH	calculated by CH	calculated by CH	same as CH	same as CH	calculated by CH
Destination	lower tier	higher tier	higher tier	lower tier	lower tier	same tier

(b) Schedule for CH tier mod 2 = 1

Figure 6.4: Schedule

its child nodes. As soon as a CH is associated with its parent CH, it sends a request for retrieving the channel offsets. In general, a parent CH keeps track of the assigned channel offsets for its child CHs, and those child CHs keep track of the channel offsets of their child CHs, and so on. As soon as a CH receives a channel-offset request, it starts calculating the channel offsets. However, this mechanism was found to be not optimal, as this approach requires a CH to keep track of all the channel offsets used. Also, when there are two Tier-1 CHs with each a child CH, there is no guarantee that both child CHs are assigned to the same channel offsets. This created the need to find a better approach leading to the colouring algorithm presented in Section 5.3.

In Section 6.2.1, the results are analysed for an increasing network density. Section 6.2.2 presents how RLL is visual observed by a user, similar to Section 4.7.

6.2.1 Increasing network density

This section demonstrates the scalability of the RLL protocol with an increasing network density while the area remains the same, with the number of devices in the network increasing from 56 nodes to 400 nodes. In Figure 6.5a, it can be seen that the latency is almost identical for all densities. For 99% of all received messages, the latency is below 200 ms. The CDF of the latency can be explained as follows. In the region of 0 to 90 ms, first, the CHs receive the message, hence there is only a minimal increase in the percentage of how many nodes have received the message. This is due to the slotframe structure and the random wait period between two bursts since a command is created every 100 ms and RLL purges the older messages stored in the outgoing queue when newer data is available. The majority of the messages are received between 100 and 200 ms. Based on the slotframe structure, lamps receive a control message at the earliest after 100 ms but latest, after 200 ms. The final 1% of the messages are received with a latency of higher than 200 ms. This is caused by the last message of a burst, which is not purged by a newer message and can, therefore, be delayed for more than 200 ms.

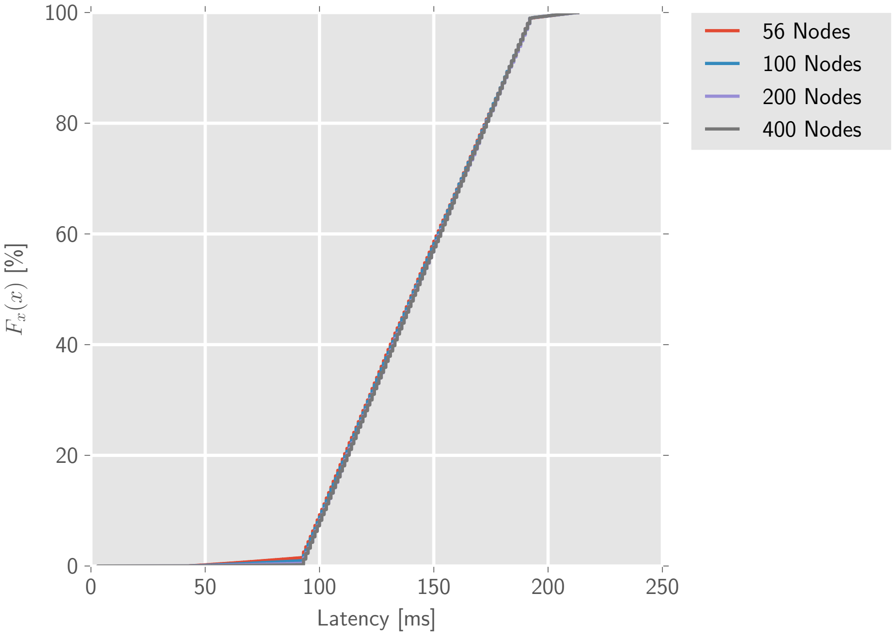
The PLR is below 1% for all densities, even below 0.01% (Figure 6.5b). Due to the schedule where each link has a timeslot and a channel assigned, no collisions occur, and only a few messages are lost due to channel conditions. The simulation results show that RLL is capable of fulfilling the requirements for a lighting control network; of course, the channel condition was nearly perfect, which made it necessary to evaluate RLL in a real-world deployment. However, while RLL performs very well, the early test showed that it is necessary to develop a better mechanism for forming the network topology. With the protocols used in this early version, the time taken to form the network sometimes exceeded several hours, which is unacceptable for a lighting control network. Also, it was learned that increasing the slotframe duration to a multiple of 10 data slots can further optimize the overall latency. These findings were taken into account while designing CIDER and selecting the colouring algorithm.

6.2.2 User perception of RLL

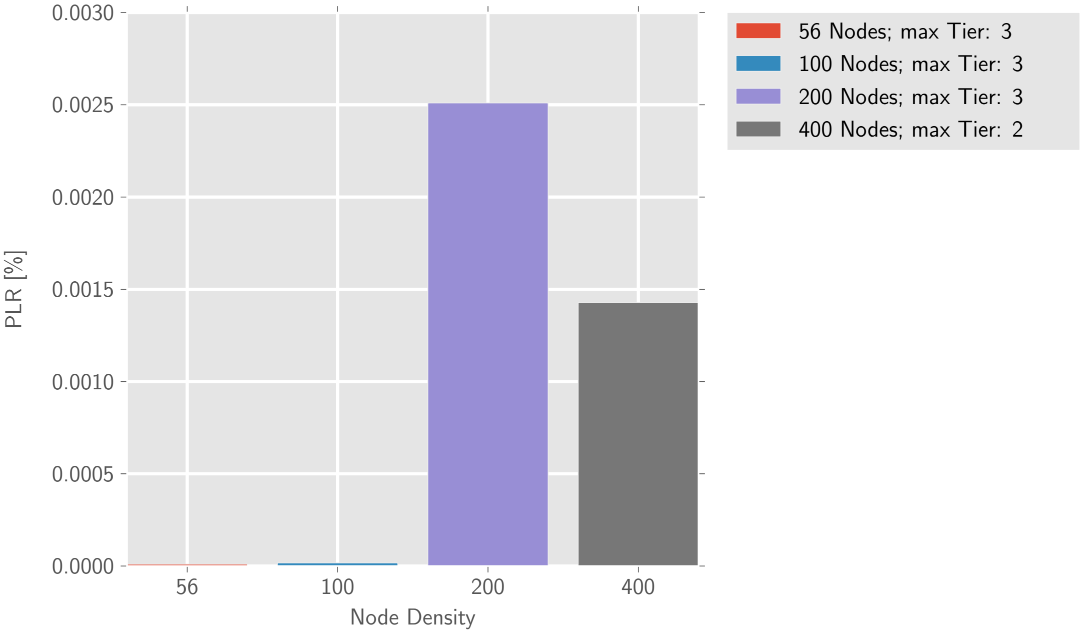
As in Section 4.7, RLL is also analysed in terms of user perception. The network setup for the visualisation is the same as before. The network contains 400 lamps, and they are distributed in a grid manner with equal distances between all lamps. The area is 80m x 40m x 3.3m (L x W x H) with the light switch located at (0m, 40m, 1.2m), representing an office space. Compared to the previous results of Flooding (Section 4.7, Figure 4.7) and Probabilistic Forwarding (Section 4.7, Figure 4.8), RLL also fulfils the requirement for expected user behaviour. Due to the usage of timeslots in RLL, it is possible to control the delivery of messages to some extent. As shown before, the majority of the nodes not only receive the message within 200ms but also in the same timeslot as shown in Figure 6.6. Using RLL, the light level changes at the same time and as a result the user expectation in terms of synchronicity and system behaviour is fulfilled.

6.3 Empirical evaluation

In this section, the performance of D^3LC is analysed, based on the experimental setup described in Section 3.2. The overall performance of D^3LC is shown based on the performance in three different environments. The key performance parameters are described in Section 3.3. In the experiments, a slotframe, consisting of 50 data slots and one beacon slot, was used with a slot duration of 10 ms. This results in a slotframe duration of 510 ms. The total of 51 timeslots per slotframe

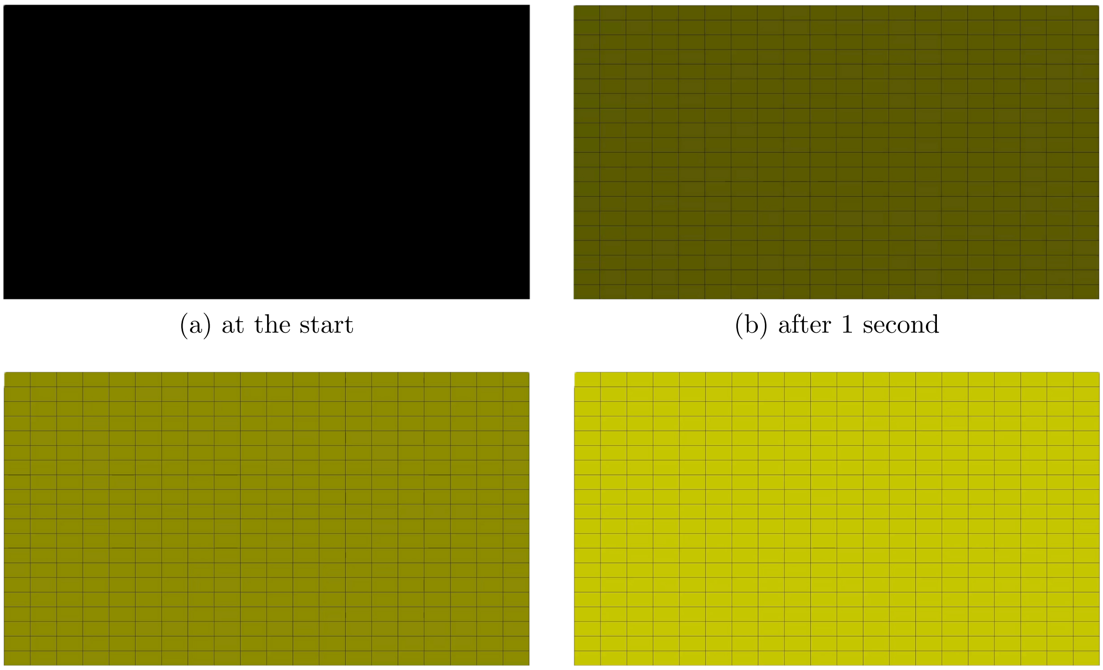


(a) Empirical CDF, RLL with increasing network density



(b) Average PLR, RLL with increasing network density

Figure 6.5: Network performance of RLL with increasing network density



(a) at the start

(b) after 1 second

(c) after 1.5 seconds

(d) at the end

Figure 6.6: Message dissemination with RLL as a dimming application

were chosen because the simulation showed that a longer slotframe duration can further optimize the overall latency. However, it is not possible to increase the slotframe duration to a very large number of timeslots since this would increase the probability that nodes lose their synchronisation with the network coordinator. In early experiments with slotframes of more than 51 slots, e.g. 100 slots and where the nodes were deployed over a large area, and/or with an interferer in the direct neighbourhood, it was found that synchronisation was lost more often. Therefore, the number of slots per slotframe was limited to reduce synchronisation failures. The remainder of this section is as follows: in Section 6.3.1, the results of experiments conducted in the Sports Hall environment are discussed. Section 6.3.2 discusses the results for the Laboratory environment, and Section 6.3.3 presents the results of experiments conducted in the Apartment environment.

6.3.1 Sports Hall

D^3LC is first analysed in a mostly interference-free environment. As stated in Section 3.2.2.1, the Sports Hall environment is used as reference experiment for this thesis. The interference level was very low while conducting the experiments with signal levels below -75 dBm. Therefore, the interference is negligible. The results are presented in Figure 6.9. First, three experiments were conducted with a large-scale setup, Experiments 1, 2, and 3. After each experiment, the network was completely reset, and all processes were started from the beginning.

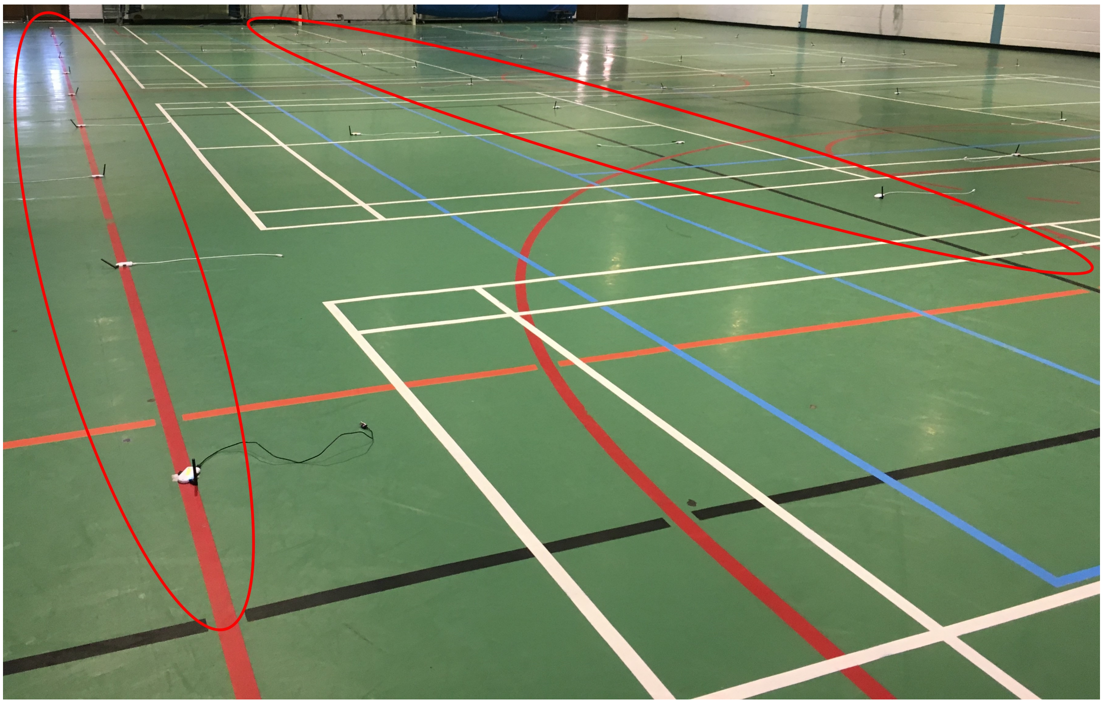


Figure 6.7: Large-scale node deployment, Sports Hall

In this setup, only one CH was elected during the network formation. After the results of each experiment were collected, the network was reset, and the network formation process was started again. The latency results (see Figure 6.9a) show that for all three experiments, the trend is similar, with only small deviation from each other. Also, the trend looks similar to the simulation results. The latency for all the three experiments is between 200 ms and 250 ms for 99% of the received messages. The overall latency is slightly higher than the one achieved during simulations. The reasons for that was found in the Contiki OS. It appeared that sometimes, Contiki was not able to process the incoming message fast enough. Hence, it was in the case in 10% to 20% of all messages that a message was not forwarded in the next timeslot, but in the equivalent of the one following that timeslot. Nonetheless, this slightly increased latency is still acceptable in terms of user expectations. The PLR is below 1% for all three experiments, see Figure 6.9b. This shows that the interference almost shows no effect on the performance. However, the first result represents the simplest setup possible with only one CH and with only two hops that needed to be covered.

The second set of experiments represents a more complex setup with multiple CHs arranged in groups (see Figure 6.8), as described in Section 3.2.2.1. Here, only one experiment was conducted (Experiment 4) due to the limited access to the sports hall. The network formation process formed a network with three CHs, one as Tier-0 CH and two as Tier-1 CH, with the Tier-0 CH as the parent. This

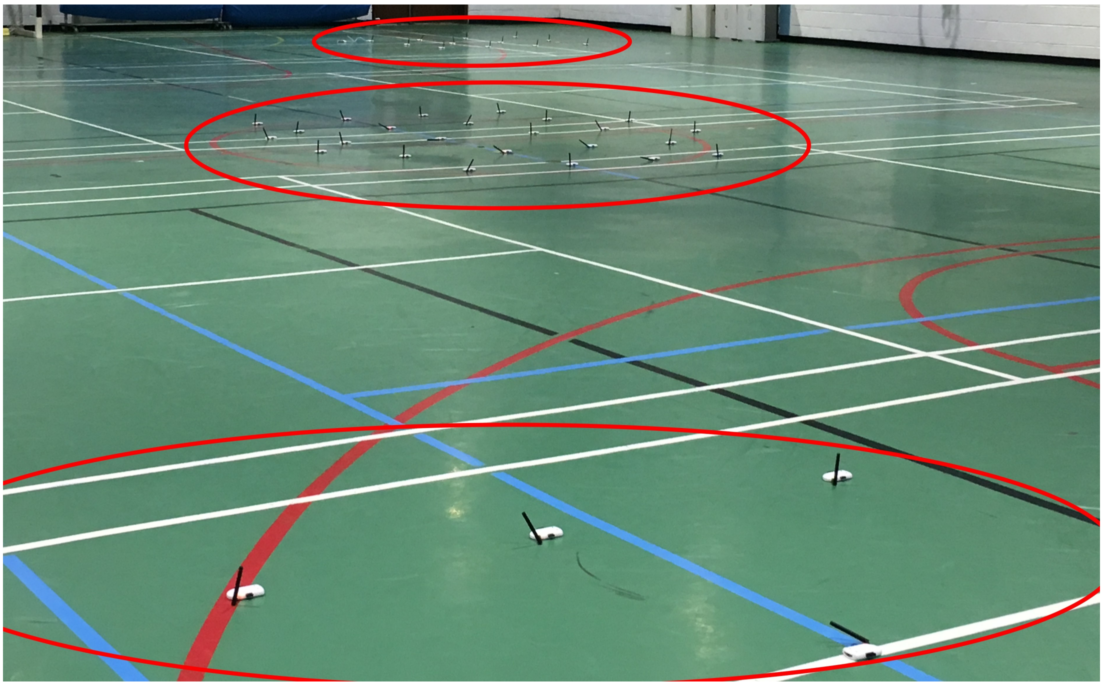


Figure 6.8: Grouped node deployment, Sports Hall

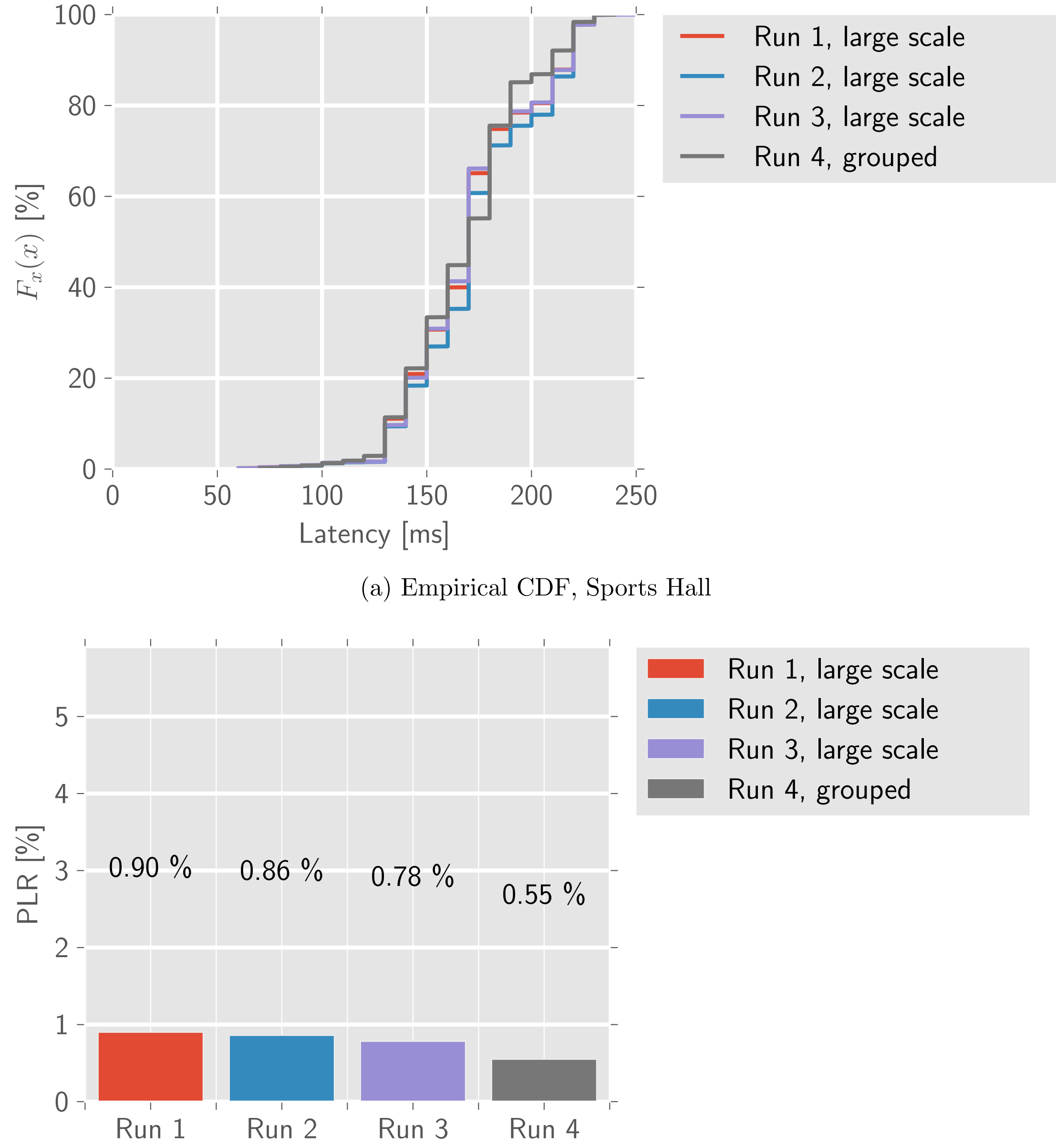
increases the number of hops needed to cover the network to three, compared to the previous setup. The results show that the additional hop has no effect on the latency and PLR. The latency result matches the ones from the large-scale experiments with 99% of messages received within 230 ms. Also, the PLR matches the results achieved earlier with a PLR below 1%. This shows that in a low interference environment, D^3LC can fulfil the requirements of a wireless lighting control network.

6.3.2 Laboratory

The laboratory was the main location for conducting the experiments. The laboratory can be compared to an open plan office, but with additional machinery such as 3D printers and the experimental setups of other research groups. Furthermore, in the laboratory, it was possible to control the interference to some extent, as it was possible to turn the WiFi AP off in the laboratory. Only the APs in the surrounding rooms and buildings affected the experiments. Several experiments were conducted in the laboratory. The first are several experiments using the large-scale setup described in Section 6.3.2.1, followed by experiments in the grouped setup described in Section 6.3.2.2.

6.3.2.1 Large Scale

The large-scale setup is used to investigate how different interference levels affect D^3LC. The first setup has all WiFi APs in the laboratory turned off (see Section 3.2.2.2). The only APs active are those which are not controllable, such as the campus WiFi. However, these APs are not located in the Laboratory, but outside the room, and the only influence was of a low level of interference. The second large-scale experiments are conducted with a high level of interference. Here, all APs in the laboratory are turned on, which is a total of 9 APs. This creates signal levels ranging from -37 dBm to -70 dBm. The nodes are deployed as shown in Figure 6.10.



(a) Empirical CDF, Sports Hall

(b) Average PLR, Sports Hall

Figure 6.9: D^{3}LC Performance, Sports Hall

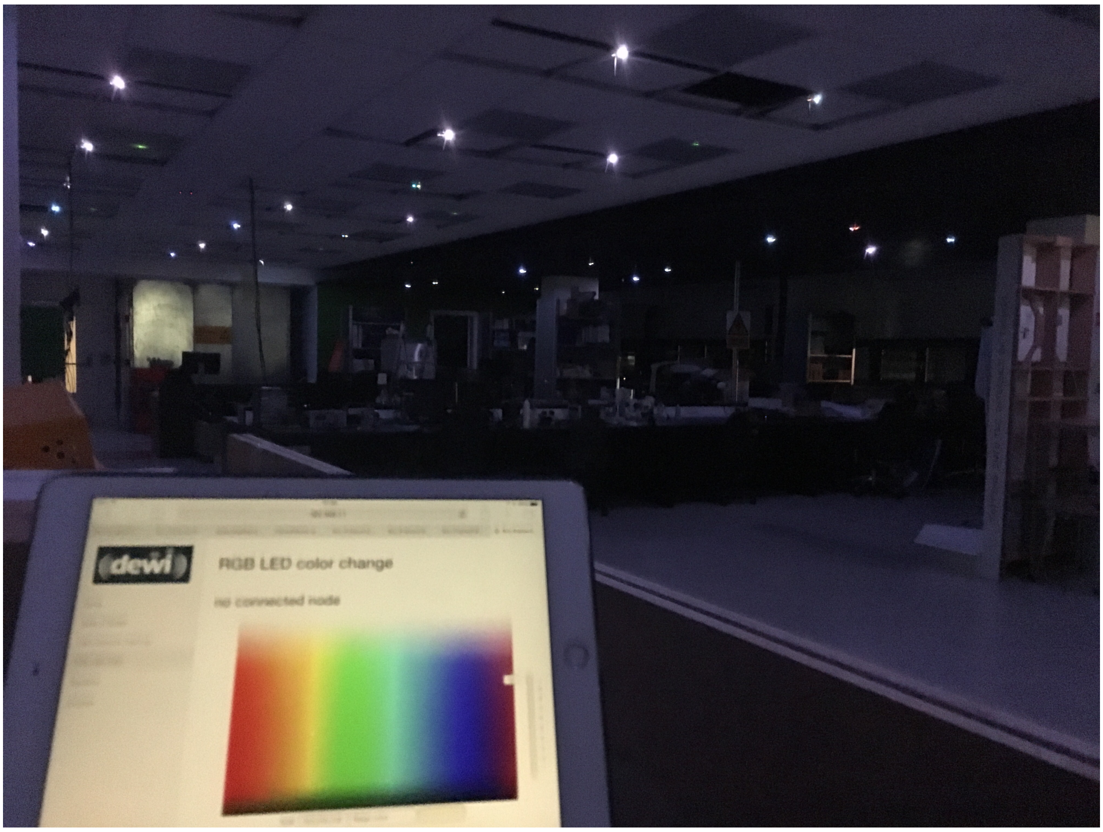


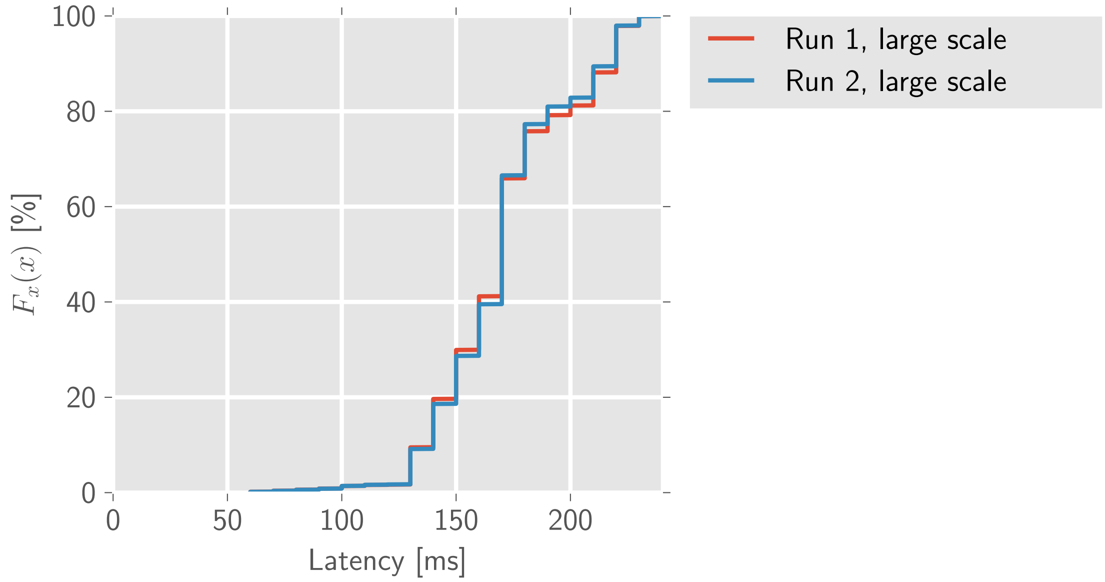
Figure 6.10: Large-scale node deployment, Laboratory

6.3.2.1.1 Low interference

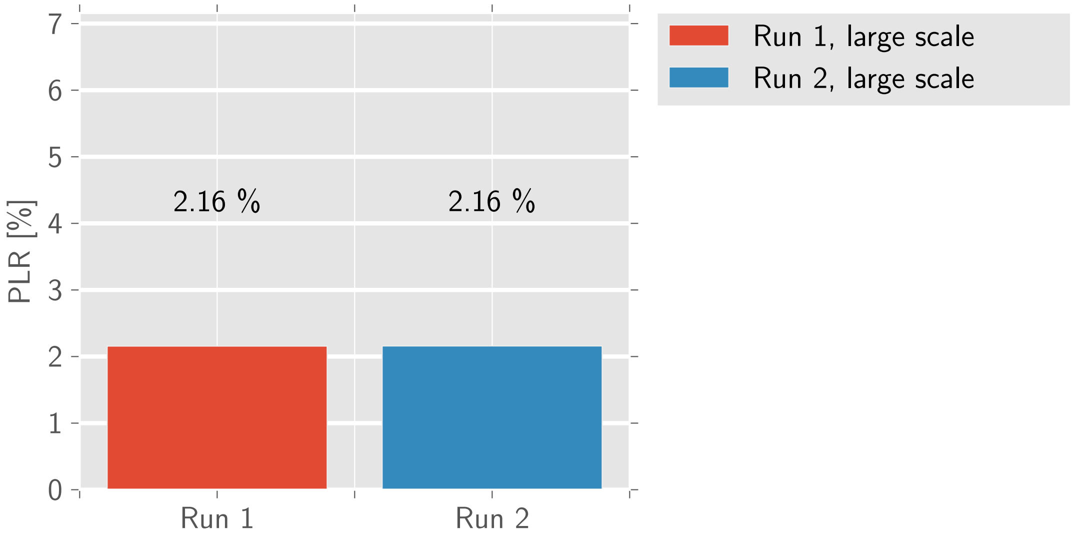
Here, D^3LC is analysed in a low-interference environment. The results are presented in Figure 6.11. Two experiments were conducted to obtain a baseline for further experiments in these environments. As before in the Sports Hall environment, in this setup, only one CH was elected during the network formation. After the results of each experiment were collected, the network was reset, and the network formation process was started again. As expected, the latency follows the same trend as the previously achieved results in the Sports Hall and the simulation (see Figure 6.11a). More than 95% of the messages are received within 230 ms. However, a difference can be noted in terms of PLR (see Figure 6.11a). The PLR is slightly higher compared to the one in the interference-free environment and is just a bit higher than 2% for both experiments. This relates to 200 messages lost of the 10000 messages created in each experiment. The slightly higher PLR is due to the interference that was present in the laboratory. Even a marginal interference level already shows an effect on the reliability. However, this effect will be investigated further in the following experiments, and at a later stage, also ideas will be presented to further improve the robustness of the overall protocol.

6.3.2.1.2 High Interference

Here, D^3LC is analysed in a high-interference environment. In the high-interference environment, all available APs are turned on, which creates signal levels ranging from -37 dBm to -70 dBm. The results are presented in Figure 6.12. Three experiments were conducted in total. As before, using this setup, only one CH was elected during the network formation. After the results of each experiment were collected, the network was reset, and the network formation process was started again. The high interference shows no effect on the latency as the result is similar to the ones achieved before (see Figure 6.12a). More than 95% of the messages are received within 230 ms. Different than before, the high level of interference causes a much larger packet loss. In Experiment 1, the PLR is just about 10%, which means that a total of 1000 messages are lost during transmission. Experiments 2 and 3 have a slightly lower PLR at about 6% and 7%. Based on the



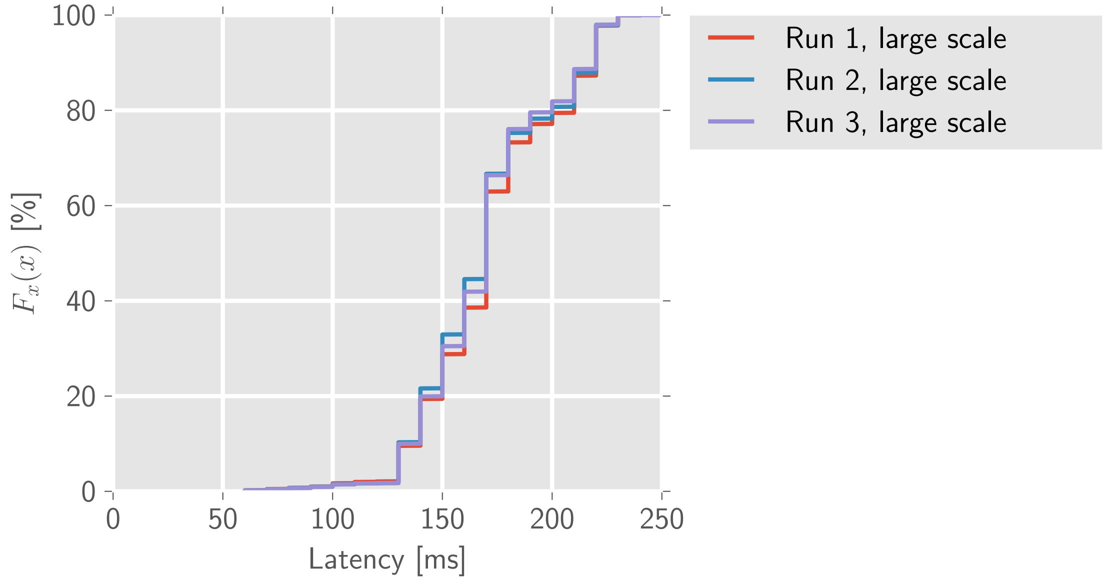
(a) Empirical CDF, Laboratory, large scale, low interference



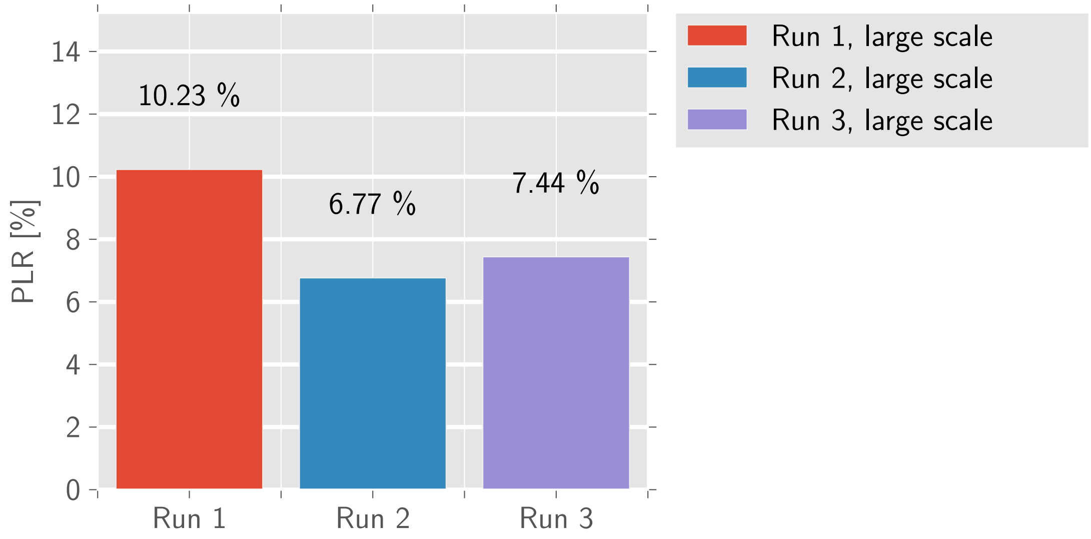
(b) Average PLR, Laboratory, Large Scale, Low Interference

Figure 6.11: D^{3}LC performance, Laboratory, large scale, low interference

requirements with a PLR below 1%, this result is not acceptable. However, as stated in the previous section, the robustness of D^3LC needs to be improved by adding additional features.



(a) Empirical CDF, Laboratory, large scale, high interference



(b) Average PLR, Laboratory, large scale, high interference

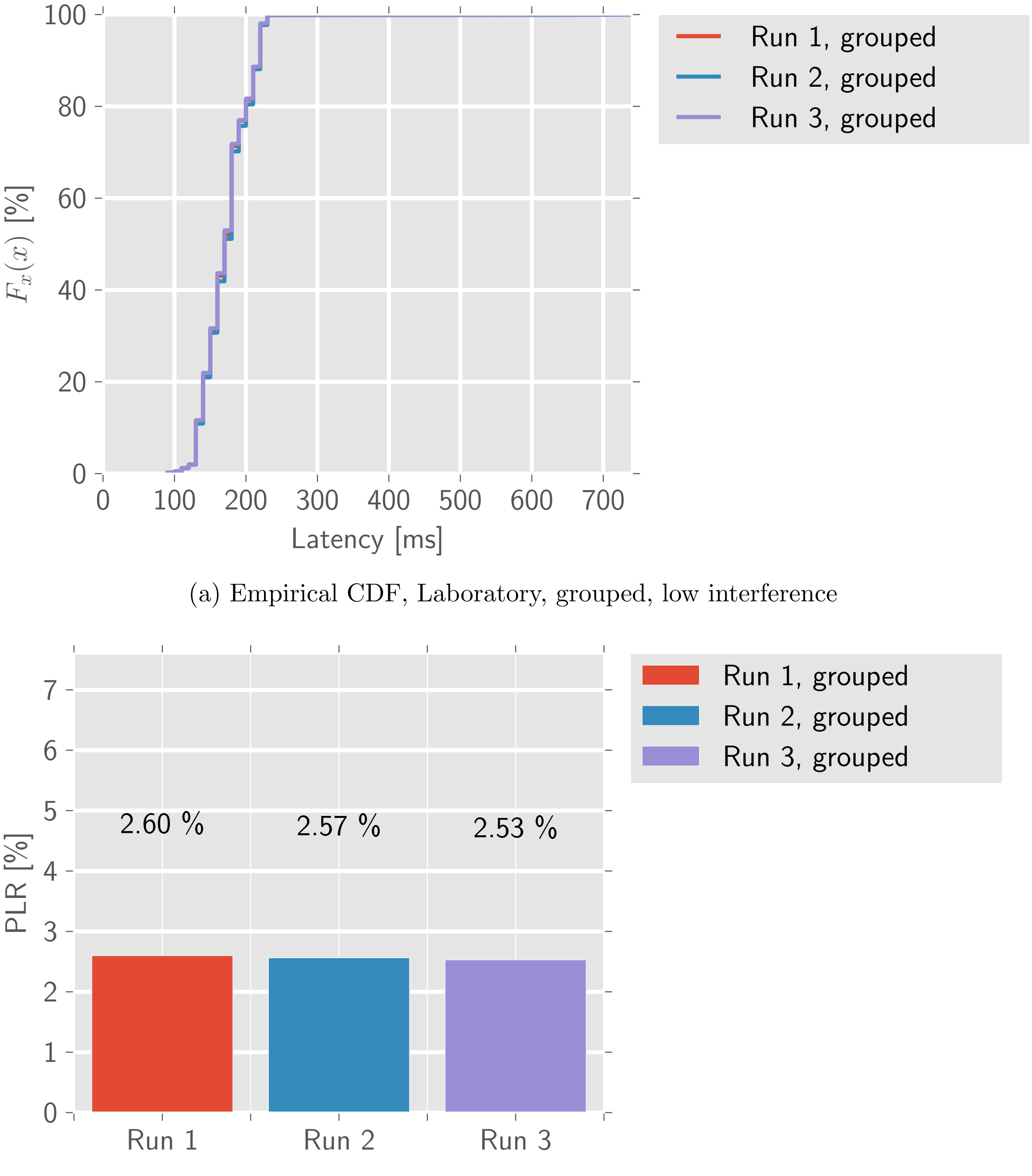
Figure 6.12: D^{3}LC performance, laboratory, large-scale, high-interference

6.3.2.2 Grouped

The large-scale experiments were used to investigate how interference affects D^3LC in a simple network setup with only one CH. In this section, it is investigated how D^3LC performs in a grouped setup influenced by different interference levels. Additionally, it is investigated how D^3LC performs when the light switch is connected to a higher-tier CH. The grouped configuration is equivalent to the one described in Section 3.2.2.2, where three CHs were elected, one on Tier 0 and two on Tier 1, with the Tier-0 CH as the parent. In total three different configurations were investigated. The first is a configuration with the same interference level as the low interference setup of the large-scale experiment with the light switch connected to the Tier-0 CH. The second configuration is the same as the first one but with an increased interference level. The interference level here is just slightly higher as the low interference setup by turning one AP in the laboratory on. The last configuration is the same as the second one, but this time, the light switch is connected to a Tier-1 CH. Using this setup, it is investigated how D^3LC performs with an increased number of hops.

6.3.2.2.1 Low interference, light switch connected to Tier 0

A total of three experiments were conducted with this configuration. All APs in the laboratory of the direct neighbourhood were turned off, and only the APs of the campus WiFi remained on. The network formation process formed a network with three CHs, one as Tier-0 CH and two as Tier-1 CHs with the Tier-0 CH as the parent. The light switch is connected to the Tier-0 CH. This increases the number of hops needed to cover the network to three, compared to the large-scale setup. The results show that the additional hop has no effect on the latency and only a minimal effect on the PLR (see Figure 6.13). The latency result matches the ones of the large-scale experiments, with 99% of messages received within 230 ms (see Figure 6.13a). Less than 1% of the messages received had an increased latency of above 230 ms. This is due to a problem with the queuing system of Contiki and will be explained in more detail in the next section. The increased number of hops did not affect the PLR (see Figure 6.13b). The PLR is similar to the one of the large-scale setup with the same interference level. This shows that a change in the topology with more hops to cover does not influence the PLR nor the latency.



(a) Empirical CDF, Laboratory, grouped, low interference

(b) Average PLR, Laboratory, grouped, low interference

Figure 6.13: D^{3}LC performance, Laboratory, grouped, low-interference

6.3.2.2.2 Medium interference, light switch connected to Tier 0

Three experiments were conducted in the second set of experiments in the Laboratory with the grouped configuration. To investigate how interference influences the performance of D^3LC with multiple CHs, the interference level was slightly increased. Compared to the low interference setup, one AP in the centre of the laboratory was turned on. This AP was used by another research group to connect their devices to the Internet, which included multiple Raspberry Pi's, laptops, and smartphones. Therefore, a constant traffic was expected for this AP. Figure 6.14 shows how the AP operating on WiFi channel 1 overlaps with the four TSCH channels. Even though TSCH uses channel hopping, where TSCH hops over all available channels periodically, this AP affects Channels 11, 12, 13, and 14. The latency result is reported in Figure 6.15a. It can be seen that the majority of the messages are received within 230 ms. A jump from 230 ms to 730 ms can be seen. This is due to the queueing system of Contiki. If a message is assigned in Contiki to be sent in a specific timeslot, it only can be sent in this one. However, if a message is not sent, the message is stuck in the queue for a full slotframe duration of 510 ms. This means that all messages added later will get stuck in the queue as well. As the queue in Contiki is a FiFo queue, the send process will always take the first element only. With RLL, the process should be such that only the newest command is sent, which means that the old messages

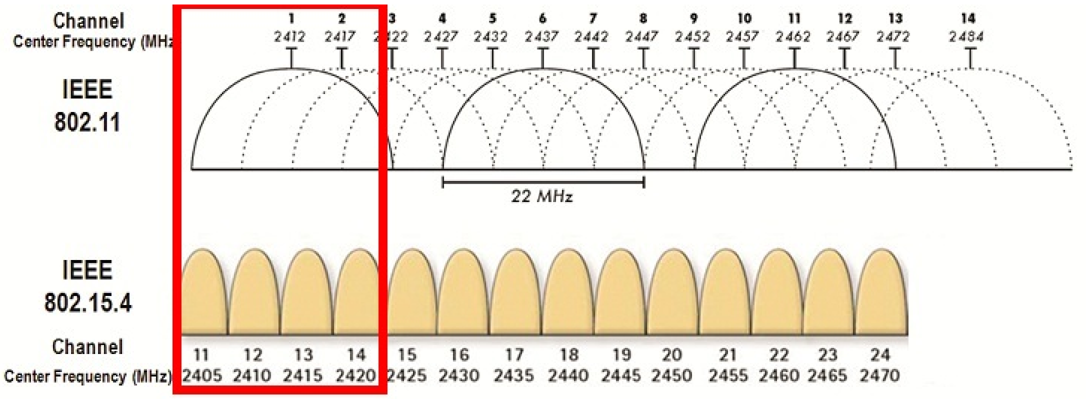
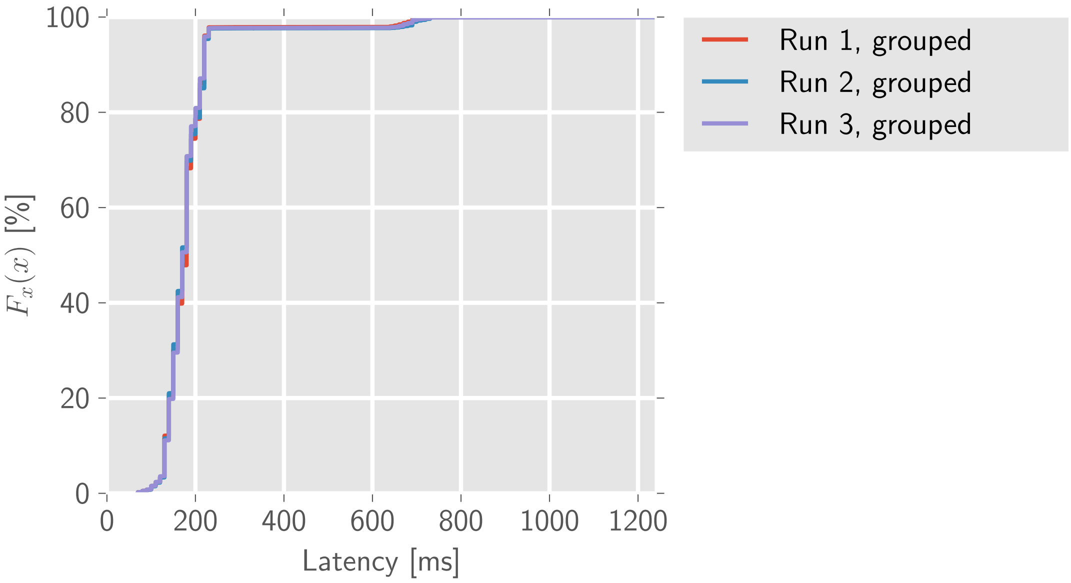


Figure 6.14: WiFi channel overlap with IEEE802.15.4 channels

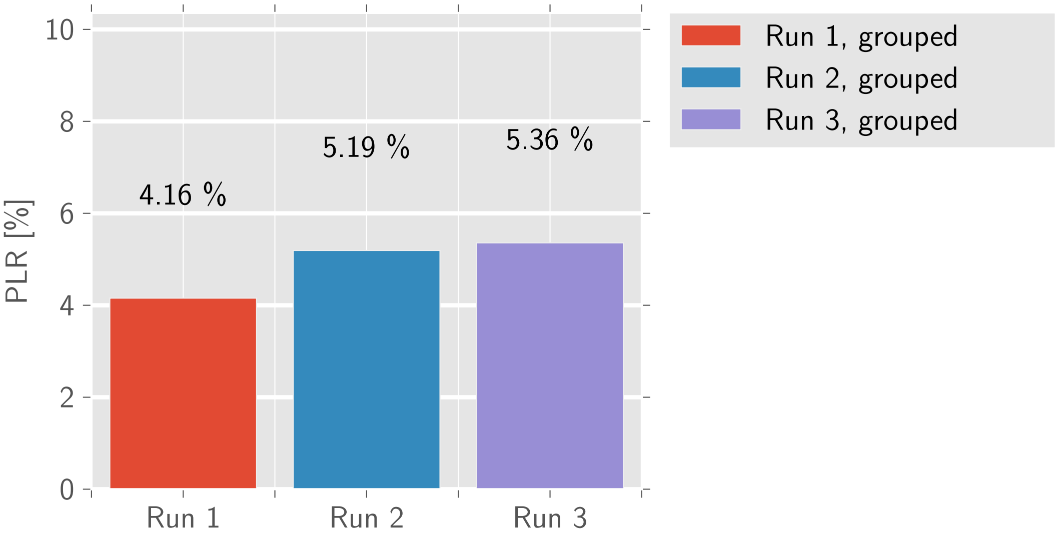
should be deleted when a new one is added. However, this was not possible with Contiki due to the software queue (the Contiki FiFo queue) and the hardware *Tx Buffer*. If a message is deleted out of the software queue, it can still be in the hardware buffer, and therefore, it is currently not possible to delete it. Fixing this particular problem in the Contiki-OS was not part of the scope of this work, but this is added to the future work list to further improve the protocol implementation. In such cases, the above-described behaviour applies, and the slotframe duration is added to the latency. In any case, this happened in only less than 3% of all messages received. The PLR for this setup is slightly higher compared to the previous experiment (see Figure 6.15b). For all experiments conducted, the PLR is between 4% and 5.4%. This shows that adding one WiFi AP to the immediate neighbourhood of a lighting control network running the D^3LC has doubled the PLR. However, as stated before, it is necessary that in the next development iteration of D^3LC, improving the robustness must be one of the major concerns.

6.3.2.2.3 Medium interference, light switch connected to Tier 1

The final set of experiments in the laboratory is a setup where the light switch is connected to a Tier-1 CH. This increases the number of hops covering the whole network to four. The interference level is the same compared to the previous set of experiments where one AP is active in the centre of the laboratory. The results are presented in Figure 6.16. It can be seen that the additional hop has a direct influence on the latency (see Figure 6.16a). Only 20% of the messages are received within 230 ms. These are basically all messages received by the nodes assigned to the same CH as the light switch. The remaining messages are received between 230 ms and 330 ms, which is an additional ten timeslots. This is due to the problem described earlier in Section 6.3.1 where Contiki appears not to be fast enough to process the incoming packets to forward to the next timeslot. This problem has much larger influence when the light switch is connected to a Tier 1 CH. Here, the message first needs to be forwarded to the Tier-0 CH and then to the next Tier-1 CH. A look at the ideal forwarding process based on the timeslots will show better why an additional ten timeslots are sometimes necessary. First,



(a) Empirical CDF, Laboratory, grouped, medium interference, Tier 0



(b) Average PLR, Laboratory, grouped, medium interference, Tier 0

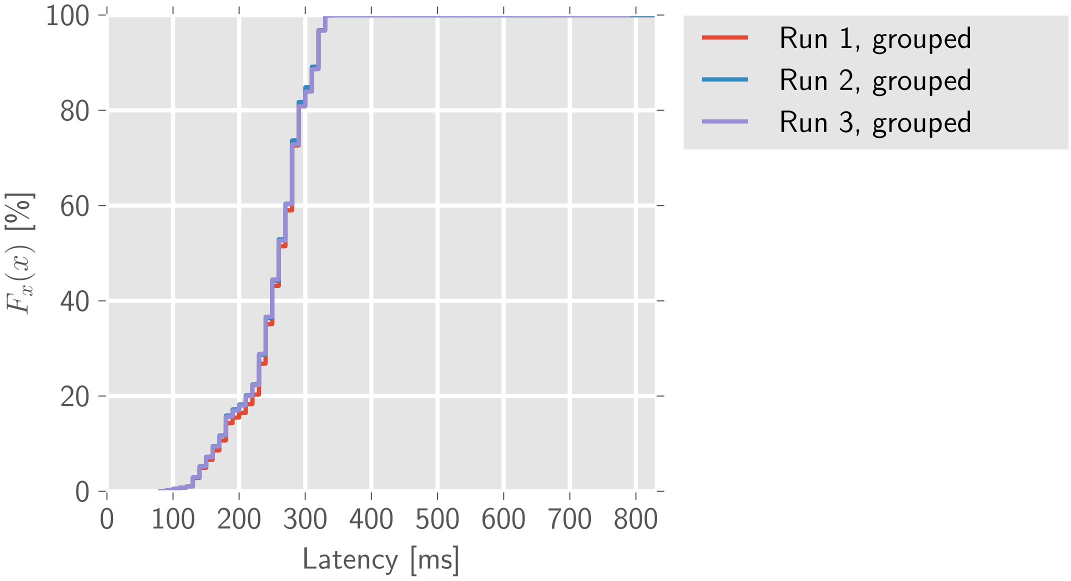
Figure 6.15: D^{3}LC Performance, Laboratory, grouped, medium interference, Tier 0

the CS transmits the message in slot R+1 to the Tier-1 CH (see Figure 5.7). Ideally, the Tier-1 CH would forward in R+3 to its parent (Tier-0 CH). The Tier-0 CH will then forward in R+6 to all other Tier-1 CHs, and finally, in R+10, to all CSs simultaneously. However, with Contiki, it happened that a message takes up to two timeslots to be processed. This means that when the first Tier-1 CH receives the message from its CS in R+1, the message might be only processed in R+3. Hence, the message cannot be forwarded in R+3, but only in the next appropriate one, which is in R+7, and in R+10 to the CSs of this CH. Reordering the slotframe design would not improve that, except for additional timeslots being added per round. However, this would increase the latency as well. The Tier-0 CH will then forward in R+10 plus the ten timeslots to its child Tier-1 CHs already elapsed, and finally, the remaining CHs forward these in R+10 to their CSs. This is a total of 200 ms of latency plus the time the message was stored in the queue, initially a total latency of 320 ms is reached. This shows that the processing time of the operating system is the key for the correct operation of D^3LC. Again, optimising the operation of Contiki was not focus of this thesis, but this should be considered in choosing a real-time operating system for the next iteration of D^3LC.

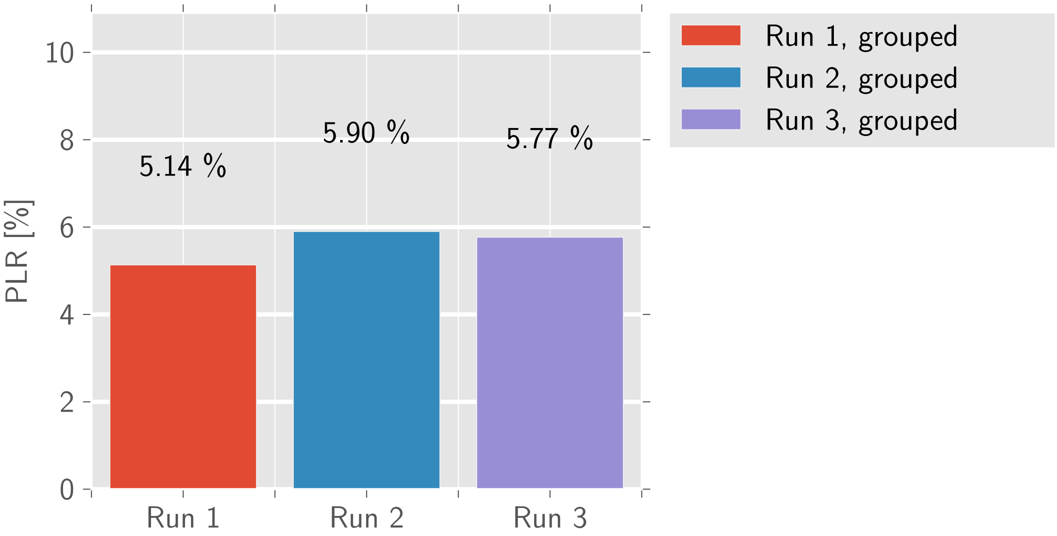
The PLR remains unchanged compared to the previous setup and is around 5% (see Figure 6.16b).

6.3.3 Apartment

The last environment used for the experiments is an occupied apartment. In total, five experiments were carried out, each at a different time of the day, Experiments 1 and 2 in the afternoon, Experiments 3 and 4 in the evening, and Experiment 5 in the late evening. This represents two different occupational patterns. The node distribution is based on the grid setup. Due to the small size of the apartment ($30m^2$), only one CH was needed to cover the whole apartment. The nodes are distributed over most of the apartment as shown in Figure 3.10. In Figure 6.17, part of the actual deployment is shown. Only two rooms were not used in this scenario, one was occupied by another person and the other one is the bathroom.



(a) Empirical CDF, Laboratory, grouped, medium interference, Tier 1



(b) Average PLR, Laboratory, grouped, medium interference, Tier 1

Figure 6.16: D^{3}LC performance, laboratory, grouped, medium interference, Tier 1

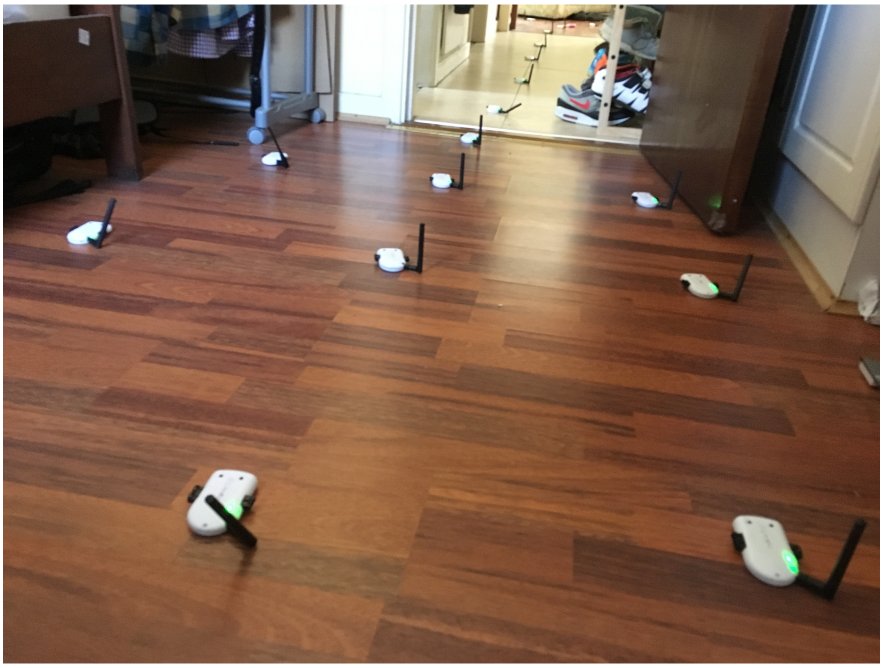


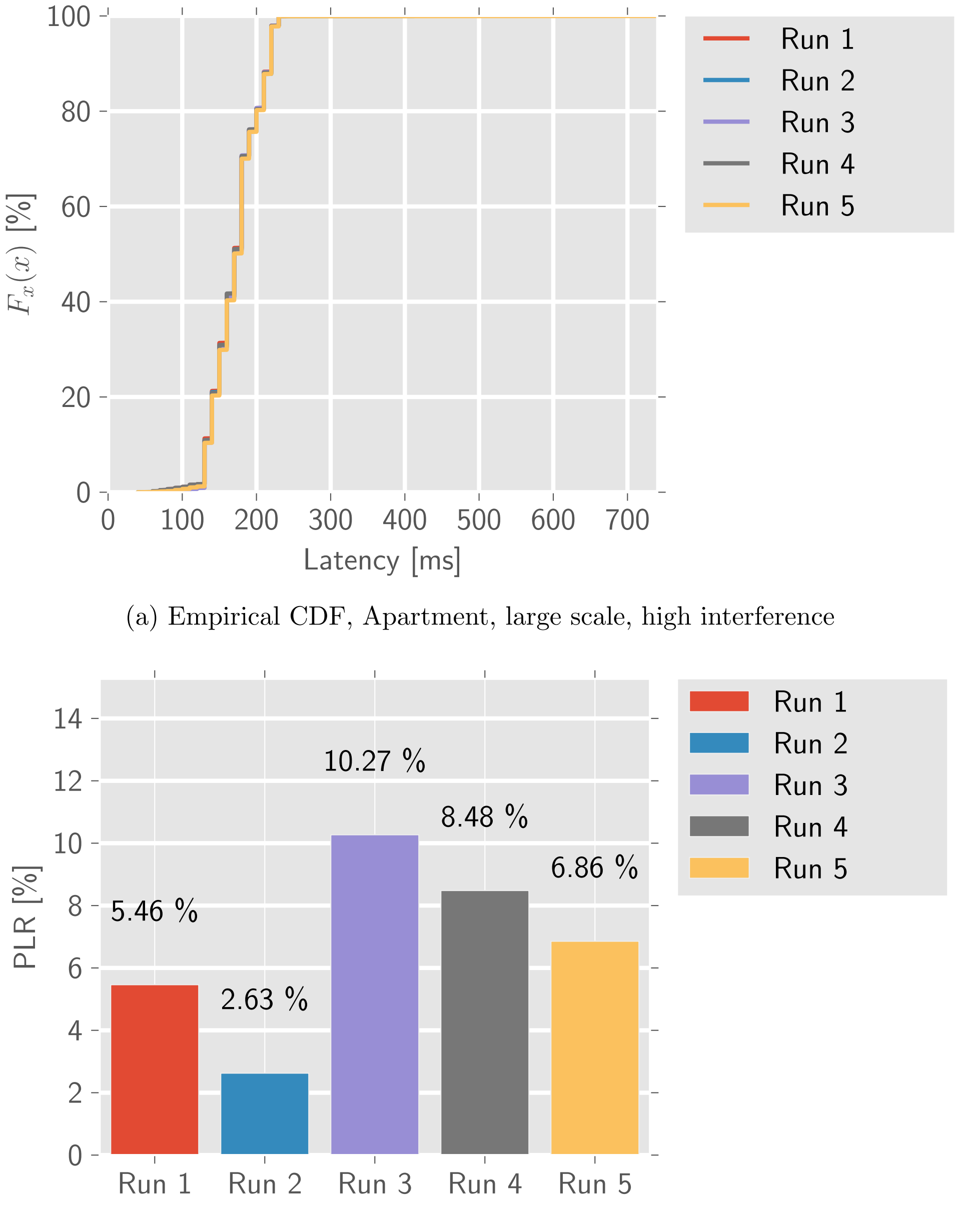
Figure 6.17: Part of the node deployment in the apartment

The apartment scenario represents a real-world scenario where not necessarily all lamps are controlled at the same time, but rooms, or even a few lamps, are organised logically in groups using a higher layer application. A special device operates as a light switch and enables a connection to smartphones, tablets, or other input devices, similar to the Hue Bridge[1]. However, even with this bridge device, D^3LC could be used as the underlying dissemination suite where all lamps would receive the message but only the ones addressed would change their state.

The results are presented in Figure 6.18. Similar to the results earlier, the latency for this setup is between 200 ms and 250 ms for all experiments. (6.18a). The PLR is different for each experiment, with the lowest PLR starting at 2.63% and the highest at 10.27% (Figure 6.18b). This can be explained by the high interference level in the apartment. It can be seen that the experiments carried out in the afternoon have the lowest PLR, the ones carried out in the evening have the highest PLR, and the one carried out during the late evening is in between. An explanation is that in the afternoon, the surrounding apartments are less

[1] Hue Bridge: `http://www2.meethue.com/en-us/p/hue-bridge/`

occupied, which would result in less WiFi traffic, while during the evening, most apartments are occupied, and therefore, the WiFi traffic is higher. However, this result again indicates that a robustness against WiFi interference is very important, and solutions need to be found [38].



(a) Empirical CDF, Apartment, large scale, high interference

(b) Average PLR, Apartment, large scale, high interference

Figure 6.18: D^{3}LC performance, Apartment, large-scale, high-interference

6.4 Low-power wireless bus

This section compares D^3LC with the LWB. The LWB is implemented without modification and using the default configuration as reported in [48]. The LWB's functionality was discussed in Section 5.4.1. The application used for LWB is the same as for experiments with D^3LC. Traffic is generated in the form of bursts where a burst represents a user dimming the light. Each burst lasts for about 2 seconds with a new command generated every 100ms. The quiet time of the light switch node between two bursts is selected randomly and ranges from 1 second to 5 seconds. This was found to be sufficient time to return the network back to a steady-state after a burst was finished. For each experiment, a total of 500 bursts were initiated; therefore, a total of 10000 messages were recorded. The environment used was the Laboratory, with a total of 50 TelosB nodes arranged using the grouped topology. First, LWB is evaluated running the D^{3}LC application without interference, i.e. by using the IEEE802.15.4 Channel 26 (see Section 6.4.1). This is followed by setting the communication channel to Channel 11, thus overlapping a WiFi access point operating on WiFi Channel 1 (see Section 6.4.2). In Section 6.4.3, the settling time between two consecutive burst is increased to a maximum of 60 seconds. This is used to investigate how the LWB performs with a much longer idle phase with no traffic. Finally, in Section 6.4.4, D^3LC is directly compared to the LWB and discussed.

6.4.1 D^{3}LC application, Channel 26

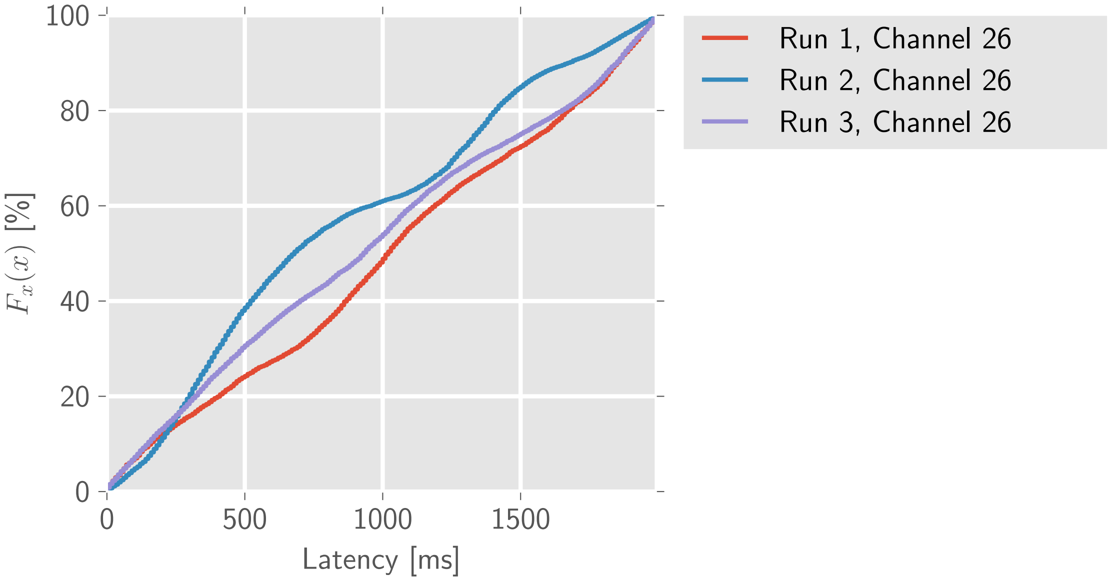
In this section, the LWB is investigated with the same application as D^3LC. In this scenario, the LWB operates on Channel 26 and, therefore, free of WiFi interference. The results are shown in Figure 6.19. Three experiments were conducted in total. With the LWB, it can be seen that less than 10% of all nodes receive the control command within 200 ms, see Figure 6.19a. It is only after 2 seconds that all nodes had received the command. This high latency is due to the LWB's round-based communication and its capability to increase the sleep time between rounds to increase the duty cycling. Based on the assumption that LWB reaches a nearly 100% packet delivery, the expectations were similar. However, for the lighting application, the packet loss with no interference is between 1.4% and 2.6%. After investigating, it was found that this is due to the LWB's node failure detection which removes a node's access to the channel when the host has not received anything from that node for a certain period of time. Hence, with the lighting application, the quiet time is just long enough that a node appears disconnected. After "losing" the connection, a node needs to request send slots again. During the time of no access to the wireless bus, a node can only store a certain number of packets in the queue. When full, the new packets will be dropped. However, this effect is investigated in more detail in a later section.

6.4.2 D^{3}LC application, Channel 11

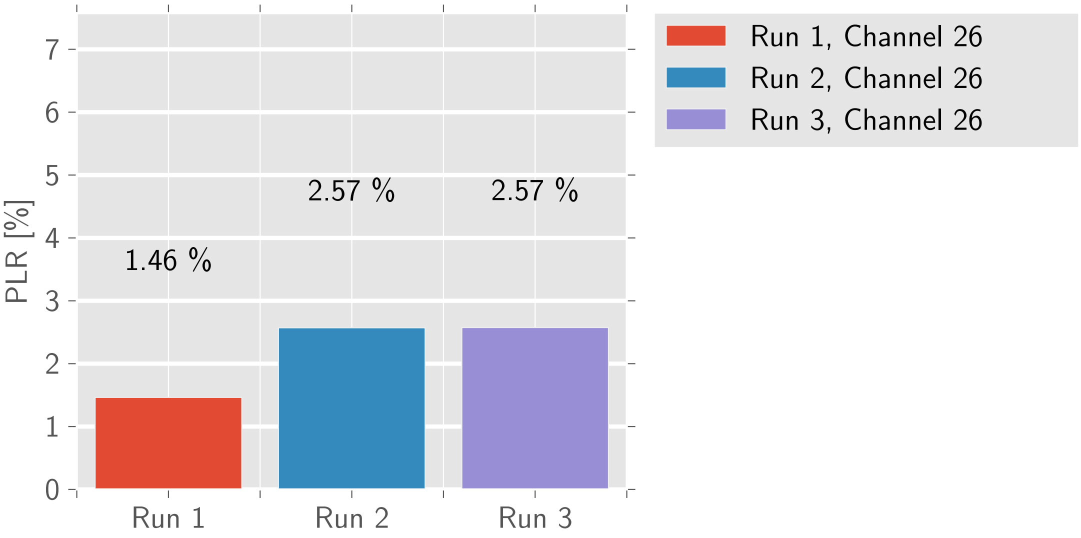
To investigate the robustness of the LWB with WiFi interference, in this set of experiments, the LWB channel is set to 11. This channel overlaps with WiFi Channel 1. As for the medium interference level experiments in Section 6.3.2.2, the same AP is active and operates on Channel 1. In total, three experiments were conducted, and the results are shown in Figure 6.20. The results for both latency and PLR are similar to the results of the LWB operating on Channel 26. The PLR is marginally higher, ranging from 2.56% to 3.77%. This shows that the LWB is able to deliver a message even when interference is present.

6.4.3 D^{3}LC application, Channel 26, long-period

As mentioned in Section 6.4.1, the LWB removes nodes from the communication schedule if there is no traffic demand received over a certain period. This is because the LWB is designed for periodic traffic where sensor data is transmitted to a central device in periodic intervals. The experiments in Section 6.4.1 showed that even with only little randomness between two bursts, a node can be removed from the schedule, in this case, the light switch. This section discusses this effect in more detail. To show the drastic increase of PLR by this effect, the quiet period range was increased to up to 60 seconds. To get an idea of the behaviour, one experiment was completed. The results of the experiment are shown in Figure 6.21. The latency trend is as expected, even if the latency increase is much more linear than before (see Figure 6.21a). This is due to the very high

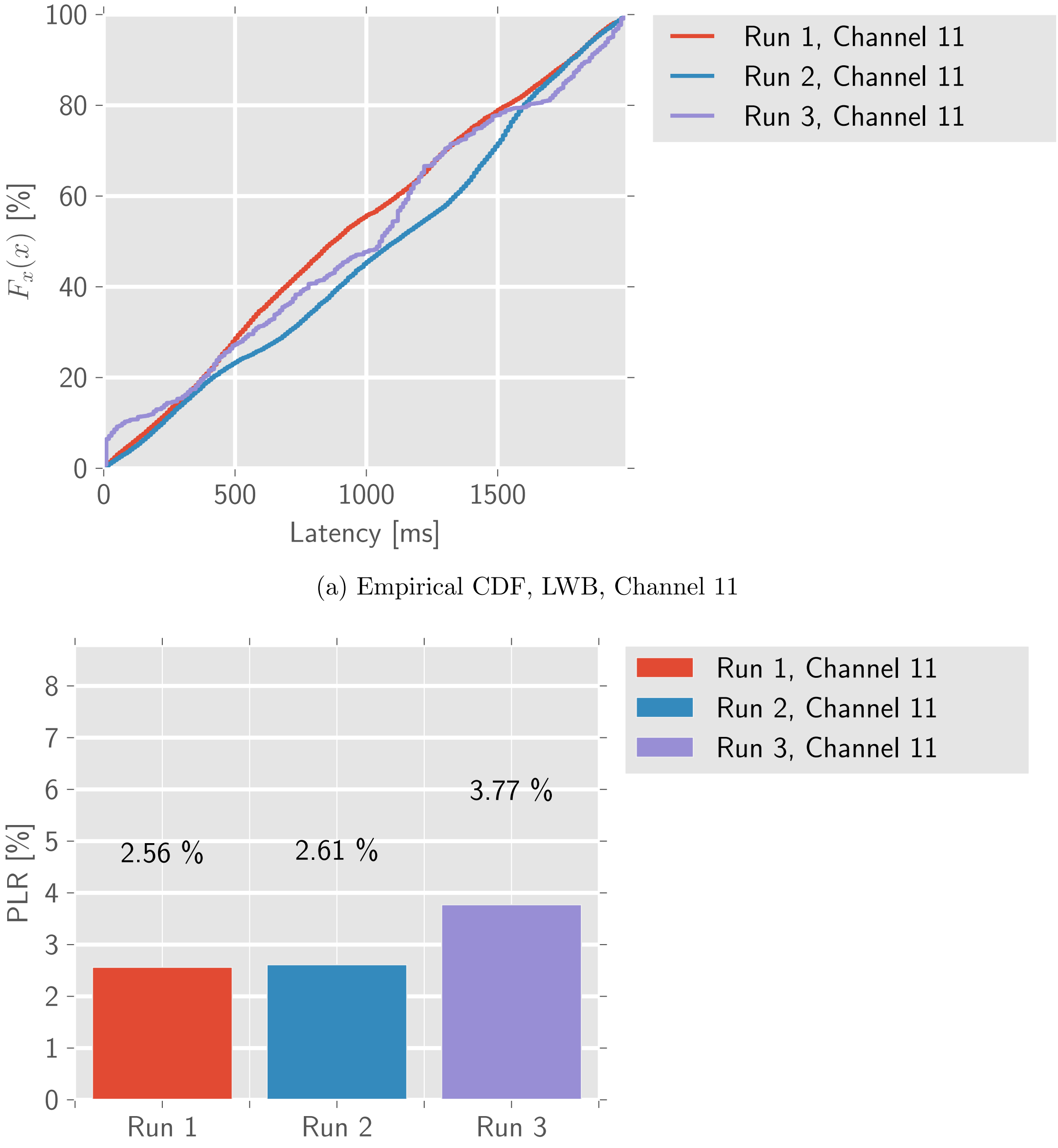


(a) Empirical CDF, LWB, Channel 26



(b) Average PLR, LWB, Channel 26

Figure 6.19: LWB performance on Channel 26



(a) Empirical CDF, LWB, Channel 11

(b) Average PLR, LWB, Channel 11

Figure 6.20: LWB performance on Channel 11

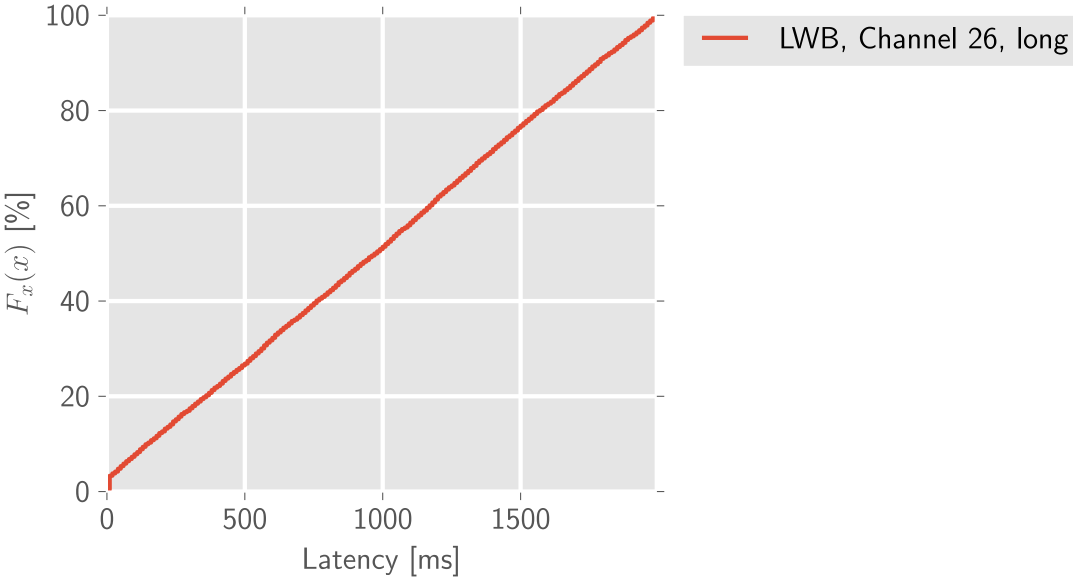
PLR of almost 50% (see Figure 6.21a). This clearly shows that the LWB is not designed for random traffic patterns. However, besides the high latency, the robustness could be improved if a different scheduling method were to be used. For example, specific devices could request fixed slots in each communication round, with which they are able to always send data if available. However, this requires modifying the LWB, which is not part of this study.

6.4.4 D^{3}LC vs. LWB

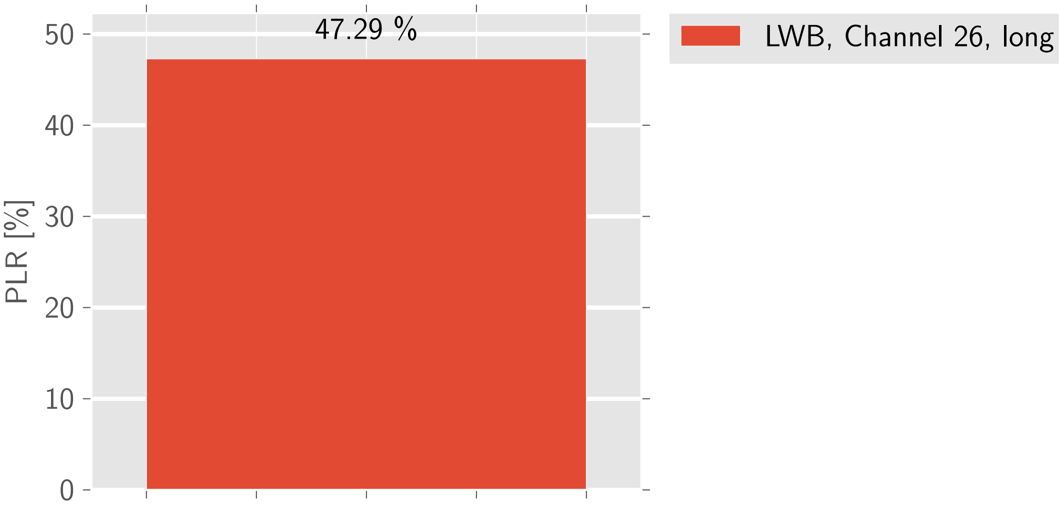
This section compares the results of the LWB under WiFi interference on Channel 1, with D^3LC at medium WiFi interference (see Figure 6.22). No additional experiments were conducted for these results; furthermore, already available results are selected for this comparison.

The results represent the mean results of the experiments conducted in Section 6.3.2.2.2 for D^3LC and the mean results of the experiments conducted in Section 6.4.2 for LWB.

In terms of latency (Figure 6.22a), D^3LC outperforms the LWB. With the LWB, 95% of the messages are received after almost 2 seconds, while with D^3LC, 95% are already received after 230 ms. Considering the queueing problem described and processing speed of Contiki, which result in an additional superframe duration to deliver a control message, D^3LC outperforms the LWB by more than one second. The PLR is marginally higher compared to the LWB (Figure 6.22b). Considering the much higher latency of the LWB, the slightly higher PLR can be considered acceptable. The comparison of the LWB with D^3LC shows that the current specification of the LWB is not well suited to a lighting application. Further improvements are needed to adapt the LWB for operation in lighting control. Most important, this comparison shows that D^3LC performs better than the current state-of-the-art.

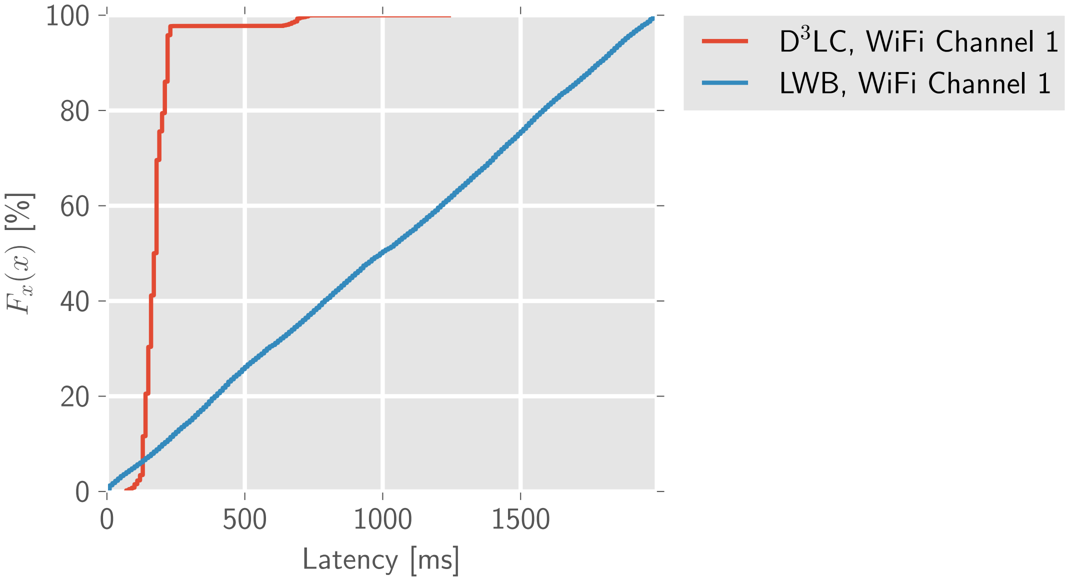


(a) Empirical CDF, LWB, Channel 26, long period

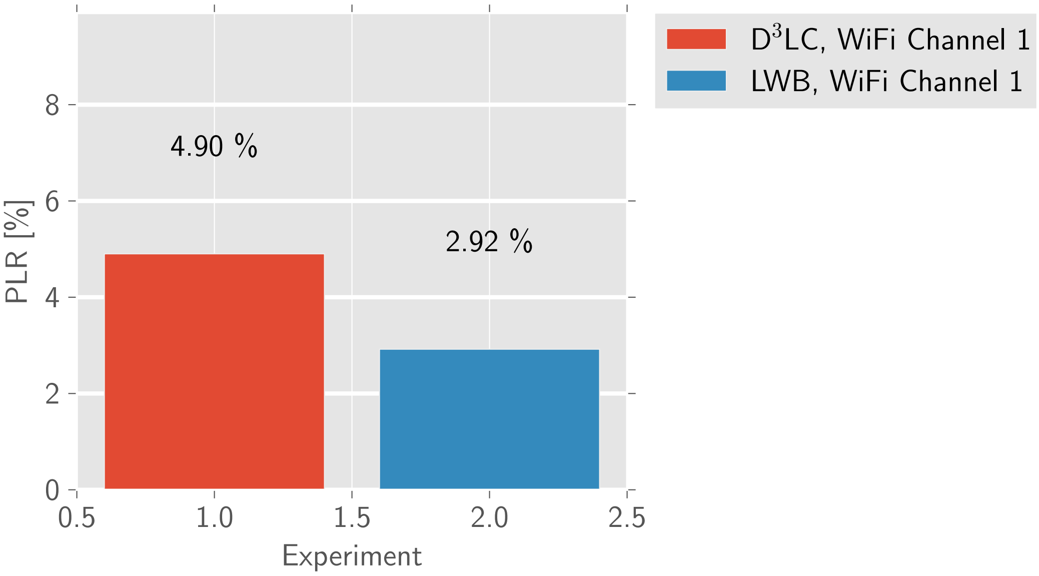


(b) Average PLR, LWB, Channel 26, long period

Figure 6.21: LWB performance on Channel 26, long period



(a) Empirical CDF, D^{3}LC vs. LWB



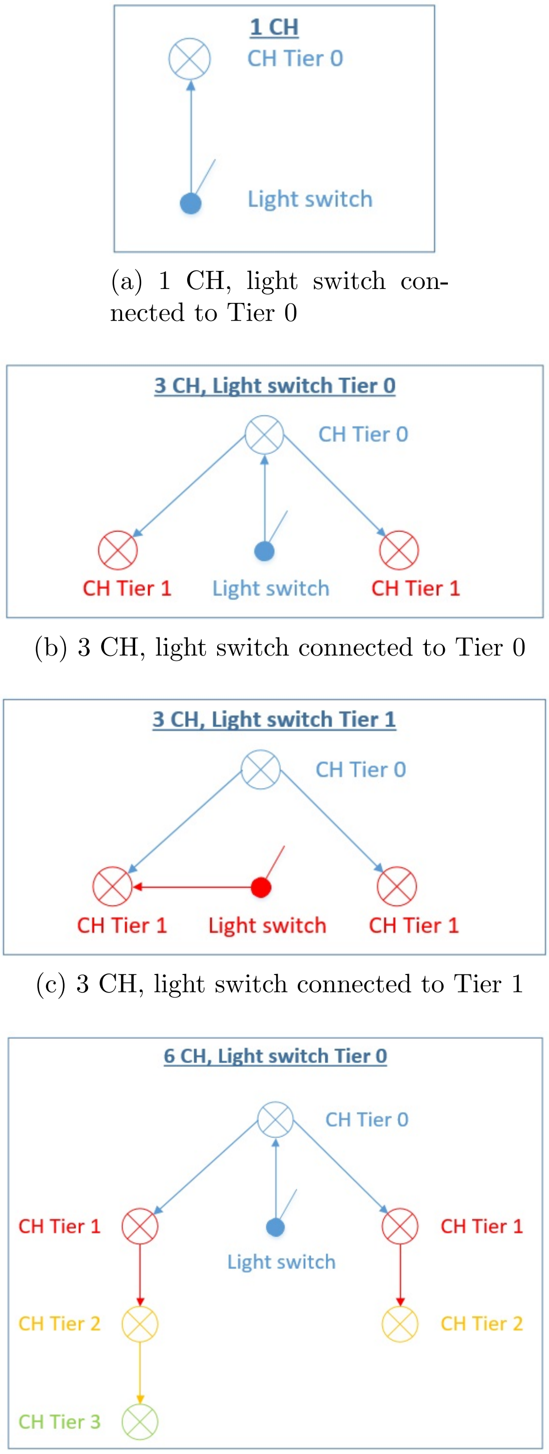
(b) Average PLR, D^{3}LC vs. LWB

Figure 6.22: D^{3}LC vs. LWB

6.5 Network load

In this section D^3LC is analysed from a different perspective. While it is known that Flooding suffers from high redundancy and one goal of D^3LC is to reduce this redundancy, here, it is analysed by how much D^3LC reduces redundancy. To analyse the reduction in redundancy it was measured how many bytes are injected into a network and how often these bytes are forwarded. Assuming a user pushes a light switch to turn the light on, this will create a message with e.g. 20 bytes. This message is forwarded a number of times by both, Flooding and D^3LC. In this section it is shown how many bytes are eventually forwarded until the end of the forwarding process. To analyse this behaviour five configurations are considered. The Flooding protocol is used for a baseline measurement where each node forwards the message exactly once. Then three configurations of the experiments conducted in the previous sections are used, i.e. the laboratory configuration with only one CH (see Figure 6.23a), the laboratory configuration with 3 CHs and the light switch connected to Tier 0 (see Figure 6.23b), and the laboratory configuration with 3 CHs and the light switch connected to Tier 1 (see Figure 6.23c). The last configuration is based on the simulation of RLL in Section 6.2 with 6 CHs and the light switch connected to Tier 0 (see Figure 6.23d).

The results of this analysis are presented in Figure 6.24. Each of the configurations described above consists of a total of 50 nodes of which one node is a light switch node which injects a control message of 20 bytes into the network. This means for Flooding a total of 49 nodes forwards this control message which results in a total of 980 forwarded bytes, since each of the 49 nodes forwards the message exactly once. In the configuration with only 1 CH where D^3LC is used as a dissemination protocol, only this one CH forwards the message. The CH receives the message from the light switch and then forwards the message to all other nodes in the network. This results in a total of 20 forwarded bytes. In the third configuration 3 CHs are used with the light switch connected to the CH located at Tier 0. The light switch forwards the control message to the CH at Tier 0, and this CH forwards the message to the two CHs located at Tier 1. All three CHs then forward the message to all other nodes in the network. The total



(a) 1 CH, light switch connected to Tier 0

(b) 3 CH, light switch connected to Tier 0

(c) 3 CH, light switch connected to Tier 1

(d) 6 CH, light switch connected to Tier 0

Figure 6.23: Topology overview

number of forwarded bytes is 80. The fourth configuration is the same as the previous one, except that the light switch is connected to a CH located at Tier 1. This change in configuration results in one additional transmission, since the light switch first forwards the message to the CH located at Tier 1, this CH forwards the message to the CH at Tier 0. The Tier 0 CH forwards the message then to all CHs located at Tier 1 and finally all CHs forward the message to all other nodes. This results in a total 100 forwarded bytes. The last configuration is based on the simulation setup of RLL with a total of 6 CHs distributed in three Tiers with the light switch connected to the CH located at Tier 0. The light switch forwards the message to the CH at Tier 0, while this one forwards the message to the CHs located at Tier 1. The Tier 1 CHs forward the message to the Tier 2 CHs and one of the Tier 2 CHs forwards the message to the remaining CH located at Tier 3. All six CHs forward the message then to all other nodes in the network. This results in a total of 10 transmissions needed to cover the network and a total of 200 forwarded bytes. This analysis shows that D^3LC is much more efficient in forwarding messages compared to Flooding. Only a minimum number of retransmissions is needed to cover the whole network. If only one CH is in a network the total network load is reduced by 98%. Even there are more CHs in the network, e.g. a total of 6 CHs the load is still reduced by 80%. In general it can be said that D^3LC significantly reduces the redundancy and therefore the network load.

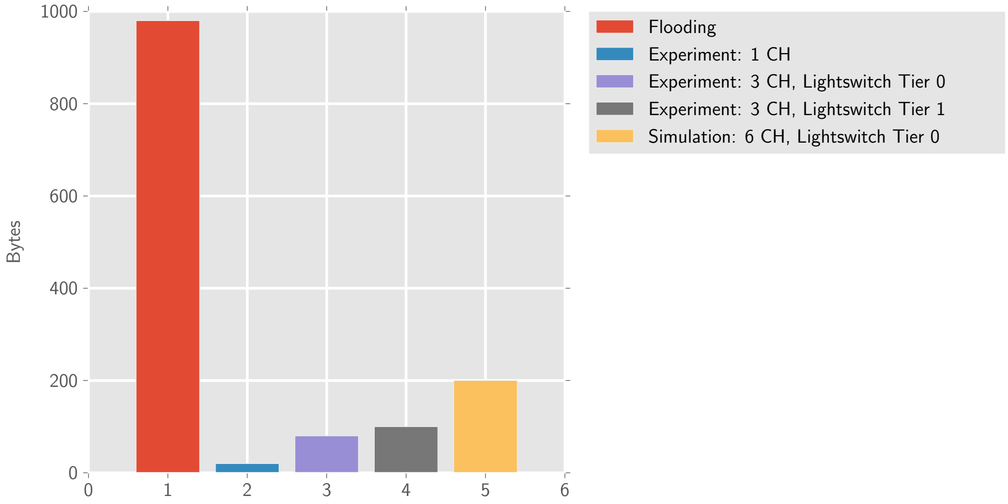


Figure 6.24: Network load comparison

6.6 Conclusion

This chapter analyses the D^3LC suite based on different aspects. First, the convergence of the network formation protocol CIDER that was developed for this work was analysed. The findings were that CIDER is able to form a network in less than 5 minutes. A simulation-based evaluation of the data dissemination protocol RLL was provided for investigating how the QoS and QoE requirements for lighting control could be fulfilled. RLL is able to provide a data dissemination mechanism with a low latency and a low PLR while at the same time, fulfilling the user expectations with ideal conditions. D^3LC was evaluated and validated in extensive real-world experiments. The results show that theD^3LC suite is capable of a real-time delivery of multi-hop message broadcasts in dense wireless lighting scenarios with message inter-arrival times of 100ms, thus enabling applications such as dimming or colour adjustment in large and dense deployments.
The observed latency at which 99% of the messages were received is between 200 and 250 ms even with multiple cluster heads, which meets the real-time performance requirements. It is expected that the latency can be reduced even further since the current implementation includes some software delays in the devices that cause some of the messages to be deferred to later time slots when

their processing was not completed ahead of the nearest transmission opportunity. Packet loss ratios are subject to interference, with very promising results in low-interference situations. Under heavy WiFi interference, packet losses increase, which highlights the need for allocating channels (both for WiFi and for IEEE 802.15.4) in a meaningful way since an excessive presence of interference will always limit the performance of wireless solutions. Nevertheless, the D^3LC suite proves to be a viable approach to address the challenges of large and dense wireless lighting control networks. To further increase the robustness of D^3LC, one possible solution for tackling the high PLR is to transmit the last command twice, which would reduce the PLR of this command to $< 1\%$. This does not solve the problem of the lack in robustness but leads towards the QoE goal for lighting networks. D^3LC was also compared to the LWB protocol, and it was found that D^3LC meets the lighting control requirements better, in particular in terms of latency.

Chapter 7

Conclusion & Outlook

7.1 Conclusion

The research work presented demonstrates the validity of the D^3LC Suite as an M2M protocol stack enabling self-organising networks and providing a time-bound message dissemination that enables wireless lighting control. The review of the current state-of-the-art literature for wireless lighting control networks aroused the need for a more sophisticated solution. Using extensive simulations, flooding and probabilistic forwarding where analysed. The first finding is that the network density has an impact on the performance of flooding. When the network becomes denser, the PLR increases as well as the latency. Furthermore, it was found that increasing the traffic rate impacts the performance of flooding as well. Another finding is that a performance improvement can be achieved by simply changing the backoff parameter to 3 instead of using the default. While investigating the performance of probabilistic forwarding, it was found that the performance slightly improves compared to flooding. However, an analysis of QoE with both protocols showed that flooding is able to achieve a constant light level at the end of a dimming process, even if many packets are lost during the process. Despite the fact that a constant light level is reached with flooding at the end of the dissemination process, flooding has drawbacks. For example, with flooding nodes keep retransmitting a message for longer than necessary, sometimes even for seconds which leads to a very high redundancy. Additionally, during the flooding process, a lot of collisions happen, leading to a high PLR and visible effects during a dimming process. Due to this flooding was found unsuitable for a wireless lighting control application. The QoE analyses of probabilistic forwarding show that achieving a better QoS does not imply that the QoE improves as well. With probabilistic forwarding, no constant light level could be attained, and some lamps never even received a single message.

To address the problems found with flooding and probabilistic forwarding, many approaches are possible, e.g. flooding with a reduced number of forwarders, flooding protocols exploiting constructive interference such as Glossy or LWB, or protocols which follow a strict dissemination plan such as INFUSE. In this thesis, the approach of clustering was used to reduce the number of forwarding nodes in a first instance and also to self-organise the network topology as a cluster tree. This also allows for topology control. For example, if with CIDER a CH fails or dies the topology can be recovered and CS nodes will not be isolated. With other protocols such as Glossy or LWB if nodes die some nodes might become isolated which creates additional problems. Due to this, CIDER was developed, which is based on a weighted clustering approach to form a cluster tree topology in a self-learning manner. This means that CIDER forms a network topology without the need of any prior knowledge. The idea of CIDER is to elect a first CH that covers the most neighbours and only add additional CHs when the range needs to be extended. A convergence analysis performed with the help of colleagues from TU Eindhoven proved that CIDER is able to form a network in less than 5 minutes for networks with up to four CHs and an arbitrary number of CSs. This proves that CIDER is very well suited to form a topology especially designed for wireless lighting control networks in a reasonable amount of time. The second protocol of the D^3LC Suite addresses the problem of assigning non-overlapping channels to the different clusters to ensure that neighbouring clusters do not interfere with one another. For this, an available greedy colouring algorithm was applied and modified to fit the needs of the D^3LC Suite. This algorithm colours the network in a way that each cluster will be assigned a unique colour. At the same time, the colour defines which channel offset is used within a cluster. Finally, to address the problem of an organised control message dissemination, RLL was designed. RLL utilizes the cluster tree topology created with CIDER. By using RLL, it is ensured that messages are disseminated in a way that the simultaneous delivery of messages to all CSs is deterministic. This means that to some extent, all lamps receive a control command at the same time. For this, a new control command is first exchanged among all CHs, and then all CHs forward the message simultaneously to all CSs using the channel offset assigned. The suitability of RLL for wireless lighting control was proven by extensive computer simulations. Networks with up to 400 devices and several CHs were simulated to prove that RLL's operation is not influenced by an increasing network density. The result was that RLL fulfils all QoS and QoE requirements with low latencies and packet loss rates below 1%.

All three protocols are incorporated in the D^3LC Suite. To ensure that the D^3LC Suite also works as expected in real-world environments, extensive experiments were carried out. The experiments showed that the latency achieved was close to the simulation results. It was found that due to the processing limitations of the operating system, the full potential of RLL could not be exploited. Especially, the need for an immediate forwarding of messages by the CH could thus not be guaranteed. This limits RLL to covering only up to 3 hops with a latency of 230 ms while in an ideal system, this coverage could be doubled. However, three hops were always enough coverage for any setup investigated in this thesis. A major finding of the experiments was that interference has a much larger effect on the packet loss rate of D^3LC compared to the simulations. In an environment with almost no interference, the PLR was below 1%, which entirely fulfils the requirement of low loss rates for wireless lighting control networks. On the other hand, under heavy interference from multiple WiFi access points on multiple channels, PLRs of up to 10% were experienced. This shows that it is necessary to improve the robustness against WiFi interference of D^3LC. A first simple mechanism to further improve the delivery of the last message is to simply transmit the last control command twice. This would at least lower the probability of lamps being in different states at the end of a control process. This does not improve the general robustness, but it is a step towards the system behaviour expected by a user. Nevertheless, the D^{3}LC Suite promises to be a viable approach to address the challenges of large and dense wireless lighting control networks.

The LWB was analysed to show and further demonstrate that D^3LC is a suitable approach. The LWB claims to be one of the fastest and most reliable flooding protocols available. First, the LWB was analysed using different configurations. The findings are that the LWB may operate very well with applications where periodic traffic can be expected, but in the experiments conducted here, it is shown that the LWB has problems in achieving acceptable results with non-periodic traffic, such as event based traffic which occurs in lighting networks when a switch may be pressed at any time. While the PLR was similar compared to that of D^3LC in the same environment, the latency exceeds that of D^3LC by up to two seconds. Bringing the performance of the LWB into context with D^3LC shows that D^3LC performs better than the current state-of-the-art.

In this thesis it was not only shown that D^3LC is a viable approach for dense wireless lighting networks, furthermore this thesis provides a better understanding on how certain network parameters affect the wireless lighting network. For example, it was shown that if the latency varies throughout the network, e.g. nodes on one side of the network receive a control message earlier than other nodes, this results in an unpredictable lighting behaviour where lamps are not always in the same state. A user will perceive such effects as a kind of flickering which is an unwanted effect. Additional to the impact of latency on the lighting performance, the effects of high PLRs are presented. In networks using Flooding as the dissemination protocol, a high packet loss can be seen during the dimming process while at the end of a dimming process the lamps usually are in the same state. This higher packet loss is not directly noticeable by an user here. However, with other dissemination protocols such as probabilistic forwarding, the effects of a high PLR result in lamps which do not turn on. A user would then potentially replace a light bulb even though the light bulb is not broken. If more nodes are added to a lighting installation these effects worsen. Hence, the network density directly influences the network performance of state-of-the-art protocols where the latency and PLR increases if the network density increases.

The operation of D^3LC was demonstrated during two live demonstrations. The first demonstration was during the CeBIT 2017 where an exhibition visitor could control several lamps via a tablet. The same setup was used in a second event, the public day of the DEWI project in Graz 2017, where the demonstrator was examined by two external reviewers.

7.2 Outlook

The evaluation of D^3LC showed that it is a well-suited protocol stack to address the challenges of wireless lighting control. However, there are a number of measures for further improving the performance of D^3LC in the short term. The problems found within Contiki need to be addressed. This is best done by improving the processing time of Contiki itself. This requires an in-depth study of Contiki for understanding how the architecture is designed. Another option is to investigate whether the slotframe structure can be modified in a way that it still achieves a performance similar to the current implementation but that the processing time will not impact the overall performance.

As previously mentioned, the weighting parameters, as well as the timing parameters of CIDER, were often arbitrarily selected. To gain a better knowledge of how timing and weighting parameters influence the protocol operation and performance, more experiments are necessary using different sets of parameters. At the same time, additional environments could be selected to determine whether this also influences the performance, and candidate environments can be an outdoor application or a deployment in a theatre.

The current slotframe structure for RLL is designed to support a total of 8 hops. With this it is possible to cover a relatively large network. However, since D^3LC is designed to support all kinds of networks with more or less hops, it needs to be explored if a more dynamic slotframe structure can be implemented. For example, in networks with only a small number of CHs the slotframe could be shorter, which would also improve the overall network latency. This dynamic slotframe structure could reuse the information available from CIDER and RLL and could then set the slotframe structure accordingly.

As for the presented experiments, only 50 devices were available; an option for investigating how D^3LC performs with an increasing network density is to implement D^3LC in a simulation environment. This model could also be used to analyse the effects of the weighting and timing parameters.

The LWB's current implementation was not found to be suitable for lighting applications due to not being able to adapt quickly enough to non-periodic traffic. The LWB follows a communication schedule where assigned slots will be removed from the schedule when a node is inactive. This limits LWB to mostly periodic traffic applications or to applications where a higher latency is acceptable. To solve this problem, one option is to introduce fixed communication slots for specific devices such as a light switch. To this end, an in-depth study is necessary to understand the LWB processes in detail and to see if such changes are possible.

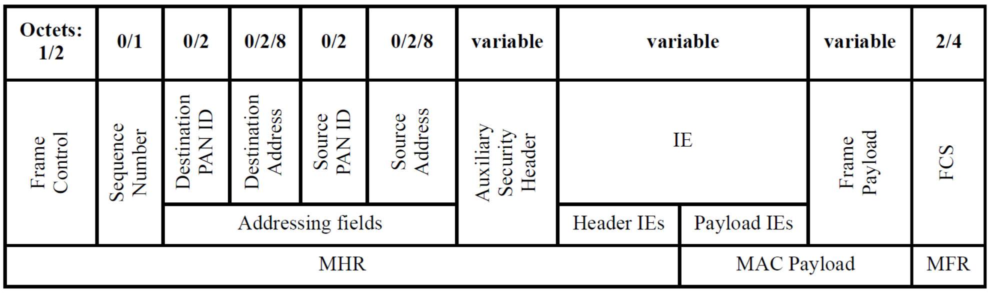
In the long term, D^3LC could be modified for designing building management systems integrating both the control of luminaires and the collection of sensor data. Using the current scheduler, many lamps remain in an idle state while messages are only exchanged within the CHs. Also, a lighting control process is usually a rather infrequent process which occurs only several times a day. Hence, the complete lighting network is in idle mode most of the time. By designing a new smart scheduler, the lighting network could be used as a backbone network for sensor data collection while still being able to meet the QoS and QoE requirements of lighting control applications. Especially the idle CSs could be used for a backbone network in which these nodes are responsible to collect sensor data. It would be important for such a network that the reaction time and robustness of the lighting network remain unaffected.

The increased PLR under heavy WiFi interference highlights the need to allocate channels (both for WiFi and for IEEE 802.15.4) in a meaningful way because an excessive presence of interference will always limit the performance of wireless solutions. Usually, WiFi and IEEE 802.15.4 networks work separately, and cooperative strategies need to be developed. One approach could be to blacklist channels for IEEE 802.15.4 networks with bad Signal-to-Noise Ratio (SNR) measurements. Where the measurements are carried out during active transmissions. By blacklisting the specific channels, these are excluded from the hopping sequence of TSCH, and for this, it is necessary to reconfigure the network with a new hopping sequence. A mechanism of how to propagate the new hopping sequence needs to be defined. If a device detects an unusual and sudden change in the SNR on a specific channel, this channel could be blacklisted for increasing the overall robustness of the system.

Appendix A

Frame Format of D^3LC messages

The IEEE802.15.4 general frame format is shown in Figure A.1. In the standard, it is defined that the frame size should not exceed 127 bytes; however, with IEEE802.15.4, it is possible to define a variable payload of up to 124 bytes [70], depending on the configuration of the MAC frame. With D^3LC, IEEE802.15.4 within Contiki is configured in such a manner that the payload has a total size of up to 112 bytes. The configuration of the IEEE802.15.4 frame used for this work is shown in Table A.1; however, these are suggestions given by the standard, while the format of the D^3LC frame is the suggestion of this work. All fields except the $D^3LCPayload$ are set by the implementation of TSCH within Contiki, which means these parameters are always used for all frames transmitted. Only the payload is variable and can be changed. The format of the D^3LC payload is shown in Table A.2 and represents the payload of the IEEE802.15.4 frame. The payload is generated by the higher layers and is included only in an



Octets: 1/2	0/1	0/2	0/2/8	0/2	0/2/8	variable	variable		variable	2/4
Frame Control	Sequence Number	Destination PAN ID	Destination Address	Source PAN ID	Source Address	Auxiliary Security Header	IE		Frame Payload	FCS
		Addressing fields					Header IEs	Payload IEs		
MHR								MAC Payload		MFR

Figure A.1: IEEE802.15.4 General frame format [70]

Byte	1-2	3	4-5	6-7	8-9	10 - 11	12 - 123	124 - 127
Type	Frame Control	Sequence Number	Destination PAN ID	Destination Address	Source PAN ID	Source Address	D^3LC Payload	FCS

Table A.1: Contiki IEEE802.15.4 frame including the D^3LC payload

IEEE802.15.4 frame. The identifier field defines to which protocol the frame belongs, e.g. CIDER. The *SRC* and *DST* fields includes the source and destination address of the D^3LC message. This addresses are the same as the ones for the MAC protocol in this work, but can be different if necessary, e.g. in Internet Protocol (IP) networks. The payload field includes further fields defined by the different protocols (CIDER, RLL, and Colouring).

Byte	1	2 - 3	4 - 5	6 - 112
Type	Identifier	SRC	DST	Payload

Table A.2: D^3LC Payload format

A.1 Identifier field

The possible values of the identifier field are defined in Table A.3. The identifier field has the size of one byte

Identifier value	Description
0x01	CIDER message
0x02	RLL message
0x03	Colouring message

Table A.3: Values of the identifier field

A.2 SRC field

The SRC field includes the source address of the transmitter. The source address has the size of two bytes and is in the range of 0x0001 to 0xFFFE. The *source address* field of the IEEE802.15.4 frame is the same as the source address field of the D^3LC frame. This address is needed for processing in the higher layers.

A.3 DST field

The DST field includes the destination address of the packet. The destination address has the size of two bytes. For unicast transmissions, the destination address is in range of 0x0001 to 0xFFFE. The broadcast destination address is defined as 0xFFFF. The destination address is the same as the *destination address* field of the IEEE802.15.4 frame. This address is needed for processing in the higher layers.

A.4 Payload field

The payload field includes the content of the sub-protocols (CIDER, Colouring, and RLL). The size of the payload is variable. Bytes 6 to 112 are reserved for the payload in a D^3LC frame.

Appendix B

Frame format of CIDER messages

A general frame format of a CIDER message is shown in Section B.1. The specific frame format for all CIDER messages is shown in Section B.2 and the sections following it.

B.1 General Frame Format

The CIDER frame is embedded into the payload field of the D^{3}LC frame. The format of a CIDER frame is shown in Table B.1. The frame is composed of an identifier field and a payload field.

Byte	1 - 2	3 - 106
Type	Identifier	Payload

Table B.1: CIDER frame format

B.1.1 Identifier field

The possible values of the identifier field are defined in Table B.2. The identifier field is two bytes long.

Identifier Value	Description
0x01	PING
0x02	Neighbour Update
0x03	Utility Update
0x04	CH Competition
0x05	CH
0x06	CS PING
0x07	CS
0x08	CH Promote
0x09	CH Promote Ack
0x64	Complete
0x65	Incomplete
0xc8	Undefined

Table B.2: Values of the identifier field

B.1.2 Payload field

The Payload field includes the content of the different CIDER messages. The size of the payload is variable. Bytes 3 to 106 is reserved for the payload in a CIDER message.

B.2 PING message

The PING message is used by a node to advertise its presence while in the PING state. The identifier field of the CIDER message is set to 0x01 for a PING message. The PING message has no payload.

Byte	1 - 104
Type	Not used

Table B.3: PING message format

B.3 Neighbour Update message

The Neighbour Update message is used by a node while in the Neighbour Update state to transmit a node's neighbour and cluster degrees. For a Neighbour Update, the identifier field of the CIDER message is set to 0x02. The message format consists of two bytes for the neighbour degree as well as two bytes for the cluster degree. The remaining bytes are not used.

Byte	1 - 2	3 - 4	5 - 104
Type	Neighbour degree	Cluster degree	Not used

Table B.4: Neighbour Update message format

B.4 Utility Update message

The Utility Update message is used by a node to transmit its utility while in the Utility Update state. For a Utility Update, the identifier field of the CIDER message is set to 0x03. The message format consists of two bytes for the utility value. The remaining bytes are not used.

Byte	1 - 2	3 - 104
Type	Utility	Not used

Table B.5: Utility Update message format

B.5 CS PING message

The CS PING message is used by a node while in the CS PING state to advertise that it has not yet been assigned to any cluster. For a CS PING, the identifier field of the CIDER message is set to 0x06. The message format consists of two bytes for the utility value, two bytes for the neighbour degree, and two bytes for the cluster degree. The remaining bytes are not used.

Byte	1 - 2	3 - 4	5 - 6	7 - 104
Type	Utility	Neighbour degree	Cluster degree	Not used

Table B.6: CS PING message format

B.6 CH Competition message

The CH Competition message is used by a node while in the CH Competition state to advertise that it is a potential CH. For a CH Competition, the identifier field of the CIDER message is set to 0x04. The message format consists of two bytes for the utility value. The remaining bytes are not used.

Byte	1 - 2	3 - 104
Type	Utility	Not used

Table B.7: CH competition message format

B.7 CH message

The CH message is used by a node while in the CH state to advertise that a node is a CH. For a CH message, the identifier field of the CIDER message is set to 0x05. The message format consists of two bytes for the utility value, one byte for the tier, one byte for the colour, and 100 bytes for the list of a maximum of 50 cluster members.

Byte	1 - 2	3	4	5 - 104
Type	Utility	Tier	Colour	list of cluster members

Table B.8: CH message format

B.8 CS message

The CS message is used by a node while in the CS state to advertise that it is assigned to a CH. For a CS message, the identifier field of the CIDER message is set to 0x07. The message format consists of two bytes for the utility value, two bytes for the neighbour degree, two bytes for the cluster degree, and two bytes for the link address of the parent node. The remaining bytes are not used.

Byte	1 - 2	3 - 4	5 - 6	7 - 8	9 - 104
Type	Utility	Neighbour degree	Cluster degree	Parent	Not used

Table B.9: CS message format

B.9 CH Promote message

The CH Promote message is used by a node while in the CH Promote state to promote a selected node to act as CH. For CH Promote, the identifier field of the CIDER message is set to 0x08. The message format consists of two bytes for the link address of the parent node and one byte for the tier of the parent node. The remaining bytes are not used.

Byte	1 - 2	3	4 - 104
Type	Parent	Tier	Not used

Table B.10: CH Promote message format

B.10 CH Promote ACK message

The CH Promote ACK message is used by a node while in the CH Promote ACK state to acknowledge the reception of a CH Promote message. For CH Promote ACK, the identifier field of the CIDER message is set to 0x09. The CH Promote ACK message has no payload.

Byte	1 - 104
Type	Not used

Table B.11: CH Promote ACK message format

B.11 Complete message

The Complete message is used by a node while in the CH state to determine whether the clustering process is finished or not. For a Complete message, the identifier field of the CIDER message is set to 0x64. The Complete message has no payload.

Byte	1 - 104
Type	Not used

Table B.12: Complete message format

B.12 Uncomplete message

The incomplete message is used by a node while in the CH state to inform other CHs that there are still nodes not assigned to a CH in its neighbourhood. For an Uncomplete message, the identifier field of the CIDER message is set to 0x65. The Uncomplete message has no payload.

Byte	1 - 104
Type	Not used

Table B.13: Uncomplete message format

Appendix C

Frame format of Colouring messages

The general frame format of a COLOUR message is shown in Section C.1. Following this, the specific frame format for all COLOUR messages is shown.

C.1 General Frame Format

The COLOUR frame is embedded into the payload field of the D^3LC frame. The frame format of a COLOUR frame is shown in Table C.1. The frame consists of two fields, an identifier field, and a payload field.

Byte	1 - 2	3 - 106
Type	Identifier	Payload

Table C.1: COLOUR frame format

C.2 Update message

The Update message is used by all nodes to update their neighbourhood. For an Update, the identifier field of the COLOUR message is set to 0x01. The payload of an Update message is defined in Table C.2

Byte	1 - 2	3 - 4	5 - 6	7 - 104
Type	$d(u)$	sd(u)	rand(u)	Not used

Table C.2: Update message frame format

C.3 Release message

The Release message is used by all nodes to update their neighbourhood with the selected colour. For a Release, the identifier field of the COLOUR message is set to 0x02. The payload of a Release message is defined in Table C.3

Byte	1 - 2	3 - 104
Type	COLOUR	Not used

Table C.3: Release message frame format

C.4 Complete message

The Complete message is used by the Tier 0 CH to determine whether the colouring process has finished or not. For a Complete message, the identifier field of the COLOUR message is set to 0x65. The Complete message has no payload.

Byte	1 - 104
Type	Not used

Table C.4: Complete message frame format

C.5 Uncompleted message

The Uncompleted message is used by the Tier 0 CH to inform other CHs that its colour has not yet been selected or is not unique. For an Uncompleted message, the identifier field of the COLOUR message is set to 0x66. The Uncompleted message has no payload.

Byte	1 - 104
Type	Not used

Table C.5: Uncompleted message frame format

Appendix D

Frame format of RLL messages

The frame format of an RLL frame is based on the D^3LC frame (see Table A.2). However, for the RLL Identifier, the field is set to *0x02*. The *SRC* field represents the source address of the current forwarding node while the *DST* field represents the destination address, which is always set to the broadcast address (*0xffff*) for all experiments carried out in this thesis. In general, RLL can be used in different applications; therefore, it is possible to use RLL also for Unicast messages by changing the destination address. The RLL payload can be up to a total 106 bytes. This is based on the 127-byte IEEE802.15.4 frame format. A full RLL frame is presented in Table D.1.

Byte	1	2 - 3	4 - 5	6 - 112
Type	Identifier: *RLL*	SRC	DST: *0xffff*	Payload

Table D.1: RLL frame format

Appendix E

Simulation configuration for Broadcast Storm simulations with OMNeT++

```
[General]
ned-path = /home/user/Workspace/DEWI/DEWI/simulations/;/
   ↪ home/user/Workspace/DEWI/DEWI/src
network = dewi.simulations.UseCase1_Lighting.Dimming.
   ↪ nonBeaconCSMA.Network
num-rngs = 2
repeat = 10000
cmdenv-express-mode = true
tkenv-plugin-path = ../../../Etc/plugins
description = "Dimming Application"
sim-time-limit = 15d
#Debug
parallel-simulation = false
**.debug = false
**.coreDebug = false

output-vector-file = ${resultdir}/vector/${runid}.vec
output-scalar-file = ${resultdir}/scalar/${runid}.sca
```

```
###########################################################
##############Configure NodeLocation#######################
###########################################################
*.lamp[*].mobility.distanceToWall = 2.5m
*.lamp[*].mobility.distanceToCeiling = 0.3m

*.lamp[*].X = 80m
*.lamp[*].Y = 20m
*.lamp[*].Z = 3.3m

**.lightSwitch.mobility.initFromDisplayString = false
**.lightSwitch.mobility.initialX = 0m
**.lightSwitch.mobility.initialY = 20m
**.lightSwitch.mobility.initialZ = 1.2m

###########################################################
############## Configure DataCenter #####################
###########################################################
**.DataCenter.recordValues = true

###########################################################
##############Configure ChannelControl#####################
###########################################################
**.channelControl.coreDebug = false
**.channelControl.alpha = 3 #// path loss coefficient
**.channelControl.carrierFrequency = 2.4GHz #// base
   ↪ carrier frequency of all the channels (in Hz)
**.channelControl.maxInterferenceDistance = -1m ##6m for
   ↪ 240 nodes ##26m 16 nodes
**.channelControl.numChannels = 16 #// number of radio
   ↪ channels (frequencies)
**.channelControl.pMax = 1mW #// maximum sending power used
   ↪  for this network (in mW)
```

```
**.channelControl.propagationModel = "NakagamiModel" #("
   ↪ FreeSpaceModel","TwoRayGroundModel","RiceModel","
   ↪ RayleighModel","NakagamiModel","
   ↪ LogNormalShadowingModel")
**.channelControl.sat = -95dBm #// signal attenuation
   ↪ threshold (in dBm)
**.channelNumber = 11

############################################################
##########Configure WLAN Module###########################
############################################################
#Configure WLAN Queue
**.wlan.ifqType = "DropTailQueue"
**.ifq.frameCapacity = 50
**.rxSetupTime = 0.00108s

#Configure WLAN PHY
**.wlan.phy.usage_radio_recv = 23mA
**.wlan.phy.usage_radio_idle = 0.021mA
**.wlan.phy.usage_radio_sleep = 0.001mA

############################################################
#############Configure PHYLayer##########################
############################################################
#Configure PHY-Layer
**.wlan.phy.channelNumber = 11
**.wlan.phy.NoBitError = false

**.wlan.phy.sensitivity = -95dBm #[dBm]
**.wlan.phy.receptionThreshold = -110dBm
**.wlan.phy.thermalNoise = -110dBm #[dBm]
**.wlan.phy.pathLossAlpha = 3
**.wlan.phy.TransmissionAntennaGainIndB = 0dB
**.wlan.phy.ReceiveAntennaGainIndB = 0dB
```

```
**.wlan.phy.SystemLossFactor = 0dB
**.wlan.phy.nak_m = 2.9
**.wlan.phy.aMaxPHYPacketSize = 127
**.wlan.phy.aTurnaroundTime = default #312.5
**.phy.snirThreshold = 4dB

###########################################################
##############Configure MAC Layer##########################
###########################################################
**.wlan.mac.useMACAcks = true
**.wlan.mac.sifs = default #0.005001s #0.005s

**.wlan.mac.macMaxFrameRetries = 7
**.wlan.mac.macAckWaitDuration = default
**.wlan.mac.ccaDetectionTime = 0.000128s
**.wlan.mac.rxSetupTime = default
**.wlan.mac.aTurnaroundTime = default #0.005s
**.wlan.mac.aUnitBackoffPeriod = default
**.wlan.mac.ackLength = default
**.wlan.mac.headerLength = default
**.wlan.mac.backoffMethod = "exponential"
**.wlan.mac.contentionWindow = 2
**.wlan.mac.macMinBE = default
**.wlan.mac.macMaxBE = default

###########################################################
##############Configure InterfaceTable#####################
###########################################################
**.interfaceTable.displayAddresses = false

###########################################################
##############Configure NetworkConfigurator################
###########################################################
**.configurator.addDefaultRoutes = false
```

```
**.configurator.addStaticRoutes = false
**.configurator.addSubnetRoutes = false
**.configurator.assignAddresses = true
**.configurator.assignDisjunctSubnetAddresses = true
**.configurator.dumpAddresses = false
**.configurator.dumpConfig = "ConfigTest"
**.configurator.dumpLinks = false
**.configurator.dumpRoutes = false
**.configurator.dumpTopology = false
**.configurator.optimizeRoutes = false

###########################################################
####################Configure UDPApp#######################
###########################################################
####Sender####
**.lightSwitch.udpAppType = "UDPLightSwitch"
**.lightSwitch.numUdpApps = 1

**.lightSwitch.udpApp[0].excludeLocalDestAddresses = true
**.lightSwitch.udpApp[0].chooseDestAddrMode = "perBurst"
**.lightSwitch.udpApp[0].destAddrRNG = 1
**.lightSwitch.udpApp[0].localPort = 1234
**.lightSwitch.udpApp[0].destPort = 1234
**.lightSwitch.udpApp[0].startTime = 5s
**.lightSwitch.udpApp[0].stopTime = 11d
**.lightSwitch.udpApp[0].messageLength = 72B
**.lightSwitch.udpApp[0].burstDuration = 2s
**.lightSwitch.udpApp[0].sleepDuration = 3600s
**.lightSwitch.udpApp[0].sendInterval = 0.1s
**.lightSwitch.udpApp[0].delayLimit = 200s
**.lightSwitch.udpApp[0].outputInterface = "wlan"
**.lightSwitch.udpApp[0].outputInterfaceMulticastBroadcast
   ↪ = "wlan"
**.lightSwitch.udpApp[0].setBroadcast = true
```

```
##all other nodes##
**.lamp[*].udpAppType = "UDPLamp"
**.lamp[*].numUdpApps = 1
**.lamp[*].udpApp[0].localPort = 1234
**.lamp[*].udpApp[0].destPort = 1234
**.lamp[*].udpApp[0].messageLength = 72B #lenght in Bytes
**.lamp[*].udpApp[0].sendInterval = 5s #+ uniform(-0.001s
   ↪ ,0.001s,1)
**.lamp[*].udpApp[0].burstDuration = 2s #uniform(1s,4s,1)
**.lamp[*].udpApp[0].stopTime = -1s
**.lamp[*].udpApp[0].startTime = 5s
**.lamp[*].udpApp[0].delayLimit = 200s
**.lamp[*].udpApp[0].setBroadcast = true
**.lamp[*].udpApp[0].outputInterface = "wlan"
**.lamp[*].udpApp[0].outputInterfaceMulticastBroadcast = "
   ↪ wlan"
**.lamp[*].udpApp[0].isSource = false
**.lamp[*].udpApp[0].isSink = true
**.lamp[*].udpApp[0].resendBroadcast = true
**.Method = "NONE"

**.udpApp[0].*_E2ED_*.vector-recording = true
**.udpApp[0].*_HopCount_*.vector-recording = true
**.udpApp[0].*_HopCount_*.with-akaroa = true
**.udpApp[0].*_ReceivedMSG_*.scalar-recording = true
**.udpApp[0].*_ReceivedMSG_*.with-akaroa = true
**.udpApp[0].*_ReMSG_*.scalar-recording = true
**.udpApp[0].*_ReMSG_*.with-akaroa = true
**.udpApp[0].*_DRMSG_*.scalar-recording = true
**.udpApp[0].*_DRMSG_*.with-akaroa = true
**.lightSwitch.udpApp[0].*_MSGSent_*.scalar-recording =
   ↪ true
**.lightSwitch.udpApp[0].*_MSGSent_*.with-akaroa = true
```

```
**.lightSwitch.udpApp[0].*_BurstsSent_*.scalar-recording =
   ↪ true
**.lightSwitch.udpApp[0].*_BurstsSent_*.with-akaroa = true
**.**_numCollision_*.scalar-recording = true
**.udpApp[0].*_numCollision_*.with-akaroa = true
**.**_nbDroppedFrames_*.scalar-recording = true
**.udpApp[0].*_nbDroppedFrames_*.with-akaroa = true
**.**_nbRxFrames_*.scalar-recording = false
**.udpApp[0].*_nbRxFrames_*.with-akaroa = true
**.**_nbTxFrames_*.scalar-recording = true
**.udpApp[0].*_nbTxFrames_*.with-akaroa = true
**.**_nbRxFramesBroadcast_*.scalar-recording = true
**.vector-recording = false
**.with-akaroa = false

###Configure Destination Addess
**.lightSwitch.udpApp[0].destAddresses = "255.255.255.255"
   ↪ #moduleListByPath("**")

###########################################################
####################Output Vectors#########################
###########################################################
**.scalar-recording = false
```

Appendix F

Simulation configuration for RLL simulations with OMNeT++

```
[General]
network = dewi.simulations.UseCase1_Lighting.Dimming.RLL.Network

ned-path = /home/user/Workspace/DEWI/DEWI/simulations/;/home/user/
    ↪ Workspace/DEWI/DEWI/src
num-rngs = 2
cmdenv-express-mode = true
#tkenv-plugin-path = ../../../Etc/plugins
result-dir = /home/user/DATA/test/
description = "Dimming Application"
sim-time-limit = 15d

#description = "RLL"

repeat = 10000

#Debug
parallel-simulation = false
debug-on-errors = true
**.debug = false
```

```
**.coreDebug = false
**.vector-recording = false
**.scalar-recording = false
#Number of Nodes
**.numHosts = 56

###########################################################
##############Configure NodeLocation#######################
###########################################################

**.lightSwitch.app.numberOfBursts = 5000

*.lamp[*].mobility.distanceToWall = 2.5m
*.lamp[*].mobility.distanceToCeiling = 0.3m

*.lamp[*].X = 80m
*.lamp[*].Y = 40m
*.lamp[*].Z = 3.3m

**.lightSwitch.app.LightSwitch = true
**.lightSwitch.app.StartTime = 1000s
**.lightSwitch.networklayer.capablePanCoor = false
**.BurstPause = 1s
**.lightSwitch.mobility.initFromDisplayString = false
**.lightSwitch.mobility.initialX = 0m
**.lightSwitch.mobility.initialY = 20m
**.lightSwitch.mobility.initialZ = 1.2m

**.gateWay.mobility.initialX = uniform(0m,80m)
**.gateWay.mobility.initialY = uniform(0m,40m)
**.gateWay.mobility.initialZ = 0m

*.gateWay.mobility.initFromDisplayString = false
```

```
##########################################################
##############Configure ChannelControl#####################
###########################################################
#**.channelControl.coreDebug = false
**.channelControl.alpha = 3 #// path loss coefficient
**.channelControl.carrierFrequency = 2.4GHz #// base carrier
   ↪ frequency of all the channels (in Hz)
**.channelControl.maxInterferenceDistance = -1m ##6m for 240 nodes
   ↪  ##26m 16 nodes
**.channelControl.numChannels = 27 #// number of radio channels (
   ↪ frequencies)
**.channelControl.pMax = 1mW #// maximum sending power used for
   ↪ this network (in mW)
**.channelControl.propagationModel = "TwoRayGroundModel" #("
   ↪ FreeSpaceModel","TwoRayGroundModel","RiceModel","
   ↪ RayleighModel","NakagamiModel","LogNormalShadowingModel")
**.channelControl.sat = -95dBm #// signal attenuation threshold (
   ↪ in dBm)
**.channelNumber = 11

###########################################################
###########Configure WLAN Module###########################
###########################################################
#Configure WLAN Queue
**.wlan.ifqType = "Ieee802154eQueue"
**.ifq.frameCapacity = 256

#Configure WLAN PHY
**.wlan.phy.usage_radio_recv = 23mA
**.wlan.phy.usage_radio_idle = 0.021mA
**.wlan.phy.usage_radio_sleep = 0.001mA

###########################################################
##############Configure PHYLayer###########################
```

```
###########################################################

#Configure PHY-Layer
#**.wlan.phy.channelNumber = 11
**.wlan.phy.NoBitError = false

**.wlan.phy.sensitivity = -95dBm #[dBm]
**.wlan.phy.receptionThreshold = -95dBm
**.wlan.phy.thermalNoise = -110dBm #[dBm]
**.wlan.phy.pathLossAlpha = 3
**.wlan.phy.TransmissionAntennaGainIndB = 0dB
**.wlan.phy.ReceiveAntennaGainIndB = 0dB
**.wlan.phy.SystemLossFactor = 0dB
**.wlan.phy.nak_m = 2.9
**.wlan.phy.aMaxPHYPacketSize = 127
**.wlan.phy.aTurnaroundTime = default #312.5
**.phy.snirThreshold = 4dB

**.transmitterPower = 0.1mW

###########################################################
##############Configure MAC Layer##########################
###########################################################
**.wlan.mac.backoffMethod = "exponential"
**.wlan.mac.contentionWindow = 2
**.wlan.mac.macMinBE = 0
**.wlan.mac.macMaxBE = 7
**.wlan.mac.useIeee802Ctrl  = true   # to use the Ieee802Ctrl
   ↪ Network messages
**.lightSwitch.wlan.mac.useTSCH    = false   # use the TSCH mode (
   ↪ Std 802.15.4e-2012)
**.gateWay.wlan.mac.useTSCH    = true
**.lamp[*].wlan.mac.useTSCH    = false   # use the TSCH mode (Std
   ↪ 802.15.4e-2012)
```

```
**.lamp[*].wlan.mac.setRxOnWhenIdle  = false   # default: true;
**.wlan.mac.setRxOnWhenIdle  = false   # default: true;
**.gateWay.**.isPANCoor   = true
**.**.mac.isPANCoor   = false
**.**.mac.useTimeslotID   = 0   # default: 0 (problem at the std
   ↪ default with the 2200 RxWait time (in the 6TSCH is it
   ↪ declaired as 2000 RxWait time))
# use: 2 for "corrected" default or 1 for the 6TSCH minimal
   ↪ example
**.**.mac.useHoppingSequenceID = 0   # default: 0 (all 16 channels
   ↪ )
**.**.mac.useMetrics   = true
**.mac.useCCA = false
**.mac.panCoorName      = "gateWay"
**.mac.BO = 5
**.mac.SO = 3


##########################################################
#############Configure InterfaceTable#####################
###########################################################

**.interfaceTable.displayAddresses = false

###########################################################
#############Configure NetworkConfigurator################
###########################################################

**.configurator.addDefaultRoutes = false
**.configurator.addStaticRoutes = false
**.configurator.addSubnetRoutes = false
**.configurator.assignAddresses = true
**.configurator.assignDisjunctSubnetAddresses = true
**.configurator.dumpAddresses = false
```

```
**.configurator.dumpConfig = "ConfigTest"
**.configurator.dumpLinks = false
**.configurator.dumpRoutes = false
**.configurator.dumpTopology = false
**.configurator.optimizeRoutes = false

[Config _56_0dBm]
**.numHosts = 56
result-dir = /home/user/DATA/Sim_Results/Lighting/Dimming/RLL/
   ↪ LeftMiddle/raw/_0dBm/_56
**.wlan.phy.transmitterPower = 1mW

[Config _100_0dBm]
**.numHosts = 100
result-dir = /home/user/DATA/Sim_Results/Lighting/Dimming/RLL/
   ↪ LeftMiddle/raw/_0dBm/_100
**.wlan.phy.transmitterPower = 1mW

[Config _200_0dBm]
**.numHosts = 200
result-dir = /home/user/DATA/Sim_Results/Lighting/Dimming/RLL/
   ↪ LeftMiddle/raw/_0dBm/_200
**.wlan.phy.transmitterPower = 1mW

[Config _400_0dBm]
**.numHosts = 400
result-dir = /home/user/DATA/Sim_Results/Lighting/Dimming/RLL/
   ↪ LeftMiddle/raw/_0dBm/_400
**.wlan.phy.transmitterPower = 1mW
```

Bibliography

[1] History of IoT — Background Information and Timeline of the Trending Topic. 2

[2] PRISM - Probabilistic Symbolic Model Checker, 2017. 146

[3] S. Banerjee A. Mishra and W. Arbaugh. Weighted coloring based channel assignment for WLANs. 2005. 121

[4] H. S. AbdelSalam and S. Olariu. BEES: BioinspirEd backbonE Selection in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, **23**(1):44–51, jan 2012. 102

[5] Nicola Accettura, Maria Rita Palattella, Gennaro Boggia, Luigi Alfredo Grieco, and Mischa Dohler. Decentralized Traffic Aware Scheduling for multi-hop Low power Lossy Networks in the Internet of Things. In *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 1–6. IEEE, jun 2013. 117, 119

[6] M. Mehdi Afsar and Mohammad-H. Tayarani-N. Clustering in sensor networks: A literature survey. *Journal of Network and Computer Applications*, **46**:198–226, nov 2014. 96

[7] A. Manjeshwar Agrawal and D. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. *in:Proceedings of 15th International Parallel and Distributed Processing Symposium(IPDPS'01)*, **00**(C):2009–2015, 2001. 97

[8] L. Alexandru, P. Valentin, and S. Stefan. Adaptive Channel Selection Algorithm for a Large Scale Street Lighting Control ZigBee Network. *Electronics and Electrical Engineering*, **19**(9):105–109, nov 2013. 117

[9] W. R. Alling. The Integration of Microcomputers and Controllable Output Ballasts-A New Dimension in Lighting Control. *IEEE Transactions on Industry Applications*, **IA-20**(5):1198–1205, sep 1984. 15

[10] J.M. Alonso, J. Ribas, J.J. Del Coz, A.J. Calleja, E. Lopez, and M. Rico-Secades. Intelligent control system for fluorescent lighting based on LonWorks technology. In *IECON '98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.98CH36200)*, **1**, pages 92–97. IEEE, aug 1998. 15

[11] Amazon. Amazon Dash Button. 2

[12] A. Ariza-Quintana, E. Casilari, and A. Triviño Cabrera. Implementation of MANET Routing Protocols on OMNeT++. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, Simutools '08, pages 80:1–80:4, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 38, 39

[13] Mahesh (Umamaheswaran) Arumugam. Infuse. In *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys '04*, SenSys '04, page 281, New York, New York, USA, 2004. ACM Press. 131

[14] Emmanuel Baccelli, Oliver Hahm, Mesut Gunes, Matthias Wahlisch, and Thomas Schmidt. RIOT OS: Towards an OS for the Internet of Things. In *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 79–80. IEEE, apr 2013. 44

[15] Sachin Bhardwaj, Aly A. Syed, Tanir Ozcelebi, and Johan Lukkien. Power-managed smart lighting using a semantic interoperability

architecture. *IEEE Transactions on Consumer Electronics*, **57**(2):420–427, may 2011. 16, 22

[16] K.K. Kim B.J. Leung. Frequency assignment for IEEE 802.11 wireless networks. 2003. 121

[17] Remi Bosman, Johan Lukkien, and Richard Verhoeven. An Integral Approach to Programming Sensor Networks. In *2009 6th IEEE Consumer Communications and Networking Conference*, pages 1–5. IEEE, jan 2009. 22

[18] Athanassios Boulis. Castalia. In *Proceedings of the 5th international conference on Embedded networked sensor systems - SenSys '07*, SenSys '07, page 407, New York, New York, USA, 2007. ACM Press. 38, 39

[19] Daniel Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, **22**(4):251–256, apr 1979. 121, 122

[20] P. A. Brown. Power Line Communications - Past Present and Future. *Proceedings of International Symposium on Power-line Communications and its Applications*, 1999. 16

[21] D Caicedo, S Li, and A Pandharipande. Smart lighting control with workspace and ceiling sensors. *Lighting Research and Technology*, page 1477153516629531, feb 2016. 16, 18

[22] Matteo Ceriotti, Michele Corra, Leandro D'Orazio, Roberto Doriguzzi, Daniele Facchin, Sc Tefan Guna, Gian Paolo Jesi, Renato Lo Cigno, Luca Mottola, Amy L. Murphy, Massimo Pescalli, Gian Pietro Picco, Denis Pregnolato, and Carloalberto Torghele. Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels. *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 187–198, apr 2011. 16, 26

[23] Xinjie Chang. Network simulations with OPNET. In *Proceedings of the 31st conference on Winter simulation Simulation—a bridge to the future - WSC '99*, **1** of *WSC '99*, pages 307–314, New York, New York, USA, 1999. ACM Press. 37, 38

[24] Soamsiri Chantaraskul. Multi-channel Utilization Algorithms for IEEE 802.15.4 based Wireless Network: A Survey. *Engineering Journal*, **17**(3):119–126, jul 2013. 117

[25] M. Chatterjee, S. K. Das, and D. Turgut. An on-demand weighted clustering algorithm (WCA) for ad hoc networks. In *Globecom '00 - IEEE. Global Telecommunications Conference. Conference Record (Cat. No.00CH37137)*, **3**, pages 1697–1701 vol.3. IEEE, nov 2000. 107, 108

[26] Mainak Chatterjee, Sajal K. Das, and Damla Turgut. WCA: AWeighted Clustering Algorithm forMobile Ad Hoc Networks. *Cluster Computing*, **5**(2):193–204, apr 2002. xxiii, 100, 103, 108

[27] Chee-Yee Chong and S.P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, **91**(8):1247–1256, aug 2003. 1

[28] Feng Chen and Falko Dressler. A simulation model of IEEE 802.15. 4 in OMNeT++. *Proc. of the 6th GI/ITG KuVS Fachgesprach Drahtlose Sensornetze (FGSN)*, pages 35–38, 2007. 43

[29] Peng Cheng, Fan Zhang, Jiming Chen, Youxian Sun, and Xuemin Shen. A Distributed TDMA Scheduling Algorithm for Target Tracking in Ultrasonic Sensor Networks. *IEEE Transactions on Industrial Electronics*, **60**(9):3836–3845, sep 2013. xxiii, 123

[30] Wonchang Choi and Miae Woo. A Distributed Weighted Clustering Algorithm for Mobile Ad Hoc Networks. pages 73–73, feb 2006. 108

[31] Contiki-OS. Contiki: The Open Source Operating System for the Internet of Things. 43

[32] Conrad Dandelski, Xiangyu Wang, and Jean-Paul Linnartz. Private Conversation with Philips Lighting B.V., 2016. 8

[33] Conrad Dandelski, Bernd Ludwig Wenning, Michael Kuhn, and Dirk Pesch. Broadcast storm problem in dense wireless lighting control networks. In *Proceedings of the International Symposium on Wireless Communication Systems*, **2016-April**, pages 91–95, 2016. xviii, 83, 84

[34] Paul Darbee. INSTEON: the details. *Smarthome Technology*, pages 1–64, 2005. 33

[35] Upeka De Silva, Adisorn Lertsinsrubtavee, Arjuna Sathiaseelan, and Kanchana Kanchanasut. Named Data Networking Based Smart Home Lighting. In *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference - SIGCOMM '16*, SIGCOMM '16, pages 573–574, New York, New York, USA, 2016. ACM Press. 16, 24

[36] Jean-Jacques DeLisle. Major M2M and IoT Challenges You Need to Know, 2015. xviii, 3, 4

[37] Fuhu Deng. *An Autonomous Channel Selection Algorithm for WLANs An autonomous channel selection algorithm for WLANs*. PhD thesis, 2014. 117

[38] Indika Sanjeewa Abeywickrama Dhanapala, Ramona Marfievici, Sameera Palipana, Piyush Agrawal, and Dirk Pesch. Modeling WiFi Traffic for White Space Prediction in Wireless Sensor Networks. In *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, pages 551–554. IEEE, oct 2017. 172

[39] Ping Ding, JoAnne Holliday, and Aslihan Celik. Distributed Energy-Efficient Hierarchical Clustering for Wireless Sensor Networks. In *Lecture Notes in Computer Science*, Lecture Notes in Computer Science, pages 322–339. Springer, Berlin, Heidelberg, jun 2005. xxiii, 97, 98

[40] Marc Domingo-Prieto, Tengfei Chang, Xavier Vilajosana, and Thomas Watteyne. Distributed PID-Based Scheduling for 6TiSCH Networks. *IEEE Communications Letters*, **20**(5):1006–1009, may 2016. 117, 120

[41] Dongkyun Kim, Chai-Keong Toh, J.-C. C. Cano, P. Manzoni, Dongkyun Kim, Chai-Keong Toh, J.-C. C. Cano, and P. Manzoni. A bounding algorithm for the broadcast storm problem in mobile ad hoc networks. In *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, **2**, pages 1131–1136. IEEE, mar 2003. 64

[42] A. Dunkels. Protothreads. 44

[43] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *29th Annual IEEE International Conference on Local Computer Networks*, pages 455–462. IEEE (Comput. Soc.), nov 2004. 44

[44] Adam Dunkels. Rime A Lightweight Layered Communication Stack for Sensor Networks. In *Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session*, Delft, The Netherlands, jan 2007. 44

[45] Simon Duquennoy, Beshr Al Nahas, Olaf Landsiedel, and Thomas Watteyne. Orchestra. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems - SenSys '15*, SenSys '15, pages 337–350, New York, New York, USA, 2015. ACM Press. 117, 119

[46] M.P. Durisic, Z. Tafa, G. Dimic, and V. Milutinovic. A survey of military applications of wireless sensor networks. *Mediterranean Conference on Embedded Computing (MECO) 2012*, pages 196–199, 2012. 1

[47] Atila Ertas and Jesse C. Jones. *The Engineering Design Process, 2nd Edition*. 1996. 8

[48] ETH Zürich. Low-Power Wireless Bus (LWB), sep 2017. 174

[49] Z M Fadlullah, M M Fouda, N Kato, A Takeuchi, N Iwasaki, and Y Nozaki. Toward intelligent machine-to-machine communications in smart grid. *IEEE Communications Magazine*, **49**(4):60–65, apr 2011. 3

[50] R.F. Fernandes, C. C. Fonseca, D. Brandao, P. Ferrari, A. Flammini, and A. Vezzoli. Flexible Wireless Sensor Network for smart lighting applications. In *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pages 434–439. IEEE, may 2014. 16, 27, 28

[51] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with Glossy. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 73–84, apr 2011. 67, 69

[52] Federico Ferrari, Marco Zimmerling, Luca Mottola, and Lothar Thiele. Low-power wireless bus. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems - SenSys '12*, SenSys '12, page 1, New York, New York, USA, 2012. ACM Press. 132

[53] Internet Engineering Task Force. IPv6 over the TSCH mode of IEEE 802.15.4e (6tisch) -, 2015. 117

[54] T. Yan T. He J.A. Stankovic G. Zhou, C.D. Huang and T.F. Adbelzaher. MMSN: Multi-frequency media access control for wireless sensor networks. 2006. 121

[55] A. T. Gambutan and M. Popa. Smart homes. A solution for a smart lighting system. In *2009 5th International Symposium on Applied Computational Intelligence and Informatics*, pages 205–210. IEEE, may 2009. 22

[56] Geng Wu, S Talwar, K Johnsson, N Himayat, and K D Johnson. M2M: From mobile to embedded internet. *IEEE Communications Magazine*, **49**(4):36–43, apr 2011. 3

[57] Carles Gomez and Josep Paradells. Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, **48**(6):92–101, jun 2010. 31

[58] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, **29**(7):1645–1660, sep 2013. 2

[59] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, **1**(4):660–670, oct 2002. 97, 98

[60] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking - MobiCom '99*, MobiCom '99, pages 174–185, New York, New York, USA, 1999. ACM Press. 130

[61] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, **vol.1**, page 10. IEEE Comput. Soc, jan 2000. xxiii, 96

[62] Cong Hui Huang, Yih Feng Su, Yu Tang Su, Wei Hao Su, Jun Jie Chen, and Wei Jie Zhou. Design and Application of ZigBee Wireless Techniques for an Intelligent House Lighting Control System. In *2014 International Symposium on Computer, Consumer and Control*, pages 19–22. IEEE, jun 2014. 16, 25

[63] Hesong Huang and Jie Wu. A probabilistic clustering algorithm in wireless sensor networks. In *Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd*, **3**, pages 1796–1798, sep 2005. 97, 99, 110

[64] Zucheng Huang and Feng Yuan. Implementation of 6LoWPAN and Its Application in Smart Lighting. *Journal of Computer and Communications*, **03**(03):80, mar 2015. 16, 29

[65] Jonathan W Hui and David 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 '04*, SenSys '04, page 81, New York, New York, USA, 2004. ACM Press. 130

[66] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, **9**(3):90–95, 2007. 45

[67] AbdulRahman H. Hussein, Amer O. Abu Salem, and Sufian Yousef. A flexible weighted clustering algorithm based on battery power for Mobile Ad hoc Networks. In *2008 IEEE International Symposium on Industrial Electronics*, pages 2102–2107. IEEE, jun 2008. 108

[68] IEEE Computer Society. *IEEE Standard Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, **2011**. 2011. 85, 90

[69] IEEE Computer Society. *IEEE Standard Part 15.4e: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer*, **2012**. 2012. 90

[70] IEEE Computer Society. IEEE Standard Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Standards for Local and Metropolitan Area Networks*, **2015**:1–709, 2016. xx, 90, 92, 93, 192

[71] Matthias Ihmig and Peter Steenkiste. Distributed Dynamic Channel Selection in Chaotic Wireless Networks. In *13th European Wireless Conference*, pages 1–7, 2007. 117

[72] Earth Policy Institute. Data Highlights - 15: World on the Edge by the Numbers Shining a Light on Energy Efficiency — EPI, 2011. 17

[73] Texas Instruments. CC2538 A Powerful System-On-Chip for 2.4-GHz IEEE 802.15.4-2006 and ZigBee Applications — TI.com. 109

[74] ITU. G.9959 : Short range narrow-band digital radiocommunication transceivers - PHY, MAC, SAR and LLC layer specifications, 2015. 32

[75] Jianfeng Wang. Zigbee light link and its applicationss. *IEEE Wireless Communications*, **20**(4):6–7, aug 2013. 32

[76] Kan Zheng, Fanglong Hu, Wenbo Wang, Wei Xiang, and M. Dohler. Radio resource allocation in LTE-advanced cellular networks with M2M communications. *IEEE Communications Magazine*, **50**(7):184–192, jul 2012. 3

[77] Taewook Kang, Jangkyu Yun, Hoseung Lee, Icksoo Lee, Hyunsook Kim, Byunghwa Lee, Byeongjik Lee, and Kijun Han. A Clustering Method for Energy Efficient Routing in Wireless Sensor Networks. In *Proceeding EHAC'07 Proceedings of the 6th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications*, EHAC'07, pages 133–138, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS). 97, 98

[78] I. Katzela and M. Naghshineh. Channel assignment schemes for cellular mobile telecommunications: a comprehensive survey. 1996. 121

[79] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot. Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pages 1451–1459. IEEE, 2007. 117

[80] R. Kawano and T. Miyazaki. Distributed coloring algorithm for wireless sensor networks and its applications. 2007. 121

[81] Giannis Kazdaridis, Stratos Keranidis, Adamantios Fiamegkos, Thanasis Korakis, Iordanis Koutsopoulos, and Leandros Tassiulas. Novel metrics and experimentation insights for dynamic frequency

selection in wireless LANs. In *Proceedings of the 6th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization - WiNTECH '11*, page 51, New York, New York, USA, 2011. ACM Press. 117

[82] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. Klein Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin. Simulating Wireless and Mobile Networks in OMNeT++ the MiXiM Vision. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, Simutools '08, pages 71:1–71:8, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 38, 39

[83] P. Chanda D.P. Agrawal K.R. Chowdhury, N. Nandiraju and QA. Zeng. Channel allocation and medium access control for wireless sensor networks. 2009. 122

[84] Marta Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of Probabilistic Real-time Systems. In *CAV'11 Proceedings of the 23rd international conference on Computer aided verification*, CAV'11, pages 585–591, Berlin, Heidelberg, 2011. Springer-Verlag. 146

[85] D.J. Leith and P. Clifford. A Self-Managed Distributed Channel Selection Algorithm for WLANs. In *2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1–9. IEEE, 2006. 117

[86] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. TinyOS: An Operating System for Sensor Networks. In Werner Weber, Jan M. Rabaey, and Emile Aarts, editors, *Ambient Intelligence*, pages 115–148. Springer-Verlag, Berlin/Heidelberg, 2005. 17

[87] Philip Levis, Neil Patel, David Culler, and Scott Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. *Proceedings of the First Symposium on Networked Systems Design and Implementation*, pages 15–28, 2004. 129

[88] Mingfu Li and Hung-Ju Lin. Design and Implementation of Smart Home Control Systems Based on Wireless Sensor Networks and Power Line Communications. *IEEE Transactions on Industrial Electronics*, **62**(7):4430–4442, jul 2015. 16, 23

[89] Charlston Lights. LED Lighting Requirement Calculator - Charlston Lights. 95

[90] Yu-Ju Lin, H. A. Latchman, Minkyu Lee, and S. Katar. A power line communication network infrastructure for the smart home. *IEEE Wireless Communications*, **9**(6):104–111, dec 2002. 16

[91] Jean Paul Linnartz. Wireless Communication, The Interactive Multimedia CD-ROM, 2001. 40

[92] Jean Paul Linnartz. D502.001_Ph1 User requirements (HW/SW). Technical report, 2013. 8, 39

[93] Rongduo Liu, Wei Wu, Hao Zhu, and Dacheng Yang. M2M-Oriented QoS Categorization in Cellular Network. In *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–5. IEEE, sep 2011. 3

[94] Matthew Loy, Raju Karingattil, and Louis Williams. ISM-Band and Short Range Device Regulatory Compliance Overview. 2005. 20

[95] Kenneth C Mansfield Jr and James L Antonakos. *Computer Networking for LANs to WANs: Hardware, Software and Security.* Cengage Learning, 2009. 59

[96] S. Mccanne, S. Floyd, and K. Fall. *ns2 (network simulator 2).* 37

[97] MDT Technologies. MDT Dimming Actuator, 2014. 78

[98] Geoff Mulligan. The 6LoWPAN architecture. In *Proceedings of the 4th workshop on Embedded networked sensors - EmNets '07*, page 78, New York, New York, USA, jun 2007. ACM Press. 33

[99] Yoshitaka Nakamura, Yutaka Takahashi, and Osamu Takahashi. A selection method of optimal channel in wireless network by the dynamic control of the Duty Cycle threshold. 2014. 117

[100] L. Narayanan. Channel assignment and graph multicoloring. 2002. 121

[101] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking - MobiCom '99*, **8**, pages 151–162, New York, New York, USA, 1999. ACM Press. 62, 63, 64, 70

[102] Dusit Niyato, Lu Xiao, and Ping Wang. Machine-to-machine communications for home energy management system in smart grid. *IEEE Communications Magazine*, **49**(4):53–59, apr 2011. 3

[103] Maria Rita Palattella, Nicola Accettura, Mischa Dohler, Luigi Alfredo Grieco, and Gennaro Boggia. Traffic Aware Scheduling Algorithm for reliable low-power multi-hop IEEE 802.15.4e networks. In *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, pages 327–332. IEEE, sep 2012. xviii, xix, 91, 92, 117, 118

[104] Maria Rita Palattella, Pascal Thubert, Xavier Vilajosana, Thomas Watteyne, Qin Wang, and Thomas Engel. 6TiSCH Wireless Industrial Networks: Determinism Meets IPv6. In *Internet of Things. Smart Sensors, Measurement and Instrumentation*, Smart Sensors, Measurement and Instrumentation, pages 111–141. Springer, Cham, 2014. 117

[105] Ashish Pandharipande, David Caicedo, and Xiangyu Wang. Sensor-Driven Wireless Lighting Control: System Solutions and Services for Intelligent Buildings. *IEEE Sensors Journal*, **14**(12):4207–4215, dec 2014. 16, 18, 19

[106] Wei Peng and Xi Cheng Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *2000 1st Annual Workshop on Mobile and Ad Hoc Networking and Computing, MobiHOC 2000*, MobiHoc '00, pages 129–130, Piscataway, NJ, USA, 2000. IEEE Press. 64

[107] Andrea Peruffo, Ashish Pandharipande, David Caicedo, and Luca Schenato. Lighting control with distributed wireless sensing and actuation for daylight and occupancy adaptation. *Energy and Buildings*, **97**:13–20, jun 2015. 16, 19

[108] Milena F. Pinto, Guilherme M. Soares, Thiago R. F. Mendonca, Pedro S. Almeida, and Henrique A. C. Braga. Smart modules for lighting system applications and power quality measurements. In *2014 11th IEEE/IAS International Conference on Industry Applications*, pages 1–8. IEEE, dec 2014. 16, 28

[109] Plasma. The Evolution of M2M Technology From M2M to IoT, 2015. 1

[110] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys '04*, page 95, New York, New York, USA, 2004. ACM Press. 24

[111] Theodore S Rappaport. *Wireless communications: principles and practice.* Prentice-Hall of India, New Delhi, 2008. 40, 42

[112] Christian Reinisch, Wolfgang Kastner, Georg Neugschwandtner, and Wolfgang Granzer. Wireless Technologies in Home and Building Automation. In *2007 5th IEEE International Conference on Industrial Informatics*, **1**, pages 93–98. IEEE, jul 2007. 16, 22

[113] George F. Riley and Thomas R. Henderson. The ns-3 Network Simulator Modeling and Tools for Network Simulation. pages 15–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. 37

[114] Riot-OS. RIOT - The friendly Operating System for the Internet of Things. 43

[115] Hyun-Jun Shin, Min-ho Jeon, and Chang-Heon On. Network Selection Algorithm at the Starting Point of Mobile Device in Indoor Wi-Fi Environment. *International Journal of Control and Automation*, **7**(9):97–108, sep 2014. 117

[116] Shubham. Machine to Machine Communication Examples and Applications, 2015. 2

[117] M. K. Simon and M. S. Alouini. A unified approach to the probability of error for noncoherent and differentially coherent modulations over generalized fading channels. *IEEE Transactions on Communications*, **46**(12):1625–1638, dec 1998. 16

[118] R. Sivakumar, P. Sinha, and V. Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, **17**(8):1454–1465, aug 1999. 99

[119] Robert Sombrutzki and Anatolij Zubow. A Practical Approach to Reliable Flooding in Mobile Ad hoc Networks, 2014. 62

[120] Steven Spielberg. Minority Report, 2002. 1

[121] Changsu Suh and Young-Bae Ko. Design and implementation of intelligent home control systems based on active sensor networks. *IEEE Transactions on Consumer Electronics*, **54**(3):1177–1184, aug 2008. 16, 24

[122] Tien-Wen Sung, Ting-Ting Wu, Chu-Sing Yang, and Yueh-Min Huang. RELIABLE DATA BROADCAST FOR ZIGBEE WIRELESS SENSOR NETWORKS. *International Journal on Smart Sensing & Intelligent Systems*, **3**(3):504–520, 2010. xxiii, 65

[123] P. Suresh, J. Vijay Daniel, V. Parthasarathy, and R. H. Aswathy. A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment. In *2014 International Conference on Science Engineering and Management Research (ICSEMR)*, pages 1–8. IEEE, nov 2014. 2

[124] Hoda Taheri, Peyman Neamatollahi, Ossama Mohamed Younis, Shahrzad Naghibzadeh, and Mohammad Hossein Yaghmaee. An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic. *Ad Hoc Networks*, **10**(7):1469–1481, sep 2012. 97, 100

[125] Mahmoud Talebi, Jan Friso Groote, and Conrad Dandelski. Modelling and verication of a cluster-tree formation protocol implementation for the IEEE 802.15.4 TSCH MAC operation mode. In *2nd Workshop on Models for Formal Analysis of Real Systems*, 2017. 145

[126] TechTarget. Machine-to-machine (M2M) communications and intelligence, 2018. 1

[127] Alessandra Toninelli, Susanna Pantsar-Syväniemi, Paolo Bellavista, and Eila Ovaska. Supporting context awareness in smart environments. In *Proceedings of the International Workshop on Middleware for Pervasive Mobile and Embedded Computing - M-PAC '09*, page 1, New York, New York, USA, 2009. ACM Press. 22

[128] P.W.M. Tsang and R.W.C. Wang. Development of a distributive lighting control system using Local Operating Network. *IEEE Transactions on Consumer Electronics*, **40**(4):879–889, nov 1994. 15

[129] Andras Varga. OMNeT++. pages 35–59. Springer Berlin Heidelberg, 2010. 37, 38

[130] Pawan Kumar Verma, Rajesh Verma, Arun Prakash, Ashish Agrawal, Kshirasagar Naik, Rajeev Tripathi, Maazen Alsabaan, Tarek Khalifa, Tamer Abdelkader, and Abdulhakim Abogharaf. Machine-to-Machine (M2M) communications: A survey. *Journal of Network and Computer Applications*, **66**:83–105, may 2016. xxi, 2, 3

[131] F. Viani, A. Polo, F. Robol, E. Giarola, and A. Ferro. Experimental validation of a wireless distributed system for smart public lighting management. In *2016 IEEE International Smart Cities Conference (ISC2)*, pages 1–6. IEEE, sep 2016. 16, 30

[132] Federico Viani, Alessandro Polo, Paola Garofalo, Nicola Anselmi, Marco Salucci, and Enrico Giarola. Evolutionary Optimization Applied to Wireless Smart Lighting in Energy-Efficient Museums. *IEEE Sensors Journal*, **17**(5):1213–1214, mar 2017. 16

[133] H. Wang. M2M Communications (Abstract only). In *IET International Conference on Communication Technology and Application (ICCTA 2011)*, pages 2–2. IET, 2011. 1

[134] Q Wang, X Vilajosana, and T Watteyne. 6TiSCH Operation Sublayer (6top) draft-wang-6tisch-6top-sublayer-01. Technical report, Internet-Draft. IETF, 2014. 120

[135] Yu-Xuan Wang and Forrest Sheng Bao. An Entropy-Based Weighted Clustering Algorithm and Its Optimization for Ad Hoc Networks. In *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007)*, pages 56–56. IEEE, oct 2007. 108

[136] Thomas Watteyne, Maria Rita Palattella, and Luigi Alfredo Grieco. Using IEEE802. 15.4 e TSCH in an LLN context: Overview, Problem Statement and Goals. *IETF draft, draft-watteyne-6tisch-tsch-00*, 2013. 117, 118

[137] Y-J. Wen and AM Agogino. Control of wireless-networked lighting in open-plan offices. *Lighting Research and Technology*, **43**(2):235–248, jun 2011. 16, 18

[138] WhatIs. What is latency?, 2017. 58

[139] Ruitao Xu, Gaotao Shi, Jun Luo, Zenghua Zhao, and Yantai Shu. MuZi: Multi-channel ZigBee Networks for Avoiding WiFi Interference. In *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, pages 323–329. IEEE, oct 2011. 117

[140] G. Zhou Y. Wu, M. Keally and W. Mao. Traffic-Aware Channel Assignment in Wireless Sensor Networks. 2009. 121

[141] Yao-Jung Wen and Alice M. Agogino. Wireless networked lighting systems for optimizing energy savings and user satisfaction. In *2008 IEEE Wireless Hive Networks Conference*, pages 1–7. IEEE, aug 2008. 16, 17

[142] O. Younis and S. Fahmy. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, **3**(4):366–379, oct 2004. xxiii, 97

[143] Z-Wave. About Z-Wave Technology, 2017. 32

[144] Masha Zager. MGM City Center Uses Tech To Wow Guests. *Broadband Properties*, **31**(7):90–91, 2010. 5

[145] Xiao-Long Zheng and Meng Wan. A Survey on Data Dissemination in Wireless Sensor Networks. *Journal of Computer Science and Technology*, **29**(3):470–486, may 2014. 128

[146] Zolertia. RE-Mote — Zolertia. 43