Model-based Analysis and Optimisation of Wireless Networks

Dissertation

zur Erlangung des Doktorgrades
"Dr. rer. nat."

der Fakultät für Wirtschaftswissenschaften
der Universität Duisburg-Essen

vorgelegt von

Sascha Jungen

aus

Bottrop, Deutschland

Betreuer:

Prof. Dr. Pedro José Marrón
Networked Embedded Systems Group (NES)
Institut für Informatik und Wirtschaftsinformatik (ICB)

Essen, Dezember 2021

Gutachter:

Prof. Dr. Pedro José Marrón
Prof. Dr. Gregor Schiele

Tag der mündlichen Prüfung:

12. August 2022

DOI: 10.17185/duepublico/78838
URN: urn:nbn:de:hbz:465-20230825-085920-7

Acknowledgements

I would like to express my gratitude to my advisor Prof. Dr. Pedro José Marrón for providing me the opportunity to delve into research and to complete this thesis in his group. Likewise, my thanks go Prof. Dr. Gregor Schiele for agreeing to review my research.

I would also like to thank my colleagues from the Networked Embedded Systems Group of the University of Duisburg-Essen for providing me valuable support, keeping me on the path to finish this thesis. Especially, I am deeply grateful to Matteo who accompanied my entire research, encouraging me to always keep progress and providing invaluable advice whenever required. Special thanks also go to Richard (who opened me the door to the world of research and this group), to Robert (who let me step inside), to Valentin and Alexander (for having many fruitful lunch discussions and countless cheerful jokes) and to Elke (who supported me with any administrative work).

Finally, I would like to thank my parents for always encouraging me to pursue my wishes and goals in life.

Abstract

In the last two decades, wireless communication has evolved to one of the most important techniques for exchanging information. Providing much more flexibility and possibilities compared to its wired alternative, applications utilising wireless communication require elaborated conceptions of planning the underlying network to achieve desired operational requirements and ensure reliability. However, realising these objectives requires to exploit network-specific knowledge, e.g., about the environment with its distinct characteristics or the used device hardware and applications, possibly entailing further efforts and costs to obtain, e.g., by performing time-consuming measurement campaigns. Thus, solutions are often required being practical and adapted to a given situation. Goal of this thesis is the development of universal approaches for analysing and improving wireless networks without constraining their applicability by underlying assumptions while usefully exploiting available network information. Breaking down the process of knowledge generation and exploitation into consecutive steps based on the amount of available information allows to provide granular approaches able to adapt to a wide range of scenarios considering their highly individual conditions. Distinguishing and addressing four leverage points, this thesis makes the following contributions to further understand and optimise wireless networks: (1) *identifying* the network topology and localise its devices, (2) *modelling* wireless communication in a fast and accurate way, (3) *understanding* the cause-effect relationship for changing network behaviour and exploit the knowledge for (4) *improving* the network with regard to specifiable performance metrics. The evaluation was conducted in several real-world scenarios, demonstrating the viability of automated general solutions for network analysis and optimisation at different levels of knowledge, managing to realise universal, network-independent applicability on one side and adaptability to specific systems and environments on the other side. The presented approaches are able to flexibly operate with variable amounts of information, ranging from the necessary minimum up to a large amount of further relevant details, enhancing the quality of results. Conclusively, each contribution of this thesis provides an approach to analyse a different aspect of the network and extract valuable knowledge to exploit. Altogether, they cover the complete range of knowledge acquisition to assist users in gaining insights about their network to ultimately improve the performance of their setup.

Table of Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

1.1 Background and Motivation

In the last two decades, wireless communication has evolved to one of the most important techniques for exchanging information. Development has yielded a wide range of standardised technologies, each tailored to specific requirements. Around these technologies several areas of application for wireless communication have evolved finding their way into everyday life, such as cellular networks, the Internet of Things (IoT), Cyber-Physical Systems (CPS) and Industry 4.0. In spite of their distinctions, all these applications have in common relying on electromagnetic waves carrying information from a sender to a receiver wirelessly over the air. For operating properly, systems heavily depend on successful, unimpeded communication between the involved devices, which in their entirety represent a network. By propagating through space, to achieve a successful transmission of information, the signal must be able to reach its destination with sufficient power, defying environmental conditions, e.g., intermediate obstacles, interfering noise or climatic changes. Providing much more flexibility and possibilities compared to its wired alternative, applications utilising wireless communication require elaborated conceptions of planning the underlying network as there are many relevant aspects to consider [p82, p157]. Therefore, an adequate understanding of electromagnetic wave propagation as well as of involved influencing factors of the environment, in which the devices operate, is crucial to ensure a functional application. However, taking all factors into account is a complex and challenging task. While the influence of one single factor might be comprehensible in many cases, the combination of all these factors result in highly individual conditions for each pair of wirelessly communicating devices, hence creating unique scenarios. As a consequence of distinct scenario characteristics, solutions are often required being practical and adapted to a given situation. When planning and realising a new deployment, this diversity hampers a direct transfer of solutions used in other scenarios highlighting the necessity and importance of obtaining and exploiting site-specific knowledge to create a tailored solution.

Identified by researchers as persisting challenge for the upcoming decade [p32], a thorough planning and deployment of a network is crucial to achieve desired operational requirements and ensure reliability even in case of occurring disturbances. Acquiring knowledge of network and environment requires expertise and is a costly and time-consuming effort usually obtained by sophisticated experimental campaigns [p102, p27]. For this reason, many networks lack such in-depth preplanning being prone to disregarded interferences during operation. Instead, devices are often placed based on an intuitive understanding of “good” positions for wireless reception, e.g., in central positions or at higher spots to achieve either better coverage or avoid obstacles. In many cases this approach works well, but in others it may not achieve the desired effects, e.g., when unknown environmental properties or interfering sources influence wireless communication. This missing information is implicitly reflected in the behaviour of the network and thus can be inferred by applying further analysis to actual data, e.g., the measured signal strength between devices. Exploiting such information then facilitates the development of optimisation plans steering the network performance into the desired direction. Offering the tools providing knowledge in realising a general and automated approach of analysis and optimisation applicable to new and also existing infrastructures would simplify the process of deployment and adjustment significantly by saving time as well as costs by eliminating the necessity of domain experts.

1.2 Challenges

Developing universal approaches for analysing and improving wireless networks without constraining their applicability by underlying assumptions while usefully including available network information though creates several challenges.

Heterogeneous Networks The variety of equipped hardware, e.g., boards, sensors and radios, provides diverse types of analysable data to gain information about the device, the network and the environment. However, especially when investigating the IoT consisting of large numbers of potentially different devices participating in a network, the heterogeneity aggravates assumptions on finding a basic set of commonly available information to agree on. This underlines the importance of establishing common standards to ensure proper operation. As a consequence, a large variance of exploitable information can be expected not only across multiple networks but also over different devices in the same network. Therefore realising approaches working in the most general way necessitates a reduction of expected information to a common and indispensable minimum, each network should be able to provide, such as the received signal strength between two devices or the communication frequency of sent signals. In spite of information scarcity, optimisations and knowledge gain still need to remain possible.

Costs of Information Exploiting available information is essential for knowledge acquisition and network optimisation. Self-evidently, a larger amount of information comprises more potential for achieving better results in the quality of optimisation approaches. However, obtaining additional information beyond the available one implies costs which have to be assessed in relation to their benefit. Information gain and costs in this context may occur in various ways. On the one hand it can arise through manual external effort, e.g., performing deployment site measurements to acquire positional or structural information of network and environment properties, requiring financial resources, time and expertise. On the other hand it can emerge through an automated internal effort, deriving implicitly contained knowledge from the available information, e.g., obstacle influences on temporal signal changes, involving input and computation times of the analysis tools as primary costs. Since an automatic approach for information extraction already comprises incorporated expertise, it is preferable due to a more general applicability and potentially lower costs. This is achieved by an omission of otherwise recurring required personal expertise in case of a manual realisation. As a consequential requirement, developed algorithms have to cope with variable amounts of information as well as taken design and deployment decisions and exploit them comprehensively. This flexibility then is crucial for achieving the best results with fewest possible effort and costs.

System Uniqueness Experimental campaigns reveal invaluable insights into the properties of a network and its environment, aiding a sophisticated network optimisation. However, as these campaigns are usually conducted to reveal traits of a specific network, the transferability of such insights is limited and universally applicable knowledge hard to derive as each network is situated in a unique setting. Such setting comprises a large number of factors to consider, such as the devices, their hardware and its properties, the environment containing structural features, positions of walls, other obstacles, static and mobile, their construction materials and other possibly interfering signal sources. The unique interplay of these factors is highly unlikely to occur at any other system in the same way again, always preventing an unrestricted transfer of insights to other networks.

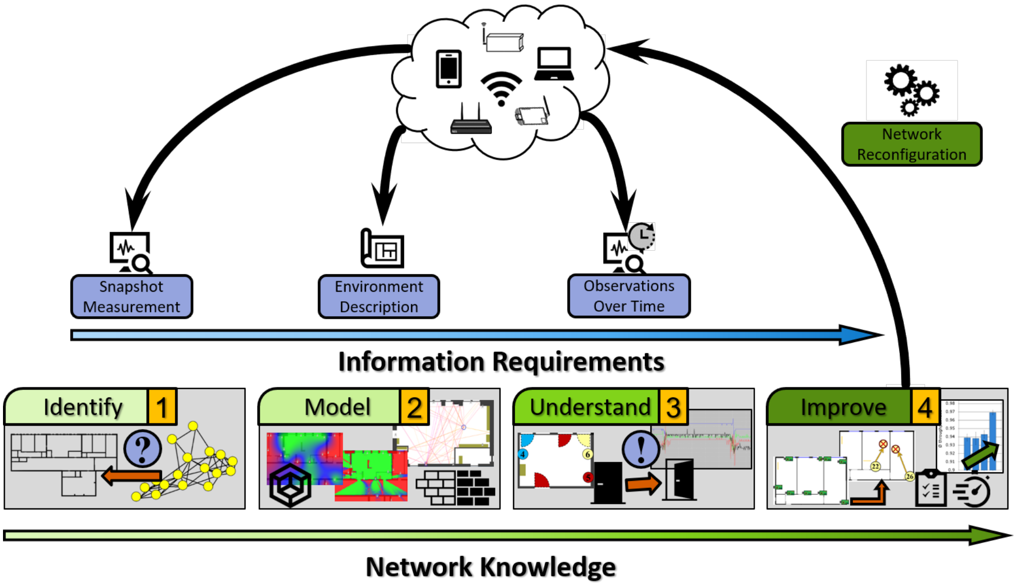


Figure 1.1: *Visualisation of contributions of this thesis. For each phase in the depicted process of gaining network knowledge, an approach is proposed to increase and utilise the available knowledge.*

Therefore general solutions for network optimisation cannot be as granular as dedicated ones and also require network-independent approaches at first, adaptable to system and environment of a specific network in a second step.

1.3 Goals and Contributions

Goal of this thesis is to extend the state of the art in understanding and optimising wireless networks in a universally applicable manner by providing flexible approaches tackling different stages of knowledge generation and exploitation. Figure 1.1 visualises the addressed consecutive levels of network knowledge in conjunction with an increasing amount of required information about the system, either provided by the network itself or externally. In detail, this thesis focuses on how information provided by operating wireless networks and their environments can be analysed to automatically derive valuable knowledge about their properties. Ultimately, this knowledge in turn becomes exploitable to enable an optimisation of the network behaviour towards specific performance goals. This thesis provides solutions for topology determination and device localisation based on varying amounts of information, as well as techniques to identify causes of changing network behaviour by incorporating a description of the environment a network is located in. Furthermore, it evaluates how these insights and their implications foster optimisation strategies based on relocation of devices to achieve better network performance. As a secondary objective, the introduced approaches are kept practical and simple to apply while requiring minimal effort to configure inputs as well as exhibiting acceptable computation times, facilitating their applicability on common systems and by inexperienced users. Addressing the aforementioned challenges, this thesis makes four contributions for extracting and exploiting network knowledge in a more general and automatised way. These contributions are briefly stated in the following, referencing to the corresponding stage numbers of Figure 1.1.

(1) Topology Identification and Device Localisation with Minimal Information Repeatedly emerging and changing infrastructures impose a challenge for ensuring a proper operation of a network while fulfilling its performance requirements, such as reliability, robustness or timeliness. Knowledge of device locations is a key element for understanding and improving a network. Not in all cases, positions have been tracked or known beforehand. In addition, the heterogeneity of networks, especially in the case of IoT, does not allow for many assumptions on a network and the provided information without significantly restricting the applicability of an employed approach. This thesis introduces FLoW [a4] a general approach able to work with different amounts of information, ranging from the absolute basic information, the strength of a received signal, each radio chip is able to measure, over an outline of the deployment scenario up to the knowledge of a few locations of devices using them as anchors. The proposed approach is able to generate a topology estimation of the network using minimal information and refine it in an incremental process by estimating the device positions when provided with aforementioned supplementary information.

(2) Fast and Accurate Runtime-Adaptable Wireless Communication Modelling Being able to accurately predict the signal quality at arbitrary locations is an effective way to prevent a costly trial-and-error method when exploring new positions for devices. In general, modelling of wireless communication can be realised in two ways, an empirical, network-based approach and a deterministic, environment-based one utilising physics of signal propagation independent of the actual network. The more fruitful option regarding generality, predictability and knowledge gain potential is the latter as it comprises a description of the environment enabling to take environmental features into account and interrelate them with network observations in a second step. An increasing amount of provided details commonly improves the accuracy of a model but also increases computation time. To keep a balance between acceptable speed and sufficient exactness this thesis proposes solutions for achieving both. Either by compensating missing details through fusing observations and physics in a hybrid approach or by utilising massive parallel architectures of modern graphic processing units [a1, a3, a2]. These models, joint together in a set called MOVE, serve as foundation for the subsequent contributions of this thesis.

(3) Cause-Effect Relationship for Changing Network Behaviour The importance of understanding the behaviour of wireless communication in practice has promoted many experimental studies, characterising signal propagation in specific environments. Despite the gathered knowledge, the perceived discrepancy between abstract models and actual network performance has supported the belief that system design and debugging can rely only on direct experience and trial-and-error. As a result, reasoning about wireless systems is nowadays a tedious, manual process. This contribution, called FOLLOW [a2] walks the playground between model and reality to make wireless networks understand their own behaviour in the environment where they operate. It provides a technique based on ray tracing incorporating a description of its deployment scenario without the necessity of providing highly detailed accuracy. The approach is able to estimate the impact of obstacles, typically found in indoor scenarios, e.g., doors, windows or elevators on the network by exploring possible wireless signal paths interacting with an obstacle. To circumvent an accuracy loss of model predictions caused by an inaccurate environment description, a coarser classification between strong and weak expected signal strengths is performed rather than a qualitative analysis. Based on the estimated impact, it becomes then possible to automatically assign observed changes during the operational phase to the corresponding obstacles identifying them as a reason.

(4) Network Performance Optimisation The design and deployment of networked embedded systems is a challenging task. In particular, the environment in which the system operates has a severe impact on the final performance. As fourth contribution, MOTION [a3] is presented, an approach exploiting the aforementioned hybrid modelling technique able to characterise indoor wireless communication in the target environment based on measurements and descriptions. Investigations of arbitrary displacements of devices become feasible aiding in exploring and evaluating the optimisation potential of networks. By controlling the relocation of a single node based on the information provided by the model, it is possible to improve performance metrics, such as latency, throughput and energy consumption of typical applications. Ultimately, this contribution constitutes the final step of enabling a directed situated optimisation of wireless systems by exploiting the available network knowledge obtained in previous steps.

1.4 Outline

The remainder of this thesis comprises the following. Chapter 2 covers the background of wireless communication. Chapter 3 describes the related work of the subsequent contributions presented in this thesis. Chapter 4 discusses FLoW, a technique for analysing network communication generating topology information [a4]. Chapter 5 introduces MOVE, a set of models for wireless signal propagation [a1, a3, a2]. Chapter 6 describes Follow, a method to identify obstacles as reasons for temporal changes in signal reception [a2]. Chapter 7 explains MOTION, an approach for optimising network application performance by relocating devices [a3]. Chapter 8 contains the evaluation of the previous contributions. Chapter 9 concludes the work and presents an outlook of future research directions.

2 Background of Wireless Communication

This chapter provides the substantial background of wireless communication, explains associated terms and elaborates on the physical properties relevant for modelling signal propagation. Furthermore an introduction to ray tracing is given, the models covered in this thesis are based on. Finally, challenges of modelling wireless communication are discussed, justifying the design choices made, described in the succeeding chapter.

2.1 Wireless Communication

Wireless communication has evolved to one of the most important techniques for exchanging information and fundamentally influenced the technological development during the last decades. Besides the obvious advantage of not requiring any physical connection between communicating devices and thus facilitating scenarios including mobility, wireless communication has several further advantages over the wired alternative. Due to the ability of radio waves passing solid materials, connecting to a network via stationary access point is rather bound to an area, instead of being limited to single positions providing sockets, massively increasing availability and flexibility. In addition, to establish a network connection, further hardware except for the radio chip of the communicating device becomes dispensable, reducing the costs as well as opening the possibility to offer services at a larger scale [p82].

However, ubiquitous availability and increased flexibility comes at a cost. The first noteworthy field not covered in this thesis, but mentioned for the sake of completeness is security. Although mostly designated to a specified receiver, the radio signal containing a message spreads on the entire area, allowing third persons to more simply overhear communication and possibly extract confidential information. Taking the risks of a freely receivable communication into account, security is an essential topic in wireless communication [p111, p80]. For the common standardised transmission technologies many publicly available encryption standards exist, protecting exchanged messages from being readable [p140].

Even though security is a steadily prevailing topic, this thesis focuses on a different yet similarly relevant area of problems. It aims at extending the state of the art of wireless communication by identifying and improving the network properties through focusing on the physical aspects of signal propagation and their implications to identify the reasons behind observed network behaviour by an automated analytical approach. In this context, some commonly used terms are introduced and explained in the following.

RSSI The strength at which a signal is received determines whether a radio chip is able to differentiate between the sent signal and the background noise to successfully extract the sent information (for details see Section 2.2.2). This strength is measured by most radio chips upon successful reception and provided as an integer value, the so-called *received signal strength indicator* (RSSI). Although containing minor inaccuracies depending on the used hardware, the RSSI still provides a viable estimation of the signal quality. Therefore it facilitates to evaluate signal propagation and coverage with any wireless device, delivering valuable information about environment and network. Especially the fact being one of the few commonalities among the otherwise heterogeneous variety of wireless devices predestines the RSSI to be utilised for analysis as one key metric. Consequently, all contributions of this thesis directly or indirectly use the RSSI for deriving information.

Link By being able to receive a signal at sufficient strength, the two involved devices establish a unidirectional *link* from the sender to the receiver. Accordingly, if a reception is reversely possible too, the devices share a bidirectional link. However, in the context of this thesis, the term link is used to describe the unidirectional way of signal transmission, resulting in counting two links in case of successful bidirectional transmissions of signals.

WSN A wireless sensor network (WSN) describes a set of autonomous wirelessly communicating devices equipped with sensors monitoring environmental characteristics, such as temperature, humidity or light levels. Due to the mostly mobile usage in large numbers of up to thousands of devices [p52] and the small device size, hardware capabilities are restricted to fulfil basic computations and sensing tasks only. Being dependent on battery sources, devices are mostly designed for operating over long periods of time prolonging the necessity of a costly battery exchange if scheduled at all. The devices organise themselves in networks to deliver their gathered data to a central collecting device, the sink, which usually has larger computational capabilities for processing and analysing this data. The technical standard of the Institute of Electrical and Electronics Engineers (IEEE), 802.15.4 [p70] provides a basis for low-data-rate wireless communication, also taking devices with limited energy resources into account by confining to implementations of very low complexity at hardware level. It also regulates communication on three possible frequency bands, operating at 868 MHz, 915 MHz and 2.4 GHz. Building on this standard, several protocol specifications have emerged, such as ZigBee [p73] and Thread [p150], providing further implementations of higher communication layers. Appropriate due to their versatility and flexibility, WSNs have been used in a large range of scenarios, e.g., environmental sensing [p152, p163], structural monitoring [p28, p29], health care [p85, p31], education [p172], disaster detection [p21] and industrial maintenance [p151, p167].

WiFi Dedicated to wireless communication in a less resource constrained context and usually with higher transmission powers, the term WiFi describes a set of protocols that have evolved based on the IEEE 802.11 standard [p69]. For WiFi, frequency bands of 5 GHz and 2.4 GHz are used, partially overlapping with the frequencies used in IEEE 802.15.4. A large portion of RSSI measurements used for evaluating the contributions of this thesis were collected from WSN testbeds and WiFi networks.

2.2 Properties and Physics

In this section, the relevant basics of radio signal propagation are summarised, based on the contents of [p164], [p62], [p121] and [p123] if not otherwise stated. These are important to understand the principles, which the presented physical models use to perform their predictions.

2.2.1 Electromagnetic Waves

Transmitting information from a source to a destination requires a medium, to which this information is attached. In wireless communication, electromagnetic waves are utilised to convey information (*signals*), over the air.

Electromagnetic waves consist of two synchronously sinusoidal oscillating perpendicular fields, an electric and a magnetic one, containing energy and transporting it through space, as shown in Figure 2.1. These propagate at the speed of light $c = 299.792.458\ m\ s^{-1}$ in a vacuum and at reduced speed in matter based on the permeability. An electromagnetic wave is characterised by an amplitude and a frequency of oscillation f, measured in hertz (Hz (s^{-1})), or alternatively the wavelength λ in metres, i.e., the distance at which an oscillation repeats. Frequency and wavelength are convertible through the equation:

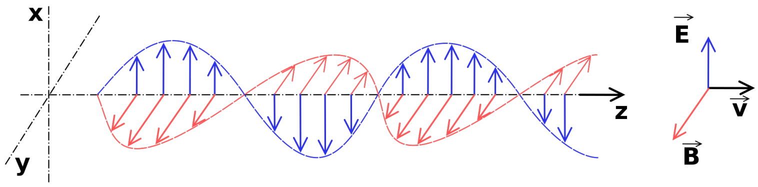


Figure 2.1: *Oscillating electromagnetic wave consisting of an electric ($\vec{E}$) and a magnetic ($\vec{B}$) field carrying energy through space in the direction of $\vec{v}$ (image source: [m13]).*

$$f = \frac{c}{\lambda} \; . \tag{2.1}$$

For electromagnetic waves the amplitude denotes the maximum field strength of electric and magnetic fields whose energy is proportional to the squared amplitude. The total energy of a wave in turn is the sum of smallest units (quanta) with discrete amounts of energy, called photons. According to the quantum theory, photons exhibit characteristics of both particles and (electromagnetic) waves. They have a fixed amount of carried energy E, measured in joule (J ($kg\ m^2\ s^{-2}$)), which is determined by their frequency according to the Planck-Einstein equation

$$E = h\ f \tag{2.2}$$

with $h = 6.62607015 \times 10^{-34}\ J\ s$ as the Planck constant. This constant defines the minimum amount of energy for an electromagnetic action to take place. In order to increase the amount of transmitted energy of a wave, either the frequency or the quantity of emitted photons has to be increased. Electromagnetic waves do not influence each other, allowing an overlapping of multiple waves while propagating in space. Due to this applicable superposition principle, electric and magnetic fields of each particle contribute to existing respective fields of other sources at the same location. When propagating in the same direction, multiple particles add up their energy forming one large observed electromagnetic wave accumulating the individual energies. The rate of energy transfer (*power*) is measured in watt (W ($J\ s^{-1}$)) [p149].

Electromagnetic waves are a result of oscillation or acceleration of electrically charged particles emitting energy outwards. Once emitted, waves carry the radiated energy away from their source and distribute it through space in an expanding wavefront. A wavefront describes the surface on which all points have the same time of travel regarding their source. The actual shape of a wavefront depends on the type of emitting source. Assuming a single point source, the wavefront propagates away from that point equally in each direction as a sphere. Upon permeating matter during propagation, an energy transfer back to other particles is possible, depending on the wave frequency and the type of matter.

The general principle of wireless communication builds on the exploitation of aforementioned electromagnetic wave properties. Although emerging under various conditions, the relevant method for emitting electromagnetic waves in the context of wireless communication is to accelerate charged particles by regulating the current flow through an antenna. This technique enables a controlled and precise way of electromagnetic wave generation to transmit signals with certain frequency and power (strength). Common frequencies used for wireless communication range from a few Hz up to many GHz [p35]. Common transmission powers are usually given in the range of milliwatts (mW), due to regulations of wireless communication, prohibiting the use of stronger signals above a few watts for civil usage. E.g., in wireless local area networks using the 2.4 GHz band, the maximum allowed transmission power in Europe is limited to 100 mW [m7].

2.2.2 Signal Path Loss

During the transition from transmitter to a destination, signals experience losses resulting in less received power, caused by the expanding coverage of the wavefront in space and the interaction with obstructions on its path, called *attenuation*. The difference between transmitted and received power is called *path loss* and is a key value in predicting wireless signal propagation. It is measured in decibel (dB), a relative unit to represent the ratio L of a value of power (or field) quantity P to another reference value P_0 using a logarithmic scale [p148, p68]:

$$L_P = 10 \ \log \frac{P}{P_0} \tag{2.3}$$

According to Equation 2.3, a difference of, e.g., $L_P = 1$ dB corresponds to a power ratio of $10^{\frac{1}{10}}$ for P compared to the reference value P_0. If used as absolute value, the unit of reference is commonly appended as suffix. In case of watts, the unit dBW is used, in case of milliwatts the unit dBm. A transmission power of 0 dBm is then equivalent to 1 mW.

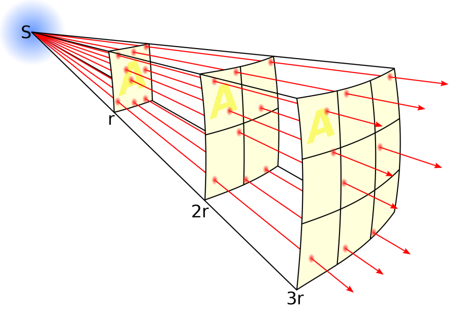


Figure 2.2: *Section of a spherical wavefront propagation with quadratic areal increase in relation to the distance r (image source: [m6]).*

Free Space Propagation

Having a clear line of sight between transmitter and receiver without any obstructions, Friis free space Equation 2.4 [p50] can be applied, describing the received power P_r as a function of distance:

$$P_r(d) = \frac{P_t\ G_t\ G_r\ \lambda^2}{(4\pi)^2\ d^2\ L} \tag{2.4}$$

with P_t as the transmitted power, G_t and G_r denoting the transmitter and receiver antenna gains (dimensionless) and L as losses related to hardware (dimensionless, $L \geq 1$). As an electromagnetic wavefront propagating away from a source expands in space, the power of a received signal decreases in dependency of the square of the distance between transmitter and receiver according to the inverse square law. With an increasing distance, the initial transmission power P_t is distributed on a larger area of the wavefront, reducing the power density and thus the received power. Figure 2.2 illustrates the quadratic areal growth and the decreasing power density of a wavefront emitted by a source S for an increasing distance r.

Expressing the free space power density reduction as path loss (PL) in decibels while assuming no losses in hardware ($L = 1$) gives Equation 2.5:

$$PL(dB) = 10\ \log \frac{P_t}{P_r} = 10\ \log \left[\frac{(4\pi)^2\ d^2}{G_t\ G_r\ \lambda^2} \right] \tag{2.5}$$

Optionally, the effects of antenna gains may be excluded to simplify the above equation even further, assuming unity gain $G_t = G_r = 1$. Reshaping the above equation produces the following, simply to compute equation:

$$\begin{aligned} PL(dB) &= 10\ \log\left[\left(\frac{4\pi d}{\lambda}\right)^2\right] = 20\ \log\left(\frac{4\pi df}{c}\right) \\ &= 20\ \log(d) + 20\ \log(f) + 20\ \log\left(\frac{4\pi}{c}\right) \\ &= 20\ \log(d) + 20\ \log(f) - 147.55 \end{aligned} \tag{2.6}$$

Equation 2.6 is utilised by the ray tracing model (Chapter 5) to compute the share of the distance-based losses on the overall path loss for each possible signal path to a destination. The equations based on Friis free space model only produce valid predictions for P_r if the distance d is located in the so-called far-field of the transmitter antenna. This means d has to be greater than the far-field distance d_f which is defined as:

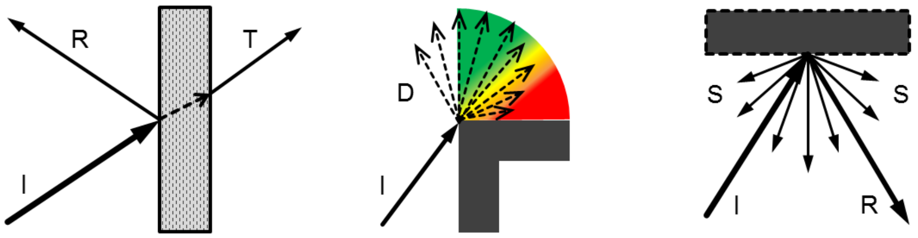
$$d_f = \frac{2D^2}{\lambda} \qquad\qquad \text{if } d_f \gg D \text{ and } d_f \gg \lambda \tag{2.7}$$

with D as the largest physical linear dimension of the antenna. At closer (near-field) distances, interactions between antenna and electromagnetic fields complicate an accurate prediction of the actual distance losses. To take the far-field region into account, for modelling path losses a reference distance d_0 is chosen which is larger than d_f, but smaller than any distance d, for which predictions have to be performed. The computation of the received power $P_r(d)$ at distance d can then be expressed in relation to the received power $P_r(d_0)$ at the reference distance, giving Equation 2.8:

$$P_r(d) = P_r(d_0)\left(\frac{d_0}{d}\right)^2 \qquad d \geq d_0 \geq d_f \tag{2.8}$$

According to [p123], typical reference distances are 100 m or 1 km in outdoor and 1 m in indoor environments when describing systems with low-gain antennas. The received power at reference distance $P(d_0)$ can either be computed using Equation 2.4 or determined empirically by performing measurements in the modelled network. In the context of this thesis, received power levels are expressed in units of decibels using milliwatts as reference unit (dBm) to depict their large range:

$$P_r(d)\ [dBm] = 10 \log\left[\frac{P_r(d_0)}{0.001\ W}\right] + 20 \log\left(\frac{d_0}{d}\right) \tag{2.9}$$



(a) (Specular) reflection (R) and transmission (T)

(b) Diffraction (D)

(c) Scattering (S) on a rough surface

Figure 2.3: *Basic signal propagation effects of an incident wave (I).*

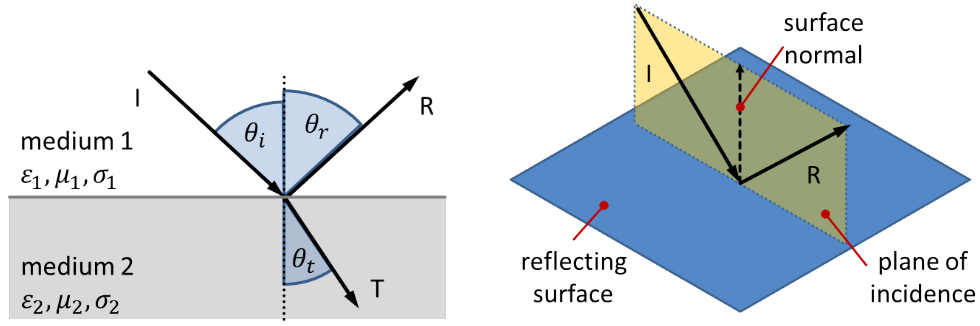
2.2.3 Signal Propagation and Attenuation

In addition to the plain propagation in space through one medium, electromagnetic waves may also encounter obstructions of different media with varying properties affecting their propagation behaviour. Three basic effects can be distinguished influencing both propagation direction and power density of a wavefront: *reflections*, *diffractions* and *scattering*. These effects are depicted in Figure 2.3 and described in the following.

Reflection/Transmission

Reflection describes the process of a change in direction of a propagating electromagnetic wavefront occurring on contact with a plain surface of a different medium that has a large size compared to the wavelength. Depending on the medium, the wavefront is split into a reflected and a transmitted part with different fractions of field intensity (power). While propagating in a new medium, a wavefront may also experience absorption losses due to the material properties of this medium. E.g., if the wavefront impinges on a perfect dielectric, i.e., a medium with high permittivity and zero conductivity (terms are explained in next paragraph), it is split but does not have any absorption losses. In case of a perfect conductor, i.e., a medium with infinite conductivity, all the energy is reflected without any transmission or absorption losses. Figure 2.3 (a) generally visualises this effect of wavefront separation. The electric field intensities of both, the resulting reflected and transmitted waves, depend on the Fresnel reflection coefficient Γ. Γ is a function depending on the involved material properties as well as the angle of incidence, wave frequency and polarisation, visualised in Figure 2.4 (a).

Relating to reflections, three properties of a material are relevant: conductivity σ, permeability μ and permittivity ϵ. Conductivity describes the property to let an electric current pass through. A high conductivity results in an easier flow of current with less resistance to overcome. Permeability



(a) Relevant parameters for reflection computations

(b) Plane of incidence

Figure 2.4: *Reflection of an incident wave.*

states the characteristic to resist to an applied magnetic field, determining the degree of magnetisation it experiences as consequence. Permittivity on the other hand describes the property of a material to react to an applied electric field by polarising its particles, i.e., the alignment of electric charges within the material. High permittivity relates to a larger tendency of polarisation transferring more energy into the emerging electric field. The complex-valued dielectric constant ϵ of a material stating its permittivity is often expressed with a relative value ϵ_r in relation to the permittivity of the vacuum which is given as $\epsilon_0 \approx 8.854 \times 10^{-12} F\ m^{-1}$:

$$\epsilon = \epsilon_r \epsilon_0 - i\epsilon^{'} = \epsilon_r \epsilon_0 - i\frac{\sigma}{2\pi f} \tag{2.10}$$

with i as the imaginary unit and f as the frequency of an incoming wave. The imaginary part $\epsilon^{'}$ describes the absorption effect, occurring upon propagation of an electromagnetic wave through a lossy dielectric medium ($\sigma > 0$). In case of a perfect (lossless) dielectric ($\sigma = 0$), ϵ can be expressed as $\epsilon = \epsilon_r \epsilon_0$ since the imaginary part $\epsilon^{'}$ becomes zero.

In case of well conducting media ($f < \sigma/(\epsilon_r \epsilon_0)$), permittivity ϵ_r and conductivity σ are insensitive to the frequency, but for lossy dielectric media this can be different. Thus, these material properties need to be provided for each of the investigated frequency ranges individually.

In the context of electromagnetic waves, polarisation describes the geometrical orientation of oscillation of the electric field component. For reflections, only two relevant types of polarisation need to be distinguished, the parallel and perpendicular electric field polarisation. The electric field component either oscillates parallel or perpendicular to the plane of incidence, which is spanned by the vector of the incoming wave and the normal vector of the reflection surface, as outlined in Figure 2.4 (b). Each polarised wave can be expressed as the sum of two linear polarisations, which are spatially orthogonal, allowing a reduction to these two cases. The Fresnel reflection coefficients [p49] of both polarisation directions are then given by:

$$\Gamma_{\perp} = \frac{E_r}{E_i} = \frac{\eta_2 \cos\theta_i - \eta_1 \cos\theta_t}{\eta_2 \cos\theta_i + \eta_1 \cos\theta_t} \qquad \text{for perpendicular polarisation} \tag{2.11}$$

$$\Gamma_{\parallel} = \frac{E_r}{E_i} = \frac{\eta_2 \cos\theta_t - \eta_1 \cos\theta_i}{\eta_2 \cos\theta_t + \eta_1 \cos\theta_i} \qquad \text{for parallel polarisation} \tag{2.12}$$

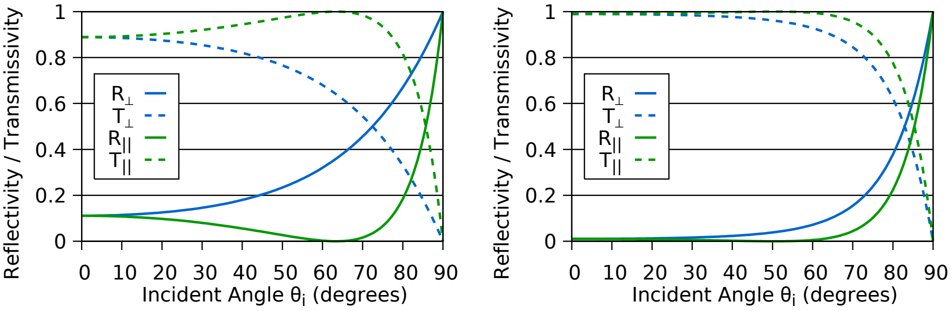
describing the ratio of electric field amplitudes of the reflected (E_r) and incident wave (E_i). η is the intrinsic impedance of a medium, which is a complex number for lossy media. For perfect dielectrics, the computation of η is reduced to $\sqrt{\mu/\epsilon}$.

Upon transition into another medium, incoming waves are *refracted*, changing their direction depending on the angle of incidence and the change of propagation speed between the two media. Snell's law relates the alteration in direction to the change of propagation speed:

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{v_1}{v_2} = \sqrt{\frac{\mu_2\epsilon_2}{\mu_1\epsilon_1}} \tag{2.13}$$

where the velocity is given by $v = 1/\sqrt{\mu\epsilon}$. This relation also implies the law of reflection: for the reflected part of a wave, incident angle θ_i and reflected angle θ_r have to be equal since the reflected wave stays in the same medium, i.e., $v_1 = v_2$. Equation 2.13 can be applied to substitute one of the angles by the other, allowing a computation of the reflectivity using the incident angle only.

To obtain the fraction of reflected power or *reflectivity* R which is more relevant for modelling wireless communication, Γ has to be squared since the power is proportional to the squared amplitude of an electric field, yielding $R = |\Gamma|^2$. The opposite, i.e., the share of transmitted power or *transmissivity* T is given as $T = 1 - R$, which is a direct consequence of energy conservation. This relation does not consider subsequent absorption losses due to wave propagation in a potentially lossy medium. Figure 2.5 visualises both coefficients R and T in dependency of the incident angle and wave polarisation for the transition from air to either glass or plywood [p159].



(a) Transition from air to glass **(b)** Transition from air to plywood

Figure 2.5: *Reflectivity and transmissivity of perpendicularly and parallelly polarised waves for transition from air to either glass (left) or plywood (right), based on relative permittivity values of [p159].*

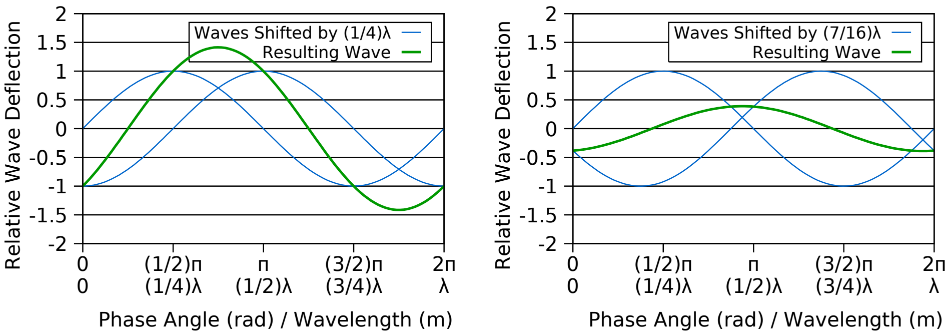
For waves with parallel polarisation exists an angle θ_B, at which a wave is entirely transmitted into a second dielectric medium without any reflection. This angle, for which $\Gamma_{||} = 0$, is called Brewster angle, satisfying:

$$\cos \theta_B = \sqrt{\frac{\epsilon_1}{\epsilon_1 + \epsilon_2}} \, . \tag{2.14}$$

For the two example transitions, depicted in Figure 2.5, the Brewster angle from air to glass is $\theta_B = 63.43°$, from air to plywood $\theta_B = 49.80°$.

Diffraction

Diffraction describes the effect of an electromagnetic wavefront bending around the surface of an obstacle, then being able to reach regions behind an obstacle which are not in line of sight to the source. While advancing deeper into this shadow region, a larger angle of deviation compared to line of sight successively reduces the received signal power. This is exemplarily shown in Figure 2.3 (b). Diffraction is attributable to Huygen's principle [p66] where each point on a wavefront is considered to be the source of another, secondary wave propagating into all directions. After passing an obstacle, e.g., an edge of a wall, these secondary waves allow the signal to propagate around it causing the diffraction effect.



(a) Constructive interference

(b) Destructive interference

Figure 2.6: *Wave interference effects in dependency of their phase difference.*

Especially in outdoor scenarios without line of sight connections and large distances to cover, diffraction effects play a significant role when predicting signal strengths since they apply to geographic features, e.g., hills, mountains and the curvature of the earth. Also indoors, a consideration of diffraction effects contributes to more accurate signal predictions as the combination of propagation effects produces small-scale fading (Section 2.2.4).

To understand the influences of obstructions on a diffracted wavefront and determine the potential losses of the received power, the possible paths of waves arriving at the receiver at the same time need to be investigated. Especially the differences in travelled path lengths have a significant influence on the received field strength of a wave as they determine the phase shift of multiple arriving waves. The *phase* describes the current position of a wave within its periodic sinusoidal oscillation. Depending on the phase difference, these waves can either *interfere constructively*, increasing the resulting wave amplitude or *interfere destructively*, decreasing the amplitude. The interference effect is depicted in Figure 2.6, showing two shifted waves of varying difference and same amplitude, adding up both phases in a resulting wave. If the difference of the travelled path length between two waves of the same frequency arriving at a specific position is an even multiple of their half wavelength, both will be in the same phase, creating the best possible constructive interference. The opposite occurs if shifted by an odd multiple of the half wavelength, entirely cancelling out both waves in the worst case.

To demonstrate and explain the effect of diffraction, the following explanations base on signal propagations in outdoor scenarios covering large distances. Considering the geometry shown in Figure 2.7, the excess path length Δ compared to a straight wave propagation can be approximately described as:

$$\Delta = \frac{h^2 \ (d_1 + d_2)}{2 \ d_1 \ d_2} \tag{2.15}$$

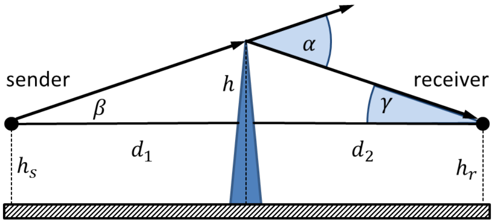


Figure 2.7: *Diffraction example at a knife-edge obstructing the line of sight between sender and receiver.*

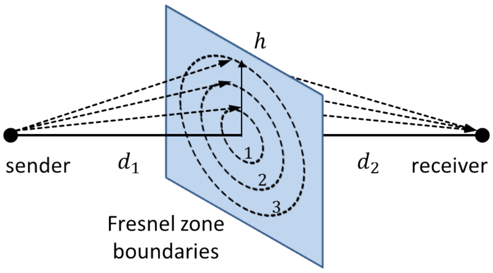


Figure 2.8: *Profile view of the ellipsoid Fresnel regions at a specific position between sender and receiver showing the Fresnel zone boundaries marked by the dashed circles.*

for $d_1, d_2 \gg h$ and $h \gg \lambda$. In case of different sender and receiver heights, large distances likewise allow an approximation by assuming $h_s \approx h_r$. The resulting phase difference in dependency of Δ is then given by:

$$\Phi = \frac{2\pi\Delta}{\lambda} = \frac{2\pi}{\lambda}\frac{h^2\ (d_1 + d_2)}{2\ d_1\ d_2} = \frac{\pi}{2}v^2 \tag{2.16}$$

with v as the Fresnel-Kirchhoff diffraction parameter defined as:

$$v = h\sqrt{\frac{2(d_1 + d_2)}{\lambda d_1 d_2}}\ . \tag{2.17}$$

Referring to the mechanism of interference, to illustrate the diffraction loss, the concept of Fresnel zones is elucidated. Fresnel zones describe disjoint ellipsoid regions with the positions of sender and receiver as their focal points, surrounding the direct line of sight path between both. They delimit the regions in which secondary waves arriving at a receiver have an excess path length of $n\lambda/2$ compared to the shortest line of sight distance. Figure 2.8 depicts the Fresnel zone boundaries on a profile view of the ellipsoid regions.

Waves arriving from consecutive Fresnel zones alternately contribute to the received amplitude with constructive and destructive interference. The radius r_n of the n-th Fresnel zone circle at an arbitrary position between sender and receiver can be approximated by:

$$r_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}} \tag{2.18}$$

for $d_1, d_2 \gg r_n$. The maximum Fresnel zone radii are obtained on the half distance ($d_1 = d_2$), decreasing when approaching one of the focal points. Equation 2.18 emphasises the relevance of the wave frequency as well as the position of an obstruction when determining the impact of diffraction on a received signal. These diffraction losses arise if parts of Fresnel zones are blocked by obstacles, preventing a fraction of waves and their energy from reaching the receiver. Most energy is transmitted by waves from within the first Fresnel zone. If this zone is kept mostly clear, obstructions of other Fresnel zones result in negligible losses only. Practical experience in designing line of sight microwave links has shown that an obstruction of up to 40% is tolerable without detecting remarkable losses [p33].

An exact computation of diffraction losses in practice is not possible due to the difficulty of mathematically describing complex and irregular environments. Thus simplified considerations applicable to certain scenarios have been derived. One of these simplifications is stated as the knife-edge model, in which the obstacle, a diffraction is occurring at, is treated as a sharp edge. Fresnel zones can be used to estimate diffraction losses. The ratio of an electric field strength E_d of a diffracted wave to the field strength of an unobstructed wave in open space E_o is given by the complex Fresnel integral $F(v)$:

$$\frac{E_d}{E_o} = F(v) = \frac{(1+i)}{2} \int_v^\infty \exp((-i\pi t^2)/2) dt \ . \tag{2.19}$$

The diffraction gain G_d resulting from a knife-edge is then given by:

$$G_d(dB) = 20 \log |F(v)| \tag{2.20}$$

which is mostly approximated by numerical solutions, as done by Lee [p93] for different ranges of v:

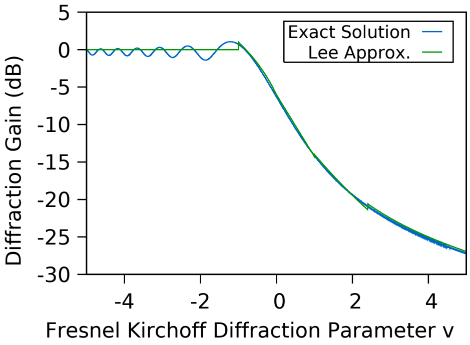


Figure 2.9: *Diffraction gain in dependency of the Fresnel-Kirchhoff parameter v.*

$$G_d(dB) = \begin{cases} 0 & \text{for } v \leq -1 \\ 20\log(0.5 - 0.62v) & \text{for } -1 \leq v \leq 0 \\ 20\log(0.5\exp(-0.95v)) & \text{for } 0 \leq v \leq 1 \\ 20\log(0.4 - \sqrt{0.1184 - (0.38 - 0.1v)^2}) & \text{for } 1 \leq v \leq 2.4 \\ 20\log\left(\frac{0.225}{v}\right) & \text{for } v > 2.4 \end{cases} \tag{2.21}$$

Approximation and exact solution in dependency of v are shown in Figure 2.9. This computation describes the effect of one single diffraction on the propagating wave. If multiple consecutive diffractions occur, instead of computing each diffraction separately, their gains can also be computed in a combined way [p44, p38]. However, while this is often applicable to outdoor scenarios, e.g., signals being diffracted at several hills without further intermediate effects, in indoor scenarios, different propagation effects usually occur in rather arbitrary order.

Scattering

The third propagation effect, *scattering*, occurs in association with obstructions which are small compared to the wavelength. These are, e.g., foliage and street signs outdoors, office utensils and small furniture indoors, but also in general irregularities of large, but rough surfaces. In most cases, the surface a radio wave impinges on is not perfectly smooth, causing the signal to be scattered in directions different from the specular reflection. This increases the dissemination of a stronger signal in the vicinity of the impact site beyond the direction of the angle of reflection, but also increases the loss in the main direction of propagation. Figure 2.3 (c) visualises the effect of scattering on a rough surface.

To determine whether a surface is rough, the Rayleigh criterion is used, given by:

$$h_c = \frac{\lambda}{8 \sin \theta_i} \tag{2.22}$$

defining the critical height h_c above which the maximum height difference h of an uneven surface for the given incident angle θ_i (see Figure 2.4 (a)) is expected to produce scattering effects. In this instance, to account for the occurring losses, a loss factor ρ_s is multiplied by the reflection coefficient Γ, which describes the electric field of the specular reflection on plain surfaces yielding Equation 2.23:

$$\Gamma_r = \rho_s \Gamma \tag{2.23}$$

with Γ depending on the wave polarisation and ρ_s initially introduced by [p11] as:

$$\rho_s = \exp \left[-8 \left(\frac{\pi \sigma_h \sin \theta_i}{\lambda} \right)^2 \right] . \tag{2.24}$$

The standard deviation of the surface height h is denoted as σ_h. This loss factor was later amended in [p24] by additionally considering the Bessel function I_0:

$$\rho_s = \exp \left[-8 \left(\frac{\pi \sigma_h \sin \theta_i}{\lambda} \right)^2 \right] I_0 \left[-8 \left(\frac{\pi \sigma_h \sin \theta_i}{\lambda} \right)^2 \right] . \tag{2.25}$$

Originating from the solutions of Bessel's second-order linear differential equation, which describe decaying oscillations of waves during their propagation, the multiplication of the Bessel function in Equation 2.25 allows to better fit the computed scattering losses with respect to actual experimental observations.

2.2.4 Small Scale Fading

Previous descriptions mainly focused on large scale path losses of a signal arising from the covered distance as well as the interaction of the carrying waves with obstructions. This section focuses on *small scale fading* or *fading*, describing the occurrence of remarkable fluctuations of the received field strength within a short time interval or a small distance. To observe

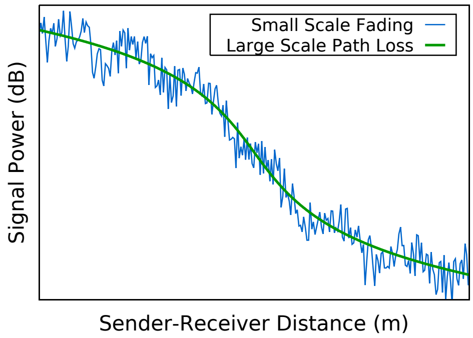


Figure 2.10: *Exemplary curve of large scale path loss in relation to small scale fading.*

fading effects, at least one mobile element is required, either in the surrounding environment altering wave propagation paths over time or moving transmitters and / or receivers in case of static environments. Fading is a result of the interference of multiple waves of the same transmitted signal, arriving at a receiver at a particular time via different paths. Due to varying travel times, these multipath waves can differ in phase and amplitude, hence causing con- and destructive interferences depending on their phase difference. Furthermore, changes in frequency also arise from movements in direction of the wave propagation, called Doppler shift. Lastly, large differences in travelling distances cause time dispersions, receiving waves of older and newer signal states overlapping at the same time, possibly preventing a correct decoding of information.

Especially in absence of a straight line of sight path between sender and receiver, whose related wave usually provides the strongest signal and is rather resistant to interference from comparably weaker indirect waves, fading can have significant impact on the received power. In urban outdoor scenarios with densely built-up areas, e.g., in case of radio broadcasts from stations many kilometres away where a signal travels large distances, many alternative paths facilitate fading. Likewise occurring at smaller spatial dimensions, fading is also relevant for indoor propagation, providing a high density of various obstructions creating many possible signal paths.

3 State of the Art

This chapter elaborates on state of the art work related to the contributions covered in this thesis on the different stages of available network knowledge. Beginning with only a sparse set of available information and increasing the amount gradually, this section introduces existing approaches for extracting knowledge out of a network, describes the addressed problems, elaborates on the challenges and requirements and discusses their advantages and limitations.

3.1 Network Analysis

Gathering basic knowledge about a network and its properties is the first important step in order to be able to conduct further analyses and gain new insights based on this knowledge. Once a wireless network has been deployed and is in operation, the arising data can contribute to a better understanding of the network behaviour. However, this data has to be gathered from the network first before it can be exploited.

3.1.1 Data Acquisition and Analysis

Analysing information on the network connectivity, e.g. the (non-)existence of links as well as the RSSI of those existent allow to draw first conclusions about the network properties and reveal possible deficiencies. To obtain and analyse such connectivity information, dedicated tools exist, aiding users in this process by automating certain steps.

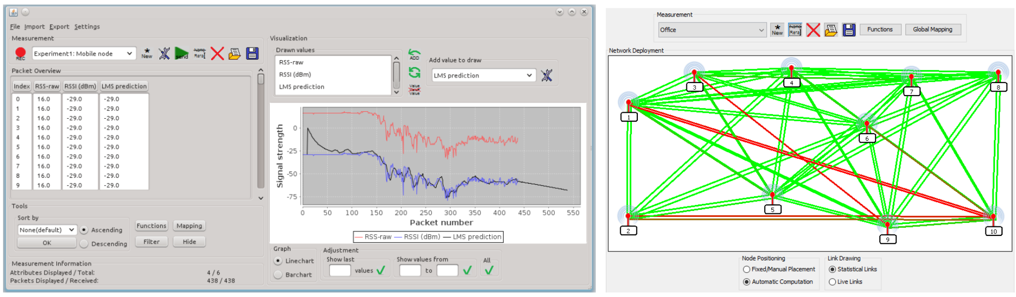
With TRIDENT [p75], a tool for in-field connectivity assessments in WSNs is proposed. Data can be gathered without requiring additional infrastructure by providing a configurable workflow, installing pre-compiled and experiment-agnostic software on the network devices, performing measurements and allowing for user interaction. Within the provided graphical user interface, several collected metrics can be visualised to gain insights into the basic network properties. E.g., the ”in-field assistant” visualises connectivities and packet delivery rates (PDR) of nodes, allowing to analyse the gathered connectivity data across multiple experiments to reveal insights about the network.

With providing these features, TRIDENT addresses users without dedicated WSN expertise and enabling them to conduct experiments by taking the part dealing with the hardware off their hands. However, due to its focus on WSN devices with their quite specific hardware and the approach to provide an automated solution requiring low user effort for obtaining information, the area of application of TRIDENT is limited. The tool supports automated assessments for two types of devices, TMote Sky, mainly used in research and Waspmote, rather used in industrial applications. The support of each new type of device requires specific adaptations due to the individual hardware capabilities including the configuration and the ability to provide different types of data. Therefore, extending the features of TRIDENT to provide automations for further devices requires noticeable efforts including additional expertise on the developer side, demonstrating the drawback of such specialised, but beneficial approaches.

A similar approach but with a larger focus of data analysis is provided by IRIS [c5]. It supports direct connections to base station nodes of TinyOS-based wireless sensor networks [p94] to download information of exchanged messages stored on the devices and to send messages into the network as well. The obtained data can then be analysed using the contained components. Having a basic set of tools available, such as a message explorer and a graph visualisation component, as shown in Figure 3.1, the modularised architecture also enables further customisation to better match the requirements of a targeted network analysis. E.g., functions can be defined and applied to metrics of a measurement in arbitrary order and afterwards visualised in different ways.

This freedom of customisation, though, comes at the cost of increased efforts on the user side when requiring computations beyond the scope of the available, more general components, but on the other hand offers more potential for obtaining the desired results and knowledge. In general, a larger set of features and capabilities introduces more complexity and increases the expertise required by the users. Depending on the actual use case, including the level of available network knowledge as well as the user expertise, providing both options, predefined and extensible components, can be beneficial to preserve applicability for most scenarios.

Aforementioned tools can be very helpful for obtaining a data basis with low effort on which further analyses can be conducted. However, as stated in Section 1.2, the capabilities of these tools to obtain and subsequently analyse certain data are limited to the ability of individual networks and



Measurement: Experiment1: Mobile node

Packet Overview

Index	RSS-raw	RSSI (dBm)	LMS prediction
0	16.0	-29.0	-29.0
1	16.0	-29.0	-29.0
2	16.0	-29.0	-29.0
3	16.0	-29.0	-29.0
4	16.0	-29.0	-29.0
5	16.0	-29.0	-29.0
6	16.0	-29.0	-29.0
7	16.0	-29.0	-29.0
8	16.0	-29.0	-29.0
9	16.0	-29.0	-29.0

Sort by: None(default) ● Ascending ○ Descending

Measurement Information
Attributes Displayed / Total: 4 / 6
Packets Displayed / Received: 438 / 438

Drawn values: RSS-raw, RSSI (dBm), LMS prediction

Add value to draw: LMS prediction

Graph: ● Linechart ○ Barchart

Measurement: Office

Node Positioning: ○ Fixed/Manual Placement ● Automatic Computation

Link Drawing: ● Statistical Links ○ Live Links

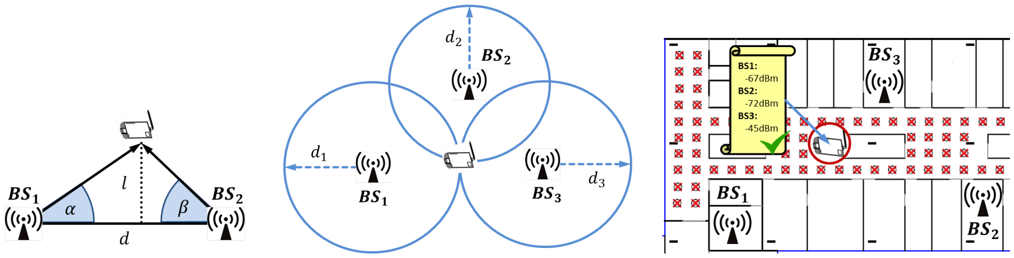
Figure 3.1: *IRIS tool for obtaining and analysing data in wireless networks (image source: [c5]).*

devices to provide such information. The same heterogeneity also aggravates a general applicability of automated approaches, e.g., supporting different types of hardware requires dedicated adaptations to be made, including time, costs and expertise. Thus, enabling the advantages of such approaches is preceded by the corresponding phases of setup and preparation. This mostly inevitable effort imposes a significant limitation when targeting at offering such features for a wider range of networks. As a consequence, when designing tools, a decision between generality and specificity with regards to the offered capabilities usually has to be made. Ideally, such approaches for analysing and obtaining network knowledge are able to offer both in order to cope with the wide range of individual conditions of each network in the best possible way.

3.1.2 Device Localisation and Topology Detection

Having the connectivity data of a network available marks the foundation for analysing and revealing basic network properties, such as a logical topology, visualising the data flow or a coarse relative distance estimation based on the signal strengths between devices. An exclusive view on these data, though, restricts the informative value due to the lack of reference to the environment. Instead, the combination with other additionally available information establishes this reference to the environment, such as device positions and maps, opens up a multitude of new possibilities and yields further valuable insights which on their part can serve as foundation for optimisations. Within this section, approaches are presented and discussed enabling the localisation of devices with the help of connectivity data and information about the environment.

Localisation is a long treated topic, in particular indoors where a variety of wireless technologies and approaches have been used to support many applications [p91, p104, p139]. In contrast to most outdoor scenarios in which satellite systems for navigation are available providing accurate posi-



(a) Triangulation **(b)** Trilateration **(c)** Fingerprinting

Figure 3.2: *Common localisation techniques.*

tion estimations like the commonly used Global Positioning System (GPS), there does not exist any comparable predominant approach for indoor scenarios. Therefore, a variety of approaches have evolved taking the diverse circumstances of each scenario into account to enable localisation, such as the available infrastructure and hardware capabilities. In case of limited data and hardware, range-free algorithms provide localisation capabilities without requiring distance measurements by using network-related information, such as the number of hops between devices. However, utilising such coarse information comes at the cost of accuracy, resulting in a positioning of nodes relative to each other. In contrast, range-based localisation approaches require the availability of inputs enabling distance calculations. Hence, the position of several so-called anchor devices or base stations is assumed to be known in order to compute the position of other devices in the same network. With such reference positions, various techniques can exploit different types of measurements which are described in the following.

Angle of Arrival

Using the Angle of Arrival (AoA) technique, the position of a device is estimated based on the direction on which a signal was received at multiple reference devices [p112]. By utilising geometric relationships the position at the intersection of lines is computed, known as triangulation, as depicted in Figure 3.2 (a). In two-dimensional space, at least two base stations as well as the corresponding angles of the device to these are required to perform a localisation. In order to obtain the necessary angle information, this approach requires dedicated hardware on each of the anchors using either rotating or multiple antennas. In case of the latter, the direction and thus the angle to the anchor device upon receiving an incoming signal is computed by measuring the time difference of arrival on the different antennas.

Although it is quite simple to compute the device position once the required information is available, applying this technique requires the aforementioned antenna hardware on the base stations to determine the angles, narrowing down the applicability. Especially in IoT networks, where the hardware capabilities are restricted in terms of equipment and performance due to limited energy resources, the Angle of Arrival is a rather inappropriate approach for localisation. In this context, other techniques are more reasonable to use, which are less relying on required hardware but on analysing the data available with respect to the given restrictions, such as the received signal strength, which is provided by almost all radio chips. Furthermore, minor errors in the angle estimation can have a significant impact on the accuracy as the possible error range linearly increases with distance. Hence, the usage of further base stations is required at larger distances to limit the error and maintain accurate localisations. Due to the previously mentioned aspects, the Angle of Arrival is rather used as supplementary technique to increase the accuracy in combination with other localisation techniques [p169, p5].

Signal Propagation Time

A different approach for estimating the position of a device is by measuring the Time of Arrival (ToA) of a sent signal [p130]. Thereto, the device to be localised includes a timestamp containing the information when the message was sent. Based on the signal propagation speed and the travel time between the sending of a message and its reception, each base station can then compute the distance to the device. The potential position of a node is then located on a circle in two-dimensional space or a sphere in three-dimensional space with the respective estimated distance as radius. To narrow down the device position to a specific point, lateration is used requiring three respectively four base stations. An example for a two-dimensional scenario is shown in Figure 3.2 (b).

Besides also considering arriving signals from indirect, non line of sight paths which might distort distance estimations, exactly measuring the travel time of a signal requires a precise time synchronisation between the involved devices. Minor deviations of a few milliseconds already result in large estimation errors because of the propagation of electromagnetic signals at the speed of light. However, especially when dealing with IoT devices, clock synchronisations are necessary quite frequently, but also challenging as well due to observed large clock drifts within short time periods and poor clock rate stabilities [p99]. Thus, using the Time of Arrival as localisation technique introduces further overhead resulting in additional effort for employing a time synchronisation protocol if not already present as well as increased energy consumption. For devices with high-quality hardware or a permanent power supply this additional costs might be limited and acceptable, but especially the increased energy demand again means that this localisation technique is not suitable for usage in IoT networks.

To circumvent the need of synchronisation between base stations and device, instead of using absolute time values, the Time Difference of Arrival (TDoA) can be measured [p138]. In this case, only the base stations are required to synchronise and then pairwise compare their measured point in time when a signal was received, resulting in the possible locations of the sending device being located on a hyperbole. Additional pairs of anchors or exclusion principles based on prior observations allow to further limit the device positions. By using this TDoA approach, the devices to be localised do not need to actively take part in the localisation process by sending timestamps or synchronising clocks, hence increasing the area of application since any device communicating with the network can be localised without implementing dedicated protocols. Nevertheless, the base stations still have the additional overhead to regularly exchange messages for time synchronisation and to perform their localisation computations.

Alternatively, to entirely eliminate the need for time synchronisation, two ways of transmitting a signal can be used in parallel, such as radio and ultrasound waves [p20]. The time difference of the reception of both signals and the known propagation speeds of both wave types allow to calculate the device distance at each base station independently. However, similar to the Angle of Arrival localisation, this approach requires dedicated hardware and thus restricts the applicability to networks with such capabilities to send signals using different wave types.

Utilising the signal propagation time to estimate the distance to other devices and use this information to localise them when combining multiple of these estimations is a simple and effective technique. However, as discussed before, a network and its devices have to meet certain requirements depending on the exact approach, either on the software or the hardware side. An approach pursuing the goal of being suitable and usable for most networks ideally should have as few hardware requirements as possible and no need to use special software or protocols. Against this background, the consideration of other techniques is appropriate, utilising different information whose collection implies fewer constraints on a network.

Received Signal Strength

A commonly available information in wireless communication is the signal strength which is measured and provided by radio chips upon reception of a signal. This information can be obtained from the devices without much effort and allows to estimate distances based on the logarithmic decay over distance, given the initial transmission power [p98]. With the distance to other devices available, lateration can be used again to perform a localisation.

However, similar to the propagation time, the signal might have not arrived at the receiver using the shortest direct path or was even influenced by multipath interferences. Therefore, also the received signal strength can provide a distorted view on the actual distances between devices when used exclusively. Nevertheless, it allows for coarse first estimates and provides valuable information about the network topology that can be refined when combined with further information. As already mentioned in Section 2.1, the wide availability of the measured signal strength qualifies this metric to be employed in general approaches for providing localisation for a wide range of networks. In addition, further network adaptations or protocols are not required when basing the analysis on this information.

A different approach of using the signal strength is to build fingerprinting maps of the environment and match the observations against pre-recorded values to localise devices [p17, p168], as depicted in Figure 3.2 (c). Thereto, the signal strength of nearby base stations is recorded at a large number of defined positions within the network environment to build up a data set. These fingerprints are then compared against measurements taken from devices which should be localised to find the best matching position. With a sufficiently large number of base stations, each recorded position is clearly distinguishable from the others and enables accurate localisations.

Although combining multiple signal measurements in a meaningful way, fingerprinting is often indicated as an impractical approach for most applications and environments. First, depending on area size and targeted resolution, a comprehensive collection of fingerprints is costly and time consuming. Furthermore, the data sets have to be updated frequently since local changes in the environment can otherwise falsify measurements and reduce the localisation accuracy. Thus, fingerprinting requires recurring efforts to maintain proper functionality. Such tasks of data collection have been partially automated by using robots [p22, p110] or crowdsourcing [p133]. Nevertheless, such possibilities are only available to a limited number of networks and therefore the manual overhead remains in most of the cases.

Other Approaches

The approaches that most closely relate to the contribution of this thesis are the ones that reconstruct topologies based on connectivity information only. [p88] shows that it is possible to deterministically identify the network boundaries and extract topologies for a self-organisation of large wireless networks in urban scenarios through the use of Voronoi clusters. In multiple distributed computational steps, the algorithms determine boundaries of a network and identify street patterns and crossings via clustering. The work, however, bases on the assumption of an ideal unit-disk communication model, i.e., a binary reception model depending on the distance to the sending device, and it is also demonstrated only in a large simulated network. Thus, the approach does not consider environmental influences and is rather suitable for outdoor modelling with many line of sight connections. In indoor scenarios, depending on the shape of the buildings, topologies are usually more compact and connectivity data is significantly influenced by propagation effects due to walls and other obstacles which might distort resulting topology graphs with respect to reality.

Multidimensional scaling (MDS) [p136] is a process used in statistics which creates graphs based on relative pair-wise dissimilarity between nodes. It arranges nodes spatially in such a way that the distances between them in space correspond as closely as possible to the surveyed dissimilarities. In case of localisation in wireless networks these values can be, e.g., the measured signal strength, the number of hops or any other metric reflecting connectivity between two devices. As MDS can be applied without any further information regarding the network except from measurements, it is subject to the same problems as the approach presented in this thesis. Without reference points, the resulting device positioning can be an arbitrarily rotated and flipped and then requires up to three anchor nodes to solve them. MDS is used as reference approach to compare against the mentioned proposed solution for localisation (Chapter 4).

Discussion

As already evident from previous approaches and also concluded in research [p91], no prevailing single localisation solution exists for indoor areas due to the different and unique properties of each network. Therefore, approaches have to be combined considering and covering a larger amount of individual circumstances in order to extend their applicability to as many networks as possible. However, many approaches rely on assumptions or have certain requirements, not all networks are able to fulfil, e.g. having two or three anchors in range of each node or knowing the position of the devices. There exist only few approaches which are able to provide localisation for a minimum amount of available information and network intervention as well. However, such approaches which in addition are able to include further optional information to improve accuracy are lacking.

The localisation approach presented in this thesis exactly addresses this issue and is based on the received signal strength which is available on most of the wireless devices. It provides the possibility to include supplementary information of environment and network to increase the accuracy. Furthermore it uses a combined approach to discover topology and localisation of a complete network at once by utilising force-directed graph algorithms, which provide an effective way of arranging graphs neatly by utilising physical laws to determine the positions of the vertices (Section 4.3.1).

3.2 Wireless Communication Modelling

Once basic information about a network are available, a next step is to gain further knowledge and deeper insights about the network and its properties. One possibility is to utilise the obtained information by employing wireless communication models. With proper modelling approaches and sufficiently detailed input, they are not only capable of reproducing the communication behaviour of the actual network, but also to enable predictions on the behaviour at arbitrary locations. Therefore, by shifting from mere knowledge acquisition to utilisation, such modelling is an essential first step offering possibilities for deeper network analysis and opening up new opportunities for optimisations. In general, it is possible to identify two alternative approaches to the problem of describing wireless communication properties, an empirical and a deterministic one.

Empirical (or statistical) models are based on statistical analyses using collected measurements as input. Their prediction accuracy is based on data quality and quantity of the gathered measurements. Usually the computations are fast and they are able to accurately reproduce observed behaviour, but provide less accurate results for unknown positions since the target environment is not taken into consideration directly. On the other hand, deterministic models base their predictions on the physical propagation characteristics and rely on knowledge of the network environment instead of network communication measurements. Due to the adherence to physical laws, they provide more accurate results than statistical models but also require detailed input data regarding the environment geometry as well as more complex computations. Depending on the available information re-

Table 3.1: *Path loss components for different environments [p123].*

Environment	Path Loss Component γ
Free Space	2
Urban Area Cellular Radio	2.7 to 3.5
Shadowed Urban Cellular Radio	3 to 5
In Building Line of Sight	1.6 to 1.8
Obstructed in Building	4 to 6
Obstructed in Factories	2 to 3

garding measurements and environment, either empirical or deterministic models should be employed, preferring the former when reproducing observations and the latter when requiring predictions at specific, unknown locations. In the following, related models of both types are adduced.

3.2.1 Empirical Modelling

Experimental studies have been largely used to characterise specific environments and technologies in details [p142, p27, p37], further highlighting the significant difference between distinct environments [p108, p102, p122, p125] or distinct weather conditions [p23]. While these studies provide invaluable insights on communication behaviour in real-world environments under specific conditions, they suggest the need of a manual experimentation in-situ before deployment. In this thesis, it is explored how modelling techniques together with observations taken in an operational system can be used to understand the reasons for specific communication properties and variations.

One of the most basic approaches, the log-distance path loss model [p123], is widely used in simulation and signal prediction, since it requires a low computational effort and considers the characteristics of signal fading in different surroundings according to the following equation:

$$PL(dB) = PL(d_0) + 10\gamma \log\left(\frac{d}{d_0}\right) \tag{3.1}$$

The path loss at a reference distance d_0 is given by Equation 2.6 and the environment-specific coefficient γ, also called path loss component, can be derived empirically from measurement data [p119, p9]. Examples for path loss exponents depending on the specific environment are given in Table 3.1. As it can be seen, these values largely vary based on the local circumstances including environment properties and also whether there exist line of sight or obstructed connections between the devices. Furthermore using such simplified equation, the model assumes a continuous decay over distance and is not able to take the dynamics of the environment into account. Hence, the log-distance model provides a general description that may deviate from the specificity of individual links, e.g., depending on their length [p83]. Therefore, more detailed approaches have been introduced adding further correction values to the predicted path loss, tailored to specific scenarios, e.g., in- and outdoor deployments with certain frequency ranges and obstruction densities [p1].

Outdoor Modelling

Designated for outdoor propagation modelling, several empirical models exist [p36, p59]. Most of them are modifications, extensions or combinations of basic models developed in the 1980s, e.g., Hata [p60], Walfisch-Bertoni [p161] and Ikegami [p71]. These models describe radio wave propagation in urban scenarios by incorporating the influence of physical effects as parameters into the path loss computations, such as antenna heights or incidence angles. Although all these models include rudiments of propagation physics they still base their predictions on empirical values without performing actual signal path computations.

The Hata model [p60] describes the influence of antenna heights on wireless communication in urban scenarios. It provides modelling approaches for areas of different population density, distinguishing three types of terrain, urban, suburban and rural areas within propagation distances up to 20 kilometres. The antenna emitting the signal is assumed to be located above adjacent roofs (30m-200m from ground) while the receiver antenna is located below (1m-10m from ground). The investigated frequencies covered by the original Hata model range up to 1500 MHz. With additional analysis and measurements, this range has been later adapted by the COST-Hata-Model [p36] to also cover frequencies between 1500 MHz and 2000 MHz. Afterwards due to the growing demand for more and thus higher frequency bands, the Stanford University Interim (SUI) models [p59] furthermore improved and extended the Hata model to frequencies above 1900 MHz.

In addition to the free space loss and the path loss component of Equation 3.1, specific correction factors have been introduced for each of the terrain types to take the individual influences of the building densities on the signal attenuation into account:

$$PL(dB) = PL(d_0) + 10\gamma \log\left(\frac{d}{d_0}\right) + X_f + X_h + S \tag{3.2}$$

where X_f is a loss correction based on and increasing with the frequency, X_h a correction based on the antenna height of the receiver and S incorporates shadowing effects. To provide accurate predictions these corrections differ depending on the modelled scenario.

With only few required information regarding the scenario characteristics and therefore rather low effort, Hata and the derived models provide meaningful general signal strength estimations in urban scenarios. However, signal propagation effects as mentioned in Section 2.2 are only implicitly covered by these models through the conducted measurements, the estimations and correction values are based on. Despite a generally acceptable accuracy, this complicates precise predictions for specific positions.

Considering further influencing factors aside from the antenna heights, the COST-Walfisch-Ikegami propagation model [p36] also takes diffraction effects into account. In urban non line of sight communication the receiver is assumed to be located in a street canyon between larger buildings where the signal, propagating above the roofs, is then diffracted at the edges of these buildings. To describe the scenario and provide meaningful estimations, more information is required such as (average) building heights and separation distances, road widths and orientations in relation to the direct radio path. Albeit required in a mostly general manner, each additional information entails further efforts for acquisition and also increases the complexity of the model. Nevertheless, this model is capable of estimating the signal quality in urban scenarios including the empirical description of diffractions while still refraining from requiring precise geometrical details about the environment.

Empirical outdoor models provide a useful approach for a fast and widespread application of signal predictions when a general overview is sufficient and precise spatial characteristics of minor importance. Requiring rather general information, adaptations to specific scenarios within the constraints of the models can be realised without much effort. Especially for covering scenarios with similar conditions, these models provide widely acceptable estimations due to the preceded sophisticated measurement campaigns, the computations are based on. However, the generality is acquired at the cost of accuracy since unspecific general information about a scenario is evidently never able to describe propagation processes in such detail as precise site-specific information would do, which of course in turn require much more efforts to obtain. Therefore for large-scale outdoor scenarios with many potential individual characteristics to consider, the usage of a more general modelling concept, as empirical models provide, is a reasonable option.

Indoor Modelling

Entailing a different set of challenges, the same comprehensiveness of approaches exists for indoor propagation modelling [p131, p12, p141]. While in large urban outdoor scenarios more specific individual information on, e.g., the presence of a bus stop cottage or the building materials are insignificant, the smaller scale of area shifts the focus on exactly these details. The building structure, including existence and materials of walls, doors, windows or other obstacles becomes the most essential influencing factor on the signal propagation [p97, p96]. Furthermore, indoors spatio-temporal events like moving persons or interfering noises have far more weight on the network behaviour and thus have to be considered as well to ensure a proper operation [p129].

The International Telecommunication Union (ITU) has proposed a model for indoor signal predictions covering multiple floors and implicitly including effects like transmissions through obstacles [p135]. In addition to the distance-based signal decay, the frequency and the number of travelled floors is considered for the path loss computation as well:

$$PL(dB) = 20 \log f + N \log d + L_f(n) - 28 \tag{3.3}$$

where f is the frequency in MHz, N the power loss coefficient, d the distance between sender and receiver and $L_f(n)$ the floor penetration loss factor based on the number of travelled floors. For N and L_f, the model provides different values in dependency of the used frequency and building type, i.e., residential, office and commercial buildings.

By differentiating between path losses occurring on the same, but also on the transition to a different floor, the ITU model combines two basic aspects of indoor signal propagation in a simple formula to calculate. However, specific effects caused by the individual constellation of obstacles defined by the building floor plan and furniture are only reflected indirectly within the empirically determined values. Thus, the universally applicable modelling is again achieved at the cost of reduced accuracy compared to other models including more specific details of the environment [p63] like the one described next.

Taking the significance of obstacles on the signal propagation in indoor scenarios into account, the Motley Keenan multi-wall model [p81] proposes a more detailed method to consider losses which are caused by walls. By distinguishing different coarse wall categories, the model sums up their type-specific attenuations depending on the number of walls located between the communicating devices. Since the original model only considers certain reference walls with fixed attenuations, it has been refined in later studies by considering the actual wall thickness and its effect on the attenuation as well [p96]:

$$PL(dB) = PL(d_0) + 10\gamma \log\left(\frac{d}{d_0}\right) + \sum_{i=1}^{n} k_i \, L_{0i} \, 2^{\log_3\left(\frac{e_i}{e_{0i}}\right)} \tag{3.4}$$

where k_i is the number of walls of type i, L_{0i} the penetration loss of the reference wall, e_{0i} the thickness of the reference wall and e_i the thickness of the actual wall. With this approach, the experimentally determined attenuations of specific wall types can be better tailored to a broader set of scenarios resulting in an increased estimation accuracy while still limiting to certain general wall categories.

To properly apply the Motley Keenan model, general knowledge of the floor plan, the type of existing walls and the positions of the devices have to be present, though. Additionally including the thickness of individual wall types requires even further detailed knowledge about the environment. Due to the complexity of indoor communication, many empirical models are supplemented with increasingly specific environmental information to improve accuracy. Given the network-wide availability of such details, the usage of a deterministic model could also be a reasonable option at a certain point, depending on the targeted accuracy. Since transmissions and therefore wall attenuations play a significant role in indoor signal propagation, the general concept of Motley Keenan has also been adopted for the ray tracing models, presented in Chapter 5.

Indoor scenarios pose a special challenge to propagation models. As a result of the much higher density of obstacles and thus a significantly larger potential interplay of propagation effects, precise estimations in indoor scenarios on a local level are more difficult to achieve, especially for empirical models. Thus, more environmental information are taken into consideration by these models to mitigate the difficulties.

Overarching Approaches

With an increased demand and growth of large scale urban networks, indoor signal coverage also needs to be taken into account when exchanging signals with a source located outdoors. Due to availability of detailed structural information in cities, out- to indoor modelling approaches exist combining both domains using various frequencies [p89, p128, p146].

Since in this case challenges of both areas have to be considered, the computations of these models are usually composed of multiple steps, first dealing with the outdoor propagation to the respective target building and then handling the indoor propagation using different sets of equations and parameters. However, each building is individual regarding construction materials, floor plan and furnishing. Hence, the prediction accuracy of a combined modelling approach is highly dependent on the availability of such information for a modelled area. To reduce inaccuracies introduced by the increasing complexity, e.g., [p89] includes an increased amount of deterministic elements, such as a grid-based path segmentation on the basis of terrain data, angle computations and a differentiation between line of sight and non line of sight situations. As a consequence, the more details have to be taken into account, the more the boundaries between empirical and deterministic models start to fade.

Finally, recent trends on applying machine learning have not omitted propagation modelling [p15, p126]. Producing accurate predictions is possible, but time-consuming phases of collecting measurements and network training are required before the model becomes usable. Moreover, in comparison to other reference models the improved accuracy is obtained at the cost of increased resource requirements, i.e., time and memory consumption and applying such model to other scenarios with different properties again requires further training phases to be conducted. Therefore, machine learning provides useful estimations, once trained properly, but are lacking flexibility. Hence models based on this approach suffer from similar problems of scenario specificity as their manually configured counterparts.

Discussion

Although relying mostly on measurements, upon targeting more accurate predictions, empirical models still require an increasing amount of information about the environment. As more and more additional factors are considered, the model complexity naturally increases, especially when keeping them applicable and valid for a multitude of parameters, e.g., frequencies, distances or environment-specific settings. Setting up and validating such equations for each of the regarded cases on the other hand require many measurements to be performed. In addition, conducted measurements might deliver insights which are not fully applicable to other scenarios since certain important characteristics are different. Depending on the required details and targeted accuracy as well as considering the growing availability of topological information about cities and even building interiors, at some point the usage of deterministic models constitutes a reasonable alternative to empirical ones. Apart from that, most empirical models already contain deterministic elements in order to improve their predictions so that a clear categorisation of such models is difficult in practice.

Furthermore, their dependency on experimentally determined input parameters for achieving accurate predictions restricts empirical models for wireless communication to rather static scenarios with only marginal changes over time. From a global point of view they provide good general approximations for signal quality and thus are convenient for predictions on a macro-scale, e.g., planning where to place cellular transmitter in an urban environment. However, they lack in details, e.g., considering specific positions of smaller but relevant obstacles, necessary to accurately predict signal quality at a local scale. Empirical models are incapable of taking environmental dynamics into account, as these again would require further adaptations of the model parameters to be considered. This renders such models inappropriate for describing temporal effects such as spatially limited influences on the signal propagation, like moving traffic or persons. As a consequence, different types of models have to be specified allowing for more flexible but still precise predictions.

3.2.2 Deterministic Modelling

Wireless communication bases on radio waves that propagate and interact with objects in the environment according to laws of physics. Focusing on these, deterministic models provide realistic predictions and are not reliant on experimentally determined input parameters but rather on site specific information of the environment. However, reflection, diffraction and scattering are effects that are very hard to model due to impact of, e.g., object shapes and materials as well as of the exact location of senders and receivers [p171]. Hence to achieve accurate predictions, deterministic models require detailed information, are usually complex to implement and computationally expensive, especially for 3D models [p124, p166]. For obtaining predictions of wireless communication in a given environment several approaches exist, focusing on different aspects of wave propagation. Many of them base on geometrical optics, such as the approaches described in the following section.

Figure 3.3: *Images using ray tracing for rendering light effects (image sources: [m15, m14]).*

Ray Tracing

Simulating the physical propagation of waves in a computer model by regarding and computing continuous wavefronts as discrete amounts of individually traceable waves, named rays, is called *ray tracing*. This technique has been used in many applications, in-, outdoor and hybrid scenarios both in two and three-dimensional realisations [p134, p156, p6, p48, p72, p30, p132, p137, p154]. Behaviour and properties of a generated wavefront at any position in a given environment are reconstructed by tracking the paths of each ray leading to this position. Applying laws of physics on the interaction with obstacles and the propagation in space for each ray, the expected attenuation and thereby the coverage and energy of the wave at arbitrary positions can be computed. Although varying on the specific characteristics in interaction with obstacles at different frequencies, the basic wave propagation effects remain the same. Therefore ray tracing can be used to simulate phenomena involving waves of different wavelengths. One prominent application is to model effects of visible light, i.e., waves with frequencies between 430 THz and 750 THz, in images and gaming applications to provide a realistic impression of an illuminated scenario [p65]. Figure 3.3 displays two examples of image rendering using ray tracing to compute light effects, such as reflection and refraction. Besides the energy of traced rays and the resulting shown intensity, colour information, i.e., the exact wavelengths, have to be considered in image rendering as well when dealing with visible light.

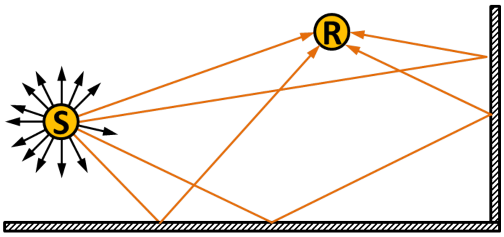


Figure 3.4: *Rays computed by aimed (orange) and launched methods (orange & black).*

In contrast to visible light, applying ray tracing to model signal propagation in wireless communication usually requires to analyse signals given at one specific wavelength. Minor frequency shifts due to propagation effects are neglectable since radios are designed to receive signals within a tolerance around the designated frequency. Therefore when modelling wireless communication, a prediction of the signal power at a given position is the most relevant information to obtain.

For implementing ray tracing, there exist two basic concepts how rays are computed: a quantitative launched and a qualitative aimed approach. Figure 3.4 demonstrates the difference in the amount of computed rays between aimed and launched approaches. In the context of this thesis, both approaches, aimed and launched ray tracing, are utilised for modelling wireless communication, combining benefits of both methods (see Chapter 5).

Launched Ray Tracing In the launched approach numerous rays are shot in all directions around the sender and tracked whether arriving at the receiver [p134, p30]. For each ray, computations mostly consist of finding the closest obstacle crossing the current propagation direction of the tracked ray. If found, depending on the provided detail of the ray tracing implementation, one or more propagation effects are applied at the point of impact, changing the state and possibly the direction of the investigated ray. This process is repeated until the ray either arrives at the receiver or is marked as invalid due to given limitations, e.g., exceeding the maximum travel distance, the number of considered effects or leaving the scene boundaries.

Launching a large but finite amount of rays at arbitrary directions in a trial and error principle implies many irrelevant path computations for rays not reaching the receiver and also involves the risk of false misses. Despite drawbacks and risks, due to being limited to subsequent and independent obstacle collision tests, the tracking of launched rays requires considerably less computational resources compared to an aimed approach. Furthermore computations can benefit from parallelisation, central processing units

(CPUs) and graphic processing units (GPUs) are able to provide. Especially the latter GPU parallel architectures, which intended usage is to efficiently handle 3D environments with respect to light waves, offer ideal preconditions for performing radio wave computations.

Aimed Ray Tracing In contrast, an aimed approach focuses on constructing only valid ray paths [p156, p6, p48, p137]. Starting from the sender, instead of tracking a fixed number of previously generated rays with specific directions, surrounding obstacles are analysed for possible interactions providing ray paths either reaching or reducing the distance to the receiver. Promising paths are further pursued, unfavourable ones discarded, successively constructing valid paths to the receiver while involving further obstacles.

Assuming unrestricted computational resources and sufficient details of the environment available, the aimed approach is able to perform an exhaustive search reliably discovering all possible ray paths from sender to receiver. However, constructing ray paths entails significantly more computational effort than just launching rays and following their path, as each path has to be recursively tracked to assess its validity when reaching the receiver. Considering each and every combination of obstacle interactions for a single ray quickly creates large path trees with countless branches depending on the number of present obstacles. Consequently, an introduction of limitations and the usage of accelerating shortcuts and algorithms are inevitable to make an appropriate trade-off between accuracy and computation time [p7].

Area-based Propagation Modelling

In contrast to determining single-point signal propagation paths, the following models compute signal paths using areal approaches to provide their estimations.

Entirely focusing on reflections, a technique similar to ray tracing, computing entire sets of rays instead of individual ones, is called beam tracing [p147, p47]. All positions a ray is able to reach after being reflected on a certain surface are described as a cone. Starting from the sender, these cones are constructed recursively using the image method (Section 5.3.2) to determine reflected paths to the receiver. This approach was proposed to quickly trace reflection paths in indoor scenarios.

As experienced during the implementation of the ray tracing approach (Chapter 5), restricting path computations to reflections exclusively indeed speeds up the computation time compared to combined approaches considering other propagation effects as well. However, as the authors also noted themselves, neglecting the similarly important effects of diffraction and transmission might affect prediction accuracy. Thus a more general and comprehensive model needs to take the latter into account as well.

Utilising another algorithm used in computer graphics, a radiance model is proposed in [p107]. Radiosity is a rendering procedure determining surfaces on which light sources are reflected diffusely to illuminate entire scenes in a more realistic way. Transferring this concept to signal propagation, the model uses facets and edges of buildings in urban scenarios to identify areas in which radiance transfers can occur via reflection and diffraction. By combining a three-dimensional grid-based spatial division of the geometric elements to limit intersection tests with a precomputed visibility graph, paths from sender to receiver can be constructed.

For the computations of the conducted evaluation, equal reflective properties for all 303 buildings have been assumed in the investigated area of approximately 1 km^2. To further increase the accuracy of estimations, the individual reflective properties of each building have to be taken into account, which the authors have refrained from. While the geometry can be retrieved from now publicly available databases, however, obtaining construction materials and properties of each building at an urban scale is difficult. Consequentially, the full potential of the given approach cannot be exploited.

A different approach is presented in [p55], called multi-resolution frequency domain ParFlow. The underlying concept is to consider wave propagation in an environment as an energy flux. Dividing the site area into smaller segments, preferentially alongside and around obstacles like walls or pillars, the exchange of energy between these segments is computed. To obtain a prediction, the paths of energy flow from the segment of the sender to the segment of the receiver is determined by traversing the precalculated tree-based structure of adjacent segments. The computation time is comparable to unoptimised basic ray tracing methods [p56] and provides a similar prediction accuracy.

To perform predictions in a reasonable time, ParFlow relies on the precomputed segmentation of the environment which is the most time-consuming part of the computations. Ideally such segmentations have to be done once, but changes in the environment can invalidate the optimisations and require to trigger repeated adaptations. Thus, such approaches have difficulties in adapting to frequently occurring dynamic changes, e.g., moving obstacles or persons, without sacrificing either accuracy by ignoring these changes or speed by discarding or repeating precalculations.

Discussion

The previously introduced models use individual physical approaches to determine the expected signal strength at target positions. Areal modelling approaches offer the benefit of providing predictions for larger regions with fewer computations. Especially in cases of a high scenario complexity where the computation becomes increasingly more expensive due to many obstacles to be taken into account, these models are advantageously when requiring exhaustive predictions. On the other hand, targeted approaches such as ray tracing allow for selective and locally restricted, but fast computations on, e.g., dynamically changing obstacles, moving transmitters and receivers. Compared to areal approaches they usually require less complex computations for a single iteration and thus can better adapt to changes in the environment.

As a consequence of their more detailed way of computation based on geometrics of the environment and physical laws, deterministic models have correspondingly higher requirements for necessary details. With the increasing complexity of larger scenarios, obtaining such specifics could become costly, reaching a point where the procurement may no longer be proportionate to the benefits. Hence, if refraining from taking the necessary effort, a reduced prediction accuracy has to be accepted instead. Furthermore, the utilisation of such details is reflected in increased computation times, requiring optimisations and shortcuts to keep the approaches practical. Besides potential trade-offs in accuracy, especially shortcuts based on intensive precalculations also bear the risk of reducing the flexibility of the underlying approaches and thus narrow down their areas of application. For this reason, it should be assessed individually for each application purpose which of the available approaches can be considered for modelling the given scenario with which required level of detail.

Tackling the aforementioned conflict of finding an appropriate balance between requirements and targeted accuracy with regard to the availability of information in a given scenario, this thesis proposes an adaptive approach which preserves the flexibility of usage (Chapter 5). Consisting of multiple communication models of both types, deterministic and empirical, which are either used individually or jointly, the advantages of the different models are combined to extend the applicability and create synergies which improve the accuracy of the given models. Hence, a modelling of wireless communication is made available to a wide range of scenarios, allowing to further increase the network knowledge by discovering new device positions.

3.3 Understanding Changes in the Environment

Insights and results gathered by the approaches mentioned in the previous sections during the earlier stages of knowledge acquisition support network operators in obtaining a basic understanding of their network topology and discover potential optimisation spots. These improvements are mainly based on single snapshots of the respective networks. Since the communication behaviour of most networks varies over time due to changes in the environment, additionally including temporal aspects in network analysis helps to reveal coherences and thus to better understand the network behaviour. This in turn is beneficial to make educated decisions on possible optimisation steps. Within this section, a general overview of existing approaches is given analysing and understanding network dynamics caused by changes in the environment.

3.3.1 Impact of Obstacle Movements on Wireless Signals and their Detection

Considering a specific point in space, one of the main reasons for occurring fluctuations of the signal quality over time is the movement of objects or obstacles, in the following also called *events*, temporarily or permanently altering the propagation paths and thus affecting the measured signal strength. These obstacles can be classified into three categories: static, semi-dynamic and mobile. Static objects are not expected to often alter their positions (if at all) as for example any building structures outdoors or larger furniture indoors. Semi-dynamic obstacles are able to move locally, but within limited degrees of freedom, e.g., doors, windows or elevators. Lastly, mobile objects are capable of freely moving through the environment without any restrictions, e.g., humans, animals, vehicles or robots. With an increasing mobility, the predictability on the network behaviour naturally decreases as more object states and positions have to be taken into consideration. However, in order to obtain a full picture of the network environment, reveal possible risks and implement counter-measures, detailed investigations regarding potential impacts on the network performance are required. An overview regarding current research of the previously mentioned object categories is provided below.

Table 3.2: *Mean path loss for three classroom configurations [p127].*

Configuration	Empty	Furnished	Furnished and Persons Present
∅ Path Loss	73.05 dB	79.38 dB	80.69 dB

Static Objects

The impact of walls and other fixed building structures on the signal propagation behaviour has already been tackled in Section 3.2. Instead, research on the effects of rather static, but potentially movable objects is now examined in more detail, as these are of greater relevance exerting influence on the network behaviour during operation anyhow.

The general effect of larger interior objects on infrared frequencies has already been investigated in simulation [p2]. By extending the prediction model described in [p19] computing multiple signal reflections in rectangular rooms with the geometry of furniture, predictions could be better aligned with observed behaviour compared to simulations using only basic walls. Providing first insights on the complexity of indoor signal propagation, this approach tackles furniture effects in an elementary way. The influence of furniture for predicting wireless signals is further examined empirically in [p127]. Wireless signal strength is measured at 24 different positions in an either completely empty, furnished or both furnished and crowded classroom. The results are shown in Table 3.2. Evaluations have shown that the difference in signal quality between an empty and a furnished classroom without persons inside is much larger compared to the difference between a furnished, but deserted and a furnished and crowded classroom. This highlights the importance of considering such obstacles for understanding communication properties of wireless networks.

Similar to providing site-specific attenuation coefficients in urban scenarios depending on the building density, other research integrates furniture into empirical models in the same way [p16]. By providing a coverage index, the average furniture density per room is incorporated into the attenuation computation. Nevertheless, due to the unspecific general index, this way of considering furnishing does not allow to draw any conclusions about the influence of particular objects on the signal propagation. Here the weakness of the empirical models again becomes apparent in return for a simplified calculation, only providing a coarse depiction of furniture. However, to enable further analysis beyond the pure consideration in predicting expected signal strengths, a detailed geometrical representation and precise positioning of each object within the modelled environment would be required.

In the context of wireless networks, static obstacles are a fundamental part of the environment. As they have the most significant impact on the network behaviour, the initial positioning of network components is carried out on their determining basis. Therefore, static obstacles should be thoroughly investigated during the planning phase and the device positioning adjusted accordingly. However, due to a rather rare relocation, they have the highest predictability when analysing temporal fluctuations. Thus, for a detection of behavioural changes during operation they are of minor relevance.

Semi-dynamic Objects

Turning now to the category of semi-dynamic objects which are expected to regularly undergo movements. Taking into account the capability of objects to alternate between different states and positions implies a potential influence on network links in their vicinity upon transition which has to be investigated. There exists several research on the effect of typical semi-dynamic indoor obstacles, such as doors and windows [p90, p84, p160, p8] as well as elevators [p100, p103].

Using a geometry-based modelling approach, the effect of typical indoor obstacles is simulated at a larger scale for entire buildings in [p160]. A three-dimensional ray tracing model is utilised to create WiFi signal predictions involving a building database to obtain the obstacle geometry. Coming back to static objects once again, it is investigated, how missing objects in the database, such as walls or furniture, affect the accuracy of the used models as well as the influence of adjacent buildings. Furthermore, this work also examines the expected effect of changing door states on the signal quality by changing the door state in the virtual representation of the environment and performing a prediction afterwards, observing slow fading effects on the links in the vicinity. However, these changes were only investigated in a simulated (incomplete) environment.

Verifying these insights in reality, a follow-up research examines the effects of door movements by conducting measurements in a small scenario consisting of two adjacent rooms [p8]. The rooms are separated by a concrete wall and each room has one door leading to an outside corridor without any direct connection between the two rooms. With changing transmitter and receiver positions, the effects of the door states on the link quality were recorded and analysed. Confirming the observations on the significance of changing door states on the signal strength in simulation from [p160], the insights were then transferred to the model by introducing an empirical parameter reflecting the door's influence and thus improving the prediction accuracy.

The significance of semi-dynamic elements has been identified and confirmed in simulation and reality, providing important insights on possible influences on a network during its operation. Nevertheless, these rather general investigations are only a first step since they mainly focus on signal changes and predictions on link level and also not consider such effects during regular operation and on a network-wide scale.

Mobile Objects

Focusing on the last category of obstacles, mobile ones have the highest potential of temporarily affecting network properties in a considerable way. As these are able to move to any position within the environment and therefore might influence the network in arbitrary and unpredictable ways, it is difficult to entirely asses their actual impact on a specific network without performing comprehensive experimental campaigns. The latter, however, is often not feasible in larger scenarios. Thus, the general properties of mobile objects should be explored thoroughly to estimate risks beforehand and consider them during network planning in conjunction with an observation during operation.

Especially in indoor and public WiFi scenarios, but also in other wireless applications, humans are often located in the network environment and thus are a relevant aspect with respect to the network behaviour. As the human body significantly attenuates wireless signals, understanding and modelling such impact allows to increase localisation and signal prediction accuracy. The attenuation of the signal strength between a WiFi access point and a held mobile device in dependency to the orientation of the near human body is investigated more closely in [p46]. The results are then used to create an attenuation model for the human body eliminating the need to collect fingerprint measurements for multiple orientations at the same position and therefore reducing the amount of necessary data to be gathered. Focusing rather on the distance than on the orientation to a mobile device, the influence of persons on links between access point and mobile devices in WiFi networks is examined in [p10]. Based on the observations, a *people presence influence distance* is defined, within which a human body significantly influences the signal strength, diminishing gradually with an increasing distance to the device.

Obtaining general knowledge as done in the previous two examples is an important first step addressing the numerous possibilities of mobile objects to influence the network behaviour in rather unpredictable ways. The application of this knowledge, integrated into models and other analysis components, can serve as basis for more extensive analyses, gaining further knowledge, particularly with respect to the network in operation. This comprises, e.g., localising and identifying observed events with humans involved or enabling new applications to detect human features [p78, p4] through the appropriate analysis of RSSI measurements.

Non-solid Influences Aside from interfering moving objects, also environmental conditions affect the signal quality which are not attributable to solid matter. Depending on the the used hardware, temperature and humidity can have a measurable impact on radio chips used in typical sensor nodes. With increasing temperatures above 30 $°C$ and humidities of more than 42 %, respectively, the reception capabilities decrease, producing more bit errors and in the worst case even turning an otherwise reliable link into a non-existent one [p101]. Especially when planning networks in scenarios with fluctuating and extreme weather conditions, these effects have to be taken into account as well. Fortunately, these conditions can be easily measured using respective sensors. In typical indoor scenarios, however, such effects rather have minor relevance as the conditions are typically stable.

Discussion

The significant impact of moving obstacles on the link quality has been proven in virtual and real scenarios providing valuable insights for understanding the network behaviour. Nevertheless, utilising this general knowledge is only one advanced part of understanding the causes of observed and initially unknown events in an operating network. The first preceding step is the detection of actual changes, followed by a spatial determination of these. Once this is done, an analysis can be performed, applying knowledge of the obstacles and their properties to determine the most probable reason.

In the previously mentioned research, the comparison of the observations with the resulting conclusions was performed manually afterwards. In this case, the presented approaches are limited to providing prediction results and not capable of correlating observations to possible causes. Requiring a subsequent manual analysis on the results aggravates an extensive application in other scenarios. Therefore, a comprehensive approach should also be able to identify changes and relate them to possible causes in order to provide insights with as little user intervention as possible.

However, identifying the exact reason of a network change in the final step is difficult since the determined location of occurrence still has to be associated with other information, such as a map containing local characteristics. As a consequence, a model of the network environment is necessary which integrates descriptions of the obstacles to be identified as potential causes in order to being able to relate them to any observations.

To the best of the author's knowledge, however, specific models that describe how wireless waves exactly propagate in the case of typical objects that populate indoor environments and exploit this information to understand the interplay of objects and network, are lacking. With this respect, the work presented in Chapter 6 provides a first step towards building a map of events happening in the environment against corresponding observed fingerprints, thus paving the way for further advancements in the modelling of environments that change over time.

3.4 Network Optimisation

Acquiring and utilising knowledge of a network ultimately enables developing elaborate strategies to optimise the behaviour towards specified goals. Self-evidently, the more complete the picture of the environment and its influencing factors is, the better optimisation measures can be implemented considering these aspects, yielding more persistent improvements. Conceptually two optimisation approaches exist, either modifying the software components to adapt the communication behaviour of a device to the characteristics of the environment or to relocate the physical position of a device and therefore locally changing the properties of the surrounding. Examples for both approaches are given in this section. Since these approaches are orthogonal, a combination of both is possible and entails the most potential for achieving results if one approach alone would not be sufficient.

3.4.1 Software Tuning

For optimising wireless networks, one solution is to perform adjustments to the software running on the devices. This includes modifications to parameters of underlying communication protocols, e.g., changing various thresholds to adjust the operational behaviour, or of hardware configurations, e.g., controlling the transmission power. The resulting effects on the network behaviour are then used to steer the relevant metrics towards the desired direction.

Beginning with optimisations on link level, [p51] proposes an approach, locally executable on a device, to detect link anomalies using a Bayes threshold on the RSSI value and initiate individual transmission power level adjustments for compensation. Using this type of local optimisation, evaluated by and regulated on the device itself, introduces the beneficial capability of self-correction in the preservation of links and has the advantage of not requiring any external coordination. Thus additional communication costs are spared which would otherwise be required for synchronisation. However, individual adaptations to detected and locally restricted changes might lead to cascaded adaptations on adjacent devices taking some time before the network has stabilised. Furthermore, the limited overview of each device spatially restricted to adjacent devices in communication range aggravates a realisation of potentially more effective network-wide optimisations. On the other hand, the latter requires a complete overview of the network and its current performance, which is not always available. Hence, device-local approaches offer more flexible deployment options and thus provide a reasonable complement to globally coordinated optimisation approaches, presented next.

The medium access control (MAC) layer, the second lowest layer of the network protocol stack, regulates the interaction of the device hardware with the transmission medium. MAC protocols ensure that multiple devices have an organised access to the common medium which cannot be used simultaneously, minimising the risks of data loss. The chosen parameters of such protocols impact the performance of the applications and thus are chosen to optimise certain metrics. However, changes in the network behaviour might require adaptations to these parameters to preserve the performance. Focusing on the optimisation of such parameters for sensor network applications, [p175] provides a framework for automated adaptation during runtime. Based on a model of the network stack and real-time information of the system, the approach continuously monitors the network performance. Once the monitoring detects a degradation of the performance below the specified requirements, an adaptation is triggered. Depending on the system dynamics, it adjusts different MAC protocol parameters, i.e., the time for and between radio channel checks as well as the maximum number of retransmissions, to restore the performance. The optimal trade-off solution between the considered performance metrics, lifetime, latency and reliability is determined mathematically by solving an optimisation problem. A different approach of finding the best parameters for a given scenario is investigated in [p145]. Network simulations are utilised to discover the optimal application configuration by collecting connectivity data from the deployed application and configuring the simulation environment to match the operational one. Running a multitude of simulations with different protocol parameters, the ones with the most promising outcome regarding observed performance metrics are chosen to applied in reality.

Mainly targeting different protocols and scenarios, similar approaches to modify MAC parameters are also pursued in other research. In [p106], an adaptation framework is proposed for specifying algorithm and application requirements for WSNs using the TinyOS operating system [p94]. Having a set of pre-evaluated configurations and their impact on several performance metrics available, such as power consumption, latency or delivery ratio, the framework then decides on the configuration best matching the required behaviour. Similarly, the packet length can be adjusted depending on the channel stability [p42]. Based on user requirements and a set of models for environment, hardware and protocols, [p115] automates the parametrisation of IoT protocols. By employing mathematical optimisation techniques, this approach selects a protocol configuration depending on the environmental conditions to influence single network properties.

Aforementioned approaches analyse a network in its entirety in order to perform more effective optimisations on a network-wide level requiring a centralised coordination of optimisation measures. The benefit of such holistic approaches, though, comes at a cost. A network-wide coordination includes exchanging messages for constantly measuring performance metrics and sending adaptation commands, entailing constant overhead. To moreover enable adaptations on the network layers, implementations often require individual modifications or additions to the code running on the devices in order to gain access to certain data or enable modifications during runtime. This implies individual solutions depending on the hardware and protocols used and aggravates a widespread application.

As shown in research, changes made to the software affecting the behaviour of the executing devices constitute a beneficial approach to counteract temporal variances in network performance, especially if infrastructural changes are undesirable or infeasible. However, such changes are not able to exert any direct influence on the surrounding environment of the devices, only to mitigate experienced issues. Indeed, software tuning can be sufficient in many cases, although the effect of such changes is limited. Eventually, physical adjustments of network components become necessary due to long-term changes in the environment which severely affect the performance and as a consequence question the position of certain devices.

3.4.2 Physical Adjustments

In this section, optimisation approaches are based on precise descriptions of network environments as well as accurate estimations of communication, enabling physical modifications of the network structure by adding, removing or relocating devices. Requiring aforementioned larger amount of details for implementation, the potential benefits of modifying the network structure are likewise greater. Furthermore, besides pure optimisation, approaches focusing on physical adjustments open up possibilities for new applications, such as designing entire networks while considering and satisfying given requirements or assisting users during device deployment.

An incremental approach for the initial indoor deployment of IoT devices is presented in [p41], with the goal of creating a fully connected network. Requiring a model of the environment to perform the planning, a user can interactively mark cells on a two-dimensional grid containing walls to define the outline of the scenario, place existing devices and define areas in which relay nodes might be deployed to interconnect the former. The deployment is then performed within three phases. In a first offline phase, relay nodes positions are determined within the specified areas to connect all existing nodes on paths leading through the building, ignoring possible paths outside. The second online phase involves the physical placement of the identified relay node positions and a subsequent validation of existing connectivity between these relays. As solution, disconnected relays are moved closer to the other nodes one at a time until a connection is established, eventual emerging gaps are again filled with further relay nodes until all deployed nodes are connected. Finally in the last phase, the connectivity is optimised by eliminating weak links and redundant relay nodes.

Offering a large advantage for users to design entire networks, considering and interconnecting desired device positions, this approach, however, aims at creating connected networks with sufficiently strong links above a given RSSI threshold rather than focusing on individual network performance metrics. This may result in subsequent adjustments for fulfilling potential application requirements. Furthermore, other elements of the environment except devices, walls and free space are not considered. These are of minor relevance in this case as the path losses are computed using a modified version of the empirical log-distance path loss model [p123], additionally considering the number of walls in the direct path between sender and receiver. Thus, physical effects like reflection and diffraction are not considered at all when determining suitable positions for relay nodes.

Focusing on improving already deployed networks, a coverage analysis of WiFi access points is performed in [p79] using the Motley Keenan multi-wall model [p81], distinguishing three different types of materials to find the best positions for a varying number of available devices, maximising the channel capacity in an area of interest. This approach optimises the coverage of sending devices which are independent of each other, but it does not consider the interaction among the devices itself as it is required for optimising, e.g., application performance metrics.

Dealing with an expansion of the network range as well, the effective placement of relay nodes to circumvent large obstacles of arbitrary shape, such as rock formations in outdoor scenarios, is investigated in [p158]. In two-dimensional space, an approach for creating a minimal set of relay nodes bypassing the obstacle and connecting the network is proposed. Considering the network as a graph with nodes as vertices and links between nodes as edges, a minimum Steiner tree [p67] is computed, describing a graph with the shortest possible connection between all existing vertices. In order to circumvent existing obstacles, additional vertices can be considered to be placed as junction points.

Similar goals are considered in [p170], which uses simplistic descriptions of the deployment scenario to estimate the minimum number of relays necessary to build a connected network. By using a weight-based approach determined via path-loss predictions of a multi-wall model [p36], weak links are identified to be bypassed by relay nodes. After determining and evaluating candidate positions for such links, a minimisation step is performed to merge redundant relay positions.

Pursuing the same idea as the approach presented in this thesis, improving a deployed network by utilising models for signal path prediction, the aforementioned work focuses on improving reliability of weakly connected and disconnected networks, but does not consider other metrics on a network or application level, similar to the previously mentioned work [p41]. Hence, it covers only a subset of possible performance optimisations and moreover does not verify its functionality in a real deployment, only in simulation.

Following this general idea and going one step further by incorporating reality into the process as well, Mo$\rho\phi\varepsilon\upsilon\varsigma$ [c7] demonstrates the capabilities of network simulations creating results that correspond to reality in order to evaluate them finding potential device positions for optimisation. In contrast to the approach mentioned in the previous section [p145], this work focuses on the effects of node placements instead of protocol parameters on the resulting network performance. Bottlenecks of a given network are first identified by simulating the network behaviour based on obtained connectivity data. By adding, removing or relocating simulated devices at certain positions in the virtual network, the effect on application performance metrics, such as latency, throughput or radio on-time are then evaluated. Promising configurations with outcomes showing improvements are then transferred to reality. There, the suggested configurations were able to improve the targeted application metrics significantly as well, confirming the simulated observations.

Optimisations attributable to physical interventions on the network structure demonstrate their capabilities of improving various aspects of a network. Supplemented by simulations to cover and manage the countless number of possible options for modifications, both provide a powerful combination, assisting users in pursuing their optimisation tasks in a convenient way obviating the need for time consuming trial and error approaches. Exploiting the potential and removing the limitations, Μορφευς is subject to, this thesis presents an approach independent of specific requirements and application metrics (Chapter 7), supporting general wireless system optimisation goals while taking details of the environment into account.

4 Network Analysis

Thanks to the ease and flexibility of deployment, a forthcoming future is envisioned where the Internet of Things and cyber-physical systems will pervade everyday life with a multitude of low-power wireless devices. The scale at which the technology is expected to enter environments will quickly make it impractical to both operate and keep track of each individual device. Furthermore, new infrastructures will have to be quickly deployed as well as merged with existing ones in a continuously evolving ecosystem. The maintenance of this technology becomes then a crucial challenge that needs to be addressed in order to make the vision practical.

One of the key properties of a wireless infrastructure is the location of each device. In fact, from this information depends not only the ability to make sense of the observations performed by the sensors, but also the capacity of understanding system behaviour and resolving failures. However, keeping a record of the device positions in evolving and dense networks made of wireless devices is problematic, even more with an increasing system scale. As a result, an effective monitoring solution to know where devices are deployed should rely on as few information as possible.

Analysing information on the network connectivity, the (non-)existence of links as well as the signal strengths of those existent allow to draw first conclusions about the network properties and reveal possible deficiencies. Fusing connectivity information with other additional information, such as device positions, floor plans or obstacle properties yield further valuable insights which on their part can serve as foundation for optimisations. This chapter introduces an approach for localising wireless devices in indoor environments based on an adapted force-directed algorithm, originally used for visualising graphs.

4.1 Challenges

Realising a localisation approach which is able to incorporate different information to achieve more accurate results entails some challenges regarding the handling of the availability as well as the quality of information, explained in the following.

Reconciling Individual Conditions

As already stated in the previous chapter, there does not exist a universal solution for indoor localisation, as the conditions of each scenario are too different. The heterogeneity of devices as well as continually changing infrastructures aggravate making assumptions on information, obtainable from and about a network. Thus a general solution needs to regard and comprise different amounts of available information to dynamically adapt to a given scenario and optimally exploit its features. Nevertheless, it is necessary to agree on a common set of basic information, each scenario should be able to provide as a minimum requirement. Beyond that, however, there should be as few other requirements as possible to preserve applicability.

Addressing this issue, the localisation approach presented in this chapter is based on the received signal strength which is available on most of the wireless devices. In a multi-stage process, the algorithm is able to include a flexible amount of information, beginning with the very basic connectivity data (RSSI), extendable with optional information of a floor plan and anchor node positions, to increase the quality of localisation.

Ambiguity of Sparse Information

Allowing a solution to work with a flexible amount of details to preserve applicability requires to examine the edge cases. As a consequence of (exclusively) using the most common and basic information, the results of analysis might be inaccurate or subject to certain ambiguities. Taking the RSSI, which combines many influencing aspects of hardware and environment in one single number, the informative value of such is limited. Considered individually, it is difficult to derive meaningful conclusions on the reasons of having observed a specific value, e.g., if it is was measured due to distance losses, obstructions on the path, a defect radio chip or for any other plausible reasons. Thus, more context is required in order to resolve such ambiguities. This can either be achieved by relating more information of the same type or by incorporating other types.

Given the unavailability of the latter, the presented localisation approach which utilises a force-directed algorithm, is able to implicitly relate all of the measured RSSI values due to the underlying physical principles. Based on the RSSI value, a link exerts an influence on the two devices it relates to, but also on other devices in the direct vicinity. In this way, individual local inaccuracies are corrected by the entirety of data. However, although mitigating the problem, certain inaccuracies are still inevitable and have to be accepted in such cases in which further details are lacking.

4.2 Approach

The idea of modelling a network as energy system with forces depending on connectivity information has already been exploited for localisation in some approaches. [p64] utilises a mass-spring relaxation algorithm to localise nodes in a self-organising robotic network and improve the floor plan maps, generated by the robotic sensors while traversing the environment and discovering devices. Sensor drifts of the robots resulting in imprecise and displaced mappings could be corrected by the force-directed algorithm as well as measurements from multiple robots aligned. A similar approach is used in [p92], but it assumes the pair-wise distances between nodes to be known beforehand. These solutions define decentralised algorithms able to integrate distance information in force-directed graphs and demonstrate the benefits of such algorithms within the given scenarios quite clearly. In this work, however, it is aimed at experimenting with various amounts of information to understand the impact on the resulting accuracy. Moreover, in the analysis part it is refrained from changing the existing system infrastructure, exploiting only available data.

Departing from the body of work done on device localisation based on wireless measurements and dedicated infrastructures (Section 3.1), this chapter presents FLoW, a Force-directed Localisation of Wireless Devices algorithm able to work with different amounts of provided information [a4]. FLoW starts by unfolding a graph where vertices repel each other and edges build attraction based on connectivity information. This topology without physical awareness can then be positioned inside a map of the scenario, if available, and expanded to best fit the specific environment shape. In case in which the physical position of some node is provided, this knowledge can be exploited to resolve ambiguities. In exploring each of these steps, this work (1) defines techniques able to exploit information to the extent available in order to practically infer the position of wireless devices, (2) analyses the impact of information on the achievable accuracy and (3) demonstrates to what degree localisation can be achieved in real systems without dedicated infrastructures and protocols. Exploring the trade-offs between localisation accuracy and the amount of information about the system and the environment made available, the investigation focuses on scenarios, whose complex structures with strong impact on low-power wireless communication affects the ability to match simple measurements with physical distances. By realising the aforementioned, FLoW makes the following contributions:

Position Inference with Basic Information and Incremental Refinement Using the basic force-directed graph mechanism with connectivity information, FLoW enables to obtain a first estimation of device positions without any further required knowledge about network or environment. Thus it can help to gain a basic understanding of the network topology and the spatial arrangement of nodes in the environment without much effort. However, as the accuracy of the output is limited when using coarse basic data, further information can be added as needed. FLoW exploits information from different sources, network and environment, in a flexible process to achieve the most accurate results with the given data, individual to each scenario.

Localisation Independent of Dedicated Hardware While certain approaches, e.g., AoA or ToA, require the availability of specific equipment or software features, FLoW enables a localisation without any constraints on used hardware. The only requirement is the possibility to measure the signal strength, which each radio is capable of, in order to gather the required connectivity information of a network. Since other exploited information used in further computation steps refers exclusively to the environment, the localisation of FLoW is applicable to a wide range of wirelessly communicating networks, as it can be employed independently of the used devices.

To obtain device position estimations and exploit different amounts of available information, FLoW uses a multi-stage incorporation process, successively adding data and thus refining the accuracy. This process is shown in Figure 4.1, starting with the basic RSSI values as input, producing a relative node displacement utilising the adapted force-directed algorithm, described in Section 4.3.1. The output of this step, however, lacks any reference points and cannot be transferred to actual positions or distances. By adding a floor plan outline in the second refinement step, the device positions are adjusted to fit the given shape. Maintaining the relative displacements of the first step, this is achieved by transforming the whole graph using simple operations, i.e., translation, rotation, mirroring and scaling. As a result, absolute device coordinates are obtained which can be refined by specifying positions of anchor nodes in the third step, producing more accurate results. These stages are described in detail in the following sections, beginning with a general description of the utilised force-directed algorithm, followed by the steps taken to estimate device positions and finally refining them with the usage of anchor node positions.

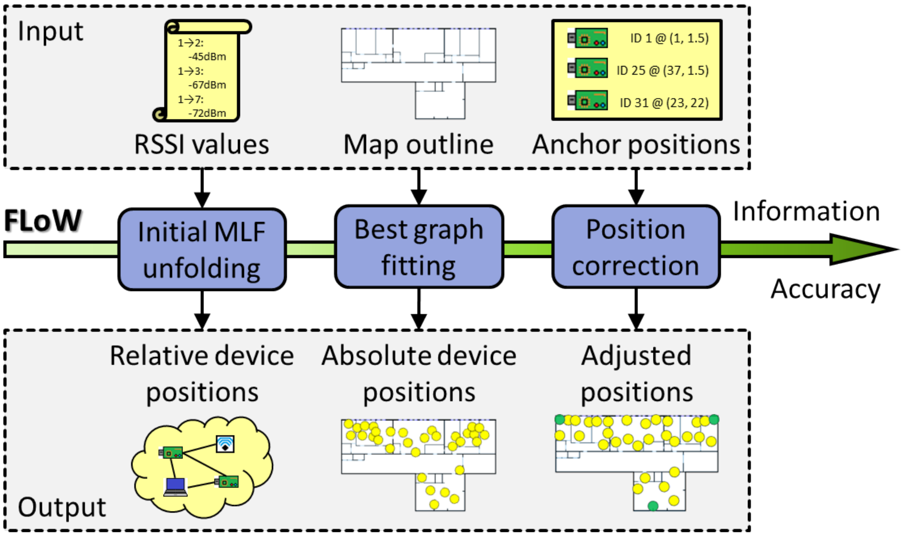


Figure 4.1: FLoW *multi-stage output refinement process using variable amount of information. The first step uses a multilevel force-directed (MLF) algorithm [p162] to create the initial graph.*

By applying FLoW in real-world deployments as well as in challenging artificial setups (Section 8.3), it is investigated how the approach is able to exploit available domain knowledge to estimate the position of devices and to increase the accuracy when successively provided with additional information. Furthermore, restrictions of the approach resulting from the usage of limited information are revealed and analysed.

4.3 Implementation

In this section, the three output stages of FLoW are described in detail. It is described how force-directed algorithms help to localise devices in unknown deployments just by providing connectivity information and an outline of the scenario. It is also discussed how further system knowledge can be used to resolve ambiguities.

4.3.1 Force-directed Algorithms

Force-directed algorithms [p18, p86] provide an effective way of arranging graphs neatly by utilising physical laws to determine the positions of the vertices. To realise a proper dispersion, vertices are treated as particles that repel each other. Edges, instead, imply an attractive force between the corresponding vertices, preventing them from drifting apart infinitely. Combining both properties result in a graph which, once an equilibrium is reached, occupies a given space with vertices whose proximity depends on the interconnecting edges.

This property of force-directed algorithms makes it very suitable for visualising wireless networks. Nodes can be represented as vertices and connectivity information as edges of the graph. The only constraint is that the network graph is fully connected, without isolated nodes or network partitions. Otherwise these partitions would repel each other and impede a proper graph expansion within given boundaries.

Going into more detail, the general basic two-dimensional force-directed algorithm used and extended by FLoW is depicted by Algorithm 4.1 and works as follows. Initially, vertices are placed randomly in a given drawing area. Then the following steps are repeated several times (rounds) until an equilibrium is established, i.e., vertex displacements have to be below a given threshold thr. For each pair of vertices, their vectorial repulsive forces are computed according to the variable function $f_r(x)$ which is based on their absolute distance x. In case of existing edges between two vertices, opposing attractive forces are computed as well using $f_a(x)$. If modelling heterogeneous vertices with different properties, a weight factor w can be additionally considered, individually influencing forces on each vertex. The sum of all repulsive and attractive forces yields the displacement of a vertex which is applied either after each individual vertex computation or after all vertices have been processed, depending on the implementation. Achieving a controllable graph expansion and constraining the occupied space in dependency of the total number of vertices, an equilibrium length k is introduced. This variable determines the distance at which repulsive and attractive forces of connected vertices are equal, cancelling out each other. In complex graphs, certain tensions between connected vertices are inevitable, causing them to move back and forth between each computation step. To deal with such occurrences and nevertheless achieve a convergence, a damping factor t, called temperature, is furthermore introduced, limiting the maximum allowed vertex displacement in each step. Starting with large values allowing strong movements at the beginning of the algorithm, the temperature is successively reduced in further steps by applying a cooling function until converging. By customising the aforementioned algorithm parameter, described in the following Section 4.3.2, the output of force-directed algorithms can be adapted to certain requirements.

The reference approach [p162] uses a multilevel force-directed (MLF) algorithm to optimally expand a connected graph in a two-dimensional space. To avoid local force minima and thus possible incorrect unfolding steps of a graph, the initial set of vertices is reduced stepwise by conflating each

Algorithm 4.1 Basic force-directed algorithm, referring to [p162].

```
function forceDirectedPlacement($Vertices, Edges, t$)
    $converged := false$;
    while ($\neg$ $converged$) do
        $converged := true$;
        for ($v \in Vertices$) do
            $\Theta := 0$;
            for ($u \in Vertices, u \neq v$) do
                $\Delta := u.position - v.position$;
                $\Theta := \Theta + f_r(||\Delta||)$;
                if ($e_{u \to v} \in Edges$) then
                    $\Theta := \Theta + f_a(||\Delta||)$;
            $\Delta := (\Theta/||\Theta||) * \min(t, ||\Theta||)$;
            $v.position := v.position + \Delta$;
            if ($\Delta > thr$) then
                $converged := false$;
        $t := f_c(t)$;
```

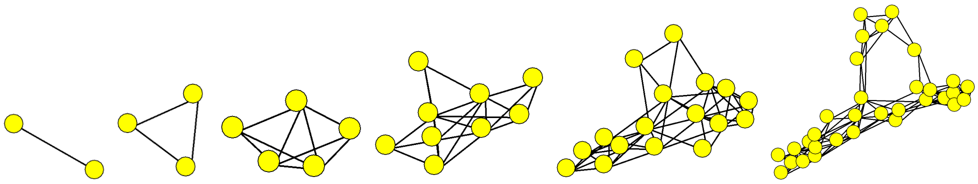


Figure 4.2: *Example of the unfolding procedure for a graph based on traces gathered from the* WSN Testbed *(Section 8.3.1).*

two most spatially correlated vertices into a new single vertex. Hence, the number of vertices is halved in each step until only two vertices remain, serving as starting point for the algorithm. Once these two vertices reach an equilibrium of forces, the conflation process is reversed by expanding the compressed nodes again at their current position and repeating the force computations until all nodes have been restored and the equilibrium established, as illustrated in Figure 4.2. This approach generates graphs significantly closer to the actual topology shape compared to a random initial placement and unfolding. The conflation process, adapted to FLoW, is described in Section 4.3.3.

4.3.2 Algorithm Parameter Adaptation

Starting from the algorithm described in [p162], functions, thresholds and constants used in the implementation of FLoW have been mostly set according to the recommended for achieving the best graph unfolding and hence positioning accuracy.

Table 4.1: *Overview of parameters and chosen values adapting the MLF algorithm to* FLoW.

Parameter	*Description*	*Value*
thr	Convergence Threshold	0.01
t_0	Initial Temperature	k
$f_c(t)$	Cooling Function (Temperature in Round i)	$0.9t_{i-1}$
$f_r(x)$	Repelling Function	$-0.2k^2/x$
$f_a(x,w)$	Attraction Function	x^2w/k
k_0	Initial Unfolding Equilibrium Distance	$\min(mapHeight, mapWidth)$
k_j	Equilibrium Distance in Unfolding Step j	$\sqrt{4/7}\ k_{j-1}$

Table 4.1 summarises these parameters and contains the corresponding values chosen. The convergence threshold is set to 0.01 metres allowing only minor vertex movements in each round before stopping the algorithm. A continuous decrease of temperature by a factor of 0.9 leads to a fast drop in the first steps followed by a smooth decay in later steps until finally converging after a finite number of rounds. Repulsive forces between each pair of vertices are computed according to $f_r(x) = -0.2 * k^2/x$, declining with increasing absolute distance x between two vertices. Attractive forces in case of an existing edge on the other hand increase at a quadratic scale using $f_a(x,w) = x^2 * w/k$. As the purpose of this algorithm is to determine distances between nodes in dependency of measured RSSI values, an additional weight factor w has been added to the attraction function incorporating this feature. Details on the weight determination follow in Section 4.3.3.

During MLF graph unfolding phase, the equilibrium distance k is successively decreased in order to counterbalance the increased space demand. According to [p162] a reduction by factor $\sqrt{4/7}$ in each unfolding step shows the best results. Furthermore, displacements are applied directly after force computations have been completed for each vertex.

4.3.3 Position Estimation

The solution just presented assumes that the graph can freely unfold, thus resulting in range-free positioning of nodes. The goal of this work is, however, to localise devices in a specific environment within its boundaries based on connectivity information. For this reason a limitation of the graph expansion to the outer boundaries of a given floor plan is required, resulting in constrained, range-based localisation. To achieve this, different possible extensions of the basic algorithm depending on the available information of the deployment scenario are explored, as described in the following sections.

For the internal handling of graph structures and force-directed computations, a corresponding data representation has been defined, listed by Algorithm 4.2. Since the MLF approach requires to deal with changing graph structures, a data type has been defined in order to describe multiple graph layouts at different levels of conflation. Additionally, a scaling factor is stored, which becomes relevant in the finalisation of the second step when fitting the graph to the floor plan. Apart from information regarding identification and position, each vertex has two optional fields describing the vertices it consists of to allow for an efficient initial unfolding procedure. Lastly, besides the vertices it connects, each edge requires to store the attraction forces both vertices exert on each other.

Algorithm 4.2 Internal data structures for graph handling.

```
struct Graph {
  List $Vertices, Edges$;
  Float $scaling$;
}

struct Vertex {
  Integer $id$;
  Vertex $subV_1, subV_2$;
  Position $pos$;
}

struct Edge {
  Vertex $v_1, v_2$;
  Float $force$;
}
```

Using Wireless Connectivity as Graph Edges

Utilising the adapted MLF algorithm, measurements of the RSSI as obtained by the nodes communicating in the deployed system are employed. While in theory any connectivity metric could be used, e.g., link quality indicator or the plain existence of a link, the quality of the output topology depends on how well these values correlate with the physical distance between the nodes. By using the measured RSSI, a first estimate of distance based on the free space path loss model and use of the inverse formula as edge attraction weight is built. Needless to say, this works accurately only in clear line of sight conditions, unlikely in indoor setups. Nevertheless, to keep the approach simple and practical at the cost of a reduced accuracy, it is refrained from requiring detailed information about physical obstacles hindering the wireless signal propagation.

In addition, given that according to the path loss model the signal degrades logarithmically with the distance, links with a weak signal are more likely to introduce errors, even more in the presence of obstructing elements nearby. To mitigate this, weaker links (< -90 dBm) are not considered when generating edges, favouring stronger and meaningful RSSI values. Furthermore, the signal strengths of both link directions are averaged, ensuring a symmetric force relation between two connected vertices, fostering faster convergence. As a consequence, unidirectional links are discarded as well. Moreover, a minimum bound on the attraction forces for the lowest considered RSSI values for an estimated distance higher than 20 metres is set. However, this approach benefits by the fact that the distance between two vertices is not only influenced by their attraction through edges, but also by the repelling forces of close vertices, providing a certain degree of self-correction, especially at higher node densities.

Configuring FLoW for generating the initial graph, the given RSSI values are processed first as summarised by Algorithm 4.3. Iterating through the set of all existing links L between the nodes N of the network, the previously mentioned link filtering and averaging of the link path losses (PL) are applied first. The path loss is determined by subtracting the transmission power tx_{pwr} from the measured RSSI value. The free space distance d of a given path loss PL is then obtained by rearranging Equation 2.6 to provide the distance in dependency of the signal strength:

$$d = \frac{10^{0.05PL+7.3775}}{f} \tag{4.1}$$

The symmetric attraction weight w_{ij}, used in Table 4.1, to determine the proximity of two connected vertices i and j is then given by:

$$w_{ij} = \max(0.05, 1/d_{ij}) \tag{4.2}$$

Algorithm 4.3 RSSI-based initial edge attraction weight computation.

```
function computeEdgeAttractionWeights($L, N, tx_{pwr}, freq$)
    $WeightMap :=$ new Map()
    for ($l_{n_1 \to n_2} \in L, n_1 \in N, n_2 \in N, n_1.id < n_2.id$) do
        if ($l_{n_2 \to n_1} \notin L$) then
            continue with next $l$;
        if (($l_{n_1 \to n_2} < -90dBm$) $\vee$ ($l_{n_2 \to n_1} < -90dBm$)) then
            continue with next $l$;
        $avgPL := ((l_{n_1 \to n_2} - tx_{pwr}) + (l_{n_2 \to n_1} - tx_{pwr}))/2$;
        $d_{n_1,n_2} := 10^{0.05*avgPL+7.3775}/freq$;    (Equation 4.1)
        $w_{n_1,n_2} := \max(\frac{1}{20}, \frac{1}{d_{n_1 \leftrightarrow n_2}})$;    (Equation 4.2)
        $WeightMap.put(l_{n_1 \to n_2}, w_{n_1,n_2})$;
        $WeightMap.put(l_{n_2 \to n_1}, w_{n_1,n_2})$;
    return $WeightMap$;
```

Based on the weights, the MLF graph reduction is performed afterwards, as described by Algorithm 4.4. Starting with the initial graph g_0, containing all nodes and edges of the real network, the number of vertices is halved in each step until a reduction down to two vertices is achieved. During the conflation process, the two vertices sharing the strongest links and consequently have highest edge weight are merged to a new vertex first, maintaining edges to all vertices each of the two initial vertices were connected to. The weights of the edges are either kept or averaged if both merged vertices shared an edge to another vertex. The remaining graph generation procedure corresponds to the MLF process, alternating between equilibrium computation and graph unfolding until the complete graph has been restored.

Positioning of the Graph in the Deployment Map

Based on the RSSI information, filtered as described in the previous step, the range-free graph expansion results in a first estimate of the network topology. However, two flaws prevent the use of this graph for localisation. First, the distances between vertices are only relative and second, the graph can be arbitrarily rotated, scaled and mirrored to fit the actual scenario. In order to solve these issues, reference points to absolute coordinates are required. Due to the expansive nature of force-directed graphs, it has been decided to provide these points by setting bounds to the algorithm based on the outline of the floor plan, in which the network is located. In this way it becomes possible to find a suitable transformation of the graph as well as to obtain absolute positions of the nodes within the floor plan coordinate

Algorithm 4.4 Graph conflation process.

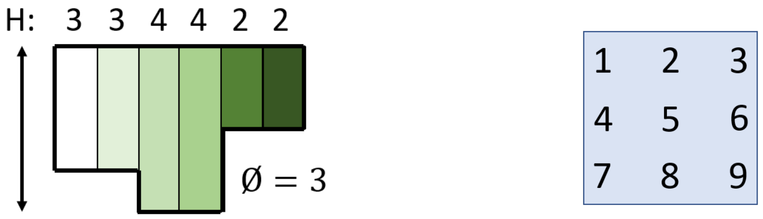
```
function ConflateGraph($g_0$)
    $ConflatedGraphList$ := new List();
    $ConflatedGraphList.add(g_0)$;
    $g_{curr} := g_0$;
    while ($g_{curr}.Vertices.size() > 2$) do
        $g_{prev} := g_{curr}$;
        $g_{curr}$ := new Graph();
        $toCluster$ := new List($g_{prev}.Vertices$);
        $id_{next} := 1$;
        while ($\neg$ $toCluster.isEmpty$) do
            $v_1 := toCluster.getFirst()$;
            $toCluster.remove(v_1)$;
            $v_2 := findClosestNeighbour(v_1, toCluster, g_{prev}.edges)$;
            if ($v_2 \neq null$) then
                $toCluster.remove(v_2)$;
            $v_c$ := new Vertex($id_{next}, v_1, v_2, null$);
            $g_{curr}.Vertices.add(v_c)$;
            $id_{next}$ += 1;

        for ($v_1, v_2 \in g_{curr}.Vertices, v_1 \neq v_2$) do
            $commonLinks := 0$;
            $forceSum := 0$;
            for ($e_{prev} \in g_{prev}.Edges$) do
                if ($((e_{prev}.v_1 = v_1.subV_1) \vee (e_{prev}.v_1 = v_1.subV_2))$
                        $\wedge (((e_{prev}.v_2 = v_2.subV_1) \vee (e_{prev}.v_2 = v_2.subV_2)))$ then
                    $commonLinks$ += 1;
                    $forceSum$ += $e.force$;
            if ($commonLinks > 0$) then
                $forceMean := forceSum/commonLinks$;
                $e_c$ := new Edge($v_1, v_2, forceMean$);
                $g_{curr}.Edges.add(e_c)$;
        $ConflatedGraphList.add(g_{curr})$;
    return $ConflatedGraphList$;
```

system. In order to decide on the best matching graph transformation, all different combinations of rotation (in steps of one degree) and mirroring of the graph shape are tested with different scaling factors and displacements. The resulting possible configurations are listed in Table 4.2.

Given mirroring and rotation of the graph, the scaling factor is determined based on four different approaches: (1) fitting the total width of the floor plan, (2) fitting the total height of the floor plan, (3) using the average of both width and height scalings, (4) averaging the scalings resulting from splitting the map into vertical segments for which the height scaling is computed individually. The latter approach, depicted in Figure 4.3 (a), should

Table 4.2: *List of transformation variants tested for each operation. The combination of all four operations results in a new configuration.*

Operation	Mirror	Rotation	Scaling	Translation	Total
Variants	4	360	4	9	51840



(a) Vertical segment scaling **(b)** Translation alignment positions

Figure 4.3: *Illustration of scaling and translation principles.*

prevent using scalings with an oversized graph just because the extent of the outline is large at certain areas. For displacements within the floor plan shape, the graph is aligned to nine different positions, which are shown in Figure 4.3 (b).

These configurations are rated based on two weighted properties: the areal coverage of the graph with respect to the floor plan and the number of vertices outside the map bounds. Thereto the outer vertices and edges of the graph form an area, which can be compared against shape and area of the actual deployment map. For this, the Jaccard coefficient [p76] is used, which measures the similarity of two given sets A and B:

$$J(A,B) := \frac{|A \cap B|}{|A \cup B|} \,. \tag{4.3}$$

A value close to 1 (or 0) denotes a high (or low) similarity between the two sets or in this case, areas. This value is then multiplied with a factor based on the number of nodes outside the map boundaries, resulting in the conformity equation:

$$\text{conformity} := J(A,B) \cdot \left(1 - \ln\left(\frac{\#n_{\text{off}}}{|N|} + a\right) - b\right) \tag{4.4}$$

where $\#n_{\text{off}}$ is the number of estimated node positions lying outside of the actual boundaries and N is defined as the set of nodes $N := \{n_1, n_2, \ldots, n_l\}$, with $|N|$ as the total number of nodes in the network. The weight of outlying nodes on the conformity value is based on an adapted logarithmic curve, penalising more the first few nodes laying outside the boundaries than the subsequent ones. Throughout tests, such curve could be fit with $a = 0.6$ and $b = 0.5$.

Comparing only the area coverage indicates how well the shapes of both areas match and therefore the likelihood that the current graph configuration has the correct orientation and scaling. However, preferably in networks with an uneven node distribution, high similarity values could be observed in cases with a significant amount of nodes lying outside of the bounding shape (the floor map). As a consequence, it seems natural to also take the outlying nodes into account in addition to the area coverage.

The complete process of graph positioning is described by Algorithm 4.5. Starting with the graph obtained from the previous step, which contains the relative positioning of the vertices, it is iterated through the different graph transformations of mirroring, rotation, scaling and translation. To avoid repeating computation steps, the graph is copied after each transformation. In order to perform scaling computation and transformation alignment correctly, in addition to the position adaptations, the bounding boxes from previous transformations have to be determined. After all transformations have been performed on the graph, the conformity value (Equation 4.4) is computed by incorporating the nodes, which are positioned outside of the floor plan (line 40) and comparing the enclosed areas of graph and map (line 44). Among all tested configurations, the one with the highest conformity value is chosen for the next step.

Fitting the Available Space

Continuing with the most conform configuration, the force-directed algorithm is applied again with an appropriate distance scaling to adjust the distribution of the nodes within the shape. In the previous step, a scaling factor has been found for the areal adaptation of the graph. However, this scaling has to be transferred to the expansion behaviour of the force-directed graph algorithm as well by regulating the temperature t_0. The adjustment procedure is described by Algorithm 4.6. The goal is to approximate the graph area produced by the algorithm to the area which was specified by the fitting process of the previous step until the deviation is less than five percent. Thereto, the scaling is successively adjusted and used as input to the force-directed algorithm. Afterwards the area of the newly produced graph is determined and the ratio between initial and current areas computed. In

Algorithm 4.5 Graph positioning on the deployment map.

```
$Mirrorings$ := {NONE, HORIZONTAL, VERTICAL, BOTH};
$Scalings$ := {WIDTH, HEIGHT, BOTH, SEGMENTS};

**function** positionGraphInDeploymentMap($g_{rel}$, $map$)
    $conf_{best} := -1$;
    $g_{best} := null$;
    **for** ($mirror \in Mirrorings$) **do**
        **for** ($rotation \in \mathbb{N}, 0 \leq rotation \leq 360$) **do**
            $g_{rotate} := g_{rel}.copy()$;
            $p_{rot,min}$ := **new** Point$(\infty, \infty)$;
            $p_{rot,max}$ := **new** Point$(-\infty, -\infty)$;
            **for** ($v \in g_{rotate}.Vertices$) **do**
                **if** (($mirror$ = VERTICAL) $\vee$ ($mirror$ = BOTH)) **then**
                    $v.pos.x -= 2 * v.pos.x$;
                **if** (($mirror$ = HORIZONTAL) $\vee$ ($mirror$ = BOTH)) **then**
                    $v.pos.y -= 2 * v.pos.y$;
                $radians := toRadians(rotation)$;
                $v.pos.x := v.pos.x * \cos(radians) - v.pos.y * \sin(radians)$;
                $v.pos.y := v.pos.y * \cos(radians) + v.pos.x * \sin(radians)$;
                $p_{rot,min}.set(\min(p_{rot,min}.x, v.pos.x), \min(p_{rot,min}.y, v.pos.y))$;
                $p_{rot,max}.set(\max(p_{rot,max}.x, v.pos.x), \max(p_{rot,max}.y, v.pos.y))$;
            **for** ($scaling \in Scalings$) **do**
                $factor := getScalingFactor(scaling, map.bounds, p_{rot,min}, p_{rot,max})$;
                $g_{scale} := g_{rotate}.copy()$;
                $g_{scale}.scaling := factor$;
                $p_{sc,min}$ := **new** Point$(\infty, \infty)$;
                $p_{sc,max}$ := **new** Point$(-\infty, -\infty)$;
                **for** ($v \in g_{scale}.Vertices$) **do**
                    $v.pos.x *= factor$;
                    $v.pos.y *= factor$;
                    $p_{sc,min}.set(\min(p_{sc,min}.x, v.pos.x), \min(p_{sc,min}.y, v.pos.y))$;
                    $p_{sc,max}.set(\max(p_{sc,max}.x, v.pos.x), \max(p_{sc,max}.y, v.pos.y))$;
                **for** ($translation \in \mathbb{N}, 0 \leq translation \leq 9$) **do**
                    $p_{trans} := getTranslationOffset(translation, map.bounds,$
                                                                      $p_{sc,min}, p_{sc,max})$;
                    $g_{trans} := g_{scale}.copy()$;
                    **for** ($v \in g_{trans}.Vertices$) **do**
                        $v.pos.x += p_{trans}.x$;
                        $v.pos.y += p_{trans}.y$;
                        $outOfBounds := 0$;
                        **if** ($\neg$ $map.bounds.contains(v.pos)$) **then**
                            $outOfBounds += 1$;
                    $area_{graph} := getAreaShape(g_{trans})$;
                    $area_{map} := getAreaShape(map)$;
                    $jaccard := \frac{|area_{graph} \cap area_{map}|}{|area_{graph} \cup area_{map}|}$;        (Equation 4.3)
                    $conf := jaccard * \left(1 - \ln\left(\frac{outOfBounds}{g_{trans}.Vertices.size()} + 0.6\right) - 0.5\right)$;
                                                                      (Equation 4.4)
                    **if** ($conf > conf_{best}$) **then**
                        $conf_{best} := conf$;
                        $g_{best} := g_{trans}$;
    **return** $g_{best}$;
```

case the new area is too large, the scaling factor is adjusted down and vice versa. To avoid a repeated alteration between increasing and decreasing the scaling factor, the adjustment steps decrease when overstepping the target ratio interval, ensuring a convergence of the algorithm.

Algorithm 4.6 Graph scaling adjustment process.

```
function ADJUSTSCALING($g, map$)
    $scalingComplete := false;$
    $acreage_{start} := getAcreage(g);$
    $acreage_{map} := getAcreage(map);$
    $adaptFactor := g.scaling;$
    $change := 0.2;$
    $lastAdjustment :=$ NONE;
    while ($\neg$ $scalingComplete$) do
        $forceDirectedPlacement(g.Vertices, g.Edges, k * adaptFactor);$
        $acreage_{curr} := getAcreage(g);$
        $ratio := acreage_{curr}/acreage_{start};$
        if ($0.95 < ratio < 1.05$) then
            $scalingComplete = true;$
        else if ($ratio < 0.95$) then
            if ($lastAdjustment =$ DOWN) then
                $change *= 0.5;$
            $adaptFactor += change;$
            $lastAdjustment :=$ UP;
        else
            if ($lastAdjustment =$ UP) then
                $change *= 0.5;$
            $adaptFactor -= change;$
            $lastAdjustment :=$ DOWN;
    return $adaptFactor;$
```

Once the adaptation factor to the temperature has been found, ensuring that the nodes will move within the given map boundaries, the force-directed algorithm is run again. This time the outer walls provided by the map serve as solid barriers to the vertices, only allowing them to move within the boundaries of the map. Vertices which were located outside of the map beforehand are relocated next to the closest vertex inside the map, ensuring all vertices being inside. As a side effect, minor deformations conflicting with the actual topology are corrected through forcefully fitting the graph into the shape. To assure a complete distribution of the vertices within the given map space, an outer wall attraction is introduced in the final step. For each wall segment, three virtual points are created, one for each end and one in the middle, providing an attraction force to their closest vertex to fill possible remaining gaps smoothening the final result. If one vertex is closest to multiple such points, it will only be attracted by the one with the least distance.

Table 4.3: *Performed map positioning computations in dependency of available anchors.*

Anchors	*Map Positioning Decision*	*Problem Addressed*
0	Max conformity	-
1	Max conformity without translation	Displacement
2	Best mirror configurations: min SSE Final configuration: max conformity	Rotation angle
≥ 3	Min SSE	Mirroring

4.3.4 Anchor Nodes

As previously indicated, topologies with an erroneous node placement can still manifest a high conformity with the floor plan, thus leading to high errors. This undesirable outcome depends on two circumstances: the symmetry of the map and the difference between the shape of the generated topology and the actual one. The former leads to ambiguities which cannot be reliably resolved by the matching algorithm without providing further information. The latter is a property of force-directed algorithms to place nodes without a common edge as far away as possible from each other within the limits of the given attraction and repulsion forces. One can think about a scenario shaped like an S; in this case the pure force-directed graph would hardly reproduce a bend in the topology (a more in depth discussion for this scenario is reported in Section 8.3.1).

To address this problem, more information about the system is necessary. In particular, if the actual position of some node is available, it can be used to anchor the graph and counteract the intrinsic limitations of the matching algorithm. Depending on the number of nodes whose location is known, it is possible to resolve the displacement within the map (with 1 anchor), determine the rotation angle (with 2 anchors) and address mirroring (with 3 anchors). To consider this, the selection of the best matching configuration has to be modified, as shown in Table 4.3.

Having exactly one anchor spares the need of shifting the graph around during the matching phase since it can be attached to the anchor, primarily reducing the configurations to test against. In the case of more anchors, the main matching process is based on finding the minimum sum of squared errors (SSE) between the estimated anchor positions and the real ones. Starting from three anchors, it is also possible to supersede the computations necessary to ensure the area coverage. For two given anchors, a combined approach is chosen that still may lead to a mirrored configuration although the anchor vertices match their positions. For each of the four possible ways of mirroring a graph, the conformity values of the configurations with the lowest squared anchor distance error are compared against each other and the configuration with the highest one chosen for the next step. To achieve a more realistic distribution and compensate for the mentioned property of providing maximum possible distances between unconnected vertices, anchor vertices are steadily dragged towards their real position in the final space-fitting phase, implicitly applying resulting position corrections also to surrounding vertices.

4.4 Conclusion - FLoW

Low-power wireless technology has allowed the free and flexible deployment of embedded systems in a variety of scenarios. As more and more systems get deployed and an increasing number of devices are introduced in scenarios with already existing infrastructures, it becomes increasingly important to understand system properties with limited information and intervention from the user. In this chapter, FLoW is presented [a4]. This approach focuses on understanding the positioning of wireless devices in an indoor environment with use of connectivity information and a physical map of the scenario as well as optional anchor devices. With limited information, it is explored how the location of devices can be estimated through the use of force-directed graphs and how specific domain knowledge can be exploited to improve the localisation process. In this context, necessary adaptations to the utilised graph approach as well as potential problems and limitations are discussed and how to address them by providing additional information. In the evaluation (Section 8.3), it is then investigated to what extent the correct positioning can be inferred by looking at different indoor testbeds and challenging simulated scenarios. Furthermore, the assumptions that needed to be made are discussed, thus exploring to what extent FLoW can be employed in practice.

From the overarching perspective of this thesis, FLoW addresses the first and earliest stage in the endeavour of gaining knowledge about a network and its properties. Having almost no information available, except connectivity information and optionally a rough understanding of the environment and its dimensions, FLoW is yet able to exploit and combine such coarse information to provide first insights regarding the positioning of devices. Thus it assists the user to fill the existing gap regarding the understanding of how the network is situated in its environment and to work towards a basis for further knowledge gain and optimisation processes, carried out by the other contributions of this thesis. For modelling wireless communication behaviour, as done in the second contribution, the integration of device positions provided by FLoW can be beneficial to perform basic predictions, assuming an unchanged rudimentary level of knowledge. However, due to the limitations given by the inherent imprecision of the basic inputs, a room specific device localisation cannot be achieved. Especially in view of the subsequent contributions, obtaining advanced knowledge, such more detailed input becomes necessary to exploit the physical properties of signal propagation by the model. Accordingly, the localisation of FLoW does not provide sufficient accuracy to meet these requirements. Nevertheless, considering the low requirements on the provided inputs, FLoW serves as a starting point for obtaining first insights for an, at that point mostly unknown network that needs to be understood.

5 Efficient and Accurate Wireless Communication Modelling

Understanding the behaviour of a wireless network is an essential first step offering possibilities for improving its properties. By modelling wireless communication, incorporating knowledge of either preceding observations or physical propagation laws and properties of the environment, predictions of expected signal quality become possible at arbitrary locations. This information is invaluable when taking action for improvements by discovering new or alternative positions for devices or basing subsequent algorithms on them. However, providing a general solution is difficult, as the available information about network and environment differs regarding quality and quantity in each scenario. Consequentially, models are required which are able to handle the given information in individual and appropriate ways.

With real-world traces of reception probabilities and corresponding features, e.g., RSSI, a statistical model can match the behaviour of wireless communication at known (observed) positions. For unknown positions, this model can only interpolate between the locations of the provided measurements. With an increased complexity of the environment and the presence of obstacles, such approximation loses accuracy. In contrast, with a large amount of computation time and resources, physical models are able to predict how the wireless signal propagates in detail. In this case, the accuracy is bound to the precision of the provided information about the physical displacement and size of objects, e.g., walls, and of their specific impact on the signal propagation.

In this chapter, an approach is introduced, providing multiple solutions for modelling wireless network communication, utilising different input data as well as computational methods, for generating predictions at a desired accuracy and complexity. These models serve as basis for the other subsequent contributions discussed in this thesis.

5.1 Challenges

Considering the goals of this thesis, comprehending wireless networks in a universally applicable manner by providing flexible approaches bears several challenges relating to modelling wireless communication. Especially pursuing a more complex physical approach, accurately reproducing the signal propagation of the real world in a virtual model requires to consider the interplay of multiple physical effects. As mentioned in Section 2.2, in order to correctly simulate certain effects, specific details of the target environment need to be provided. However, as the available information could considerably differ in each scenario, these details might not be available. Therefore, communication models need to provide capabilities dealing with both situations, adjusting the accuracy based on the availability and the given amount of information. This conflict of model design, preciseness versus practicability, is illustrated by the two subsequent example cases which also have been dealt with in the context of this thesis.

Interdependencies of Properties

Achieving good signal coverage in a wireless network requires either well prepared planning, comprising knowledge of the environment or time-consuming trial and error to accomplish an effective placement of access points. Pursuing the former elaborated approach, obtaining knowledge to perform an accurate physical modelling also comprises considering material characteristics in relation to the chosen frequency of wireless communication.

Each frequency exhibits different properties, e.g., the occurring propagation loss when passing through specific materials or the ability to bend around sharp edges depending on the wavelength. Existing communication frequencies in almost any range from a few Hz (submarine communications [p153]) up to 100 GHz (5G cellular network technology [m2]) require extended measurement campaigns to accurately model the signal interaction with obstacles of different materials and thus are costly. According to [p144], conducting a comprehensive experimental study on radio signal attenuation, testing frequencies between 0.5 GHz and 8 GHz penetrating 20 different common construction materials, an increasing frequency in general increases the propagation loss for most of the materials. However, there are some exceptions to take into account, e.g., glass and plywood, where at certain discrete higher frequencies resonance effects even increase the received power. These effects may also vary over different pieces of the same material depending on the manufacturing process or the type of performed experiments as seen in a similar study on material properties [p165], exacerbating a precise modelling even more.

As a consequence, models either have to be designed considering influencing factors in the minutest details, requiring large amounts of information providing an individually tailored solution to one specific scenario in the end or following a more general approach applicable to various scenarios at the cost of accuracy. Complying with the goals of this thesis, limiting the amount of necessary information to achieve a flexible and practical solution, the more general approach is pursued and implemented to model wireless communication.

Interplay of Effects and Required Details for Their Modelling

Conversely, in order to realistically model certain propagation effects, the availability of specific details is essential. E.g., to accurately model scattering effects and their impact on signal propagation, the surface roughness of the involved materials is required as well as the wave polarisation (see Equations 2.22, 2.23). In absence of such information, the influence of scattering can at most be estimated, if considered at all. Con- and destructive interferences on the other hand depend on the phases of multiple waves superimposing at the same point in space and time. As amplification and cancellation differ only by half of a wavelength, corresponding to a distance of either around 17 cm for 900 MHz or 6 cm for 2.4 GHz frequencies, a precise signal path length computation is indispensable. Hence, the descriptions of the modelled environments have to be of the highest possible accuracy regarding device and obstacle positions. Computed interferences may otherwise predict the exact opposite of what is observed in reality in the worst case, rendering them useless.

Accordingly, a trade-off has to be made, either to provide a sophisticated modelling close to reality by considering a large number of physical effects at the consequence of increased detail requirements or to refrain from the necessity of such details, again, at the cost of accuracy. In the context of this work, the latter approach has been chosen, minimising the requirements for maintaining an applicability in a large number of scenarios where such information may not be available or require a disproportional effort to obtain.

5.2 Approach

Modelling of actual wireless communication in real environments can be based, in practice, on statistical, experimental observations of signal properties between devices once deployed in a specific setup [p108]. This statistical approach has the benefit of being fast to compute once measurements are available; no prior, detailed information is necessary about the scenario and, once a system is deployed, the obtained estimations adhere closely to the real existing environment and system structure. However, the measurements make it hard (if possible at all) to generalise the gathered description to other scenarios or even to other device positioning inside the same scenario. Similarly, it is impossible to reason about the causes of changes in the wireless signal as no description of the scenario is available.

On the other side, the physical properties of the environment and the laws of electromagnetic wave propagation offer the possibility, in theory, to exactly model the expected wireless features. This comes, however, at the cost of complexity in terms of both the degree of details that need to be provided in order to accurately describe the surroundings as well as the computational resources necessary to process the actual model features throughout the environment.

Evidently, the more knowledge of a network is available, the more profound predictions can be made when modelling wireless communication at unknown positions. Nevertheless, in each scenario the available amount of information differs, aggravating the usage of one specific type of model when targeting to achieve the most accurate estimations for a given input. Consequentially, the used approaches have to provide the flexibility to adapt to the information available. Since both types of models, statistical and physical, use orthogonal data as their inputs, each of these are valuable to use when dealing with heterogeneous scenarios with varying information about network and environment.

Pursuing the goal of providing practical and general solutions, applicable to a wide range of networks, this chapter introduces Models for Network Optimisation in Versatile Environments (MOVE), a collection of models for predicting wireless signal propagation [a1, a3, a2]. With the main focus on the physical aspect of modelling due to better exploitability of locally accurate predictions for gaining network knowledge, MOVE provides four models in total, appropriate for communication modelling under different conditions: (1) a two-dimensional physical model, (2) an improved three-dimensional version, (3) a statistical model and (4) a hybrid approach combining the traits of both model types, physical and statistical. Due to its individual characteristics, each of these models offer benefits which are described below.

Two-Dimensional Physical Model The first introduced model PHY bases its predictions on ray tracing to compute signal paths in a two-dimensional space. Given the availability of a description of the environment, it offers the possibility to perform accurate predictions and thus to investigate signal properties at arbitrary positions within a network. By implementing optimisations of ray path investigation and parametrisations on the computation limits (see Section 5.3.2), this model is able to provide estimations on the communication behaviour at an adjustable quality and computation time. Intended for execution on a CPU and thus suitable for use on almost any hardware, it represents the base model of MOVE and is furthermore utilised in subsequent contributions to increase the knowledge about the network.

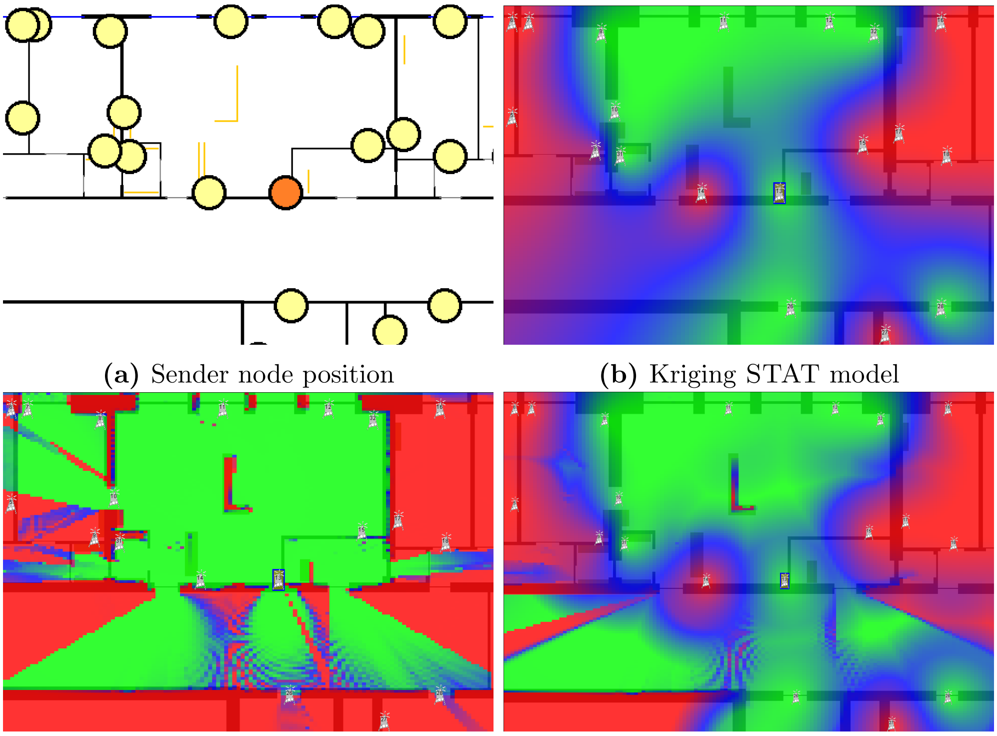
Three-Dimensional Physical Model Pursuing physical modelling approaches while having limitations on any resource requires a trade-off between accuracy and accepted costs, e.g., efforts for obtaining further network knowledge in case of lacking details, or execution time in case of limited computational resources. To mitigate this problem in favour of improved accuracy and/or reduced costs, one approach is to improve the efficiency of computations by adapting them to the given hardware used for the execution of model predictions. Considering the required computational effort to generate an accurate model by including more physical details, a three-dimensional ray tracing model 3D GPU has been developed, executable on the GPU and utilising its parallelisation capabilities to handle the problem complexity (Section 5.3.3). In this way, it becomes possible to increase the accuracy while retaining the calculation time compared to the previous approach using the CPU. However, as appropriate graphics cards, albeit widely used, cannot be assumed to be available in any scenario, this approach comes at the cost of less applicability and is considered as an optional improvement of the otherwise applicable two-dimensional CPU model.

Statistical Model The third model of MOVE introduces a statistical approach STAT, based on Kriging interpolation [p87], applying a logarithmic interpolation between known data points (Section 5.3.4). In absence of a description of the environment, but existing measurements from the network, this model is able to perfectly reproduce the network behaviour within a fraction of time compared to the physical models of MOVE, thus acting as their complement. Compared to other statistical models, this model does not require any empirically determined coefficients as input, which have to be set specific to the environment. Device positions and traces collected from the network are sufficient for performing predictions. However, as STAT is not aware of any local environment properties, accuracy decreases with an increasing distance to known positions, aggravating predictions at positions with sparse coverage of data points. Nevertheless, this model comes into play when physical modelling is not applicable, thus providing a valuable addition to MOVE.

Hybrid Model Investigating the potential of combining benefits of both model types, the fourth model introduces an approach, fusing PHY and STAT into one HYBRID model (Section 5.3.5). To generate predictions, a weighted output of both models is used, favouring one or the other in dependency of the distance to known positions. Due to combined usage, the weaknesses of both models are mostly compensated by the strengths of the respective other. The inherent impreciseness of PHY, attributable to inaccurate descriptions and unconsidered effects, is locally corrected by observations taken from STAT. On the other hand, the decreasing prediction quality of STAT at larger distances is counteracted by the environment description utilised by PHY. Hence, in case a scenario offers extensive information enabling the usage of either STAT and PHY, HYBRID can be used instead to fuse the data obtaining predictions at an enhanced accuracy.

Figure 5.1 visualises the signal strength predictions of the different models of MOVE for a sending device highlighted in orange in (a), which is located in the main evaluation scenario WSN Testbed (Section 8.1). Due to their individual ways predicting signal strengths, the outputs of STAT (b) and PHY (c) look different. While STAT produces continuous interpolations between known device positions, the signal strengths of PHY might change abruptly depending on the obstacles located in the environment. As being a combination of the two former models, the visual representation of HYBRID (d) contains properties of both. Considering the characteristics of each individual model as well as the added value of the complete set, MOVE makes the following contributions:

Modelling Fitting the Conditions of a Scenario MOVE provides suitable modelling approaches for a variety of scenarios with different information and capabilities available. Having either traces of network communication or a description of the environment exclusively available, MOVE enables a modelling in both cases by providing either a statistical or physical model. Upon the availability of computational resources, the degree of detail can further be adjusted by increasing the complexity and thereby the accuracy of the predictions. Thus, MOVE utilises the individual circumstances given in each scenario to enable a modelling in the most suitable way.



(a) Sender node position

(b) Kriging STAT model

(c) Ray tracing PHY model

(d) HYBRID model

Figure 5.1: STAT, PHY *and* HYBRID *estimations of reception probabilities for the sending node highlighted in orange in (a). The colours denote an expected clear reception (green), bad connectivity (red), or a transitional region (blue).*

Accurate Physical Modelling Without Detailed Descriptions As described in the challenges, modelling certain physical effects correctly requires a minimum amount of detail, e.g., centimetre accuracy for computing interference effects. Instead, the physical models of MOVE demonstrate that it is possible to achieve a satisfiable accuracy without incorporating such effects and thus, although certainly beneficial, not requiring the minutest details as well.

Synergy Exploitation by Combining Different Inputs Considered alone, both types of model play out their individual strengths. However, by providing and combining details on both, traces and environment, the advantages of both models can be exploited and synergies created. Realising the fusion of these two classes of models into a single, hybrid description of signal propagation, MOVE allows to benefit from the availability of different types of information about a network by gaining increased accuracy.

The models of MOVE are evaluated using data traces collected from multiple indoor scenarios (Section 8.1). Different qualities in the available measurement data, as well as environmental information, show that MOVE provides a suitable model for each scenario, exploiting the given data and achieving accurate predictions for entire networks within few seconds (Section 8.4), even with the more complex physical models. Furthermore, the results demonstrate that the introduced HYBRID technique is able to accurately follow the best models for the given scenario and the available information. The predictions of HYBRID are computed in the same amount of time than the two models, it consists of, demonstrating the practicality of the solution. Finally, the transferability of the physical model to networks with different hardware setups and communication properties is examined in more detail, revealing further insights about influences on the link quality predictions as well as capabilities and limitations of the chosen approach.

5.3 Implementation

In this section, the specific contributions are described to make physical modelling practical both in 2D and 3D descriptions, for the latter in particular exploiting the use of GPU computational power. Afterwards, a hybrid solution is introduced, enhancing the physical model by incorporating empirical observations via usage of a likewise introduced statistical model, combining the benefits of both modelling types and improving the prediction accuracy.

5.3.1 Definition of a Map

The most important information for physical modelling is the map of the environment, including the objects present in it. While autonomous 3D mapping of environments is making significant progress [p45], it is still challenging to perceive the specific features of relevance for techniques such as ray tracing. In particular, the understanding of the main involved materials is a tedious task that still relies on manual supervision. In this thesis, an effort is made in simplifying and reducing the information that needs to be gathered and provided as input to the various implemented modelling engines.

For indoor environments, a two-dimensional representation of main elements is necessary such as inner and outer walls. Depending on the level of accuracy required, information about surrounding buildings that could reflect signals might also be provided. In addition, relevant objects like doors, windows and ceilings are expected to be described with their height and, if

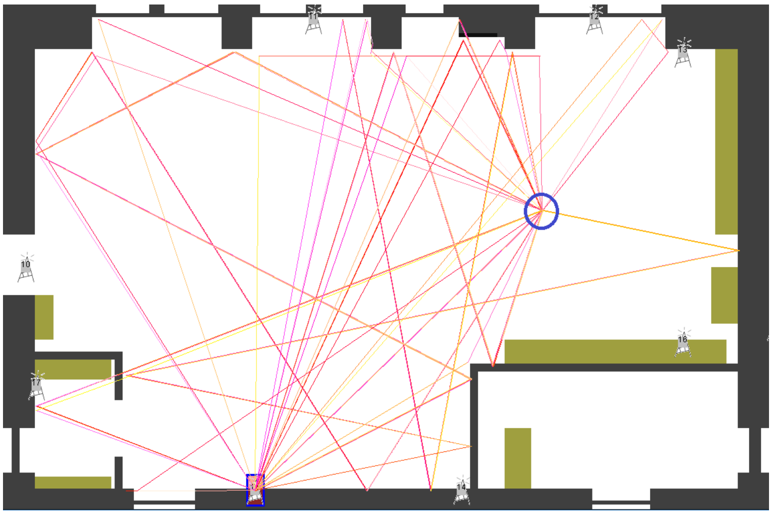


Figure 5.2: *Example image of a 2D ray tracer showing different paths from the sender at the bottom to the destination (blue circle) located in the middle of the room.*

applicable, tagged as objects that can change their state from closed to open and vice versa. The identification of the involved materials with their corresponding attenuation and reflection coefficients can be limited to coarse categories [p165]. In particular, ten types of objects are distinguished: two types of doors (thick and thin), five categories of walls (thick, very thick, brick, dry and outer walls), windows and two different closets (wooden and metallic). While more details, e.g., chairs or tables could further increase the model accuracy [p34], it is refrained from including them. Instead, the representation is limited to include big visible metallic furniture like cupboards. Finally, the position and optionally the height of wireless devices are can be exploited.

This information can easily be obtained through a floor map and a quick survey of the scenario, providing a machine-readable description that can be easily supported with computer-aided editing tools. Even though the amount of details to be provided is restricted, it must be clear that the accuracy of the supplied scenario descriptions is essential in defining the resulting model adherence to reality. In particular, centimetre accuracy is required in order to correctly compute con- and destructive interference from rays taking different paths to reach a destination. However, since providing such accuracy is impractical, techniques will be introduced to circumvent the necessity of highly detailed maps.

5.3.2 CPU-based 2D Ray Tracing Model

The first introduced model is a reference approach as depicted in the literature [a3], from which an own engine has been implemented exploiting the same principles. The developed physical model PHY is based on the ray tracing model implemented in the Cooja simulator [p116]. In particular, this Java-based modelling tries to aim rays in order to reach the destination from the source directly (through line of sight in free space or through transmission) or indirectly (through diffraction or reflection) by identifying surrounding obstacles. An example of the computed ray paths from a source to a destination is given in Figure 5.2. Given the complexity of computing a full 3D model of the propagation with just the resources offered by a (even if multi-core) CPU, the computation remains in a ”flat” 2D world. In fact, a CPU-based approach offers limited parallelism, imposing the use of an aimed approach in order to increase speed. Similarly, as rays depart from the source and encounter obstacles, the number of possible paths to be computed grows exponentially, affecting the processing time. As later evaluated, this time required to compute the model is significantly high, even for small environments with few obstacles and devices. This makes the model impractical in several use cases, e.g., for in-field system deployment and maintenance or for the integration in live drag-and-drop user design tools. Restrictions are then necessary, in particular for the maximum number of effects to consider and their combinations (e.g., reflection and diffraction are not considered together). Due to a large set of parameters, PHY offers the capabilities to adapt computations and their complexity to a given scenario. The most relevant parameters are listed Table 5.1, divided in three categories:

1) *Basic* parameters, including signal frequency, transmission output power and fixed losses for static effect computations. The former two have to be set individually for each scenario in dependency of the hardware specifications, the latter provide the base attenuation values in case material properties are not available.

2) Parameters related to *complexity*, determining the maximum number of transmissions/reflections/diffractions to consider for each ray path computation. As these values directly influence the accuracy and execution time of the ray tracing process, they should be adjusted based on multiple factors, e.g., the obstacle density, computational resources, targeted accuracy and accepted waiting time for the path computations. The impact of different combinations of effect limits is investigated in Section 8.4.2.

Table 5.1: *Parameters used by* PHY *and chosen values for evaluation.*

Parameter	*Description*	*Value*
f	Signal frequency	868 MHz / 2.4 GHz
tx_{pwr}	Transmission output power	−15 dBm to 0 dbm
att_{trans}	Fixed losses for static transmission effect	3 dBm
att_{refl}	Fixed losses for static reflection effect	5 dBm
att_{diff}	Fixed losses for static diffraction effect	10 dBm
max_{trans}	Maximum number of transmissions	5-7
max_{refl}	Maximum number of reflections	1-2
max_{diff}	Maximum number of diffractions	1
$b_{material}$	Consider individual material properties	✓
b_{angle}	Use angle-dependent diffraction loss	✓
$b_{2.5d}$	Incorporate obstacle height information	✓
$b_{radiation}$	Include device radiation patterns	×
$b_{interference}$	Compute interference effects	×

3) *Optional features* of PHY improving the prediction accuracy upon availability of additional information, comprising individual material attenuation, angle dependency, height consideration and radiation patterns. Optimisations are described in Section 5.3.2.

Utilising the aforementioned parameters, the complete computation process of PHY is visualised in Figure 5.3. Until receiving the final prediction for a given pair of starting and destination positions, three phases are passed, the initialisation, the path computations using ray tracing and the finalisation phase in which the found valid paths to the receiver are evaluated. These phases are explained below.

Initialisation

Prior to the actual computations, the model has to be initialised first, including the construction of a geometric representation in which rays can be traced. As already stated, PHY uses an aimed approach to identify paths while involving the information provided by the environment map. To significantly accelerate computations, each obstacle shape is internally represented as a rectangle of individual dimension. This simplification allows for performing very fast hit tests and angle computations to efficiently handle larger environments with many present obstacles.

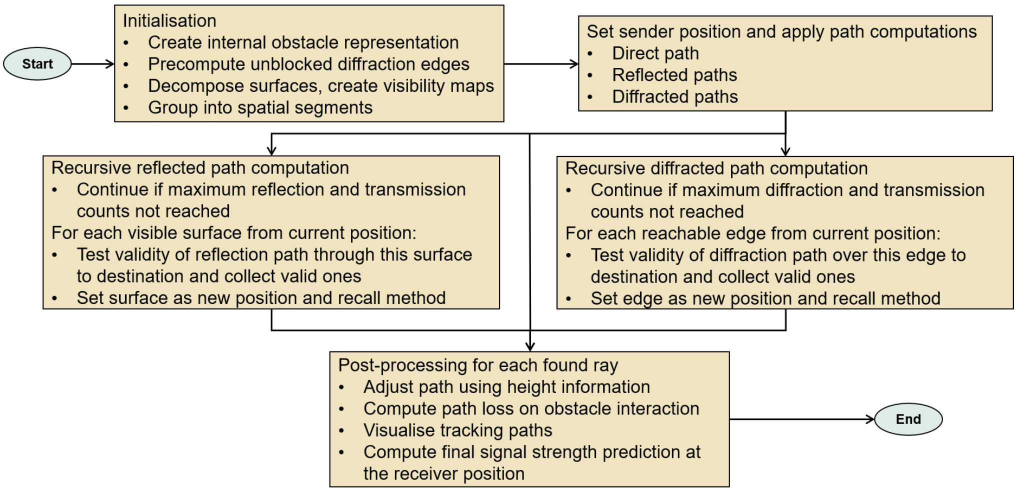

Start
Initialisation
• Create internal obstacle representation
• Precompute unblocked diffraction edges
• Decompose surfaces, create visibility maps
• Group into spatial segments
Set sender position and apply path computations
• Direct path
• Reflected paths
• Diffracted paths
Recursive reflected path computation
• Continue if maximum reflection and transmission counts not reached
For each visible surface from current position:
• Test validity of reflection path through this surface to destination and collect valid ones
• Set surface as new position and recall method
Recursive diffracted path computation
• Continue if maximum diffraction and transmission counts not reached
For each reachable edge from current position:
• Test validity of diffraction path over this edge to destination and collect valid ones
• Set edge as new position and recall method
Post-processing for each found ray
• Adjust path using height information
• Compute path loss on obstacle interaction
• Visualise tracking paths
• Compute final signal strength prediction at the receiver position
End

Figure 5.3: *Ray path computation steps of* PHY.

To decrease computation time, the paths of the signals and the visibility areas are precomputed and exploited to prune paths with minor contributions increasing usability for interactive design tools. For reflections, each obstacle is decomposed into the visible surface segments for which in turn all reachable other segments are computed. Due to the independent computation of reflections and diffractions, this allows for a fast exploration of valid reflection paths. In order to accelerate computations for diffractions, relevant edges on each obstacle are identified beforehand as well as which other edges are reachable, enabling a direct aiming at these edges when computing the ray paths similar to the aforementioned reflection computations.

Beginning with the precomputation of diffraction edges, the process is shown by Algorithm 5.1. The edges of each obstacle, i.e., top left (index 0), top right (index 1), bottom right (index 2) and bottom left (index 3), are checked for being blocked by other obstacles first and remaining free ones are set as diffraction edges, which the path computation aims at. Afterwards, the visibility to other reachable diffraction edges is evaluated by drawing a line between both points which must not intersect the other obstacle except for the diffraction point. Furthermore, although each diffraction produces a new wavefront which can move into any direction, only sharp edge diffractions are considered in PHY, bending around the obstacle to reach the shadowed region behind. E.g., for a top left edge of an obstacle, only other edges are relevant which are positioned right and above, or left and below as these are the only two directions, a ray can proceed when being bent around this edge without passing the related obstacle again. To improve readability, the visibility checks are only shown for the top left edge as the principle is similar for the other edges.

Algorithm 5.1 PHY diffraction edge precomputation.

```
function $\textsc{precomputeDiffractionEdgeVisibility}(env)$
    $O := env.getStaticObstacles()$;
    $EdgeMap := computeDiffractionEdges(env)$;
    $VisibilityMap :=$ new Map();
    for $(o_{vis} \in O)$ do
        $Edges_{vis} := EdgeMap.get(o)$;
        for $(i \in \mathbb{N}, 0 \leq i \leq 3)$ do
            if $(Edges_{vis}[i] = null)$ then
                continue with next $n$;
            $VisibleEdges :=$ new List();
            $e_{vis} := Edges_{vis}[i]$;
            for $(o_{other} \in O, o_{vis} \neq o_{other})$ do
                $Edges_{other} := EdgeMap.get(o)$;
                for $(j \in \mathbb{N}, 0 \leq j \leq 3)$ do
                    $e_{other} := Edges_{other}[j]$;
                    if $(i = 0)$ then
                        if $(\neg\, o_{other}.rectangle.intersectsLineOfPoints(e_{vis}, e_{other})$
                           $\wedge\, (((e_{vis}.x < e_{other}.x) \wedge (e_{vis}.y < e_{other}.y))$
                              $\vee\, ((e_{vis}.x > e_{other}.x) \wedge (e_{vis}.y > e_{other}.y)))$ then
                            $VisibleEdges.add(e_{other})$;
                    else if $(i = 1)$ then
                        ...
            $VisibilityMap.put(s_{vis}, VisibleEdges)$;
    return $VisibilityMap$;

function $\textsc{computeDiffractionEdges}(env)$
    $O := env.getStaticObstacles()$;
    $EdgeMap :=$ new Map();
    for $(o \in O)$ do
        $Edges := getObstacleEdges(o)$;
        for $(i \in \mathbb{N}, 0 \leq i \leq 3)$ do
            if $(env.getObstaclesAt(Edges[i]).size() > 1)$ then
                $Edges[i] := null$;
        $EdgeMap.put(o, edges)$;
    return $EdgeMap$;

function $\textsc{getObstacleEdges}(o)$
    $Edges :=$ new Array(4);
    $Edges[0] :=$ new Point$(o.minX, o.minY)$;
    $Edges[1] :=$ new Point$(o.maxX, o.minY)$;
    $Edges[2] :=$ new Point$(o.minX, o.maxY)$;
    $Edges[3] :=$ new Point$(o.maxX, o.maxY)$;
    return $Edges$;
```

The visibility precomputation for reflection surfaces is described by Algorithm 5.2. After decomposing an obstacle into its four outer segment lines: top (index 0), right (index 1), bottom (index 2) and left (index 3), these are checked for overlaps with other obstacle segments. The parts which are not blocked by other obstacles are set as reflection surfaces for which visibility computations are preformed. Considering the direction of the surface and the relative position to other obstacles, only certain surfaces can be potentially reached for further reflections. E.g., a surface facing upwards can never see other surfaces facing upwards and the other three sides only when the former is positioned below. Due to the similar process for all four sides, the depicted algorithm only shows the process for the top side.

Signal Path Computation

The first step performed by the model in predicting the expected RSSI is to determine possible signal paths from a given starting position to a destination. Each path consists of several segments, separated by the encountered effects, holding information of position, effect type, involved obstacle and incident angle. Combining these, a full reconstruction of the signal path becomes possible enabling a calculation of the expected signal attenuation on the path afterwards. A path is considered as valid if it reaches a destination within the given limits of allowed effects. Once one of these limits is exceeded during computation, this path is discarded immediately.

Depending on the types of effects to consider, the model computes paths to a destination in three different ways: directly as a straight line including transmissions and indirectly either via reflections or diffractions. In PHY, reflections and diffractions are not considered together. This decision was made due to the increased necessary computational effort at both sides, reflection and diffraction. A combination of both effects would complicate the individually optimised methods of path finding (Section 5.3.2). For reflections, path building and mirroring (explained below) need to be applied on multiple sections depending on how many diffractions are on a path. In case of diffraction, a validation only becomes possible after the entire following reflection path has been validated. In the worst case a completely computed path is then invalidated, wasting a large amount of computation time.

Signal paths resulting from scattering effects are not taken into account. Compared to the power of the incident wave and the transmitted and reflected parts, the impact of scattered waves is not insignificant, but limited [p118] and requires specific information about the surface roughness of the involved materials [p117] as well as increased computational effort to determine the radiated power. In addition, detecting scattering effects in the

Algorithm 5.2 PHY reflection surface visibility precomputation.

```
function PRECOMPUTEREFLECTIONSURFACEVISIBILITY($env$)
    $O := env.getStaticObstacles()$;
    $SurfaceMap := computeReflectionSurfaces(env)$;
    $VisibilityMap :=$ new Map();
    for ($o_{vis} \in O$) do
        $Surfaces_{vis} := SurfaceMap.get(o)$;
        for ($i \in \mathbb{N}, 0 \leq i \leq 3$) do
            $VisibleSurfaces :=$ new List();
            $s_{vis} := Surfaces_{vis}[i]$;
            for ($o_{other} \in O, o_{vis} \neq o_{other}$) do
                $Surfaces_{other} := SurfaceMap.get(o)$;
                if ($i = 0$) then
                    if (Line.$isBelow(s_{vis}, Surfaces_{other}[2])$) then
                        $VisibleSurfaces.add(Surfaces_{other}[2])$;
                    if (Line.$isLeft(s_{vis}, Surfaces_{other}[3])$) then
                        $VisibleSurfaces.add(Surfaces_{other}[3])$;
                    if (Line.$isRight(s_{vis}, Surfaces_{other}[1])$) then
                        $VisibleSurfaces.add(Surfaces_{other}[1])$;
                else if ($i = 1$) then
                    ...
            $VisibilityMap.put(s_{vis}, VisibleEdges)$;
    return $VisibilityMap$;

function COMPUTEREFLECTIONSURFACES($env$)
    $O := env.getStaticObstacles()$;
    $SurfaceMap :=$ new Map();
    for ($o_{refl} \in O$) do
        $Sides_{refl} := getObstacleSides(o_{refl})$;
        for ($o_{other} \in O, o_{refl} \neq o_{other}$) do
            $Sides_{other} := getObstacleSides(o_{other})$;
            for ($i \in \mathbb{N}, 0 \leq i \leq 3$) do
                if (Line.$overlap(Sides_{refl}[i], Sides_{other}[(i+2) \mod 4])$) then
                    $Sides_{refl}[i] :=$
                            Line.$difference(Sides_{refl}[i], Sides_{other}[(i+2) \mod 4]$);
        $SurfaceMap.put(o_{refl}, Sides_{refl})$;
    return $SurfaceMap$;

function GETOBSTACLESIDES($o$)
    $Sides :=$ new Array(4);
    $Edges := getObstacleEdges(o)$;
    for ($i \in \mathbb{N}, 0 \leq i \leq 3$) do
        $Sides[i] :=$ new Line($Edges[i], Edges[(i+1) \mod 4]$);
    return $Sides$;
```

vicinity of an investigated position would necessitate further path computations as each intersection with an obstacle surface can then be the source of numerous new signal paths in any direction, whose tracing in turn would notably increase the complexity. Thus, despite possible losses in the prediction accuracy, scattered rays are not considered to keep the amount of required details and the computation effort low.

Transmissions For the direct computation, each obstacle crossing the straight line from sender to destination is added to the path as transmission effect. After a transition of a wave into a different medium and back, there is usually a position shift of the outgoing ray due to refractions on the boundaries of the medium. However, as this angular change is dependent on the specific material type, an accurate computation is not possible. Furthermore in most cases, providing representations of large environments with centimetre accuracy is not feasible. Therefore, the refraction effect is disregarded and the comparatively insignificant additional inaccuracy is accepted at the benefit of simplified computations. In general, PHY determines transmissions during path computation when aiming at the next relevant position, i.e., obstacle surfaces for reflections, edges for diffractions or the destination.

The transmissions occurring on the straight path between two points are computed according to Algorithm 5.3. Obstacles crossing this path are added to the list, including hit point and angle determination. In order to reduce comparisons and thus to speed up the computations, obstacles are spatially grouped into smaller segments so that only those in the vicinity of the investigated direct path have to be compared. Finally, all hit obstacles are sorted by their distance to the source in an ascending order, obtaining the actual sequence of transmissions.

Reflections Taking reflections into account requires to focus on the surfaces of obstacles located in the environment. Accordingly, each obstacle is decomposed into its visible surface segments during initialisation of PHY (Section 5.3.2). For determining the next reflection, surfaces are successively targeted facing the direction of the current ray. Paths involving multiple reflections are then explored in a recursive way tracking hit obstacles. Finally, to obtain valid ray paths including reflections the image method is used, as shown in Figure 5.4. If there exists a straight line from the starting position to the mirrored destination crossing all reflecting surfaces, a path is considered. Actual positions and angles of the reflections on the surfaces only then become determinable and are added to the path information.

Algorithm 5.3 PHY transmission path computation.

```
struct PathElement$_{trans}${
    Point $hitPoint$;
    Float $hitAngle$;
}

function computeDirectPath($p_{source}, p_{dest}$)
    $Path$ := new List();
    $line_{path}$ := new Line($p_{source}, p_{dest}$);
    $O_{candidate} := getObstaclesInSegmentCrossingLine(line_{path})$;
    for ($o \in O_{candidate}$) do
        $p_{intersect} := Rectangle.getLineIntersectionPoint(o.rectangle, line_{path})$;
        if ($p_{intersect} \neq null$) then
            $angle$ := Line.$getAngle(line_{path}, getIntersectionSegment(o.rectangle))$;
            $Path.add$(new PathElement$_{trans}(p_{intersect}, angle)$);
    $Path := \{pe_1, \ldots, pe_n \mid$
        Point.$distance(p_{source}, pe_i) <$ Point.$distance(p_{source}, pe_{i+1})\ \forall pe \in Path\}$;
    return $Path$;
```

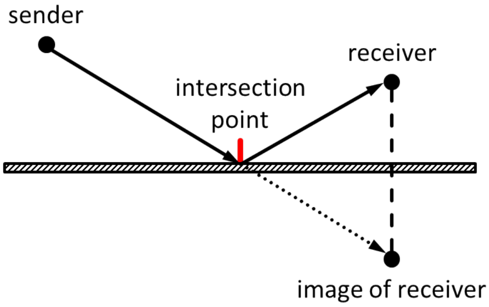


Figure 5.4: *Image method to compute intersection points of reflected rays using aimed ray tracing.*

The detailed process is described by Algorithm 5.4. In order to process the reflections efficiently, a helper object is defined, containing the original position of the surface line, a mirrored version for checking the validity of a reflection path and the exact position and angle at which the ray intersected the surface. The recursion starts by verifying the current reflection path including the latest added surface by using the image method (line 13). Beginning with the actual, unchanged positioning of elements, it is iterated through the current path of reflections in order of their occurrence. Source point and previous surfaces are mirrored using the current surface as axis. Having build up the image positions, the intersection test for each surface with the straight line between sender and destination position is performed afterwards (line 22). If not all surfaces are intersected, the path is discarded and it is proceeded with the next one. However, as it is still possible that the currently discarded path can become a valid one after adding further

Algorithm 5.4 PHY reflection path computation.

```
struct PathElement$_{refl}${
    Line $hitSurface, mirrorSurface$;
    Point $hitPoint$;
    Float $hitAngle$;
}

function COMPUTEREFLECTEDPATHS($p_{source}, p_{dest}$)
    $ValidPaths :=$ new List();
    $reflectedPathRecursion(ValidPaths,$ new Stack()$, p_{source}, p_{dest})$;
    return $ValidPaths$;

function REFLECTEDPATHRECURSION($Paths, PathStack, p_{source}, p_{dest}$)
    $p_{mirror} := p_{source}$;
    for ($i \in \mathbb{N}, 0 \leq i < PathStack.size()$) do
        $pe := PathStack.get(i)$;
        $pe.mirrorSurface := pe.hitSurface$;
        $p_{mirror} :=$ Point.$mirrorAtLine(p_{mirror}, pe.hitSurface)$;
        for ($j \in \mathbb{N}, 0 \leq j < i$) do
            $pe_{prev} := PathStack.get(j)$;
            $pe_{prev}.mirrorSurface :=$
                Line.$mirrorAtLine(pe_{prev}.mirrorSurface, pe.hitSurface)$;

    $line_{mirror \rightarrow dest} :=$ new Line($p_{mirror}, p_{dest}$);
    for ($i \in \mathbb{N}, PathStack.size() > i \geq 0$) do
        $pe := PathStack.get(i)$;
        $p_{intersect} := Line.getIntersectionPoint(line_{mirror \rightarrow dest}, pe.mirrorSurface)$;
        if ($p_{intersect} = null$) then
            goto recursionStep;
        $pe.hitAngle := Line.getAngle(line_{mirror \rightarrow dest}, pe.mirrorSurface)$;
        for ($j \in \mathbb{N}, PathStack.size() > j > i$) do
            $pe_{prev} := PathStack.get(j)$;
            $p_{intersect} :=$ Point.$mirrorAtLine(p_{intersect}, pe.hitSurface)$;
        $pe.hitPoint := p_{intersect}$;

    $count_{trans} := 0$;
    $p_{last} = p_{dest}$;
    for ($i \in \mathbb{N}, PathStack.size() > i \geq 0$) do
        $SubPath := computeDirectPath(PathStack.get(i).hitPoint, p_{last})$;
        $count_{trans} \mathrel{+}= SubPath.size() - 1$;
        $p_{last} := PathStack.get(i).hitPoint$;
        if ($count_{trans} > max_{trans}$) then goto recursionStep;
    $SubPath := computeDirectPath(p_{source}, p_{last})$;
    if ($count_{trans} > max_{trans}$) then goto recursionStep;
    $Paths.add(includeIntermediateTransmissions(PathStack))$;

    recursionStep:
    if ($PathStack.size() < max_{refl}$) then
        $p_{base} := PathStack.isEmpty()\ ?\ p_{source}\ :\ PathStack.top().hitPoint$;
        $S_{vis} := getVisibleReflectionSurfaces(p_{base}))$;
        for ($s_{vis} \in S_{vis}$) do
            $PathStack.push($new PathElement$_{refl}(s_{vis}, s_{vis}, null, 0))$;
            $computeReflectedPaths(Paths, PathStack, p_{source}, p_{dest})$;
            $PathStack.pop()$;
```

reflections, the recursion is continued at that point and only the current path is not added to list. In case a valid reflection path to the destination exists, the actual and exact hit positions of the ray are computed by reverting previous mirroring (line 29). Next, the path to the destination is determined and the transmitted obstacles between the reflections are counted (line 34). If the number of transmissions lies within the allowed limits of transmissions, the reflection path is added to the list of valid paths, including intermediate transmissions, and otherwise discarded. Finally, for the next recursion step, surfaces reachable from the current position are determined, using the precalculated visibility map (line 45). For every surface, the method is recalled with an updated stack containing the newest one to continue the exploration of the reflective paths until the maximum number of reflections has been reached.

Diffraction An identification of suitable edges is necessary to compute diffraction effects. Thereto, the edges of each obstacle are checked whether they are blocked by other obstacles. As the static elements of an environment do not change, a precomputation of relevant edges is obvious and performed during initialisation (Section 5.3.2). Suitable edges are then aimed at from the starting position. From the position of the chosen edge, it is either aimed at the destination position or another edge. Multiple diffractions are computed similarly to reflections. Depending on the direction of the subsequent path segment, the plausibility of the diffraction is reassessed. If the ray to the next position intersects the obstacle providing the edge, it is not a valid diffraction and thus the path is discarded.

Algorithm 5.5 shows the diffraction path computation PHY in detail. The general process is similar to the one for reflections, using a recursive approach. In contrast to reflections, however, the exact positions of diffractions are already fixed during the selection of the edge since they do not change with the addition of subsequent diffraction edges. The only exception is the angle of diffraction which only becomes clear once the next destination is set. Hence, only the lastly added segment of the path has to be adapted and checked for validity instead of the whole path, reducing the complexity. At the beginning of a recursion step, the latest added diffraction edge is examined regarding the reachability of the destination position (line 14). This includes checks whether the total number of transmitted obstacles does not exceed the overall limit, the diffraction is a sharp edge diffraction, bending around the obstacle at an angle between 90° and 180°, and the direct path is not intersecting the same obstacle again where the diffraction happened. Fulfilling these conditions, the angle is updated and the path considered as valid, adding it to the list, again including intermediate transmissions. On the other hand, the recursion path is aborted if either the diffraction limit

Algorithm 5.5 PHY diffraction path computation.

```
**struct** PathElement$_{diff}${
  Point $hitPoint, prevPoint$;
  Float $hitAngle$;
}

**function** computeDiffractedPaths($p_{source}, p_{dest}$)
    $ValidPaths$ := **new** List();
    $diffractedPathRecursion(ValidPaths,$ **new** Stack()$, p_{source}, p_{dest}, 0)$;
    **return** $ValidPaths$;

**function** diffractedPathRecursion($Paths, PathStack, p_{source}, p_{dest}, cnt_{trans}$)
    **if** $(PathStack.isEmpty())$ **then**
        **goto** recursionStep;
    $p_{edge} := PathStack.top().hitPoint$;
    $p_{prev} := PathStack.top().prevPoint$;
    $SubPath_{dest} := computeDirectPath(p_{edge}, p_{dest})$;
    $cnt_{newTrans} := cnt_{trans} + SubPath_{dest}.size() - 1$;
    $angle_{dest} := computeDiffractionAngle(p_{prev}, p_{edge}, p_{dest})$;
    **if** $((cnt_{newTrans} \leq max_{trans})$
      $\wedge\ (getObstacleAt(p_{edge}).rectangle.intersectsLineOfPoints(p_{edge}, p_{dest}))$
      $\wedge\ (90 \leq angle_{dest} < 180))$ **then**
        $PathStack.top().hitAngle := angle_{dest}$;
        $Paths.add(includeIntermediateTransmissions(PathStack))$;
    **else if** $((PathStack.size() = max_{diff}) \vee (cnt_{newTrans} > max_{trans}))$ **then**
        **return** ;

    recursionStep:
    $p_{base} := PathStack.isEmpty()\ ?\ p_{source}\ :\ PathStack.top().hitPoint$;
    $E_{vis} := getVisibleDiffractionEdges(p_{base}))$;
    **for** $(p_{vis} \in E_{vis})$ **do**
        $SubPath_{diff} := computeDirectPath(p_{edge}, p_{vis})$;
        $cnt_{newTrans} := cnt_{trans} + SubPath_{diff}.size() - 1$;
        **if** $(cnt_{newTrans} \leq max_{trans})$ **then**
            **continue** with next $p_{vis}$;

        **if** $(\neg\ PathStack.isEmpty())$ **then**
            $angle_{diff} := computeDiffractionAngle(p_{prev}, p_{edge}, p_{vis})$;
            **if** $((angle_{diff} < 90) \vee (angle_{diff} \geq 180))$ **then**
                **continue** with next $p_{vis}$;
            $PathStack.top().hitAngle := angle_{diff}$;

        $PathStack.push($**new** PathElement$_{diff}(p_{vis}, p_{edge}, 0))$;
        $computeReflectedPaths(Paths, PathStack, p_{source}, p_{dest}, cnt_{newTrans})$;
        $PathStack.pop()$;
```

has been reached with the latest edge or too many transmissions occurred. For the preparation of the next recursion step, visible diffraction edges are obtained from the precomputations and each edge evaluated regarding the possibility of being the next targeted position (line 25). This evaluation part basically contains the same steps as for the previous computation to the destination position, excluding the intersection test on the same obstacle as this has been checked previously during precomputations of the edge visibilities. Remaining edges forming a valid path are then pushed onto the stack one after the other to start the next recursion step.

Signal Strength Prediction

Once the valid paths have been computed, the postprocessing phase begins, evaluating each path and computing the expected path loss to finally generate an RSSI prediction. In this context, several optimisations on the paths, described in the next paragraph, are also applied. To determine the loss, each ray path is decomposed again into its segments. The total length of the path from sender to destination including the determined effects is used to compute the path loss in free space. Depending on the effects and chosen model parameters, additional losses are added either as a fixed loss constant or a dynamic value depending on the material type and thickness (transmission) or angle of direction change (diffraction). After completion, the path exhibiting the lowest expected path loss at the destination is chosen to compute the final RSSI prediction by adding the loss to the transmission power of the sender.

As they are mostly applied during the signal strength prediction of PHY, optimisations are discussed before turning to a detailed explanation of the computations. To exploit the benefits of PHY and make it usable in practice, several optimisations are introduced to either accelerate the model computation and improve its scalability, without sacrificing accuracy or improve the latter at only little additional cost. The effect of the optimisations on the model accuracy is investigated in Section 8.4.4.

Angle Dependency For each ray, the angle at which obstacles are encountered is determined in order to identify the different components that either penetrate the object or are reflected from it. However, as the reflectivity of an obstacle depends on the specific material properties, an accurate modelling is not possible when restricting to coarse material categories only. Thus angle dependent effects involving materials are not considered, excluding transmissions and reflections. In contrast, the determination of the attenuation in case of diffractions is independent of the material and included into PHY computations. When a radio wave signal is bending around an edge, the angle has a significant influence on the signal strength. While the attenuation is quite low at smaller angles of few degrees, it rapidly increases at larger angles until reaching the deep shadow area where the signal is practically not present anymore. Taking this strong angle dependency into account and not assuming a fixed attenuation on each diffraction, as done in the original version of the model, results in a more realistic modelling of signal behaviour, significantly reducing the number of false predictions.

Height Consideration (2.5D) To provide better distance estimation and, as a consequence, more accurate description of the attenuation of the signal over distance, the modelling accounts for the heights of the wireless devices and obstacles (if available), as done in [p77], saving computation time with respect to a complete 3D description. Thereto, after the ray computation in the two-dimensional space has been performed, each of the found paths is adapted, as described by Algorithm 5.6. Using Pythagoras' theorem, the height difference between sender and receiver is utilised to adjust the length of each path (line 1). Moreover, each interaction with an obstacle is rechecked whether occurring at all, taking the height of ray and obstacle at the given position into account (line 11). If a ray misses an obstacle which it should have hit, this effect is removed from the signal path and the obstacle specific attenuation not considered when computing the signal path loss. In case of missing reflections and diffractions the entire path becomes invalid and therefore is removed from the list of rays reaching the destination position. A consideration of available height information does not require complex computations and helps to improve the prediction accuracy of PHY, as shown in the evaluation.

Radiation Patterns To increase fidelity, it is possible to consider the integration of the specific antenna radiation pattern of the deployed wireless devices. In fact, the notable impact that such details could have on network behaviour is known [p174]. However, including such effect requires disproportionately more information than the benefit would provide. First, as the radiation pattern depends on the used antenna, an individual and sufficiently accurate representation is necessary for each used antenna type. Second, to compute the radiation angle correctly, in addition to its position an exact orientation of each device in the environment has to be provided as well. Therefore, although computed in two dimensions, a three-dimensional representation is required to take account for arbitrary device orientations. After some initial experimentation, is has been decided to discard this aspect for the work of this thesis and sacrifice accuracy in order to keep the approach practical.

Algorithm 5.6 PHY height consideration.

```
function includeHeightOnFullPath($pathLength_{2D}, p_{source}, p_{dest}$)
    $heightDiff := |p_{source}.z - p_{dest}.z|$;
    $pathLength_{3D} := \sqrt{(pathLength_{2D})^2 + (heightDiff)^2}$;
    return $pathLength_{3D}$;

function includeHeightOnPathSegment($segmentLength_{2D}$,
                                    $pathLength_{2D}, p_{source}, p_{dest}$)
    $pathLength_{3D} := includeHeightOnFullPath(pathLength_{2D}, p_{source}, p_{dest})$;
    $segmentLength_{3D} := pathLength_{3D} * (segmentLength_{2D}/pathLength_{2D})$;
    return $segmentLength_{3D}$;

function includeHeightOnRayPaths($AllPaths, p_{source}, p_{dest}$)
    ;
    for ($Path \in AllPaths$) do
        $pathLength_{2D} := computeFullPathLength(Path, p_{source}, p_{dest})$;
        $currLength_{2D} := 0$;
        $p_{last} := p_{source}$;
        for ($i \in \mathbb{N}, 0 \leq i < Path.size()$) do
            $pe := Path.get(i)$;
            $p_{curr} := pe.hitPoint$;
            $o := getObstacleAt(p_{curr})$;
            $currLength_{2D}$ += Point.$distance(p_{last}, p_{curr})$;
            $currHeight := p_{source} +$
                    $(currLength_{2D}/pathLength_{2D}) * (p_{dest}.z - p_{source}.z)$;
            $p_{last} := p_{curr}$;
            if ($height_{curr} > o.height$) then
                if ($pe.isTransmission()$) then
                    $Path.remove(i)$;
                else if ($pe.isReflection() \vee pe.isDiffraction()$) then
                    $AllPaths.remove(Path)$;
                    continue with next $Path$;

function computeFullPathLength($Path, p_{source}, p_{dest}$)
    $pathLength_{2D} := 0$;
    $p_{last} := p_{source}$;
    for ($i \in \mathbb{N}, 0 \leq i < Path.size()$) do
        $p_{curr} := Path.get(i).hitPoint$;
        $pathLength_{2D}$ += Point.$distance(p_{last}, p_{curr})$;
        $p_{last} := p_{curr}$;
    return $pathLength_{2D}$;
```

Turning back to the signal strength prediction, the evaluation of the computed paths is described in detail by Algorithm 5.7. Starting with the computation of aforementioned paths, the results are collected for further processing (line 2). If available, height information is subsequently incorporated to the found paths. For each found ray path, path length and losses are computed next (line 9). The former is composed of the length of all individual segments between the interactions with obstacles, the latter depends on the type of encountered effects. In case of transmissions, a fixed loss of the signal is assumed due to the reflected part of the signal as described in Section 2.2.3. Additionally, if material types are available, the signal attenuation caused by the propagation within a different medium is considered as well, depending on the travelled distance. For each reflection, a fixed value is added to the accumulated path loss, as the exact amount of reflected energy depends on specific material properties as well as the polarisation of the incident wave, which are, however, not modelled by PHY. Lastly, in case of diffraction effects, Lee's numerical formula for computing diffraction losses (Equation 2.21) is used in combination with the Fresnel-Kirchhoff diffraction parameter computation (Equation 2.17), if desired or otherwise again a fixed value added as loss. Finally, the path length is optionally adjusted to a three-dimensional one before the free space path loss (Equation 2.6) for the travelled signal distance is computed to obtain the total loss of the investigated path. The path with the lowest loss is then subtracted from the transmission power of the device to determine the expected signal strength at that position.

The original ray tracing model, which PHY builds upon, also offers the possibility of computing constructive and destructive interferences by analysing n most relevant paths and their lengths. However, the length of a radio wave of the 2.4 GHz band is around 12.5 cm. To determine how much two incoming signals are out of phase, the provided description of the environment requires centimetre accuracy to obtain a reasonable interference prediction. As already stated, due to the impracticability of providing such information, PHY does not use this method and solely relies on the signal strength provided by the strongest found path.

Algorithm 5.7 PHY signal strength prediction.

```
function COMPUTESIGNALSTRENGTH($p_{source}, p_{dest}$)
    $AllPaths$ := new List();
    $AllPaths.add(computeDirectPath(p_{source}, p_{dest}))$;
    $AllPaths.add(computeDiffractedPaths(p_{source}, p_{dest}))$;
    $AllPaths.add(computeReflectedPaths(p_{source}, p_{dest}))$;
    if ($b_{2.5d}$) then
        $includeHeightOnRayPaths(AllPaths, p_{source}, p_{dest})$;

    $pathLoss_{lowest} := \infty$;
    for ($Path \in AllPaths$) do
        $pathLength := computeFullPathLength(Path, p_{source}, p_{dest})$;
        $pathLoss := 0$;
        $p_{last} := p_{source}$;
        for ($i \in \mathbb{N}, 0 \leq i < Path.size()$) do
            $pe := Path.get(i)$;
            $p_{curr} := pe.hitPoint$;
            $p_{next} := ((i+1) < Path.size())\ ?\ Path.get(i+1) : p_{dest}$;
            if ($pe.isTransmission()$) then
                $pathLoss \mathrel{+}= att_{trans}$;
                if ($b_{material}$) then
                    $o := getObstacleAt(p_{curr})$;
                    $line_{trans}$ := new Line($p_{curr}, p_{next}$);
                    $line_{obst} := getSegmentWithinRectangle(line_{trans}, o.rectangle)$;
                    $dist_{obst} := \text{Line}.segmentLength(line_{obst})$;
                    if ($b_{2.5d}$) then
                        $dist_{obst} := includeHeightOnPathSegment(dist_{obst},$
                                                  $pathLength, p_{source}, p_{dest})$;
                    $pathLoss \mathrel{+}= o.attenuation * \min(dist_{obst}, o.thickness)$;
            else if ($pe.isReflection()$) then
                $pathLoss \mathrel{+}= att_{refl}$;
            else if ($pe.isDiffraction()$) then
                if ($b_{angle}$) then
                    $pathLoss \mathrel{+}= getLeeAttenuation(p_{last}, p_{curr}, p_{next}, f)$;
                                                  (Equations 2.17 and 2.21)
                else
                    $pathLoss \mathrel{+}= att_{diff}$;
            $p_{last} := pe.hitPoint$;
        if ($b_{2.5d}$) then
            $pathLength := includeHeightOnFullPath(pathLength, p_{source}, p_{dest})$;
        $pathLoss \mathrel{+}= getFreeSpacePathLoss(pathLength, f)$; (Equation 2.6)
        $pathLoss_{lowest} := \min(pathLoss_{lowest}, pathLoss_{curr})$;
    $predictedRSSI := tx_{pwr} - pathLoss_{lowest}$;
    return $predictedRSSI$;
```

5.3.3 GPU-based 3D Ray Tracing Model

Generating an accurate wireless model is extremely computation intensive, with a complexity that drastically increases with the amount of obstacles present in the scenario. In general, it is possible to optimise this process by trying to either identify which rays might contribute more significantly to the received signal or testing which intersections with obstacles are relevant, discarding irrelevant ray paths. An alternative solution to handle the problem complexity is to exploit current GPU parallel architectures.

The computation of each individual ray departing from the source is, indeed, independent from the others until it reaches the destination. Therefore, it is conceivable to parallelise the computation of each ray path towards the destination and offload the processing to the thousands of cores available in current GPUs. Not only this makes it much faster to compute the properties of the signal propagating from a source to a destination, but it allows to avoid prefiltering specific ray directions in favour of a full exploration of the space. As a consequence, rays instead are launched by the sender in all different directions and their paths tracked whether reaching the destination. Ultimately, the use of GPUs makes modelling wireless communication in 3D environments possible. In the remaining of this section, an own design for a GPU-based ray tracing engine, subsequently called 3D GPU, is introduced, utilising Aparapi [m4] in order to compile Java code into more efficient Open Computing Language (OpenCL) [p109] code able to run directly on GPUs. To exploit the characteristics of nowadays GPU architectures, the ray tracing engine introduced in Section 5.3.2 needs to be modified.

Environment Adaptation

First of all, the two-dimensional environment description needs to be extended to include height information for all elements including doors and windows; walls can already be assumed to reach the ceiling; floor and ceiling need to be explicitly defined. This approach does not require a pure and detailed three-dimensional representation for each obstacle. The basic two-dimensional information on the environment necessary to operate PHY is sufficient for transferring these into a three-dimensional representation by providing the mentioned additional details. Even if these details are missing, a usage in 3D GPU is still possible when applying default values, fitting most indoor environments. Pursuing the basic goals to keep the approach practical and the computational effort low, the internal representation of the environment is restricted to a few primitive solids and surfaces, allowing also fast collision tests in three-dimensional space. Receiving devices are represented as spheres, obstacles as boxes, diffraction edges, floor and ceiling as planes. Boxes are in turn decomposed into rectangular surfaces as well. Although more complex solids and shapes could be beneficial for path finding, it is refrained from including these in 3D GPU, as it would complicate collision checks and increase the required effort to provide an accurate representation of the environment.

Architecture Adaptation for GPU Execution

The libraries of Aparapi provide methods to control the execution of instructions on a GPU while hiding the code conversion and execution process from the developer. Simplified, the basic flow comprises three steps. Before execution, relevant data, e.g., parameters or structures for storing results need to be transferred from the Java program environment stored in RAM to the GPU memory. The code is then executed by specifying the number of instances to generate performing concurrent computation tasks, which are assigned to the available GPU cores. Afterwards, the data has to be fetched back from the GPU to enable further processing. GPU-executable code is written by extending a given Java class and implementing the required algorithms. Unfortunately, the conversion to OpenCL code does not support object oriented design as OpenCL and the associated performance advantages are based on simple data types like 32-bit integers or floats and one-dimensional arrays of these. Thus the mostly object oriented internal data representation of PHY needs to be entirely adapted to match these requirements, shifting the focus to a memory-centric perspective for the architecture design. Due to a lower abstraction level compared to Java, having a much simpler memory management, a careful planning and separation of memory space become necessary. During parallel execution, all GPU cores share and work on the same data stored in a continuous memory space. Coordinating proper access is crucial to avoid overlaps, race conditions and erroneous results.

In 3D GPU, relevant and constantly changing information on the computation of a ray path, which were distributed on multiple objects in PHY before, are now consecutively joined in one data block of known size. Enabling to keep track of multiple paths simultaneously, these blocks are then arranged one after the other yielding a large data array which is passed to the GPU memory. This clear structure allows to assign each computation instance its own range within the array for writing, addressed by its unique and successive id. Less critical read-only information such as obstacles or configuration settings, are passed in separate arrays in a similar way. However, the usage of static array structures introduces several challenges.

As upon creation the size of an array has to be known, the maximum number of possible rays needs to be calculated. Within each computation round, i.e., following the direction of a ray until hitting the next obstacle, the number of rays can at most double (Section 5.3.3). Based on the number of launched rays and the maximum number of computation rounds, which are both parameters of 3D GPU, as well as the number of values of a ray data block, the array length is determinable as:

$$length_{raydata} = \#startRays * 2^{\#rounds} * \#dataValues \ . \tag{5.1}$$

Nevertheless, requiring an adequate number of initial rays starting from the sender to cover the three-dimensional space in combination with a sufficiently accurate number of rounds can quickly exceed the array limit. Since a 32-bit signed integer is used as the addressing index, the limit is equivalent to 2^{31} elements. To minimise the required space of a ray data block, by default only the current computation state of a ray is stored instead of its full path. This comprises the current position and direction (three values each for x/y/z axes), the path length so far, the identification number of the last hit obstacle as well as the accumulated losses of obstacle interactions and the entire ray attenuation including the path loss, requiring a total of 10 values for each ray. An overview of the contents of a ray data block is given in Figure 5.5. A tracking of the exact ray path by storing the position of each round additionally requires space for:

$$length_{trackdata} = \#startRays * 2^{\#rounds} * 3 * \#rounds \tag{5.2}$$

elements. Thus, the tracking is an optional feature and can be enabled upon request. To not exceed the boundaries and ensure a sufficient accuracy, the default parameters have been set to 50000 initial rays and 7 computation rounds requiring approximately 2^{26} elements or 3.13 % of the maximum array space. However, with reaching the boundaries of array size, also the memory requirements increase. The above array with 2^{26} elements requires 256 MB of space for ray information and additional 512 MB when tracking the ray paths, preventing more accurate computations of 3D GPU on weaker GPUs with insufficient memory capacities.

Signal Path Computation

Instead of directing rays in specific directions, GPU architectures favour the parallel analysis of rays launched by the sender in all different directions. In contrast to an aimed approach, there is no effort made in directing rays towards the destination. Instead, paths are evaluated based on a trial and error principle, finding valid paths randomly just by the sheer number

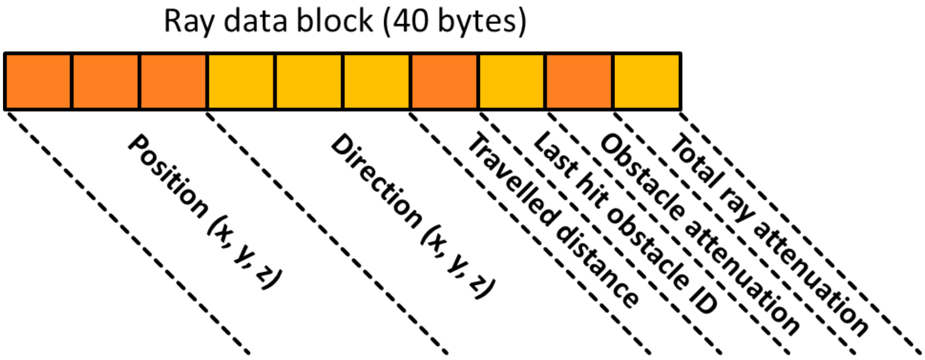


Figure 5.5: *Contents of a ray data block.*

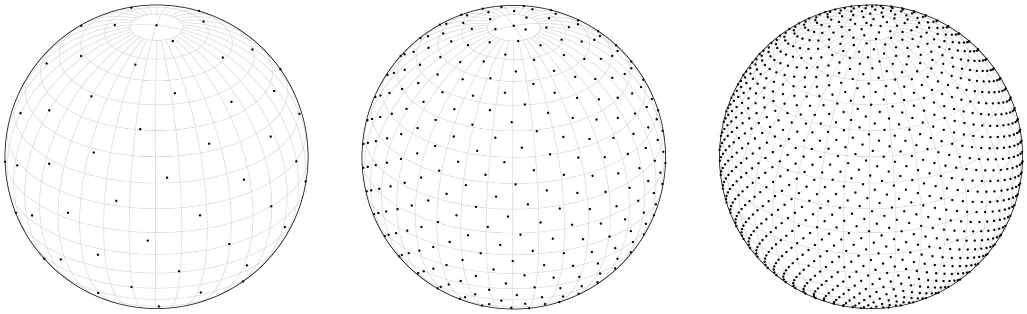
of rays. This randomness on the one hand has the disadvantage of wasting time on a computation of irrelevant rays, but on the other hand also provides the possibility of discovering paths with an arbitrary combination of effects. However, omitting aiming and the corresponding path validation mechanisms (Section 5.3.2) significantly decrease the complexity of computations, enabling fast parallel processing of ray paths, making the risk of tracking irrelevant rays tolerable. Further reducing redundant computations, 3D GPU offers the possibility of considering multiple receivers within one execution, taking advantage of the launching approach.

To provide a good spatial coverage in three-dimensional space and increase chances of reaching the destinations, in addition to a sufficient quantity, the initial directions of the launched rays are important as well. Therefore, the ray directions are distributed according to the Fibonacci lattice [p54], shown in Figure 5.6, which spreads points on a sphere at equal distance in dependency of the total number of rays. In spherical coordinates, the ith point is computed as defined by Equation 5.3:

$$\begin{aligned} lat_i &= \arcsin\left(\frac{2i}{P}\right) \\ lon_i &= 2\pi i \Theta^{-1} = 2\pi i \left(\frac{\sqrt{5}-1}{2}\right) \end{aligned} \tag{5.3}$$

which is translated to a direction vector V_i on a unit sphere with a radius of one by using Equation 5.4:

$$V_i = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos(lat_i) * \cos(lon_i) \\ \cos(lat_i) * \sin(lon_i) \\ \sin(lat_i) \end{pmatrix}. \tag{5.4}$$



(a) $N = 100$ **(b)** $N = 500$ **(c)** $N = 2000$

Figure 5.6: *Fibonacci lattice for different amounts of points.*

Starting with these directions from the sender position, each ray is tracked until it either arrives at a destination, exhibits too large signal losses or leaves the area. In order to counteract the loss of accuracy caused by both the gap between the rays launched from the sender and the distance between the devices, each receiver is represented by a sphere instead of a point. Including the benefits of PHY, a small set of rays are additionally aimed at all receivers to further decrease the chances of missing the reception of a ray due to this discretisation procedure. In particular, the first rays starting from the sender or a diffraction edge are always directed exactly towards the destinations. Indirect, single-reflected paths via floor and ceiling are also considered in this additional set of aimed rays. The discovery of more complex paths is then left to the launching approach. This partial aiming ensures having at least these rays hitting the reception spheres in case all launched rays miss them.

After a first setup, setting obstacle information and initial ray directions and transfer them to the GPU memory, the processing progresses in a sequence of steps, as shown in Figure 5.7. These are repeated multiple times depending on the number of defined computation rounds. In each round, there is an alternation between computations on the GPU and their evaluation on the CPU within the Java program. For each ray, the next collision with an obstacle (if existing) is computed by a single GPU core, updating the ray state depending on the hit object. In general, the ray position is set to the collision point, the direction changed (reflection only), the path length increased by the travelled distance and the obstacle identifier set to the last hit object. By tracking the latter two pieces of information, a continuous update on the expected ray path loss up to the point of computation becomes possible, simplifying the evaluation in a later step. More details on this are explained later. The results are then fetched back to the Java program where the ray data array is traversed and relevant information are extracted before the path computation continues with the next round. This step becomes necessary due to the chosen memory management as the

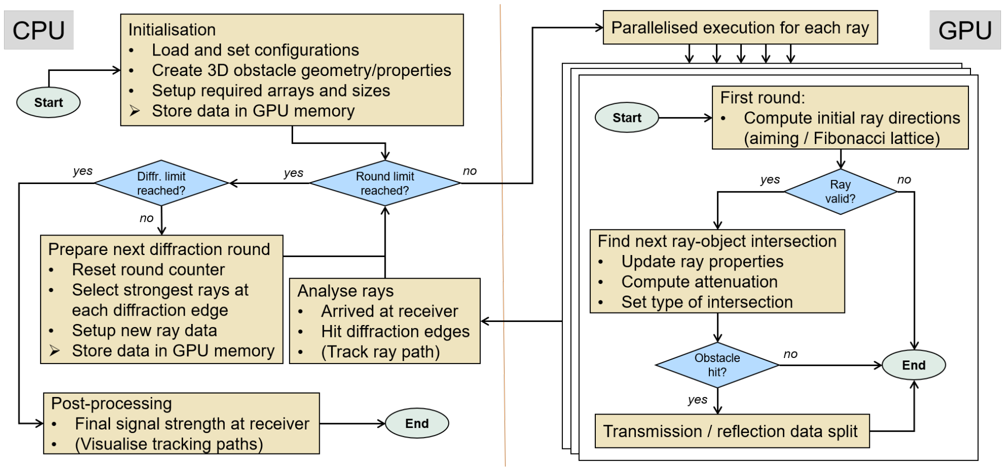

CPU
GPU
Start
Initialisation
Load and set configurations
Create 3D obstacle geometry/properties
Setup required arrays and sizes
Store data in GPU memory
Round limit reached?
yes
no
Diffr. limit reached?
yes
no
Prepare next diffraction round
Reset round counter
Select strongest rays at each diffraction edge
Setup new ray data
Store data in GPU memory
Analyse rays
Arrived at receiver
Hit diffraction edges
(Track ray path)
Post-processing
Final signal strength at receiver
(Visualise tracking paths)
End
Parallelised execution for each ray
Start
First round:
Compute initial ray directions (aiming / Fibonacci lattice)
Ray valid?
yes
no
Find next ray-object intersection
Update ray properties
Compute attenuation
Set type of intersection
Obstacle hit?
no
yes
End
Transmission / reflection data split

Figure 5.7: *Ray path computation steps of* 3D GPU*.*

reserved memory space for each ray is reused and updated in each round. Relevant information needs to be extracted before being overwritten, particularly ray interactions with virtual objects not representing solids in reality and therefore also not affecting the state of a ray directly, i.e., reception spheres or diffraction edge planes. To minimise memory transfers, except for small configuration data, e.g., the counter of the current round, data is only read from the GPU memory without any changes. After performing collision tests for each ray, four outcomes are possible, described in the following.

Signal Reception Upon hitting a reception sphere, the ray data is stored in a receiver-specific list sorted by the estimated path losses. This step is performed within the Java program on the CPU. Since the expected path losses are computed and updated on the GPU during path tracking, further computation steps are omitted, keeping the CPU time of the array traversal quite low. Due to the possibility of reaching further destinations, path computations on this ray are resumed by the GPU in the next round.

No Collision and Weak Rays Rays leaving the scene without collisions or having incurred too many losses on their path are detected during computations on the GPU. In both cases, these rays are irrelevant for the current signal prediction and thus are marked as invalid to not be considered in subsequent steps anymore. Especially in later rounds, the amount of these types of rays rapidly increases, shortening the computation time despite an increasing amount of total rays.

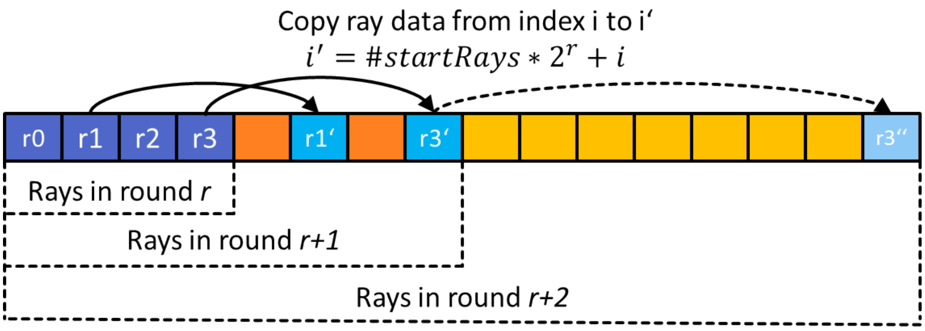


Figure 5.8: *Ray data replication strategy and array usage in dependency of the round of computation.*

Transmissions and Reflections If an obstacle is encountered, a split occurs between the transmitted and the reflected part of the original ray, each of which is processed by a different GPU core in the next step. Thereto, the ray state is copied to an empty array position, still unused in the current round r, according to Figure 5.8.

The attenuations resulting from the interaction with the obstacle are then computed based on either of the two effects, as done in PHY. For transmissions, the ray direction is kept, the position set to the location after passing the obstacle and the attenuation added based on the material type and distance within the medium. In case of the reflected part of the ray, the direction is changed according to the incident angle on the reflecting surface and the attenuation reflection coefficient is added. The omission of the validation part using the image method simplifies the reflection computations and also eliminates the necessity of post-processing on the CPU. Apart from this, a retrospective validation would only be possible with the chosen memory management if the ray path is tracked.

Diffractions For diffractions, a special case is handled. In fact, each diffracted ray potentially produces a complete new wavefront, which would require a new computation process on its own. In the aimed approach of PHY, only some specific rays of these wavefronts were selected for further investigation, which is not possible when using the rather explorative way of a launched approach. To nonetheless enable the consideration of diffraction effects in predictions, the procedure is simplified by only starting new rays in all directions of the diffraction edge basing on the strongest previous ray reaching this edge. As the aiming is omitted, declaring a single point or, transferred to three-dimensional space, a line as diffraction edge is not sufficient anymore, complicating the determination of potentially diffracted rays. Instead, on each unblocked valid diffraction edge, a rectangular plane is spanned to detect rays hitting this edge, as shown in Figure 5.9.

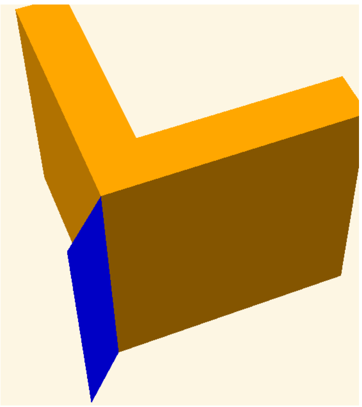


Figure 5.9: *Diffraction edge detection plane (blue).*

After each GPU computation, rays colliding with such planes are stored in edge-specific lists to determine the strongest ray reaching these edges after finishing the entire computation cycle. Depending on the configured number of considered diffractions, the ray tracing computation is repeated that many times, launching new wavefronts based on the positions of the strongest rays reaching each edge. To not require an additional computation cycle for each edge, the available space of the ray data array is split equally among the edges, having $\#startRays/\#edges$ rays to launch from each position. At this point, ray path computations continue in the same way as from the starting position over the specified number of rounds.

Once all computations on the GPU have been completed and the paths to each receiver determined, the remaining step on the CPU is to merge the results which have already been obtained during the various rounds by selecting the strongest ray reaching each receiver, equal to PHY.

GPU Code Execution Algorithm 5.8 shows the ray computation process which is executed on the GPU for each ray. For better readability, the offset computation and handling of the one-dimensional array access has been replaced by an object-like access notation. In the very first round, the values for each ray are initialised and the initial direction determined. Depending on the number of simultaneously handled receivers, the rays with the lowest ids are aimed directly or indirectly via floor and ceiling reflection towards these positions, the direction of the remaining ones are computed according to the Fibonacci lattice. In case of a newly started diffraction iteration, the three-way aiming principle remains the same, but is subject to some restrictions, described later with the diffraction code. Subsequently, the regular part of the processing begins, checking the current ray for validity. The ray computation was either discontinued in a previous step (line 18), immediately stopping the processing or the outcome of the last round needs

Algorithm 5.8 3D GPU ray computation process executed by each GPU core in every round.

```
function RUN($rayID, Rays, Obstacles, Cfg$)
    if ($Cfg.round = 0$) then
        if ($Cfg.rayOrigin =$ ORIGIN_SENDER) then
            if ($rayID < Cfg.receiverCount$) then
                $setDirectionToDest(rayID, Rays, Obstacles, Cfg)$;
            else if ($rayID < Cfg.receiverCount * 3$) then
                $setDirectionToDestViaReflection(rayID, Rays, Obstacles, Cfg)$;
            else
                $fibonacciSpherePointInit(rayID, Rays, Cfg.totalRays)$;
                                                        (Equations 5.3 and 5.4)
            $ray_{curr} := Rays[rayID]$;
            $ray_{curr}.pathLength := 0$;
            $ray_{curr}.loss_{obst} := 0$;
            $ray_{curr}.loss_{total} := 0$;
            $ray_{curr}.lastHitObst := -1$;
        else if ($Cfg.rayOrigin =$ ORIGIN_DIFFRACTION_EDGE) then
            $computeDiffractedRay(rayID, Rays, Obstacles, Cfg)$;

    if $rayIsDone(ray_{curr})$ then
        return;
    $effectCount := countRayEffects(ray_{curr})$;
    $lastEffect := lastRayEffect(ray_{curr})$;
    if (($ray_{curr}.pathLength =$ NOT_COMPUTED)
            $\vee$ ($ray_{curr}.pathLength =$ MAXIMUM_DISTANCE)
            $\vee$ ($ray_{curr}.loss_{total} + Cfg.tx_{pwr} <$ MINIMUM_RSSI)
            $\vee$ ($lastEffect =$ RAY_EFFECT_DESTINATION)
            $\vee$ ($lastEffect =$ RAY_EFFECT_DIFFRACTION $\wedge Cfg.round > 0$)
            $\vee$ ($effectCount > Cfg.maximumEffects$)) then
        $setToDone(ray_{curr})$;

    $distance := findClosestIntersection(rayID, Rays, Obstacles, Cfg)$;
    if ($distance <$ MAXIMUM_DISTANCE) then
        $updateRay(rayID, Rays, Obstacles, Cfg)$;
    else
        $ray_{curr}.pathLength :=$ MAXIMUM_DISTANCE;
```

to be evaluated regarding certain criteria (line 20): (1) the reserved ray slot has not been used (see Figure 5.8), (2) the ray did not intersect an obstacle and thus has infinite length, (3) the accumulated losses exceeded the defined limit for too weak signals, (4) the ray reached a destination position, (5) a diffraction edge was hit and the computation will be resumed in the next iteration or (6) the maximum number of considered effects has been reached. If one of these are fulfilled, the ray computation is marked as completed and the processing stopped as well. In the other case, the ray computation proceeds and the next obstacle interaction is determined (line 25). The distance to the closest intersected obstacle is returned or the maximum distance, indicating no further intersections. Upon a new intersection, the ray data is updated in dependency of the encountered effect, as described in the previous paragraphs and the computation round finished for this ray.

The process of computing the direction of diffracted rays is described by Algorithm 5.9. As previously mentioned, once a new diffraction iteration has been started, the initial direction of the rays has restrictions, similar to the ones used in PHY. This means, the ray should not change its direction by more than 90°. This holds for the aimed rays towards the destination positions as well as for the other rays for which the Fibonacci lattice is used. In case this limit is exceeded, the direction is discarded and a new one is determined until the condition is satisfied, again using the Fibonacci lattice (line 13). However to avoid computing directions which would be regularly assigned to other rays, the total amount of rays passed to the function is halved each time, creating a different distribution of points on the sphere. Afterwards, depending on the actual angle, the diffraction attenuation is added to the obstacle attenuation value. Due to the memory-efficient reusability of ray data arrays, information on the starting point of the previous ray segment are not available anymore, which are however required for computing the Fresnel-Kirchhoff diffraction parameter (Equation 2.17). Hence, instead of Lee's formula (Equation 2.21) an approach using an angle-based increase has been chosen to approximate the attenuation of diffraction for larger angles as depicted in Figure 2.9. The attenuation increases faster for smaller Fresnel-Kirchhoff parameter values ($0 < v < 1.5$) and slower for larger values ($v > 1.5$). Since the angle correlates with distances d_1 and d_2 and thus with v regarding the sharp-edge diffraction computation, the transition angle for the curve flattening has been set to 35° for adaptation. With a default attenuation of diffraction $att_{diff} = 10$dBm, the attenuation curve is approximated. If, nonetheless, no valid direction has been determined, the ray is discarded and not further processed.

Algorithm 5.9 3D GPU ray diffraction direction computation process upon the beginning of a new iteration.

```
function COMPUTEDIFFRACTEDRAYDIRECTION($rayID, Rays, Obstacles, Cfg$)
    $ray_{curr} := Rays[rayID]$;
    $dir_{old} := ray_{curr}.direction$;
    $totalRays := Cfg.totalRayCount$;
    if ($rayID < Cfg.receiverCount * 3$) then
        if ($rayID < Cfg.receiverCount$) then
            $setDirectionToDest(rayID, Rays, Obstacles, Cfg)$;
        else
            $setDirectionToDestViaReflection(rayID, Rays, Obstacles, Cfg)$;
        $dir_{new} := ray_{curr}.direction$;
        $angle :=$ Vector.$angle(dir_{old}, dir_{new})$;

    while (($angle > 90$) $\wedge$ ($totalRays > 0$)) do
        $fibonacciSpherePointInit(rayID, Rays)$;
        $dir_{new} := ray_{curr}.direction$;
        $angle :=$ Vector.$angle(dir_{old}, dir_{new})$;
        $totalRays$ /= 2;

    if ($angle \leq 90$) then
        $ray_{curr}.loss_{obst}$ += $(1 - \frac{90-angle}{90}) * Cfg.att_{diff}$;
        if ($angle \leq 35$) then
            $ray_{curr}.loss_{obst}$ += $(1 - \frac{35-angle}{35}) * Cfg.att_{diff}$;
        else
            $ray_{curr}.loss_{obst}$ += $Cfg.att_{diff} + (1 - \frac{angle-35}{65}) * Cfg.att_{diff}$;
    else
        $ray_{curr}.pathLength :=$ NOT_COMPUTED;
```

Optimisations The necessary clear memory separation between the different execution instances makes splitting of rays and building path trees restrictive, limiting the possibilities for optimisation as well. From the mentioned optimisations of PHY, 3D GPU implements the angle dependent attenuation on diffraction edges and implicitly incorporates height information by performing computations in three-dimensional environments. Precomputations are limited to surface decomposition of obstacles represented by boxes and the generation of diffraction edges. Visibility trees for building fast reflection or diffraction paths are expendable when not using an aimed approach. Finally, as previously mentioned, realising three-dimensional radiation patterns for individual devices at a sufficient resolution is impractical and thus not realised.

5.3.4 Statistical Modelling

A statistical model STAT uses data traces as input to generate a description of the observed feature, e.g., communication, using the physical location of the observations as anchor points. Such traces can be gathered also during system lifetime, refining the available information in case of system or environment changes. At the observed positions, the model reproduces the measured behaviour with high accuracy. To provide estimates at unknown positions, STAT interpolates the information based on the distance from known data points. For MOVE in particular, Kriging interpolation [p87] is applied, which uses a Gaussian process to model and predict intermediate values between known data points. The method has been adopted to solve a wide range of problems, e.g., in geology to assess the recoverable amount of ore in deposits [p43] or in WSNs to interpolate sensor data [p155].

One propagation model for each sender is computed. The RSSI traces of messages received from one sender serve as training data for the computation of the model. The Kriging interpolation method is adapted, instead of performing a linear interpolation between known points, a logarithmic decay with distance is applied as predicted by the radio signal attenuation model in open space [p123]. An example of the model for one sender, calculated from data collected in WSN Testbed described in Section 8.1, is reported in Figure 5.1 (b). The model smoothly interpolates the known values at the different receivers, independently from the actual environment shape. For the latter reason, the STAT model is implemented for solely operating in the two-dimensional space. In contrast to PHY, including height information provides only minor additional benefit to this model compared to the costs of three-dimensional computations. At most, the predicted signal strengths slightly decrease due to increasing distances related to possible height differences, similar to the 2.5D optimisation of PHY (Section 5.3.2). Other than that, the integration of height information does not have any beneficial effect on the interpolation process of STAT, which is solely based on the recorded traces as data source, not considering any resulting consequences of an additional dimension with respect to the environment except for the potential change of distance.

For the integration into STAT, a publicly available implementation of the Kriging interpolation has been used [m3]. In order to use Kriging for performing predictions on the signal quality, certain operating parameters have to be configured first. The Kriging interpolation bases its value estimations at unknown positions on semivariograms. A semivariogram is a function $\gamma(s_i, s_j)$ describing the relation of similarity between two locations s_i and s_j in dependency of their distance. In general, the closer two locations are spatially, the more similar their values are and vice versa. Hence, in com-

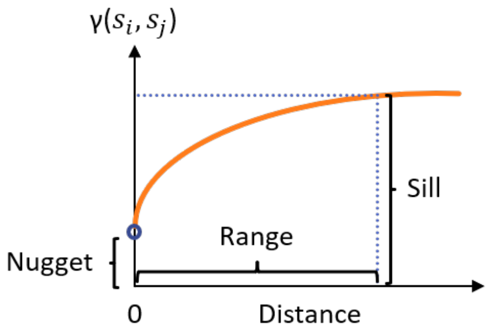


Figure 5.10: *A semivariogram with relevant parameters.*

bination with given data at known positions, the semivariogram defines the expected difference of values from these known position at a specific distance. The semivariogram used by STAT is based on empirical data and uses three key parameters: the range, the nugget and the sill. These parameters define the course of the graph, as depicted in Figure 5.10, and have to be adapted to the needs of signal modelling. The nugget describes the initial dissimilarity at zero distance and represents the initial variation or noise occurring at infinitely small distances. In terms of wireless measurements, it would be comparable to measuring the signal close by the position, it is originating from. However, even when measuring closely, the signal already suffered from a loss, which is reflected by this value and has been set to -10 dBm. For comparison, according to Equation 2.6, the path loss at a distance of 5 centimetres is already -14 dBm, justifying the above value. The second parameter, the range, describes the distance at which the graph levels off and the similarity change over distance becomes negligible. Transferred to wireless signal propagation, this means a distance at which the signal is already so weak that it is actually irrelevant as it is not received at all. Thus, the range has to be chosen in dependency of the reception capabilities of the used hardware. For the indoor scenarios used in the evaluation (Section 8.1), a range of 30 metres is sufficient, as the attenuations on the path caused by the obstacles do not allow any links beyond that distance. Lastly, the sill is the value, which the semivariogram function outputs at the defined range. Taking into account the explanations on the range, this value corresponds to a signal strength which is not measurable by the devices anymore. Since for the hardware used in the main evaluation scenarios, the lowest receivable RSSI is -97 dBm, a value of -100 dBm has been chosen. Alternatively, also the lowest received RSSI value from the measurements can be taken as reference to determine the lower bound.

5.3.5 Hybrid Modelling

The STAT and PHY models are either based on system or on environment information exclusively, resulting in the aforementioned discrepancies. However, wireless communication is affected by the environment and modelling approximations result in deviations from the predicted physical propagation. In HYBRID, the two modelling techniques are fused together, exploiting the benefits of both to generate synergies.

Optimisations The known data points in the STAT model are exploited to first calibrate the parameters of the PHY model. Configuring the key parameters of the model, e.g., the specific impact of materials on the signal propagation, is challenging, yet crucial to achieve a good accuracy. Starting from the gathered observations and the approximate descriptions of the obstacles present in the environment, HYBRID adjusts the individual parameters of the PHY model to closely match the actual values for all observed positions. This is done through a gradient descent on the accumulated squared error of all the measured receivers for a single sender through a gradual change of the individual parameters. To preserve a low computation time, alternative, more complex techniques are avoided. Anyway, even in case of a sub-optimal configuration, the described simple procedure still contributes to compensate for the errors introduced by an imprecise description of the environment.

The parameter optimisation for a specific node n_{opt} is described by Algorithm 5.10. For the exploration, a parameter interval (PI) helper class has been defined which defines the parameter to change, default and current value, upper and lower limits and the step size in which it is iterated. An interval can be defined for any parameter of PHY which should be optimised. As a reference, the model initially predicts the link qualities between the known devices (N) using the default configuration parameters. For each link, the difference between prediction and measured value (L_{real}) is computed using the same metrics as defined in the evaluation (Table 8.8). To penalise larger deviations more than smaller ones, the value is squared before it is added to the accumulated difference, which is then used for comparison to determine whether an improvement has been achieved. The parameters are now changed repeatedly one by one in either ascending or descending order as long as an improvement occurs, finishing after one full iteration over all parameters without any changes.

Moreover in PHY, small changes of a few centimetres in the position of a node can significantly impact the prediction accuracy. Figure 5.11 exemplarily visualises the impact of small distance changes on the reception probability of a device located at border distance from a sender as predicted

Algorithm 5.10 HYBRID parameter optimisation.

```
class ParameterInterval {
    Parameter $parameter$;
    Float $default, lower, upper, step, current$;

    function nextUp()
        return $\min(upper, current + step)$;
    function setUp()
        $current := \min(upper, current + step)$;
    function nextDown()
        return $\max(lower, current - step)$;
    function setDown()
        $current := \max(lower, current - step)$;
}

function optimiseParameterForNode($n_{opt}, N, L_{real}, PI, mdl$)
    for ($pi \in PI$) do
        $mdl.changeParameter(pi.parameter, pi.default)$;
    $L_{pred} := mdl.predictLinksForNode(n_{opt}, N)$;
    $diffToReality_{best} := getSquaredDiffSum(L_{real}, L_{pred})$;
    $isFinished := false$;
    $lastImprovedParam := null$;
    while ($\neg$ $isFinished$) do
        for ($i \in \mathbb{N}, 0 \leq i < PI.size()$) do
            $pi := PI.get(i)$;
            if ($lastImprovedParam = pi.parameter$) then
                $isFinished := true$;
                break for loop;
            if ($lastImprovedParam = null$) then
                $lastImprovedParam := pi.parameter$;
            $status_{param} :=$ INCREMENT;
            while ($status_{param} \neq$ DONE) do
                $hasImproved := false$;
                if ($status_{param} =$ INCREMENT) then
                    $mdl.changeParameter(pi.parameter, pi.nextUp())$;
                else if ($status_{param} =$ DECREMENT) then
                    $mdl.changeParameter(pi.parameter, pi.nextDown())$;
                $L_{pred} := mdl.predictLinksForNode(n_{opt}, N)$;
                $diffToReality_{curr} := getSquaredDiffSum(L_{real}, L_{pred})$;
                if ($diffToReality_{curr} < diffToReality_{best}$) then
                    $diffToReality_{best} := diffToReality_{curr}$;
                    $hasImproved := true$;
                    $lastImprovedParam := pi.parameter$;
                if ($hasImproved$) then
                    if ($status_{param} =$ INCREMENT) then
                        $pi.setUp()$;
                    else if ($status_{param} =$ DECREMENT) then
                        $pi.setDown()$;
                else
                    if ($status_{param} =$ INCREMENT) then
                        $status_{param} :=$ DECREMENT;
                    else if ($status_{param} =$ DECREMENT) then
                        $status_{param} :=$ DONE;
```

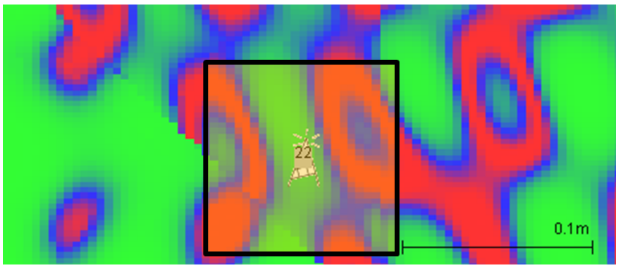


Figure 5.11: *Reception probability map of a signal emitted from a sender (not shown here) at border distance to the depicted receiving node predicted by the* PHY *model. A change in position of few centimetres decides on a successful reception. The box around the node represents the area in which nodes are relocated within the model to best match observations.*

by the PHY model. Since even detailed representations of the environment may contain small inaccuracies, nodes are moved in the range of a few centimetres around their location (box in Figure 5.11) to find the best positioning matching the measurements. The benefits of these correction as well as the parameter optimisations are evaluated in Section 8.4.4.

Weighted Output Creation After calibration, in order to combine the STAT and PHY outputs, the most informative model at each specific location has to be identified. For this reason, the HYBRID model determines a weight for each output based on the system and environment information available. In particular, due to the interpolation, the accuracy of the STAT model decreases as the physical distance from known points increases. Then the absolute distance of position p to the closest known observation $d_{min}(p)$ is used as weight. Subsequently, three regions depending on $d_{min}(p)$ are distinguished. At a distance smaller than a threshold d_{stat}, only the STAT model is considered. Similarly, at a distance bigger than a threshold d_{trans}, exclusively the output of the PHY model is used. In the transitional region in between, each model is weighted so that the contribution of the STAT model degrades linearly with distance while the PHY one increases. This reflects the ability of the STAT model to accurately describe only the close surrounding of observed positions.

Algorithm 5.11 describes the decision process on which type of model to use with which weighting depending on whether for the given positions of source and destination measurements are available. If both positions are known, then STAT is used without any involvement of PHY. In the opposite case without any measurements for both positions, PHY is the only option as STAT does not have data to perform predictions. For the remaining and more common situations, in which only one of two positions are known, a weighted combination of both model outputs is chosen, based on the aforementioned distance to other known positions.

Algorithm 5.11 HYBRID weighted decision algorithm.

```
function COMPUTEHYBRIDPREDICTION($p_{source}, p_{dest}$)
    $sourceKnown := isKnownPosition(p_{source})$;
    $destKnown := isKnownPosition(p_{dest})$;
    if ($sourceKnown \wedge destKnown$) then
        $predictedRSSI := computeStatPrediction(p_{source}, p_{dest})$;
    else if ($sourceKnown$) then
        $weight_{\mathrm{STAT}} := determineWeight(p_{source}, p_{dest})$;
        $predictedRSSI := computeStatPrediction(p_{source}, p_{dest}) * weight_{\mathrm{STAT}} +$
                $computePhyPrediction(p_{source}, p_{dest}) * (1 - weight_{\mathrm{STAT}})$;
    else if ($destKnown$) then
        $predictedRSSI := computeHybridPrediction(p_{dest}, p_{source})$;
    else
        $predictedRSSI := computePhyPrediction(p_{source}, p_{dest})$;
    return $predictedRSSI$;

function DETERMINEWEIGHT($p_{source}, p_{dest}$)
    $d_{known} := getClosestKnownPositionDistance(p_{dest})$;
    if ($d_{known} \leq d_{stat}$) then
        return 1;
    else
        $weight_{\mathrm{STAT}} := \max(0, (d_{trans} + d_{stat} - d_{known}))/d_{trans}$;
        return $weight_{\mathrm{STAT}}$;
```

Referring back to Figure 5.1, the resulting weighted computation of the wireless propagation for a sender is shown in (d), according to the implementation of HYBRID, combining the outputs of STAT (b) and the optimised implementation of the PHY model (c). The PHY model is driven by the obstacles, e.g., doors and walls. With respect to the STAT model, the difference is evident. In particular, STAT shows that some devices are unable to communicate with the sender even if PHY predicts a perfect link. On the other side, the STAT model expands areas with bad connectivity beyond the physical obstacles that most likely cause them. To address such characteristics in HYBRID, more complex weighting functions can also be used, e.g., to analyse the distance to obstacles, whose impact might be better estimated by PHY. These functions would, however, significantly increase the required processing time, striving against the set goals. In Section 8.7, the accuracy of the weighting function is discussed, showing its ability to support the design of effective network optimisation strategies.

5.4 Conclusion - MOVE

In this chapter Models for Network Optimisation in Versatile Environments (MOVE) are introduced, a collection of models for predicting wireless signal propagation. As the core contribution, MOVE provides four models, serving as basis for the other approaches presented in this thesis. While pursuing the principles of this thesis to keep all approaches practical and generally applicable without requiring expert knowledge, each model provides a different approach addressing specific conditions regarding the availability of information found in diverse networks. Thus, MOVE provides suitable solutions for a large number of scenarios.

First, the PHY model is introduced, a two-dimensional ray tracing model basing predictions entirely on a description of the environment. By applying physical laws of signal propagation in interaction with obstacles, PHY computes possible signal paths from a sender to a destination in order to estimate the expected signal strength at arbitrary positions. Being executed on a CPU, PHY is limited to a certain degree of detail as an increasing number of considered interactions exponentially increases the computation effort. To mitigate this limitation several optimisations have been implemented either reducing computation times without sacrificing much accuracy, e.g., simplifying path computations by skipping certain less relevant signal paths, or vice versa, e.g., including optional height information adjusting path lengths computed on the 2D plane. As the evaluation has shown (Section 8.4), PHY is able to provide accurate predictions of signal propagation with a sufficiently detailed representation of the environment. In the main scenario WSN TESTBED an average error per existing link between 7 dBm for the lowest and 8.5 dBm for the highest power level is achieved. In FLOW [a2] (Chapter 4), PHY is used to compute the signal quality of simulated networks from which the network topology has to be estimated in challenging artificial scenarios.

Taking advantage of the independence of each ray in computation, a 3D GPU model has been developed, tailored to the functionality and utilising the massive parallelisation capabilities of current graphics cards. It builds upon principles of the PHY model and extends it by adding a third dimension as well as increasing the number of considered effects, relaxing the restrictions, which approaches executed on the CPU suffer from. Executing 3D GPU on a modern graphic card, it turns out that computing 3D models of the environment in which the devices operate can be made practical and that its processing can be accelerated without sacrificing accuracy. The 3D GPU model is utilised in FOLLOW [a4] (Chapter 6) to identify possible reasons for observed changes in signal strength. An incorporation of the third dimension allows to discover further possible signal paths relevant to improve prediction accuracy as well as to better identify dynamic obstacles accountable for changes in signal quality.

Having no detailed description of an environment available, the STAT model enables predicting signal quality by basing entirely on observations taken from the network. To generate signal maps for each sender, it utilises Kriging interpolation calibrated with collected traces and applies logarithmic signal decay to determine expected RSSI values for unknown positions. Considered exclusively, the STAT model is able to perfectly reproduce communication behaviour for a given network with smooth transitions of signal quality between known positions. Due to its data-based nature, it does not account for any local environment properties, exacerbating predictions at positions with sparse trace coverage. However, the prediction accuracy for such uncovered positions can be improved by integrating the missing knowledge about the environment via complementary PHY model.

The last model of MOVE, HYBRID fuses the available information about the scenario together with observations taken by deployed devices into a single model describing wireless communication at each possible location in the environment. Consisting of STAT and PHY models, it takes advantage of both approaches by weighting their outputs, depending on the distance to known locations. Furthermore, incorporating recorded traces allows for a minimisation of error by serving as validation when calibrating the PHY model parameter to adjust its predictions towards observed values. The evaluation shows that HYBRID is able to exploit the most accurate information of both models available at each location with a low computational effort. MOTION [a3] (Chapter 7) then uses predictions of the HYBRID model to improve network topologies towards specific user goals by relocating devices in the environment.

In the overall view of this thesis, MOVE constitutes the second phase of increasing the knowledge about a network. Considering the individual circumstances of each scenario, the different models of MOVE provide flexible approaches for modelling wireless communication. Independent of the requirements specific to each model, a basic understanding of the location of network devices is necessary to enable predictions of the communication behaviour. These positions can be provided by FLoW with an approximate accuracy, suitable for modelling, which does not consider the environment directly, as for example the statistical model. However, for utilising the more precise physical models, positions have to be provided in a more exact way, so that they at least match the rooms, given by the likewise required description of the environment, which has to be provided by the user. Due to its ability to predict network behaviour at unknown positions, MOVE represents a key element of this thesis as it marks a significant transition from the mere collection of knowledge to an actual exploitation. This capability of prediction opens up opportunities for more advanced approaches obtaining a deeper understanding of the network and its properties. This is especially relevant for the two following contributions, FOLLOW and MOTION, as these exploit the predictions of MOVE to achieve their goals.

6 Understanding Changes in the Environment

Wireless communication is becoming an integral part of our everyday life. Not only personal devices but also industrial equipment are embracing this vision with more and more integration of wireless elements. However, while for personal use an unreliable wireless connection is acceptable (even though admittedly annoying), industrial setups require known properties of the interconnection between the different components. An increasing number of interactions with domain experts highlights how reliability, timeliness and robustness are key requirements that cannot be traded off. Understanding wireless communication properties becomes, as a consequence, an essential challenge that needs to be undertaken.

This need was recognised since long time, motivating a plethora of experimental studies (Section 3.2). These studies have, over time, replaced modelling and simulation as the reference methodology to acquire knowledge about wireless communication. Arguably, such approach was justified by the experienced discrepancy between modelled environments and real ones, corresponding to a significant mismatch between expected behaviour analysed in simulation and measured performance of real systems. As a result, the design, deployment, validation and maintenance of systems nowadays rely mostly on factual, single knowledge and individual experience [p173].

In this chapter, modelling is resumed as key methodology to reason about the behaviour of a deployed system situated in its operational environment. The goal is to look at communication in indoor scenarios where obstacles impact wireless signals in unpredictable ways depending on their shapes and construction materials. Moreover, such unforeseeable effect on network performance can change over time depending on the different states and positions of such objects during the system lifetime. A dynamic model able to capture the cause-effect relationship between changes in the scenario and the resulting variations in the wireless signal propagation in the specific operational environment would allow to reason about the network behaviour, exposing new knowledge.

6.1 Challenges

Defined by the goals and depending on the available knowledge about the environment, investigations of temporal network changes have to be conducted in different ways. Focusing on individual, predetermined objects or properties within limited boundaries, e.g., the movements of an elevator on a certain floor or the temperature in a specific room, experiments yield the most information, especially when having full control over the setup. Being able to adjust and iterate through possible states, an exhaustive analysis can be performed to gain insights on the subject. In this case, however, the cause of the observed effects is known beforehand and an assignment has already been anticipated. This important step of assigning observations to their actual cause has to be undertaken if either more general investigations without focus on specific objects are targeted or site-specific information have to be gathered at a larger scale in lack of resources to perform detailed investigations of each individual object in the whole area.

This expansion of investigations to a larger scope or area imposes two challenges regarding the combination of generality and efficiency. First, to perform detection and assignment in such a way that it works with few information and ideally requires no user intervention once in operation. Second, to conduct an investigation obtaining the most comprehensive insights despite uncertainties in the assignment.

Enabling Automated Cause Assignments

In order to realise an approach which is able to determine cause-effect relationships, certain prerequisites have to be fulfilled. Implementing an automated process, the approach has to compare and analyse measurements taken from the network in order to detect individual changes on link level, requiring a distinction between usual fluctuations in the link quality and those induced by events. Depending on the used hardware and scenario, the boundary between the two might be different. Those individual links which are then classified to be significantly changed have to be correlated afterwards to determine the location of an observed event. Since a localisation is crucial to clearly identify the reasons for network changes in a subsequent step, the usage of a detailed geometric model of the environment, including device positions, is obvious. Finally, combining the information from previous steps, the approach needs to assign the observations to a cause, making a decision between available options. This requires a decision function, which has to be defined based on the properties of the potential types of causes.

An automated assignment, however, has limitations. A system is only capable of acting within its own boundaries, i.e., recognising and assigning observations to causes which are known. The availability and utilisation of knowledge about possible influencing factors are therefore essential for a successful deployment. Consequentially, the properties about the underlying causes of the events to be detected have to be known beforehand so that such an approach is able to make unambiguous assignments at all. Obviously, the actual reasons for changing network behaviour might vary depending on the scenario. In order to cope with this multitude of potential reasons, an automated solution ideally should incorporate as much of this general knowledge as possible to ensure broad applicability. Additionally, information specific to the environment of the scenario are required as well, e.g. the position of doors, windows or furniture. Otherwise, without sufficient scenario data, only imprecise guesses based on the general knowledge would be possible.

Uncertainty of Cause Determination

The second challenge is the exploitation of available information to achieve the best possible knowledge gain for a given scenario. At the simplest, a newly detected event and its fingerprint match with a previously observed event where the cause is known. However, besides a confirmation of previous data, this case does not provide much gain in knowledge about the environment. The more interesting and valuable case is the opposite, observing events with a yet unseen effect on the network. When examining certain objects as possible causes, fingerprints regarding the actual effect of these on the links in their vicinity might not be available at all if events related to them have not been observed before. To perform a cause-effect assignment in this case, an approach has to estimate the impact based on an underlying network communication model and select the most probable. Without any confirmation on the correctness, in particular for an automated assignment, this decision is subject to uncertainty caused by inaccurate estimations due to incomplete or imprecise information about the environment. To ensure correctness and usefulness of the insights, this uncertainty has to be minimised.

6.2 Approach

Unexpected temporal changes originating from the environment can considerably affect the performance of a network. Hence, having as much knowledge as possible about potential causes is helpful to foresee and avoid adverse effects. As previously mentioned, especially in large networks it is difficult to perform exhaustive experimentations revealing the complete range of possible influences. To address this problem, such causes can be investigated by analysing the impact of certain objects in a virtual environment. Having the appropriate tools to model wireless communication in a given environment, their exploitation opens up such opportunity to gain further knowledge about the underlying network as well as the environment itself. During the computation of signal propagation paths, physical models use a snapshot of the respective network and environment reflecting their current state. However, since the communication behaviour of most networks is subject to frequent changes, investigating these temporal dynamics with the help of a model can provide additional insights. Given the model estimations on the impact of different objects in different states, observations of previously unknown changes in the communication behaviour of the network can then be matched against their possible causes.

Realising the aforementioned, this chapter introduces Follow the Rays (FOLLOW) [a2], utilising the ray tracing engines of MOVE (Chapter 5), which is able to efficiently compute the propagation of the wireless signal in an indoor environment from a basic floor map. Starting from this base, a technique is devised able to detect and localise events producing lasting changes to the interconnection between devices. Through the observation of the actual link conditions and variations, the computed model is exploited to derive which rays could have caused such changes and match them to the provided floor map, thus understanding the cause. While alternative event detection approaches exist, FOLLOW extends the state of the art by exploiting models to map the events and match the corresponding unique fingerprint, providing the following contributions:

Robust Model Signal Impact Determination In order to determine the potential influence of an obstacle on the communication behaviour, an intuitive approach would be to predict the signal strength for each link in presence and absence of the investigated obstacle and then use the differences across all affected links as unique fingerprint. However, especially for scenarios with coarse descriptions of the environment lacking centimetre accuracy, small predicted link differences of only few dBm are susceptible to errors, as these might be affected by interference effects which cannot be taken into account correctly. Addressing this issue and therefore enabling a robust determination of obstacle impacts for mentioned scenarios as well, the approach of Follow does not consider a qualitative analysis of link changes, but a quantitative one. The presence of necessary basic details of the environment given, i.e., device positions, walls, large furniture, doors and windows, just a coarse distinction between strong and weak links is made to either keep or discard links. Focusing only on strong and meaningful connections when analysing changes, those links are considered to be affected by an obstacle for which a sufficiently strong signal path exists, involving that obstacle. As a consequence, the obstacle fingerprints exclusively base on the affected links, but not on their actual change, abstracting from otherwise required details for interference computations and thereby improving the applicability of Follow.

Model-based Cause-Effect Reasoning Utilising the ability of physical models to alter the state of certain elements within the virtual representation of the environment, the impact of such changes can be estimated by Follow before they actually occur in reality. Once observed in the operating network, the subsequent comparison constitutes an implicit verification at the same time, if the actual link changes match the predicted ones. Simulation and reality are combined to enable a model-based analysis of network behaviour without the need to perform dedicated experimentation campaigns on exploring the effect of certain obstacles located in the environment. In this way, Follow provides an automated approach to understand the causes originating from changes in the environment, increasing network knowledge and eliminating the manual effort that would otherwise be required to gain such insights.

The concept of Follow is visualised in Figure 6.1, describing the two-folded process of obtaining fingerprints of events observed in the real network (left) and of obstacles based on model estimations (right) to ultimately perform a matching between an event and its possible cause. In order to derive event fingerprints from the real network, deviations with respect to the regular network behaviour have to be identified first. Thus, observations on link level become necessary for which changes have to be recognised individually. As such changes occur on different links and in different places with a temporal delay as well, these have to be correlated, yielding sets of links related to an event, which is then used for comparison. On the other hand, for obtaining obstacle fingerprints which reflect the behaviour of the network links after state changes, a description of the environment is required, including the objects to be investigated as well. Changing the position of these objects in the virtual description, the physical models

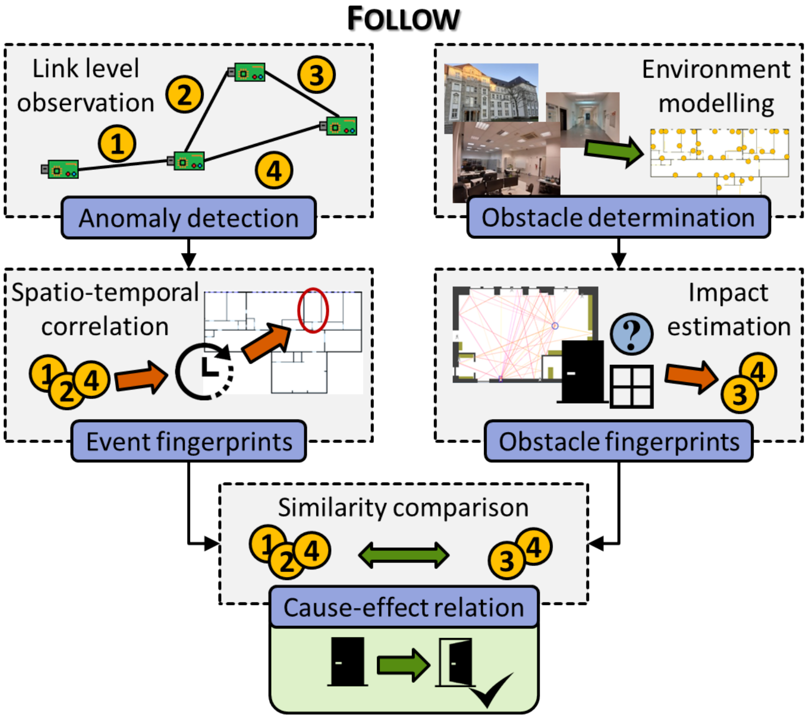


Figure 6.1: FOLLOW *concept of determining the cause-effect relation between environment and observations.*

are utilised to predict the impact on the surrounding links, producing a fingerprint for each obstacle. Comparing the event fingerprint to the ones from the predicted obstacle fingerprints, the highest similarity determines the assumed cause for the observed network changes.

To tackle aforementioned challenges, FOLLOW takes the following measures. Implementing an automated approach for the detection and assignment of events to their potential causes, FOLLOW focuses on semi-dynamic obstacles found in typical indoor scenarios, i.e., doors, windows and elevators. Since the number of possible states of this type of obstacle might be large, but is yet limited, the resulting effects on the network behaviour can be, in theory, entirely determined as well, given the required resources to enable an examination at highest possible detail, which is, though, refrained from. The spatial anchorage of semi-dynamic objects yields a defined amount of attributable objects and thus enables clearly localised assignments in comparison to completely mobile objects which might originate from outside the network boundaries and thus are not part of the environment description. In addition, the spatial limitation of movements allows for a better distinguishability between different obstacles reducing uncertainties.

FOLLOW is evaluated in an indoor testbed consisting of 40 low-power wireless devices (Section 8.6). The ability of the proposed solution to detect events and correctly map them to their location is studied with respect to the number of devices active as well as the employed transmission power, thus exploring the impact of system properties on the detection accuracy. Additionally, the analysis of the alternative models shows that it is possible to increase the computation complexity and gain predictions closer to reality by adding the third dimension at no additional costs in time. By progressively increasing transmission power, it is possible to attribute up to 91% of the measurements to the corresponding obstacles correctly. From this analysis, further use cases are identified where the approach could be exploited (Section 9.2). In conclusion, it is demonstrated that a system can automatically define an event-dependent dynamic model of its wireless communication properties mapped to in-situ gathered fingerprints.

6.3 Implementation

This section focuses on the possibility to understand the interplay between events happening in the environment with long lasting impact on communication and the wireless propagation properties between the devices. Pure statistical models based on measurements would only be able to passively observe changes in the communication properties as events happen in the environment but would not be able to trace back the actual cause of such changes. Only a model able to describe how wireless waves should be affected by a specific event can support this type of cause-effect reasoning.

A base measurement set is assumed available, which provides information about RSSI observations from all the links in an empty environment in its default state, e.g., with all doors and windows closed. Such measurements can be easily gathered at system deployment or at times in which the system is detected being in a quiet state. It is then necessary to gather individual traces for each possible event happening in the environment to complete the description of the links before and after an event. Figure 6.2 exemplarily demonstrates the effect of such an event on the RSSI of different links, opening a previously closed door. While exhibiting fluctuations during the movement from seconds 15 to 20, the selected links show a contrasting behaviour in closed and open door states. These permanent changes in the link qualities represent the unique fingerprint of each event, allowing an assignment to its possible cause as well as a later recognition. As also discussed in Section 8.6.3, this gathering procedure can be integrated in an

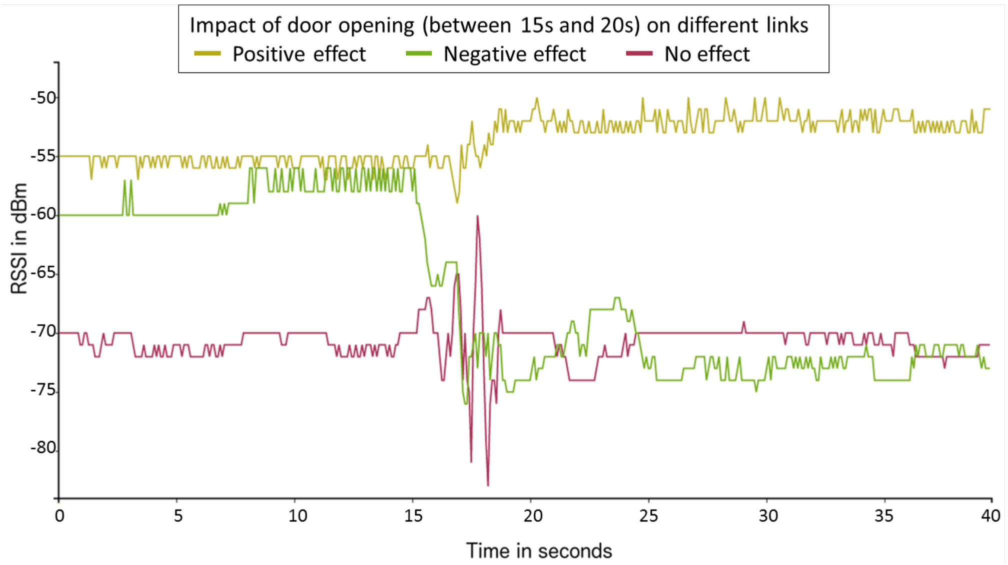


Figure 6.2: *Contrasting influence of door movements on the signal strength of different crossing links.*

online observation system to let systems calibrate themselves. The result is a mapping between localised events and corresponding fingerprints made of real-world wireless observations obtained by the deployed system. In the following, the approach to build such a mapping is described.

6.3.1 Detection of Link Anomalies

Using the base measurements as reference, event fingerprints are created out of temporally and spatially correlated link changes, which manifest a significant variation above a given threshold. An event fingerprint consists of RSSI average and variance values for all the affected and relevant links. In order to identify which links are relevant, the different phases of change in RSSI values during an event have to be identified for each individual link first. Figure 6.3 breaks down the three typical phases of an affected link, beginning with the pre-event phase usually evincing stable RSSI values with minor variance. This phase is followed by the event itself typically characterised by rapid changes in observed RSSI values with a high variance due to the movements of objects in the environment and thereby constantly changing signal paths. Finally, during the post-event phase the link quality might have changed in case of an altered environment in the vicinity of this link, e.g., an opened door. The identification of these phases enables a clear

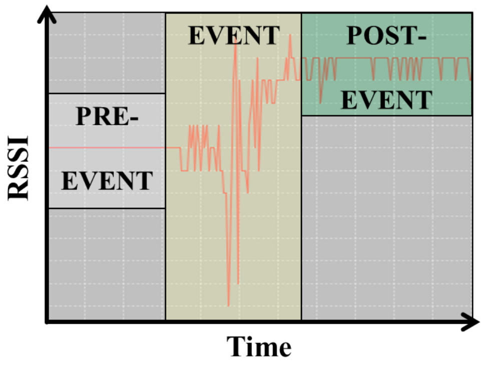


Figure 6.3: *Phases of RSSI change in case of occurring events.*

separation between RSSI samples from before and after an event to compute the difference without possible distortions in the fingerprint from the event itself. Therefore, RSSI samples related to an event have to be recognised and discarded.

To determine beginning and end of an event, an approach is required which is able to detect abnormal deviations in a sequence of collected RSSI samples. In the literature, sliding windows provide an effective and computationally inexpensive solution where RSSI values are successively added in chronological order [c1]. These windows only store the last s_{window} inserted values, while the newest value always replaces the oldest, allowing to analyse the progression of link quality over time (LoT) and to deal with a limited amount of RSSI samples only. Further computations are then performed on the contained values, such as the calculation of the mean value or the aforementioned variance which in turn can be analysed to detect events. In case of the variance, an event is assumed to take place while the value exceeds a certain threshold t_{var}. The parameters, relevant to the variance detection are listed in Table 6.1 and the process is shown by Algorithm 6.1. The anomaly detection algorithm has been taken from [c1] and adapted to the need of distinguishing the previously described event phases by identifying first and last packets during which an event took place. The used parameters, on the other hand, are not changed as these provide distinct detections of the event borders with respect to the inserted RSSI values. Especially for online observation systems and traces containing continuous event transitions, the approach of using sliding windows and observing changes in the RSSI variance is an appropriate way to identify changes of link characteristics and isolate the timespan of occurrence [p25].

Table 6.1: *Parameters and chosen values used by the link anomaly detection as described in [c1].*

Parameter	*Description*	*Value*
s_{window}	Sliding window sample size	10
w_{mean}	Weight of the mean value from the previous step	0.9
t_{var}	Event starting variance threshold	5 dBm

Algorithm 6.1 Link anomaly detection using a variance window as defined in [c1].

```
struct Event {
  Node $sender, receiver$;
  Integer $startPacket, endPacket$;
}

function DETECTLINKANOMALIES($LoT_{(n_1 \to n_2)}$)
    $Events_{(n_1 \to n_2)}$ := new List();
    $ws := s_{window}$;
    $Win$ := new Array($ws$);
    $mean := LoT_{(n_1 \to n_2)}[0]$;
    $e_{start} := -1$;
    for ($i \in \mathbb{N}, 0 \leq i < LoT_{(n_1 \to n_2)}.size()$) do
        $RSSI_i := LoT_{(n_1 \to n_2)}[i]$;
        $Win[i \mod ws] := RSSI_i$;
        $mean := w_{mean} * mean + (1 - w_{mean}) * RSSI_i$;

        if ($i < ws$) then
            continue with next $i$;

        $var := \frac{1}{n} \sum_{k=0}^{ws} (Win[k] - mean)^2$;

        if (($var \geq t_{var}$) $\wedge$ ($e_{start} = -1$)) then
            $e_{start} := i$;
        else if (($var < t_{var}$) $\wedge$ ($e_{start} \neq -1$)) then
            $Events_{(n_1 \to n_2)}.add($new Event$(n_1, n_2, e_{start}, i - 1))$;
            $e_{start} := -1$;
    return $Events_{(n_1 \to n_2)}$;
```

After identifying the time interval, in which an event occurred, the RSSI difference between pre and post-event is computed. Thereby, links that appeared or disappeared as a consequence of the event happening are recognised as well. Out of these links, the ones are preserved which are bidirectional, reliable with at least -90 dBm in both directions and where both directions change remarkably. In the experimentation, a threshold corresponding to the maximum between 2 dBm and twice the standard deviation of the link RSSI demonstrated to be sufficient to identify the relevant links.

Although event detections on a link level are a prerequisite for the following steps, they are not the main focus in the context of FOLLOW, but the ability of the models of MOVE to attribute such identified, affected links to a cause. Thus, the previously mentioned approach for detecting link anomalies is not applied during the evaluation of the succeeding matching process to prevent an interference with possibly erroneously discarded links. Instead, the utilised RSSI measurements for the evaluation have been recorded before and after an event had taken place, achieving a clear temporal separation between pre and post-event states and allowing to directly proceed with the identification of links relevant for the next steps.

6.3.2 Clustering of Links

After being able to detect events on a link level, the next step is to spatially and temporally relate these recognitions to determine all links affected by a specific event and derive the corresponding fingerprint. Thereto, a clustering algorithm is applied, shown in Figure 6.4, consisting of a first cluster generation phase and a subsequent minimisation.

First, detected link anomalies are grouped if their determined time intervals overlap by simply comparing the timestamps of the corresponding packets. Since the movement of an object through the environment might start to affect links at different points in time, especially at slow velocities, it is considered as sufficient that at least one event-related RSSI sample of both links temporally overlap. This approach minimises the amount of generated events, but on the other hand tends to create chunks of links if multiple events occur at the same time. To prevent the latter and distinguish spatially separated events, remaining links are afterwards clustered based on the rooms in which they are located. In the given map, it is possible to identify rooms by searching for areas enclosed by walls. To each room, the position in the map can then be assigned as well as the nodes and obstacles contained. This offers the possibility to better localise events by tentatively assign them to one or more rooms. First, the map is consulted to identify the specific position and room of each node interconnected by

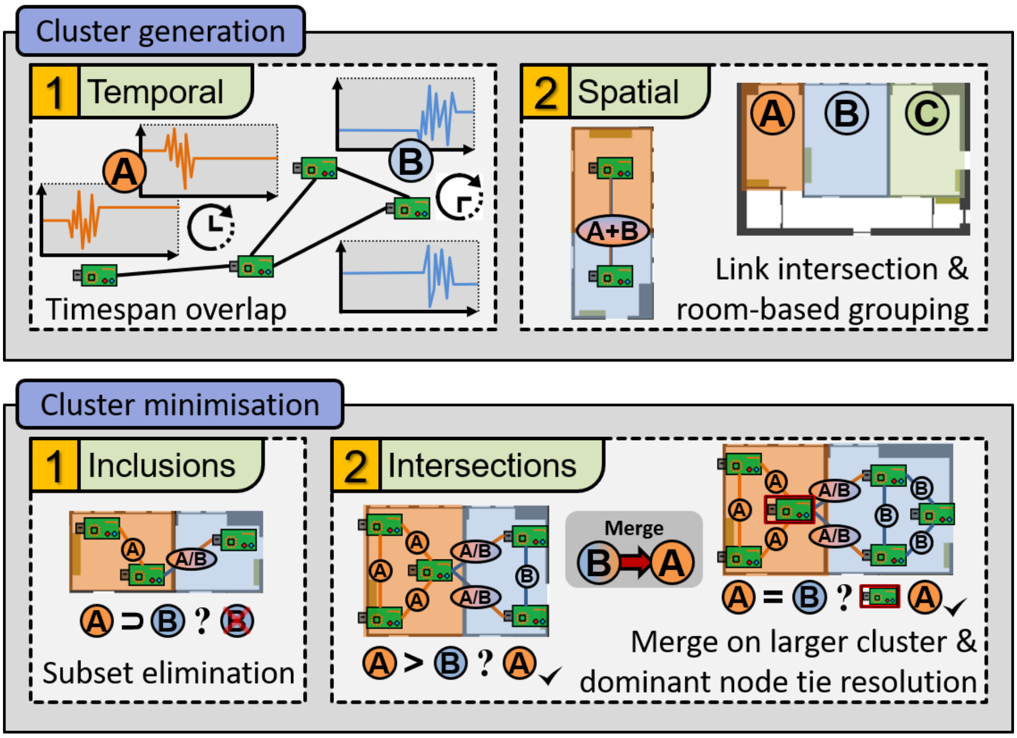


Figure 6.4: *Performed steps of the link clustering algorithm. First, clusters are generated based on their spatio-temporal correlation and then minimised in the subsequent phase.*

links marked as significantly changed. A cluster is assigned to each room including the affected links that either have at least one end point in the room or simply traverse it. According to this scheme, the same link might be present in different clusters. The room-based cluster generation is described by Algorithm 6.2. Each cluster contains a list of assigned events and the corresponding room, it is assigned to. Initially, one cluster is generated for each room. These clusters are then filled with all the event links for which the straight line between sender and receiver intersects the corresponding rooms.

Subsequently, a phase starts with the goal of minimising the considered clusters. In order to achieve that, pairs of clusters not sharing any link are kept and complete subsets of others removed. In the case in which only a partial overlapping exists, if one cluster is smaller than another but the number of common links is sufficiently big, the two clusters are merged into one that is assigned to the room of the bigger. Finally, in the case in which both clusters have the same size and share enough links, the tie is

Algorithm 6.2 Room-based cluster generation.

```
struct Cluster {
  List $events$;
  Room $room$;
}

function clusterGeneration($Events, env$)
    $Rooms := env.getRooms()$;
    $Clusters :=$ new List();
    for ($room \in Rooms$) do
        $Clusters.add($new Cluster(new List(), $room$));

    for ($e \in Events$) do
        $line_e :=$ new Line($e.sender.position, e.receiver.position$);
        for ($c \in Clusters$) do
            $linkCrossesRoom := c.room.intersectsLine(line_e)$;
            if ($linkCrossesRoom$) then
                $Clusters.events.add(e)$;
    return $Clusters$;
```

Algorithm 6.3 Cluster minimisation for total inclusions.

```
function clusterMinimisationInclusion($Clusters$)
    for ($c_1 \in Clusters$) do
        for ($c_2 \in Clusters, c_1 \neq c_2$) do
            if ($c_1.events.containsAll(c_2.events)$) then
                if ($c_1.events.size() \neq c_2.events.size()$) then
                    $Clusters.remove(c_2)$;
                else
                    $linkLength_1 := 0$;
                    for ($e_1 \in c_1.events$) do
                        $line_{e_1} :=$ new Line($e_1.sender.position, e_1.receiver.position$);
                        $linkLength_1 +=$ Line.$segmentLength($
                                    $room.getIntersectionSegment(line_{e_1}))$;
                    $linkLength_2 := 0$;
                    for ($e_2 \in c_2.events$) do
                        $line_{e_2} :=$ new Line($e_2.sender.position, e_2.receiver.position$);
                        $linkLength_2 +=$ Line.$segmentLength($
                                    $room.getIntersectionSegment(line_{e_2}))$;
                    if ($linkLength_1 > linkLength_2$) then
                        $Clusters.remove(c_2)$;
                    else
                        $Clusters.remove(c_1)$;
    return $Clusters$;
```

resolved by finding the most occurring node in the links of both clusters and assigning the merge of the clusters to the room containing such a node. What remains are the clusters of spatially correlated changed links assigned to the most probable room of reference.

Algorithm 6.3 describes the cluster minimisation by applying an inclusion check. For the existing clusters, it is checked whether one list of events fully contains the events from another cluster. If this is the case and the cluster c_1, which contains the events of the other c_2, is larger, the latter is just removed. In case both clusters contain exactly the same events, the cluster is kept, in which room the straight-line signal of a link covers the greater distance. In this way, the room with the larger potential impact on the signal quality is selected.

Algorithm 6.4 Cluster minimisation for partial intersections.

```
function CLUSTERMINIMISATIONINTERSECTION($Clusters$)
    for ($c_1 \in Clusters$) do
        for ($c_2 \in Clusters$) do
            $commonEvents := c_1.events \cap c_2.events$;
            if ($commonEvents \neq \emptyset$) then
                $share_{c_1} := commonEvents.size()/c_1.events.size()$;
                $share_{c_2} := commonEvents.size()/c_2.events.size()$;
                if (($share_{c_1} > s_{min}$) $\wedge$ ($c_1.events.size() < c_2.events.size()$)
                        $\wedge$ ($diffHasIndirectConnectionsToClusterOnly(c_2, c_1)$)) then
                    for ($e_1 \in c_1.events$) do
                        $c_2.events.add(e_1)$;
                    $Clusters.remove(c_1)$;
                else if (($share_{c_2} > s_{min}$) $\wedge$ ($c_2.events.size() < c_1.events.size()$)
                        $\wedge$ ($diffHasIndirectConnectionsToClusterOnly(c_1, c_2)$)) then
                    for ($e_2 \in c_2.events$) do
                        $c_1.events.add(e_2)$;
                    $Clusters.remove(c_2)$;
                else if (($share_{c_1} > s_{min}$) $\wedge$ ($share_{c_2} > s_{min}$) $\wedge$
                                ($c_2.events.size() = c_1.events.size()$))) then
                    $mostFrequentNode := getMostFrequentNode(c_1, c_2)$;
                    if ($c_1.room.containsNode(mostFrequentNode)$) then
                        for ($e_2 \in c_2.events$) do
                            $c_1.events.add(e_2)$;
                        $Clusters.remove(c_2)$;
                    else
                        for ($e_1 \in c_1.events$) do
                            $c_2.events.add(e_1)$;
                        $Clusters.remove(c_1)$;
    return $Clusters$;
```

Afterwards, the partial overlapping of clusters is further investigated by Algorithm 6.4. Thereto, the set of common events between two clusters is computed. In case an actual intersection exist, the share of this set with regards to the overall size of each cluster is determined. A cluster is merged with another cluster if the following conditions apply (line 8): (1) the share of the cluster is larger than s_{min}, which has been set to 60 %, (2) its size is smaller than the one from the other cluster and (3) all other event links from the first cluster, which are not part of the intersection, have a common node with an event link of the second cluster. Especially the latter condition prevents that single event links, which do not have a spatial correlation to the ones in the other cluster, are falsely merged, potentially missing a different second event occurring close to the first one. Having a sufficiently large share in both clusters and an equal number of events, the tie is resolved by determining the most frequent node in both clusters (line 17). To not give larger clusters an advantage, the node count is divided by the total number of events of each cluster first and then averaged between both clusters. Based on this value, the most frequent node is chosen and the corresponding cluster which belongs to the room of this node, is kept. The just described utility methods used for cluster minimisation are given by Algorithm 6.5.

The presented approach of grouping links is able to distinguish spatially or temporally separated events. However, the usage of RSSI values to observe changes in the environment has limitations. If multiple events overlap in space and time, a clear separation between both events and an attribution of the affected links is hardly possible as the effects of both events are inseparably intermixed within the observed RSSI values.

6.3.3 Measurements-to-Event Matching

The identification of links that changed over time and can be ascribed to the same event makes it possible to reason in more details about the actual cause for the observed link changes in addition to a first approximation of the possible event location. To make further inferences about potential causes, additional analysis is required. FOLLOW focuses on obstacles able to change their status and influence the RSSI of links in a stable long-lasting manner with limited fluctuations. This type of events are in contrast to volatile ones, manifesting high variability and hindering the detection of stable events that are co-located.

If fingerprints observed in the specific scenario for all the possible events are available, it would be possible to match the actual observations against such database. This possibility is discussed in Section 8.6.3. Without such information, traces can only be analysed in comparison to the given map,

Algorithm 6.5 Cluster minimisation utility methods.

```
function diffHasIndirectConnectionsToClusterOnly($c_{compare}, c_{base}$)
    $EventDiff := c_{compare}.events \setminus c_{base}.events$;
    $hasCommonNode := false$;
    for ($e_{diff} \in EventDiff$) do
        $hasCommonNode := false$;
        for ($e_{base} \in c_{base}.events$) do
            if (($e_{diff}.sender = e_{base}.sender) \vee (e_{diff}.sender = e_{base}.receiver)$
                  $\vee (e_{diff}.receiver = e_{base}.sender) \vee (e_{diff}.receiver = e_{base}.receiver)$)
                  then
                $hasCommonNode := true$;
                continue with next $e_{diff}$;
        if ($\neg$ $hasCommonNode$) then
            return $false$;
    return $true$;

function getMostFrequentNode($c_1, c_2$)
    $nodeCount_1 := countNodeOccurrences(c_1.events)$;
    $nodeCount_2 := countNodeOccurrences(c_2.events)$;
    $allNodes := nodeCount_1.keys() \cup nodeCount_2.keys()$;
    $frequentNode := null$;
    $freq_{best} := -1$;
    for ($node \in allNodes$) do
        $freq := \left( \frac{nodeCount_1.get(node)}{c_1.events.size()} + \frac{nodeCount_2.get(node)}{c_2.events.size()} \right) /2$;
        if ($freq > freq_{best}$) then
            $frequentNode := node$;
            $freq_{best} := freq$;
    return $frequentNode$;

function countNodeOccurrences($Events$)
    $nodeCntMap :=$ new Map();
    for ($e \in Events$) do
        $nodeCntMap.put(e.sender, nodeCntMap.getOrDefault(e.sender, 0) + 1)$;
        $nodeCntMap.put(e.receiver, nodeCntMap.getOrDefault(e.receiver, 0) + 1)$;
    return $nodeCountMap$;
```

Table 6.2: *Parameters and chosen values used by the obstacle link determination of* Follow.

Parameter	*Description*	*Value*
$t_{strongLink}$	Strong link RSSI threshold	−90 dBm
$t_{pathDiff}$	Ray path difference threshold	20 dBm

through the lenses of the ray tracing models mentioned in Section 5.2. In order to study a specific obstacle as possible cause for an observed change, the theoretical influences of each obstacle on the network links have to be known.

By exploiting the introduced ray tracing models, it is possible to precompute, for each obstacle, its expected impact on communication links if its state or position is changed, e.g., in case of a door or window if it is opened or closed. An intuitive approach would be to check if any such event would cause changes matching the actual observations. In following this approach, however, it needs to be taken into account that in most cases the scenario description does not provide centimetre accuracy for practical reasons (as in the evaluated case). Therefore, the model output cannot rely on the correct computation of interference effects, which however have been demonstrated to play a significant role in reality.

Algorithm 6.6 Obstacle link influence determination.

```
struct ObstacleLinks {
    Obstacle $obstacle$;
    Set $links$;
}

function computeObstacleLinkInfluence($N, mdl, env$)
    $O := env.getSemiDynamicObstacles()$;
    $L := mdl.predictLinks(N, env)$;
    $OL :=$ new Set();
    for $(o \in O)$ do
        $L_o :=$ new Set();
        for $(l_{(n_1 \to n_2)} \in L)$ do
            $Paths_{(n_1 \to n_2)} := mdl.getSignalPaths(n_1, n_2, env)$;
            for $(path \in Paths_{(n_1 \to n_2)})$ do
                if $(path.rssi > t_{strongLink}$
                 $\wedge (l_{(n_1 \to n_2)} - path.rssi) < t_{pathDiff}$
                 $\wedge (path.intersects(o.closedPosition)$
                      $\vee\, path.intersects(o.openPosition)))$ then
                    $L_o.add(l_{(n_1 \to n_2)})$;
                    continue with next $l$;
        $OL.add($new ObstacleLinks$(o, L_o))$;
    return $OL$;
```

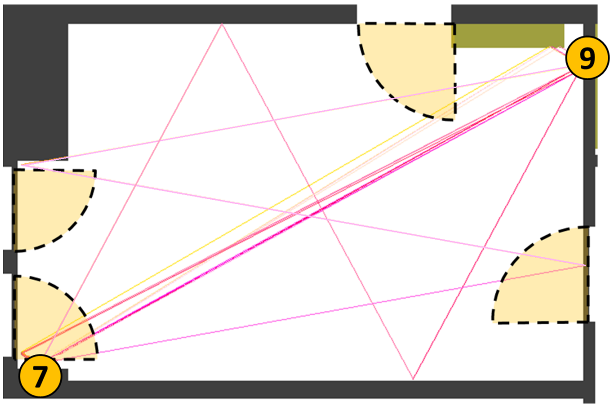


Figure 6.5: *Example ray path computation between two nodes. For each of the semi-dynamic obstacles an intersecting ray path exists, marking each obstacle within this room as a possible influence on this link.*

Without the possibility to account for the effect of interferences, the RSSI predictions computed by the model for a specific position become worthless when analysing the effect of small changes to the environment. In order to circumvent the consequences of a less accurate map on the model output as well as on the correctness of the event detection, an approach is used that depends less on the actual RSSI values. The ability of ray tracing is exploited to track each possible path of a radio signal between sender and receiver. For the needs of FOLLOW, the relevant information corresponds to the existence of a possible path to a receiver at a sufficient signal strength that interacts with the investigated obstacle, e.g., via reflection, in at least one of its possible states, e.g., if a door is open or closed. Algorithm 6.6 describes the process of determining relevant signal paths. In accordance with Table 6.2, only paths with a signal strength of above $t_{strongLink} = -90$ dBm and a difference of less than $t_{pathDiff} = 20$ dBm to the strongest received signal are considered. Figure 6.5 visualises such an example of testing possible ray paths from node 7 on the bottom left to node 9 on the upper right for intersections with semi-dynamic obstacles located in the room. In this case, all obstacles possibly have an influence on this link.

Once the sets of possibly influenced links for all the obstacles in the scenario have been generated, these can be compared against the clusters computed based on the gathered observations. The comparison process is described by Algorithm 6.7. As decision criterion, the Jaccard similarity coefficient [p76] is used to measure the similarity between two sets. The size of the intersection of both link sets divided by the size of their union builds a target coefficient (see Equation 4.3). The highest coefficient among all the set comparisons yields the decision regarding which obstacle is considered as the event cause.

Algorithm 6.7 Cause determination by link set comparisons.

```
function DETERMINEMOSTPROBLABLEEVENTCAUSE($L_{event}$, $OL$)
    $corr_{best} := 0$;
    $obst_{best} := null$;
    for ($ol \in OL$) do
        $L_{obst} := ol.links$;
        $corr := \frac{|L_{event} \cap L_{obst}|}{|L_{event} \cup L_{obst}|}$; (Equation 4.3)
        if ($corr > corr_{best}$) then
            $corr_{best} := corr$;
            $obst_{best} := ol.obstacle$;
    return $obst_{best}$;
```

6.4 Conclusion - Follow

The Internet of Things and cyber-physical systems rely on wireless interconnections. The importance of understanding the behaviour of wireless communication in practice has promoted many experimental studies, characterising signal propagation in specific environments. Despite the gathered knowledge, the perceived discrepancy between abstract models and actual network performance has supported the belief that system design and debugging can rely only on direct experience and trial and error. As a result, reasoning about wireless systems is nowadays a tedious, manual process.

The approach presented in this chapter, FOLLOW [a2], walks the playground between model and reality to make wireless networks understand their own behaviour in the environment where they operate. First, the previously introduced models of MOVE are utilised to efficiently and accurately characterise wireless signal propagation in 2D and 3D environments with obstacles. Then the model is exploited to make systems autonomously derive the impact of obstacles such as doors and windows on the link conditions. With an outline of the operational scenario and measurements from deployed devices, it is possible to effortlessly generate a situated, dynamic wireless map. FOLLOW is demonstrated in an indoor testbed, showing its practicability and effectiveness. The evaluation reveals that FOLLOW attributes up to 91% of the measurements to the corresponding obstacles correctly. Finally, it is shown how that different modelling techniques have an impact on such detection accuracy, promoting the use of 3D environment descriptions.

With FOLLOW as the third contribution of this thesis, a further advancement is made towards understanding the properties of a network. In contrast to the previous two contributions, where the network is analysed based on observations made at one specific point in time, FOLLOW incorporates observations over longer time periods to analyse temporal variations in the behaviour of links. Requiring exact signal paths to determine possible influences located in the environment, this solution exploits the physical modelling provided by MOVE. However, compared to the plain usage of the previous step, Follow has increased demands on the model to achieve its goals. To enable the physical model making temporal changes visible in its predictions, it becomes necessary to include the relevant obstacles as part of the environment description whose state can then be changed. This information has to be provided by the user in addition to the other required static obstacles. Building upon the available knowledge of how signals propagate in a given environment, the results of Follow yield further valuable insights which can be used to gain a better understanding with respect to the other steps. E.g., previously unconsidered influences might explain inaccuracies for certain links in the predictions of MOVE or certain devices and positions can be considered as more or less suitable during the optimisation phases of MOTION.

7 Network Optimisation

Planning and operating networked embedded systems is a challenge for developers and operators, especially in times of the rising Internet of Things. A large variety of devices is expected to seamlessly cooperate to make everyday scenario smarter. However, that same scenario and its specific environment have a peculiar impact on wireless communication [p108], affecting the system performance. Being able to analyse the characteristics of the environment helps to gain further knowledge about the system, identify weak spots and develop solutions to preserve and optimise its functionality and performance. Nonetheless, the design and validation of system solutions are typically based on tools that are agnostic to the environment.

Simulation [p105] can be used to test solutions in arbitrary setups, enabling general evaluation of protocols and their implementation, abstracting from the specificity of individual configurations and scenarios. Acknowledging the discrepancy between simulation and reality, the community has realised public testbeds [p58, p40], which have become an established validation methodology. Despite the possibility to demonstrate the effectiveness of solutions in concrete settings, they depict one specific environment and are unable to provide the insights necessary to configure an arbitrary system in its specific deployment site. As a result, during deployment, developers and operators resort to trial and error in order to debug and optimise running systems [p173].

Thus, for the optimization of networks, either real validation approaches tailored to specific scenarios with missing transferability or simulations independent of scenario but without reference to reality are available. Approaches that combine the versatility of simulations with the possibilities of real systems in such a way that they become applicable in arbitrary scenarios, are missing. Recognising the lack and need of holistic and universally applicable optimisation tools, this chapter presents an approach for realising such, bringing the flexibility and usefulness of simulations to reality. Performing the final step of exploiting the available knowledge gathered from the network to ultimately trigger optimisation measures, a communication model based on a description of the target environment is employed, whose outputs are translated into physical relocations of devices, impacting the system behaviour in the desired way.

7.1 Challenges

Realising the targeted automated optimisation process comprises multiple steps. First, the utilisation of the communication model as foundation for any further actions, requiring reliable outputs. Second, the evaluation of given predictions in order to develop effective optimisation strategies. And lastly, the actual conduction of physical modifications to the network relocating its devices and thus impacting the system behaviour. These steps, individually and jointly considered, pose several challenges to be met.

Divergence of Simulation and Reality Utilising virtual models for estimating network behaviour provides the huge benefit of evaluating a multitude of potential optimisation configurations without the need of testing all of them in reality. Undesired outcomes can be discarded and the set of configurations thus be narrowed down to a selection of the most auspicious ones. However, this assumes a basic level of reliability regarding the accuracy of the predictions. Considering the heterogeneity of available information in different scenarios, this unfortunately cannot be guaranteed since even minor inaccuracies or missing details in the network description might lead to deviating outcomes. Therefore, besides necessitating detailed and accurate scenario descriptions, such risks of diverging predictions have to be minimised by thorough algorithms, considering many details and desirably providing the possibility of correcting divergences from reality.

Spatial Limitations The architecture of a wireless network is planned in accordance with the structural conditions of the environment. Existing restrictions might limit possible relocation positions to specific areas which might not be optimal with respect to the targeted network metrics to be improved. Thus, an optimisation approach has to take such limitations into account and provide alternative positions from which can be chosen, finding the best trade-off between possible positions and improvements.

Mutual Impact of Multiple Physical Changes Performing profound optimisations on a network usually involves multiple physical changes to achieve the targeted effect. Each individual modification of the network topology has an effect on the performance, either imperceptible or significant. Ideally, the behaviour observed after a device relocation matches the expected one which was estimated by the model before. However, as already stated, inaccuracies in the environment model or unconsidered effects might lead to unintended side effects and hence a deviating behaviour. In an unfavourable case, such deviations then have an impact on the entire relocation plan, requiring ad-hoc adjustments. To address this issue, a comprehensive solution has to provide options for a stepwise realisation of larger optimisation strategies with the possibility of re-evaluating expected and actual states during each deployment step to perform dynamic adaptations based on new observations.

7.2 Approach

Picking up the approach described in Section 3.4.2, Moρφευς [c7] demonstrates the powerful optimisation capabilities of combining simulation and reality. Simulations execute a given application which to optimise in performance, exploring various alternative configurations by adding, removing or relocating simulated devices at certain positions in the virtual network. The results are evaluated to find the best configurations improving the target application metrics, which are then applied in reality.

For device communication, the simulation used in Moρφευς solely relies on communication properties observed at a real network, i.e., packet reception rates for all existing links between devices, which are taken as the basis when employing simulated communication. Hence, relying only on connectivity data has limits when exploring new positions in a network, neglecting possible effects of the environment on the link quality. Actually, the used communication model is not able to perform predictions for any unknown location. Since the main focus of Moρφευς was not to foresee communication behaviour, but to demonstrate the realism of simulation in reproducing an application behaviour close to reality, the investigated candidate positions were restricted to known positions where further devices were already in place and thus for which real communication data was available. Consequentially, the simulated optimisation proposals were applied in reality by (de-)activating certain present devices. However, besides the clear advantage of not requiring any details about the network environment except for the connectivity data, this also reveals the limitations of Moρφευς, restricting the explorable positions to known ones and therefore also the ability to exploit the full optimisation potential.

In fact, in the operational scenario, information is typically available from the devices already deployed at their fixed position. Foreseeing the impact of an alternative displacement of devices becomes challenging and, at the same time, crucial to satisfy the user requirements. This challenge is tackled by accurately modelling wireless communication in the operational environment starting from information gathered by the system in its current configuration and from external descriptions of the scenario. As a result, it becomes possible to reason about the impact of specific device positioning on the system behaviour, enabling the optimisation of networks situated in their operational environments.

Taking up and extending the idea of exploiting insights gained from computations of simulated scenarios and transfer them to reality by applying actual modifications to a network, this chapter presents MOTION, a Model-based Optimisation of Wireless Networks [a3]. Instead of simulating and evaluating the application itself to derive optimisation measures, MOTION uses a different approach, investigating the underlying network and focusing on device communication. A technique to support in situ deployment and maintenance of systems is introduced, utilising the HYBRID modelling of MOVE (Section 5.3.5) that fuses information from gathered measurements of wireless communication in the operational system with a physical model of the expected signal propagation. To make the technique practical and beneficial, this approach relies on the introduced optimisations to speed up the time required for the computation of the models, minimising the sacrificed accuracy (Section 5.3.2). Utilising the capabilities of MOVE to virtually explore alternative device positions in a systematic way, predictions of the expected communication behaviour are generated for each of the investigated positions. These predictions are then evaluated and the benefits of MOTION explored by devising some relocation strategy (Section 7.3) aiming at optimising the overall resulting network performance. By this means, MOTION extends the state of the art in two ways:

Network-wide Exploration of Optimisations With a model of the network defined by MOVE, it becomes possible to reason about device positioning and its impact on the system performance. Traditional approaches for network optimisation, design and analysis, as discussed in Section 3.4, focus on tuning software parameters by analysing specific network stacks or optimising defined network or application metrics. MOTION could extend all these techniques to allow the study of networks with devices at arbitrary positions, providing an accurate estimation of communication situated in the operational scenario.

Application-agnostic System Improvements MOTION focuses on optimising system performance in a pure application-agnostic fashion through a controlled relocation of devices. It is refrained from exploiting any knowledge coming from specific protocols running in the system, relying exclusively on the properties of the underlying physical topology. In this manner,

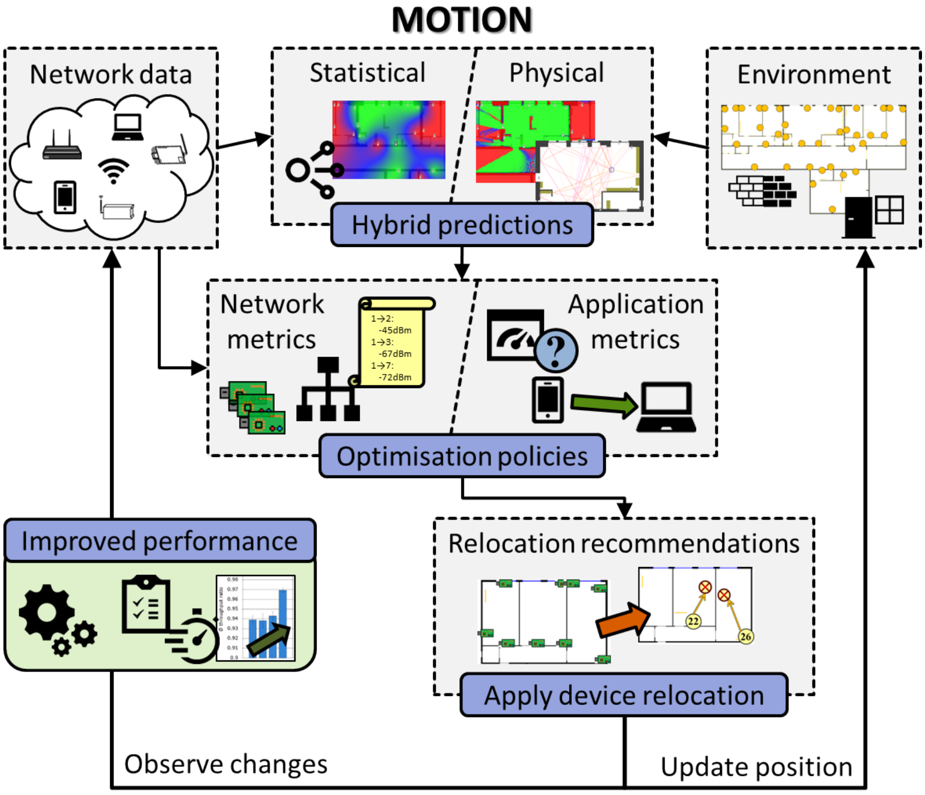


Figure 7.1: MOTION *concept of incremental network performance improvements.*

the need for any simulation of specific software implementations or any protocol model is obviated, thus gaining general applicability. Nonetheless, the ability of MOTION to accurately control the network topology can already be exploited to optimise system behaviour.

The concept MOTION is visualised in Figure 7.1, demonstrating the cycle of improvement, alternating between network analysis and device relocation: The predictions of the HYBRID model, using inputs of both recorded traces from the network and a description of the environment, are utilised by an optimisation policy (Section 7.3.2) which proposes the most promising relocation positions to the user. After choosing one option and performing the physical device relocation, the position is updated in the virtual description of the environment as well and the network performance assessed again. Based on the targeted improvements, the process can be repeated and further relocation steps undertaken.

Implementing MOTION in the aforementioned way, the described challenges of an automated optimisation process are addressed. The potential problem of a mutual impact of multiple device relocations is solved by the given optimisation cycle, including a re-evaluation of the network performance after each relocation, allowing for an adaptation of plans if actual observations deviate from the predicted ones. The closely associated problem of a possible divergence between simulation and reality is tackled by the used HYBRID model which incorporates connectivity data from known positions into its predictions, which are preferred over those generated by the physical computations. Thus a back channel is established, which allows a correction of the model in case of deviations. Furthermore, this implementation of HYBRID, observation over computation, at least allows a local repositioning even if the description is inaccurate. Finally, restrictions to the placement of devices are considered by the possibility to specify the area in which the optimisation policies search for suitable alternatives. This can be one or a combination of the following options to consider: (1) a maximum distance from the current device position, (2) a minimum distance to other nodes, (3) only the same room, in which the device is already located and (4) wall positions exclusively, in case a positioning within the room is not possible.

MOTION is evaluated on data traces collected from the WSN Testbed (Section 8.1). Predictions of HYBRID are utilised to perform device relocations comparing expected link quality and quantities against reality, proving their correctness. Furthermore, the impact of such relocations on application metrics is investigated. Based on the experimentation (Section 8.7), it is highlighted how MOTION can become a stepping stone towards the integration of environment information in the holistic, situated design of wireless systems.

7.3 Implementation

In this section, the approach of network optimisation is discussed from the technical perspective of MOTION. Re-utilising the deterministic signal propagation modelling of MOVE, the description of the environment, used by the HYBRID model, is a key component as it enables spatially precise predictions for any position and thus opens up the possibility to explore alternative locations for devices. Additionally, relocations applied to the network can easily be adjusted in the environment description and thereby reflected in subsequent computations. Since each model inevitably suffers from inaccuracies, even if these are small, the chosen incremental approach mitigates their impact by counteracting accumulating errors due to regular model updates after each relocation. As discussed and shown in Figure 7.1, MOTION performs optimisations in an iterative cycle, consisting of three steps: (1) position exploration via communication model, (2) application of an optimisation strategy and (3) user-driven relocation of one of the proposed devices. Because details regarding the functionality of the HYBRID model have already been mentioned in Chapter 5, it is proceeded with the discussion of the optimisation process.

7.3.1 Incremental Optimisation

To support network optimisation in a fully application-agnostic fashion, MOTION exploits information available only at the physical network layer. Moreover, the approach bases on measurements that can be taken by the wireless devices themselves. This allows to inspect the environment from the perspective of the specific wireless elements operating in the final system and to update the model during system lifetime. If typical radios are considered, it is usually possible to measure channel noise, RSSI and correct packet reception. While additional information, e.g., reception of corrupted messages, can also be used, this work has been restricted to a minimum set of information. Further details can be integrated in the model to increase its accuracy. Based on this information, it is possible to compute link and network metrics, e.g., the number of incoming or outgoing links per node, the average RSSI per link and the network diameter.

MOTION estimates the communication properties of arbitrary locations in the modelled environment. Therefore, the main parameter of system optimisation becomes the chosen location of devices. Utilising the obtainable network data and the predictions provided by the HYBRID model in a flexible and generic way, MOTION realises the concept of modular policies, evaluating this information with different optimisation priorities (cf. Figure 7.1). An optimisation policy could then analyse the impact of relocating devices, defining constraints and goals. The policy could impose to respect network, e.g., avoid partitioning and preserve existing reliable routing paths, or application, e.g., sensing coverage, requirements. Satisfying these constraints, the network topology could be optimised to, e.g., remove weak links, increase or decrease the number of links, increase the network average RSSI. As a result, the most promising positions for relocation improving the network behaviour with respect of the chosen policy, are computed. Among these options, the user has to choose the preferred one and then physically relocate the node to the declared position. After that, a new phase starts where policies can be applied incrementally to further adjust device positions until the network evinces the desired behaviour.

Table 7.1: *Common parameters and chosen values used by the implemented policies of* MOTION*.*

Parameter	*Description*	*Value*
$d_{relocMax}$	Maximum x/y node relocation distance	3 m
s_{grid}	Relocation position grid size	0.25 m
$d_{nodeMin}$	Minimum distance to other nodes	1 m
$b_{relocRoom}$	Restrict relocation to same room	✓
$b_{relocWall}$	Restrict relocation to wall positions	×
$b_{linkPreserve}$	Preserve existing strong links	✓
$b_{linkNoWeak}$	Disallow weak links	✓
$t_{strongLink}$	Strong link RSSI threshold	-90 dBm

Common Parameters For performing meaningful optimisations, policies can be customised to fit the requirements of a given scenario. A detailed overview of the available common adaptation parameters and the chosen values used for the evaluation is given in Table 7.1. Since they have been applied to the same scenario, both used policies share the same parameter values. To guarantee the sensing coverage of all the rooms, the possible relocations are restricted to other positions in the same room. Similarly, nodes too close to each other are avoided. Despite offering the feature of only considering positions close to walls for relocation, useful when a central placement of devices within a room is not possible, it has been refrained from using it to not further reduce the number of considered positions. Limiting the maximum relocation distance to three metres within the same room while keeping at least one metre distance to other devices already limits the number of evaluated positions, which is at most 144 with a quarter of a metre as grid size. In addition, it can be beneficial to preserve the existing network topology if reliable, preventing the strategy from destroying existing good links and creating weak links as well. The threshold for such links has been defined at -90 dbm in the given scenario due to the used hardware (Section 8.2). The chosen restrictions ensure to not change the network topology remarkably, but nevertheless demonstrate that also minor changes in position can have a significant impact on the network behaviour.

Before explaining the optimisation process in more detail, Table 7.2 lists the abbreviations which are used by the following algorithmic descriptions and gives a short explanation on their meaning to ensure a better readability due to a more compact notation. For the same reasons, the following algorithms depict the conceptional core of the realised processes and thus, despite following the described approach, might be realised differently in the actual implementation, e.g., different naming, call orders or method parameters.

Table 7.2: *Explanation of identifiers used for the algorithmic descriptions of* MOTION.

Name	*Description*
n	A node which is part of the network
N	Set of all network nodes (n)
$l_{(n_1 \rightarrow n_2)}$	A link RSSI value between two nodes (sender: n_1, receiver: n_2)
L	Set of all real or predicted existing links (l) for a network
Par	Set of parameters as defined by Table 7.1
mdl	Instance of HYBRID model
env	The environment description containing obstacle geometry
op	An optimisation policy
Opt	Set of optimisation proposals returned by a policy
p	A position within the environment
Pos	Set of positions (p) used for relocation
mtr	Network metrics for a specific node position provided by a policy

The general process, which is triggered for each optimisation is depicted by Algorithm 7.1. For each node, possible relocation positions are collected for which an improvement regarding the optimised metric is expected. First, the metrics relevant for the used policy are collected for the current state. Afterwards, potential positions for relocation are determined, for which link quality predictions are made by the model. Filtering these predictions with respect to the general requirements defined by the parameters (described below), the remaining candidate positions are then evaluated based on the policy metrics. Those who improve the current state are finally collected and returned to the user to make a decision for the relocation.

Algorithm 7.1 Optimisation policy execution process.

```
function applyOptimisationPolicy($N, L_{real}, mdl, env, op$)
    $Opt$ := new Set();
    for ($n_{opt} \in N$) do
        $mtr_{current} := op.computeMetricsForNode(n_{opt}, L_{real})$;
        $Pos := computeExplorablePositions(n_{opt}, N, env)$;
        for ($p \in Pos$) do
            $L_{pred} := mdl.predictLinksForNodeAtPosition(n_{opt}, p, N, env)$;
            if ($\neg$ $nodeLinksFulfilRequirements(n_{opt}, L_{real}, L_{pred})$) then
                continue with next $p$;
            $mtr_p := op.computeMetricsForNode(n_{opt}, L_{pred})$;
            if ($op.improvesNetworkMetrics(mtr_{current}, mtr_p)$) then
                $Opt.add($new Optimisation$(n_{opt}, p, mtr_p))$;
    return $Opt$;
```

The determination process of possible relocation positions for further examination is described by Algorithm 7.2. As a starting point, the current position of the respective device is taken. Using the current position of the respective device as centre, a grid of candidate positions is computed around it, whose size and density can be defined by parameters. Thereof now such points are valid, which do not intersect any other obstacle within the environment and optionally are located close to walls and/or in the same room as the device. Additionally, a minimum distance to other devices has to be satisfied to still benefit from the spatial difference between them. The remaining positions, fulfilling all these requirements are returned for investigation.

Algorithm 7.2 Exploration of optimisation positions.

```
function computeExplorablePositions($n_{opt}, N, env$)
    $Pos$ := new Set();
    $p_n := env.positionOf(n_{opt})$;
    for ($p \in \{p \mid |p.x - p_n.x| \leq d_{relocMax} \wedge |p.y - p_n.y| \leq d_{relocMax} \wedge$
            $\frac{|p.x - p_n.x|}{s_{grid}} \in \mathbb{N} \wedge \frac{|p.y - p_n.y|}{s_{grid}} \in \mathbb{N} \wedge \neg\ env.intersectsObstacle(p)\}$) do
        if ($b_{relocRoom} \wedge \neg\ env.roomOf(n_{opt}).contains(p)$) then
            continue with next $p$;
        if ($b_{relocWall} \wedge (env.closestWallDist(p) \leq s_{grid})$) then
            continue with next $p$;
        for ($n \in N, n \neq n_{opt}$) do
            if ($|n.position - p_n| < d_{nodeMin}$) then
                continue with next $p$;
        $Pos.add(p)$;
    return $Pos$;
```

After employing the model to predict the estimated link qualities of the new positions, the predictions are further filtered based on the optimisation requirements on link level as shown by Algorithm 7.3. In this context, two aspects are relevant. Any weak links to the target device are avoided for the investigated positions, falling below the specified threshold. Likewise, previously existing stable links are preserved. Positions violating one of these two optionally selectable rules, are discarded.

With the aforementioned processes, a basis for performing network optimisations is provided. In order to fulfil the varying requirements individual to each scenario, an implementation of different optimisation policies becomes necessary. These are described in the following.

Algorithm 7.3 Link requirement fulfilment.

```
function NODELINKSFULFILREQUIREMENTS($n_{opt}, L_{real}, L_{pred}$)
    if ($b_{linkNoWeak}$) then
        for ($l_{(n_1 \to n_2)} \in L_{pred}, n_1 = n_{opt} \lor n_2 = n_{opt}$) do
            if ($l_{(n_1 \to n_2)} < t_{strongLink}$) then
                return false;

    if ($b_{linkPreserve}$) then
        for ($l_{(n_1 \to n_2)} \in L_{real}, (n_1 = n_{opt} \lor n_2 = n_{opt}) \land$
                                  $l_{(n_1 \to n_2)} \geq t_{strongLink}$) do
            if ($((l_{(n_1 \to n_2)} \notin L_{pred}) \lor (l_{(n_1 \to n_2)} < t_{strongLink}))$
            then
                return false;
    return true;
```

7.3.2 Optimisation Policies

To demonstrate the impact of MOTION, two optimisation policies have been implemented and evaluated. Both strategies aim at improving network metrics, which in turn have the capability of positively affecting application metrics as well.

Link Quality and Quantity

The first policy focuses on improving network metrics, the overall link quality and quantity with the main goal to increase the number of strong and reduce the number of weak links (Section 8.7.1). This policy initially determines the average link RSSI of each device to prioritise the ones with the lowest values for optimisation. The strategy then analyses, for each node in the system, the links from and to the device in each possible position derived from the physical map at a chosen grid resolution and maximum distance. The estimated RSSI for each link is compared against the RSSI of the node in its original position to determine the quality of each link. The difference in the network topology is then computed, including possible new as well as removed links. Thereout, it is possible to identify the best positions to improve each individual node connectivity by enhancing the average reliability.

Figure 7.2 depicts a map of estimated relocation improvements of the marked node 11 in the upper left applying this optimisation strategy while preserving already existing links. Green areas indicate an expected improvement of overall link quality while red ones predict a decrease in quality with respect to the links to be preserved. Blue areas represent disregarded obstacle positions.

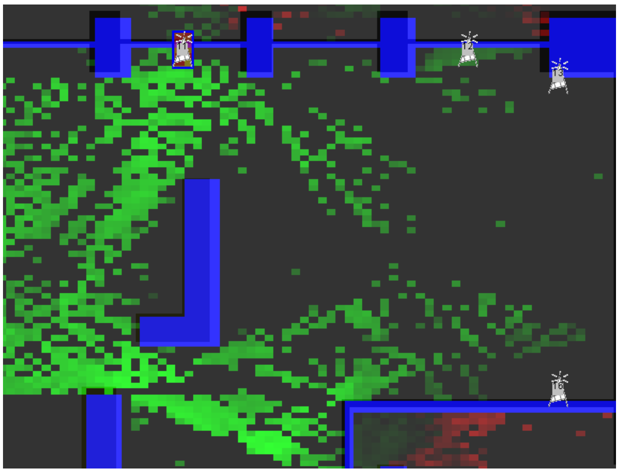


Figure 7.2: *Map of estimated relocation improvements using the network optimisation strategy of improving link quality and quantity for the selected node 11.*

The computation of metrics used the link improvement policy is listed by Algorithm 7.4. Since the relevant metrics for this policy directly base on RSSI values in order to distinguish good from weak links, the computation is straight forward. Comparing the predicted link strengths with the threshold defined via parameter, the number of good and weak links is counted and the average RSSI value across all links of the given node is computed.

Algorithm 7.4 Link improvement policy metric computation.

```
struct LinkMetrics {
    Integer $goodLinks, weakLinks$;
    Float $avgRSSI$;
}

function ComputeMetricsForNode($n_{opt}, N, L$)
    $mtr$ := new LinkMetrics(0, 0, 0);
    for ($l_{(n_1 \to n_2)} \in L, (n_1 = n_{opt}) \vee (n_2 = n_{opt})$) do
        $mtr.avgRSSI$ += $l_{(n_1 \to n_2)}$;
        if ($l_{(n_1 \to n_2)} > t_{strongLink}$) then
            $mtr.goodLinks$ += 1;
        else
            $mtr.weakLinks$ += 1;
    $mtr.avgRSSI$ /= ($mtr.goodLinks + mtr.weakLinks$);
    return $mtr$;
```

After computing the network metrics for an investigated position, these are compared against the actual metrics to determine whether a relocation to the new position is expected to improve the current state. This comparison is described by Algorithm 7.5. It is expected to have at least either an increase of good or a decrease of bad links in order to not discard the given position. Focusing on the sufficient binary quality classification of existing links rather than their exact RSSI value, the average RSSI serves as supplementary information and is thus not directly considered in metrics comparison.

Algorithm 7.5 Link improvement policy metric comparison.

```
function IMPROVESNETWORKMETRICS($mtr_{current}, mtr_{new}$)
    if ((($mtr_{new}.goodLinks > mtr_{current}.goodLinks$)
            $\wedge$ ($mtr_{new}.weakLinks \leq mtr_{current}.weakLinks$))
        $\vee$ ((($mtr_{new}.goodLinks \geq mtr_{current}.goodLinks$)
            $\wedge$ ($mtr_{new}.weakLinks < mtr_{current}.weakLinks$))) then
        return true;
    else
        return false;
```

Network Diameter

The second policy has the goal to improve the network as a whole by reducing the network diameter. In distinction from the previous policy, for each device a hop map using Dijkstra's algorithm [p39] is initially built for determining the minimum number of links (hops) to reach any other device in the network. Alternatively the hop counts can also be reduced with respect to specific devices. For the nodes having the largest hop counts, relocation positions are then evaluated, which are expected to reduce the path by establishing new reliable links to devices closer to the destination position. Regarding considered relocation positions, the same constraints from the aforementioned policy can be applied to this policy as well. Ultimately, the reduction of the network diameter has been chosen to demonstrate a positive impact on application metrics (Section 8.7.2).

Algorithm 7.6 Diameter reduction policy metric computation.

```
struct DiameterMetrics {
   Integer $maxHopCount$;
   Node $maxHopSink$;
}

function COMPUTEMETRICSFORNODE($n_{opt}, N, L$)
    $mtr$ := new DiameterMetrics($0, null$);
    for ($n \in N$) do
        $maxHops_{\rightarrow n} := getMaximumShortestReliablePathToNode(n, N, L)$;
        if ($mtr.maxHopCount < maxHops_{\rightarrow n}$) then
            $mtr.maxHopCount := maxHops_{\rightarrow n}$;
            $mtr.maxHopSink := n$;
    return $mtr$;
```

Algorithm 7.6 lists the metric computation of the diameter reduction policy. Improving the network in general, it is required to find the highest hop count from all shortest paths between any two devices. Hence, the relocated node does not need to be processed in a different way than the other nodes as each of these needs to be checked to determine the diameter. In contrast to the link improvement, this policy has to further process the predictions given by the model.

Using Dijkstra's algorithm [p39], the shortest path to a given destination node is computed for each other device of the network, described by Algorithm 7.7. Beginning from the destination node, it is successively iterated over neighbouring nodes which have a link to an already processed node to determine their hop count for only good links above the given threshold are considered. To ensure building up a reliable network, the optimisation policy used for evaluation was configured to not only reduce the diameter, but try to reduce existing and avoid new weak links as well.

Algorithm 7.7 Diameter reduction policy shortest path computation based on Dijkstra's algorithm [p39].

```
struct HopInfo {
    Integer $hops$;
    Node $predecessor$;
}

function GETMAXIMUMSHORTESTRELIABLEPATHTONODE($n_{dest}, N, L$)
    $hopInfo := $ new Array($|N|$);
    $Q := $ new Set();
    for ($n \in N$) do
        $hopInfo[n] := $ new HopInfo($\infty, null$);
        $Q.add(n)$;
    $hopInfo[n_{dest}].hops := 0$;
    while ($Q \neq \emptyset$) do
        $n_{next} := (n \in Q :$
                $hopInfo[n].hops <= hopInfo[n_{other}].hops \ \forall n_{other} \in Q, n_{other} \neq n)$;
        $Q.remove(n_{next})$;
        for ($n_{adj} \in Q : \exists l_{(n_{adj} \rightarrow n_{next})} \in L :$
                $l_{(n_{adj} \rightarrow n_{next})} > t_{strongLink}$) do
            if ($hopInfo[n_{adj}].hops > hopInfo[n_{next}].hops + 1$) then
                $hopInfo[n_{adj}].hops := hopInfo[n_{next}].hops + 1$;
                $hopInfo[n_{adj}].predecessor := n_{next}$;

    $maxHopInfo := (h \in hopInfo :$
            $h.hops >= h_{other}.hops \ \forall h_{other} \in hopInfo, h \neq h_{other})$;
    return $maxHopInfo.hops$;
```

For the sake of completeness, Algorithm 7.8 shows the metric comparison to decide on an expected improvement. Since the main goal is to reduce the network diameter, the maximum hop count is solely taken into consideration, providing a plain decision criterion. However, when optimising paths to specific nodes, further constraints might be required, e.g., ensuring certain paths to be existent with sufficient link quality.

Algorithm 7.8 Diameter reduction policy metric comparison.

```
function IMPROVESNETWORKMETRICS($mtr_{current}$, $mtr_{new}$)
    if ($mtr_{new}.maxHopCount < mtr_{current}.maxHopCount$) then
        return true;
    else
        return false;
```

7.4 Conclusion - MOTION

The design and deployment of networked embedded systems is challenging. In particular, the environment in which the system operates has a severe impact on the final performance. Existing tools trade generality for specificity with arbitrary setups, e.g., in simulation, or specific configurations, e.g., in public testbeds. As a result, the peculiar effect of the target deployment scenario on the system performance can only be experienced by hand. Optimising and adapting the system to the operational environment becomes then a trial and error, impractical challenge.

In this chapter, MOTION is introduced, a technique to optimise wireless networks by relocating devices in the environment [a3]. It utilises the HYBRID modelling technique able to characterise indoor wireless communication in the target environment from measurements and descriptions, e.g., maps, of the deployment area. MOTION can then support traditional design and analysis tools in investigating the impact of arbitrary displacements of devices in the target environment. E.g., by controlling the relocation of a single node based on the information provided by the model, it was possible to reduce the latency of a typical application by up to 21.4% (Section 8.7). Ultimately, MOTION allows situated network optimisation of wireless systems where devices can be freely moved in the environment and the corresponding impact on the network performance can be estimated beforehand.

MOTION, as the fourth contribution of this thesis, constitutes the final phase in the endeavour of acquiring knowledge about the network. In this last step, previously collected knowledge is exploited and translated into recommended actions for modifying the underlying network, with the goal to optimise its behaviour. Proposing yet unknown locations for the relocation of devices, MOTION mainly depends on the predictions provided by MOVE. As the relocations are spatially precise, the used model has to fulfil certain accuracy requirements, which requires to take the properties of the environment into account as well, suggesting the use of a physical modelling approach. Consequentially, the quality of optimisations performed by MOVE is directly affected by the quality of the provided environment description, which however does not have to be overly precise to achieve significant improvements. Also knowledge of other network properties, gained through FOLLOW regarding the effects of obstacles on the communication behaviour, can contribute to a more effective optimisation process by avoiding positions which are, e.g., potentially affected by temporal fluctuations. Once decided on a suitable position, the performed relocation influences the behaviour of the network, which in turn again affects the other components due to changing inputs. However, by adapting to the new situation, the already existing knowledge can be further extended by the insights gained from the updated device position. E.g., in case of MOVE, each new measurement point can improve accuracy by providing an additional possibility to verify predictions against reality and perform corresponding adjustments, or for FOLLOW, new device positions can help to better identify or distinguish the effects of obstacles. Hence, MOTION as one example of network optimisation, does not conclude the whole process, but marks the end of one iteration in a continual cycle of knowledge gain.

8 Evaluation

After elucidating the four contributions related to different stages of network knowledge, the evaluations of those approaches are presented in this chapter. First, a description of the utilised scenarios is given (Section 8.1). The second part briefly describes the used hardware and experimentation methodology (Section 8.2). In the following, FLoW is addressed (Section 8.3). Next, MOVE is analysed, validating the presented models (Section 8.4) and afterwards investigating the model quality for heterogeneous device configurations (Section 8.5), followed by the evaluation of Follow (Section 8.6). Finally, MOTION is assessed (Section 8.7) and the chapter concluded by a summary (Section 8.8).

8.1 Scenarios

This section introduces the real-world scenarios used to evaluate the previously introduced approaches and outlines their respective relevance by describing their characteristics. An overview of which scenario is used for each contribution is given in Table 8.1.

Table 8.1: *Summary of the evaluation scenarios and their usages in each contribution.*

Scenario	*MOVE*	*FLoW*	*Follow*	*MOTION*
WSN Testbed	✓	✓	✓	✓
FlockLab	✓	✓		
TWIST	✓	✓		
WiFi Office	✓			
WiFi Hallway	✓			

Figure 8.1: *Building of the* WSN TESTBED.

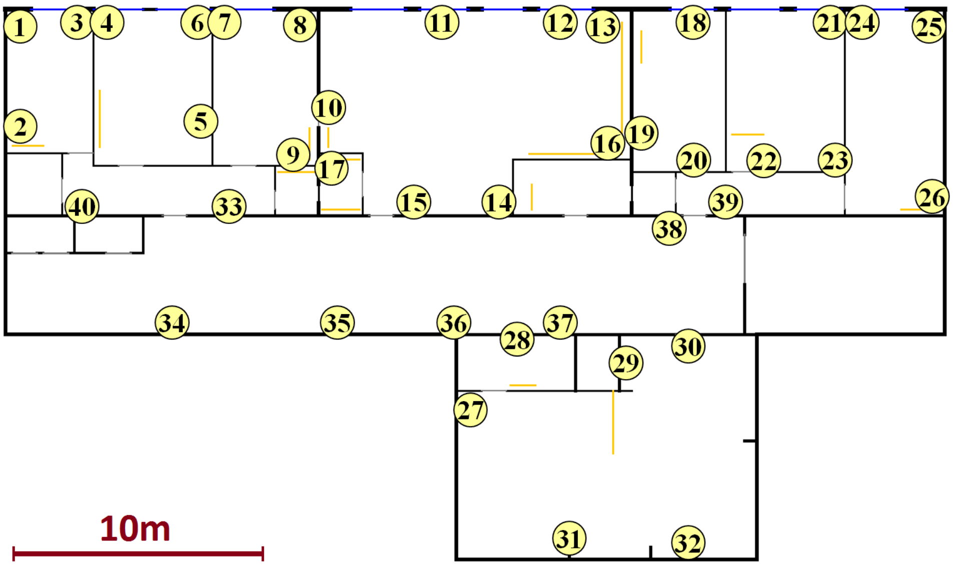
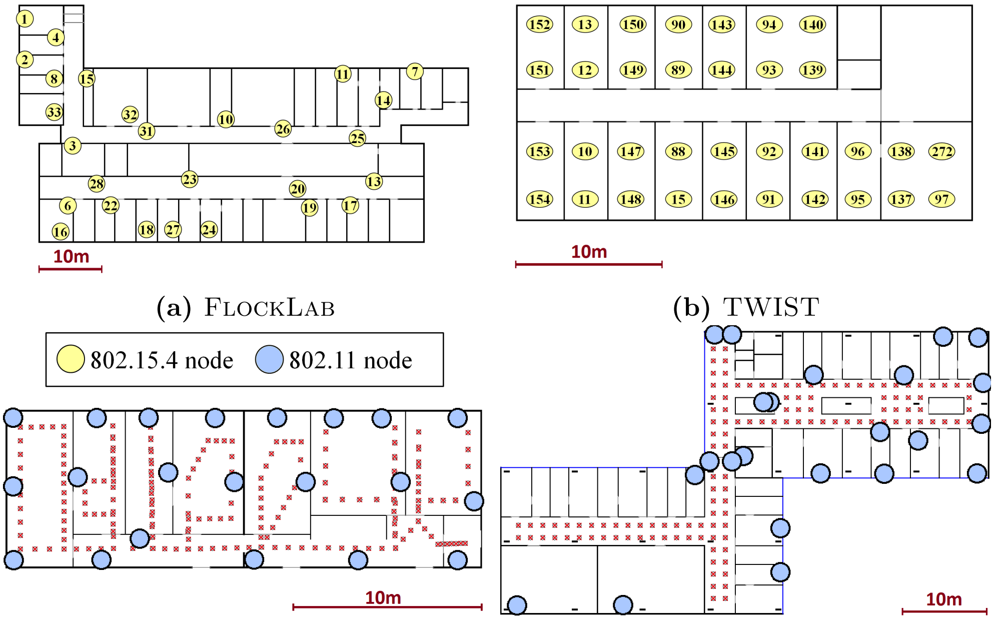


Figure 8.2: *Map of the* WSN TESTBED.

WSN Testbed

The first and main reference scenario of this thesis, shown in Figure 8.1, is a WSN testbed made of TelosB [p120] low-power 2.4 GHz wireless devices located in a building of the university from the early 20th century. The testbed contains 40 nodes, covering an area of 600 m^2 (15 $m^2/node$), distributed over ten rooms, including offices, big laboratories, small kitchens and long as well as short corridors, where also an elevator is present. A map of WSN TESTBED including the device positions is shown in Figure 8.2. This building has distinctive thick sustaining brick walls with high attenuation effect on radio signals, dividing the testbed into four areas, one of which separated from the others by a long corridor.

The placement of devices is based on the group's expertise with deployments of this type. In this scenario, the devices could be reprogrammed to perform RSSI measurements with a scheduling scheme coordinating the transmissions of beacons in order to avoid collisions and receive signals, which



(a) FlockLab

(b) TWIST

(c) WiFi Office

(d) WiFi Hallway

Figure 8.3: *Maps of all used external real evaluation scenarios. Red points mark the positions of the recorded traces (*WiFi *scenarios only).*

are solely influenced by the characteristics of the given building structure. During the experiments, all nodes were active either as senders or receivers, reporting each transmission or reception with the corresponding RSSI. Moreover, this is the only scenario where it was possible to modify the existing network infrastructure as well as to have full control over relevant details of the environment, e.g., room occupancy, furniture placement and states of doors, windows and elevator. Accordingly, WSN Testbed is used as reference scenario for all approaches of this thesis and has the most detailed environment representation of all scenarios, including structural details like wall thickness and material, height information of doors, windows and devices. The experiments conducted and covered in this thesis have been performed over a period of three years, inevitably leading to small changes in the environment. Mainly large shelves and cupboards have been moved over time causing minor fluctuations in RSSI reception throughout collected traces across different analyses. To counteract accuracy changes, adaptations on the scenario representation have been performed as well.

FlockLab

FlockLab [p95], the second scenario, depicted in Figure 8.3 (a), is a testbed located at ETH Zurich with 27 TelosB devices. It covers an indoor area of approximately 2250 m^2 (84 $m^2/node$), entailing a sparser network with fewer links than the WSN Testbed scenario. Due to the larger distances between the devices compared to the other scenarios, signals are required to be transmitted at higher transmission power levels (c.f. Table 8.3) in order to obtain a complete and sufficiently connected topology graph . For this scenario, no further details on the environment were given except for a map containing the room layout and the node positions. FlockLab is used to evaluate MOVE and the capabilities of FLoW, where an outline is sufficient to create a topology.

TWIST

The third reference scenario is TWIST [p58], a testbed at TU Berlin, shown partially in Figure 8.3 (b). With 101 TelosB devices distributed over 3 floors, this is the largest deployment in node count and density. Nodes are attached to the ceiling and arranged in a grid structure providing a uniform distribution. Same as FlockLab, the TWIST scenario is addressed to evaluate MOVE and FLoW as further information and physical control are lacking. Since the base ray tracing model PHY and FLoW were implemented for two-dimensional spaces, there is a limitation to analyse one floor at a time. Due to the almost identical layout and similar results on all three floors, the evaluation is confined to the shown fourth floor since it contains the most nodes (32) which are spread over an area of 465 m^2 (14 $m^2/node$). The more interesting property for the evaluation of FLoW is the rectangular floor plan, causing mirroring problems during the matching phase.

WiFi Office

The WiFi Office scenario (Figure 8.3 (c)) as fourth scenario bases on an existing WiFi infrastructure deployed for fingerprinting-based indoor localisation. The 2.4 GHz WiFi access points are part of a system operating in another floor of the same building as WSN Testbed, with a different arrangement of rooms over a smaller area of approximately 200 m^2. The measurements were collected from a mobile phone recording the RSSI of the beacons received from the various access points, having separate sender and receiver devices and positions and thus unidirectional records only. The observations were taken while walking through the rooms, pausing at each position in order to record one RSSI sample from all access points and their corresponding identifying name. In contrast to the previous two scenarios, the environment has been completely explored, leading to a higher density of 190 observed locations, but with only one sample per transmitter for each position as well as a significant influence of body shadowing. These notable differences in the network structure, measurement collection methodology and the quality of data qualify this as a suitable scenario for evaluating MOVE under aggravating conditions.

WiFi Hallway

The last scenario corresponds to the hallway of a campus building of the University of Mannheim, for which measurement traces are publicly available [m11]. The building consists of two shifted office wings connected through a corridor, covering an area of 1200 m^2. In this scenario, depicted in Figure 8.3 (d), different 2.4 GHz WiFi access points are spread over the area and located inside the offices. The measurements were taken with a laptop equipped with an external antenna, while walking through the hallway, similar to WiFi Office. In total, signal strengths from 130 locations have been collected and no measurements were taken inside the rooms where the access points are located. Also in this scenario, only unidirectional records are available due to decoupled sender and receiver device roles, but with multiple samples per location.

8.2 Hardware and Experimentation Methodology

This section focuses on the hardware used in the main scenario WSN Testbed and moreover elaborates on the organisation of measurements. The evaluations of the main contributions of this thesis are based on TelosB devices [p120] exchanging data wirelessly on the 2.4 GHz frequency via CC2420 radio chip [m9] using an inverted F antenna printed on the board. Due to their limited hardware resources, comparably cheap costs and mobile, battery-driven operation, TelosB devices represent typical hardware found in WSNs and are prevalent especially in the research community. They are part of several testbeds [p14, p58, p3, p95, p13, m8], from which two are also used for evaluation in this thesis, FlockLab and TWIST. However, after having performed and analysed various experiments with these devices, certain hardware-specific properties have been identified, affecting the network behaviour and predictability of link qualities. One im-

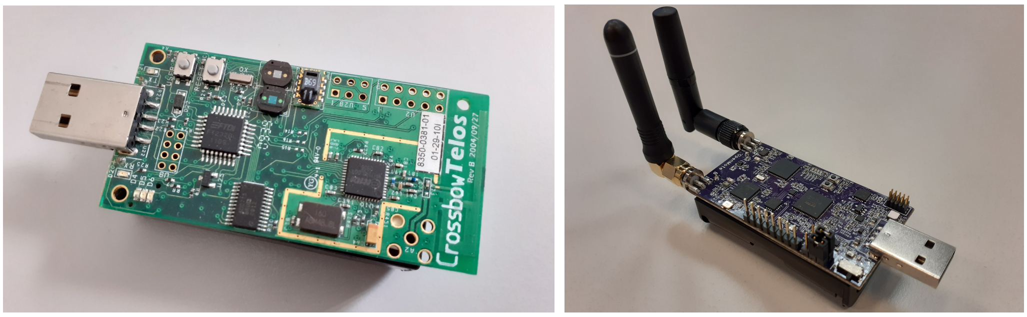


Figure 8.4: *Evaluated devices in* WSN Testbed*, TelosB (left) and OpenMoteB (right).*

Table 8.2: *Comparison of radios used in this evaluation for* WSN Testbed*.*

Device	TelosB	OpenMoteB		
Radio	CC2420	CC2538	AT86RF215	
Antenna	printed F	external dipole		
RX Sensitivity	−95 dBm	−97 dBm	−117 dBm	
Frequency	2.4 GHz	2.4 GHz	2.4 GHz	Sub-GHz
Links @ Pw7	449	534	808	958

portant aspect is the irregular radiation pattern of the printed inverted F antenna, providing significant differences in the received RSSI depending on the relative orientation to a sender. The effects of the radiation pattern on reception are analysed in Section 8.5.1.

For analysing the effect of heterogeneous hardware on the network behaviour (Section 8.5), additional OpenMoteB devices [m12] have been deployed in the WSN Testbed scenario in place of TelosB devices, both shown in Figure 8.4. These use a dual radio architecture, including a CC2538 transceiver [m10] capable of communicating on the 2.4 GHz frequency band and an AT86RF215 transceiver [m5], communicating on both Sub-GHz and 2.4 GHz. While CC2420 and CC2538 have similar receiver sensitivities of −95 dBm and −97 dBm respectively, the AT86RF215 is able to receive signals until −117 dBm increasing the effective range for receiving signals even further. In addition and contrast to the TelosB devices two external dipole antennas are attached, one for each frequency band. Furthermore, all three radio devices use the same offset quadrature phase-shift keying (OQPSK) modulation technique [p57] to send information, ensuring comparability in the performed experiments. Table 8.2 gives an overview of relevant radio characteristics and an exemplary link count recorded in WSN Testbed on the lowest evaluated power level 7 (cf. Table 8.3).

Table 8.3: *Mapping of used power levels to transmission output power.*

Power Level	Pw7	Pw11	Pw19	Pw31
Output Power	−15 dBm	−10 dBm	−5 dBm	0 dBm

Main areas of interest are the network topology and the quality of existing links between the devices, measured by recording RSSI values and sources of received transmissions for each individual device. For the evaluations of this thesis, measurements from different scenarios have been used. These were either pre-recorded at certain transmission powers or, in case of having control over the deployment, performed on various power levels. Using three scenarios, in which TelosB devices are deployed, the power level notation is oriented towards the available settings of these devices, ranging from the weakest transmission power level 3 up to the highest power level 31. As power level 3 turned out to be too weak to establish an operational network in the main scenario WSN Testbed, this level has been omitted, beginning with power level 7 instead, increasing in linear steps up to power level 31. To give an understanding of the nominal power output in dBm and ensure a comprehensive analysis of measurements from different scenarios, a mapping is given in Table 8.3.

8.3 Network Analysis

After discussing FLoW in Chapter 4 and how it allows to realise a localisation with as few information as possible, the approach is validated in this section. To thoroughly assess the capabilities of FLoW to deal with different topologies, varying in shape, size and density, this evaluation is based on real as well as on simulated scenarios, each featuring unique characteristics. An overview of the selected scenarios including network relevant parameters is given in Table 8.4. Aside from measuring the accuracy in real-world deployments, topologies testing FLoW under specific conditions are also analysed, e.g., large networks or unfavourable floor plans. The evaluation is organised in three steps, increasing the amount of given information to assess the corresponding influence on the localisation accuracy. In the experimentation, also the behaviour of MDS [p136] is assessed, a comparable algorithm capable of localising devices only through connectivity data.

Table 8.4: *Summary of the FLoW evaluation scenarios.*

Scenario	*Type*	*Nodes*	*Area*	*Node Coverage*	*∅Links/Node*
WSN P7	real	40	600 m^2	15 $m^2/node$	9.7
WSN P31	real	40	600 m^2	15 $m^2/node$	21.8
FlockLab	real	27	2250 m^2	84 $m^2/node$	8.2
TWIST (4^{th} floor)	real	32	465 m^2	14 $m^2/node$	22.2
S-Shape	sim	57	2850 m^2	50 $m^2/node$	19.0
H-Shape	sim	52	2800 m^2	52 $m^2/node$	9.44
Square	sim	200	22500 m^2	112 $m^2/node$	55.3-58.9

8.3.1 Scenarios of FLoW

To evaluate the viability of FLoW in real-world scenarios, three different testbeds were chosen, for which access to floor plans and multiple connectivity measurements were given, WSN Testbed, FlockLab and TWIST (Section 8.1). While for WSN Testbed two power levels 7 and 31 were considered, for the latter two, measurements were conducted only using the highest power level 31.

In addition, three artificial setups with specific features are also identified constituting a challenge for FLoW, described in the following. In each simulated scenario, the nodes are uniformly distributed and the signal strength of wireless links is derived by the PHY ray tracing model. Even if simulated and characterised through idealised RSSI values, these scenarios allow to stress specific behaviours not evident in the real deployments at disposal.

S-Shape The first artificial scenario is based on an environment with 57 nodes, shaped in the form of the letter S. There exist no bridging connections between nodes in different segments which are not in line of sight resulting in a tube shaped graph which is required to be bent in order to fit the shape.

H-Shape An environment in the form of the letter H, containing 54 nodes is used as second scenario. The possible problem arising here is an intrinsically flipped graph which cannot be solved by mirroring, but rather by identifying and performing local changes on the affected parts of the graph.

Square Lastly, the evaluation scenario of MDS [p136] has been chosen: a large square shaped area with no other obstacles. On the one hand the intention here is to compare FLoW against MDS running under ideal circumstances. On the other hand another goal is to evaluate the performance when dealing with larger networks and a constantly sparse amount of additional information, i.e., anchors. Thereto in contrast to the other scenarios, except for four anchor nodes placed at each corner, 196 of the 200 used nodes were randomly placed in an area of 150 m x 150 m and five different runs were tested.

8.3.2 MDS: Multidimensional Scaling

Many localisation algorithms rely on more information than the mere connectivity data or are based on strict assumptions. E.g., tri- [p114] or multilateration algorithms require to have at least three anchors located in communication range. Similar arguments apply to other range-free algorithms like APIT [p61], Centroid [p26] or DV-Hop [p113]. Others require angular information to work, e.g., in [p112]. Thus a comparison is not possible without providing detailed information. This additional knowledge may increase the accuracy in localisation but also reduces the generality of the approach if such information is not available or can be obtained only with additional effort.

To the best of the author's knowledge, the reference algorithm in the literature relying only on connectivity data is the multidimensional scaling-based approach, MDS [p136]. It requires a completely filled $|N| \times |N|$ matrix, representing the pairwise distance dissimilarities between each two nodes. Without anchor nodes, the use of absolute distances is avoided in the first evaluation cases. As an alternative, the RSSI values can be used. For indirect connections, the minimum hop count is used as distance metric, as done in the original paper. For the implementation, the available MDS Java library [m1] is employed. Both FLoW and MDS are then computed centrally.

8.3.3 Localisation only with Connectivity Information

It is first evaluated if it is possible to understand the physical location exclusively through connectivity information. As mentioned, the lack of reference points results in topologies that are arbitrarily rotated, scaled and mirrored, challenging the direct comparison with a real topology. In particular, the use of absolute metric errors would offer a distorted perspective as the algorithms at this stage can only find the correct topology shape rather than the exact distances. Therefore, for this step, the performance is analysed through a specific metric.

Comparison Metric

To obtain a meaningful comparison, an appropriate scaling factor for the computed topologies needs to be identified, approximating both graphs in size. Since the generated graphs contain different errors on each individual link, it is impossible to find a uniform scaling factor fitting all distances. Therefore, for each pair of nodes the relative distance deviation is computed:

$$\delta_{ij} := \frac{dist(n_{i,real}, n_{j,real})}{dist(n_{i,algo}, n_{j,algo})} \tag{8.1}$$

where $dist(n_a, n_b)$ is the difference between the position of two nodes either in reality or positioned by the algorithm. Then Δ is defined as the set of all pairwise scaling quotients:

$$\Delta := \{\delta_{ij} \mid 1 \leq i < j \leq |N|\} \tag{8.2}$$

The complete set Δ may contain outliers influencing the scaling negatively when computing the average. This error would also propagate to later comparison steps if the graph is significantly over- or undersized. This influence can be minimised by removing c highest and lowest values:

$$c := \left\lfloor \frac{k \cdot |\Delta|}{2} \right\rfloor \tag{8.3}$$

For the evaluation, $k = 0.3$ is used, which provides suitable scaling factors for all given scenarios. Using the set Δ_c recursively defined as:

$$\begin{aligned} \Delta_0 :=& \Delta \\ \Delta_m := \Delta_{m-1} \backslash &\{\min \Delta_{m-1}, \max \Delta_{m-1}\} \end{aligned} \tag{8.4}$$

with $1 \leq m \leq c$ the average of the remaining scaling quotients is computed leading to the final scaling factor s:

$$s := \frac{1}{|\Delta_c|} \sum_{\delta \in \Delta_c} \delta \tag{8.5}$$

The positions of the nodes of the computed graph are then multiplied with the scaling factor s. In the end, the relative error between two nodes can be computed as:

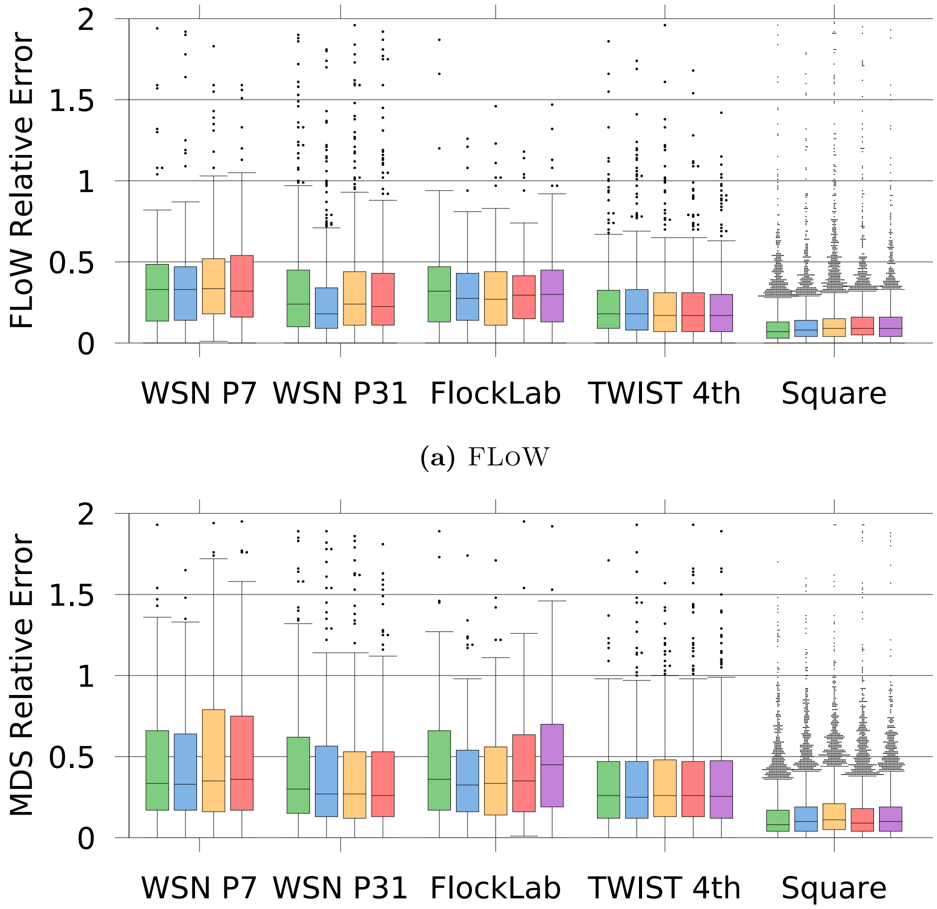
$$\text{relerr}_{ij} := \left| 1 - \frac{s}{\delta_{ij}} \right| \tag{8.6}$$

To not cancel out positive and negative deviations when computing the average error, only the absolute values are considered. Accordingly, e.g., a value of 0.1 indicates that the estimated distance between two nodes differs by either +10 % or −10 % of the original length.

Scenario Comparison

The relative distance errors of both FLoW and MDS, when provided only with connectivity information, are shown in Figure 8.5. For the real scenarios, four to five measurements, taken at different times have been used for evaluation, marked with different colours. In addition to the effects of the permanent static obstacles, influences of different day times and room occupations are reflected in slight fluctuations of the RSSI for each link between the measurements. On the other side, for the artificial scenario SQUARE, data sets have been generated by PHY. Thus, although utilising model predictions, this scenario contains undistorted RSSI values, as they only depend on the distance between devices without any further propagation effects as the map does not contain any obstacles. Hence, these values actually reflect the distance between two nodes, allowing to evaluate the potential of FLoW under ideal conditions. In order to obtain data sets with different RSSI values, device positions have been randomly altered in this case. The error for each pair of nodes is then computed according to Equation 8.6.

In all real scenarios and the MDS reference scenario SQUARE, the median error is similar for both algorithms. However, FLoW exhibits less relative errors for the majority of the links, yielding a graph closer to the real topology. Across different data sets for the same scenario, smaller changes in RSSI and existing links may lead to different graph layouts and link errors. With FLoW, minor changes are counterbalanced due to the force-based mutual influence of vertices, explaining the more uniform results. Noticeable is the slight improvement between the two power levels of WSN TESTBED. By doubling the average amount of links per node from 9.7 to 21.8 (cf. Table 8.4), the average relative link error decreases by 0.10 (0.53 vs. 0.43) as more edges imply more certainty for positioning a vertex in relation to the surrounding ones. This only partially works since, as the nodes become more connected, the graph successively loses its distinctive shape, ending up in a circle in the case of full connectivity. Lastly, in SQUARE both



(a) FLoW

(b) MDS

Figure 8.5: *Relative pairwise distance errors of* FLoW *and MDS algorithms applied on multiple data sets (different colours) from both, real scenarios and the MDS reference scenario* Square. *With* WSN Testbed *P7 and P31, cases are distinguished in which a transmission power level 7 (*−15 *dBm) or 31 (*0 *dBm) were used. Note that the y-axis range has been capped at a value of 2 due to a better visualisation, although few values above this limit exist.*

algorithms show their potential. Without any obstructions, the average relative error over all five data sets is 0.15 for MDS and 0.12 for FLoW, with the latter having more outliers on the one hand, but a lower distance error for the large majority of device pairs on the other hand. Since the area of Square is quite large compared to the other scenarios, this minor difference in the relative error translates to an absolute distance improvement of around one metre in accuracy in favour of FLoW.

Having only RSSI values available, FLoW is able to reproduce the distance relationships between node pairs more accurately than MDS, even in the reference scenario. However, the accuracy of both heavily depends on the influence of obstacles and propagation effects reflected in deviating RSSI values, compared to the pure signal loss over distance. Nevertheless, FLoW is able to better compensate for such distortions due to the mentioned mutual correction of adjacent nodes.

8.3.4 Localisation with the Help of Floor Plans

The use of the floor plan allows to translate relative positions to absolute ones by matching areas and find the most suitable graph configuration. In case of minor disparities in shape, the matching of FLoW forcefully fits the graph into the floor plan. MDS has no anchor node available and it is unable to exploit the information provided by a map of the scenario, leaving no chance to obtain absolute coordinates. To nevertheless enable a comparison, the same FLoW matching is applied on the MDS output. In this case, no further adaptation to the floor plan through graph deformation is possible without losing relative position information. The results are shown in Table 8.5.

When successfully fitting the graph into the floor plan, FLoW achieves better accuracy for node placement than MDS. However, in most scenarios the mere floor plan is insufficient to ensure a reliable matching of the graph, as exemplified in Figure 8.6. Due to the edgy shape, the FLoW matching algorithm succeeds in placing the graph correctly in the WSN Testbed P7 scenario in two cases, but in none for the P31 case. In all of the unsuccessful attempts, the graph is flipped. Increasing the overall connectivity loses distinctive characteristics, reducing the chances of a correct matching. Although the matching may already provide a correct fitting, there are examples for which a correct result is highly affected by ambiguities caused by symmetries intrinsic in the map, as shown in Figure 8.7.

Table 8.5: *Analysis of the impact of the floor plan information. The average accuracy (*∅Acc*) bases on all available traces;* RelErr *denotes the relative error of the average accuracy in relation to the diagonal of the floor plan bounding box; the mismatch column (*Mism.*) shows the number of traces whose graph representation was incorrectly matched or flipped. Due to the different focus, for the simulated* SHAPE *scenarios only one set of connectivity data was generated.*

Scenario	*Algorithm*	*∅Acc*	*RelErr*	Acc_{Best}	Acc_{Worst}	*Mism.*
WSN P7	FLoW	6.55 m	14.9 %	3.24 m	9.77 m	2/4
	MDS	7.57 m	17.2 %	4.14 m	10.97 m	2/4
WSN P31	FLoW	9.22 m	21.0 %	8.53 m	9.89 m	4/4
	MDS	7.96 m	18.1 %	4.75 m	11.04 m	2/4
FLOCKLAB	FLoW	20.05 m	22.5 %	8.22 m	36.73 m	2/5
	MDS	28.83 m	31.2 %	10.16 m	42.41 m	4/5
TWIST	FLoW	15.82 m	45.9 %	8.28 m	18.09 m	5/5
	MDS	10.45 m	30.3 %	8.35 m	17.48 m	4/5
S-SHAPE	FLoW	51.79 m	52.5 %	-	-	1/1
	MDS	52.44 m	53.2 %	-	-	1/1
H-SHAPE	FLoW	41.60 m	49.0 %	-	-	1/1
	MDS	38.78 m	45.7 %	-	-	1/1
SQUARE	FLoW	67.02 m	31.6 %	8.50 m	112.54 m	4/5
	MDS	87.88 m	41.4 %	76.95 m	122.53 m	5/5

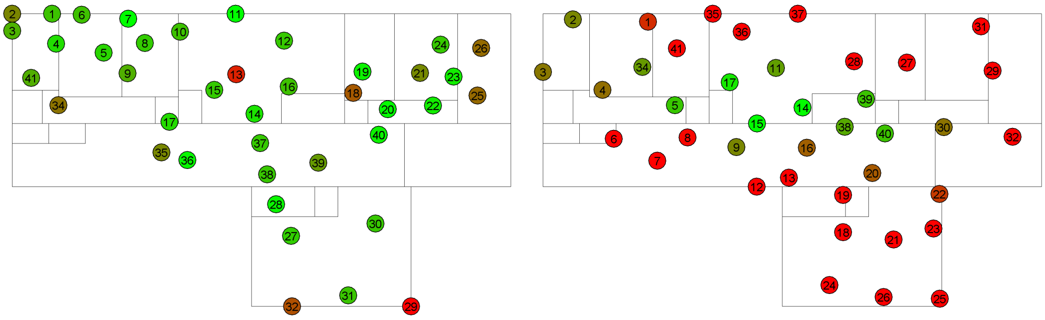


Figure 8.6: FLoW *matching results for* WSN TESTBED *P7 (left) and P31 (right), for which the graph is flipped w.r.t. the correct placement. The colour denotes the difference between shown computed and real position. A pure green (red) colour marks errors smaller than 2.5 % (bigger than 20 %) of the floor diagonal.*

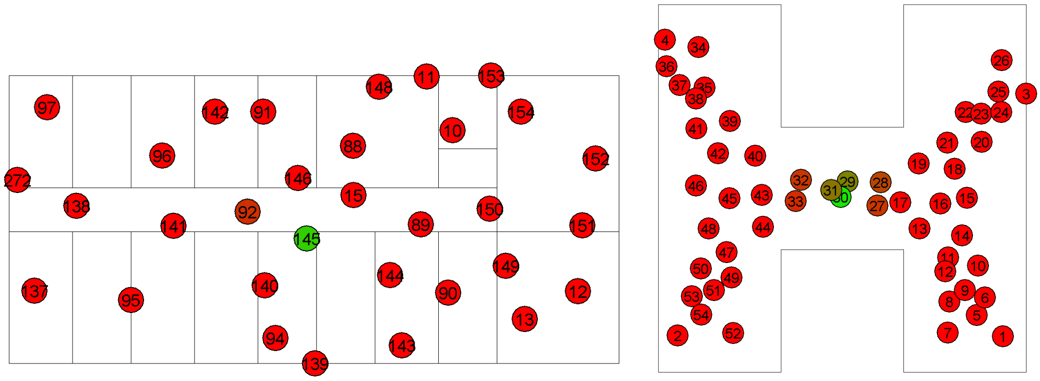


Figure 8.7: *Examples of symmetric floor plans with a successful shape matching but an incorrect node placement caused by the inability to resolve ambiguities. The* TWIST *scenario is shown on the left, the* H-SHAPE *scenario on the right.*

In conclusion, exploiting the information about the floor plan enables the matching of relative to absolute distances and therefore of the estimation of node positions. The success of the matching, however, heavily depends on two factors: the shape of the floor plan and the average network connectivity. First, the less symmetric the shape is, the easier an unambiguous matching can be found. Second, although improving the local positioning of nodes, an increasing connectivity reduces the number of distinctive shape characteristics, complicating the fitting. To compensate for these issues, further information on the positions of some anchor node are indispensable.

8.3.5 Localisation Assisted by Anchor Nodes

Now it is evaluated how a limited number of anchor nodes enables FLoW to resolve ambiguous situations. For each scenario ten sets of random anchor nodes were chosen containing one to four anchors each. The impact of manually selecting anchors placed at crucial positions is also evaluated, ensuring an ideal distribution within the floor plan. Except for a better matching, MDS does not benefit from adding anchors as it cannot be deformed without losing its relative node positions. Thus the presentation of MDS results is restricted to four anchors to still enable a comparison with FLoW. A complete overview of the results is given in Table 8.6.

In real scenarios, FLoW manifests errors within an average range of 6 % to 9 % of the floor plan diagonal. Two examples of successful outcomes are given in Figure 8.8. It is clear that once successfully matched, further anchor nodes do not provide any significant benefit. Minor improvements

Table 8.6: *Impact of an increasing number of anchors (#*A*). The average accuracy (∅*Acc*) is based on all available traces, each evaluated with ten different randomly chosen anchors.* ∅SelAcc *denotes the average accuracy for anchors manually selected close to the corners of the respective floor plans.*

Scenario	*#A*	Acc_{Best}	$RelErr_{Best}$	Acc_{Worst}	*∅Acc*	*∅SelAcc*
WSN P7	1	3.00 m	6.8 %	10.31 m	4.92 m	3.36 m
	2	2.94 m	6.7 %	15.97 m	7.30 m	5.41 m
	3	2.99 m	6.8 %	14.72 m	4.07 m	3.20 m
	4	3.06 m	7.0 %	4.85 m	3.61 m	3.27 m
MDS	4	3.94 m	9.0 %	8.03 m	5.08 m	4.64 m
WSN P31	1	3.17 m	7.2 %	10.05 m	5.63 m	5.51 m
	2	3.00 m	6.8 %	18.78 m	8.43 m	6.25 m
	3	2.64 m	6.0 %	14.77 m	5.65 m	3.31 m
	4	3.05 m	6.9 %	14.39 m	5.05 m	3.19 m
MDS	4	3.92 m	8.9 %	13.93 m	5.72 m	5.36 m
FlockLab	1	5.74 m	6.4 %	36.84 m	10.64 m	9.88 m
	2	5.84 m	6.5 %	26.98 m	14.62 m	11.25 m
	3	5.45 m	6.1 %	26.39 m	8.94 m	6.19 m
	4	5.73 m	6.4 %	19.91 m	8.41 m	6.38 m
MDS	4	9.00 m	10.1 %	30.39 m	12.19 m	10.73 m
TWIST	1	3.19 m	9.3 %	18.29 m	6.79 m	4.78 m
	2	3.16 m	9.2 %	19.21 m	4.86 m	3.84 m
	3	3.12 m	9.1 %	14.44 m	5.06 m	3.82 m
	4	2.91 m	8.4 %	7.81 m	4.05 m	3.86 m
MDS	4	4.21 m	12.2 %	8.65 m	5.53 m	7.14 m
S-Shape	1	16.98 m	17.2 %	53.14 m	25.56 m	18.12 m
	2	16.29 m	16.5 %	24.35 m	19.95 m	39.50 m
	3	17.04 m	17.3 %	34.88 m	21.50 m	15.35 m
	4	14.83 m	15.0 %	21.63 m	18.22 m	12.01 m
MDS	4	19.58 m	19.9 %	44.59 m	24.23 m	22.42 m
H-Shape	1	15.72 m	18.5 %	18.41 m	16.50 m	17.07 m
	2	15.39 m	18.1 %	33.01 m	22.96 m	14.38 m
	3	10.84 m	12.8 %	16.70 m	15.23 m	11.65 m
	4	9.53 m	11.2 %	16.42 m	14.44 m	4.90 m
MDS	4	6.58 m	7.8 %	10.14 m	8.26 m	6.82 m
Square	1	5.48 m	2.6 %	69.81 m	14.62 m	35.29 m
	2	6.76 m	3.2 %	80.68 m	33.88 m	32.85 m
	3	5.50 m	2.6 %	19.94 m	8.87 m	9.84 m
	4	5.32 m	2.5 %	17.17 m	8.10 m	7.80 m
MDS	4	4.47 m	2.1 %	13.39 m	6.01 m	5.59 m

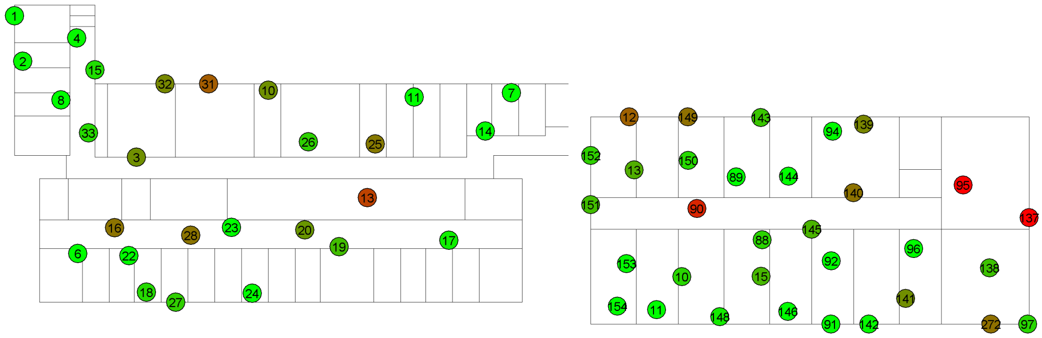
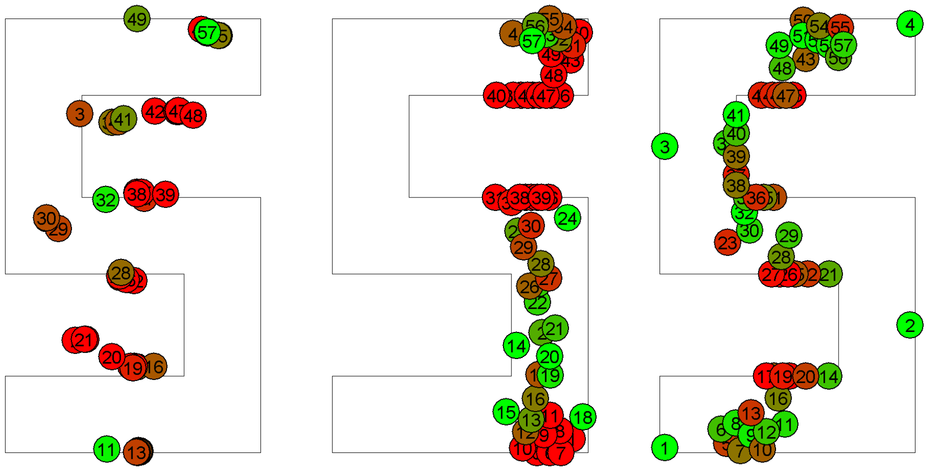


Figure 8.8: *Successful matching examples of the* FlockLab *scenario (left) and the* TWIST *scenario (right).*

are seen for TWIST, where the graph shape has no distinctive characteristic. For the Square scenario, more anchors have no noticeable effect. This can be attributed to the simpler floor plan shapes compared to the other more complex simulated ones where an increasing number of anchors successively reduces the error. In general, the more the shape of a floor plan approaches a convex shape, the better a force-directed algorithm is able to spread the graph throughout the available area. In the majority of the scenarios, FLoW outperforms MDS already with a single anchor. MDS performs better in the specific scenario of reference for its original evaluation (Square) and otherwise only in the simulated H-Shape for randomly chosen anchors.

An increasing number of anchors reduces the chances of wrong graph matches, reflected in a decreasing error in all worst cases. For the real scenarios, three anchors are sufficient to produce correctly matched results for all of the available data sets. Nevertheless, if chosen randomly, it is possible that an adverse set of anchors still yields an incorrect result. Manually selected anchor nodes, instead, provide good results with an error below the average of a random anchor placement.

In general, scenarios with more complex shapes benefit from a deliberate anchor choice. Figure 8.9 shows the specific case of S-Shape. The large difference between the graph generated by FLoW having the shape of a tube and the actual shape of the floor plan pose a challenge to the matching part. In the middle figure, an exemplary result of an inconvenient anchor choice is shown with a graph only occupying the right side of the floor plan. Slightly better distributed, but still not optimal is the result depicted on the right figure for manually selected anchors. In this case, the anchor nodes force the graph to be stretched horizontally. The attraction force of a single node compared to the combined force of several surrounding nodes is too



(a) MDS Random **(b)** FLoW Random **(c)** FLoW Manual

Figure 8.9: *Difference between localisations with random anchors (a/b) and manually selected anchors (c) in the* S-Shape *scenario.*

weak for a better distribution. Also MDS fails to address this problem, in particular considering its inability to account for the scenario shape, as demonstrated by the free placement of devices also outside the boundaries of the physical map.

8.3.6 Computation Complexity

Lastly, the computation complexity of the approach is analysed for each different step and scenario. As reported in Table 8.7, listing the computation time for different scenarios, MDS scales better than FLoW especially in scenarios with a high connectivity where FLoW requires to perform more computations due to an increased number of attraction forces to consider. In general, the behaviour is justified by considering that the processing executed by MDS depends exclusively on the number of nodes n ($\mathcal{O}(n^3)$ [p136]). The performance of the graph generation of FLoW is also affected by the number of links l, thus impacting its scalability for more connected scenarios. During the computation of forces, the position of nodes is updated incrementally, requiring to perform a pairwise node comparison and then to check for the existence of a corresponding link between these two. Hence, the complexity is $\mathcal{O}(n^2\ l)$. Furthermore, because of the unfolding procedure, the force-directed algorithm is applied $r = \lceil \log_2(n) \rceil$ times until the full graph is restored. Due to the reduced number of nodes, each computation only requires a fraction of effort compared to the final one containing all nodes. The total amount of considered nodes can be described

Table 8.7: *Average processing times of different computation steps for each evaluation scenario, performed on a single threaded i7-4550U CPU running at 2.1 GHz.*

Comp.Step	First Graph		Matching (Map)		Matching (Anchor)	
Scenario \ Alg.	FLoW	MDS	FLoW	MDS	FLoW	MDS
WSN P7	48 ms	21 ms	3859 ms	3132 ms	1002 ms	735 ms
WSN P31	105 ms	33 ms	4328 ms	4128 ms	1571 ms	1144 ms
FlockLab	15 ms	5 ms	3460 ms	3273 ms	968 ms	871 ms
TWIST	34 ms	10 ms	2042 ms	1935 ms	1057 ms	858 ms
S-Shape	267 ms	90 ms	5664 ms	3538 ms	3061 ms	1415 ms
H-Shape	154 ms	32 ms	5918 ms	4974 ms	1790 ms	885 ms
Square	29737 ms	9829 ms	195822 ms	29079 ms	293734 ms	26691 ms

as a geometric series $\sum_{k=0}^{r-1} \frac{1}{2^k}$. As it converges to 2 for $r \to \infty$, this repeated execution contributes to the complexity by at most doubling the effort. In addition to that, during the incremental process of graph fitting, scaling and position corrections, the algorithm is run multiple times. Since the main focus of FLoW does not lay on scalability, but rather on a general demonstration of the approach to perform precise localisation with a variable amount of information, additional optimisations indeed have the potential of reducing computation times. A link-based computation with a concurrent update of node positions and a corresponding reorganisation of data structure for example could decrease the computational complexity in this case. However, the influence on the accuracy of FLoW would then have to be further investigated as well.

8.3.7 Discussion

The evaluation has shown that FLoW can localise nodes of a network within a given floor plan using very limited information with a relative error below 9 % for the investigated real scenarios. The main challenge resides in translating the graph produced by the force-directed algorithm to a correct positioning within the floor plan, especially in the case of symmetric shapes. Depending on the scenario, four reasonably placed anchor nodes are sufficient to eliminate ambiguities.

In general, FLoW works better with floor plans having a convex shape with less concave elements. This facilitates both matching and filling the map. On the other hand, it also fosters ambiguities through emerging symmetries, which can be resolved with anchor nodes. Furthermore, force-directed

Table 8.8: *Used metrics for estimating the model quality.*

Model / *Reality*	Link Existent ($\text{RSSI}_{\text{MODEL}} \geq \text{RSSI}^{\text{MIN}}_{\text{REAL}}$)	Non-Existent ($\text{RSSI}_{\text{MODEL}} < \text{RSSI}^{\text{MIN}}_{\text{REAL}}$)	Invalid Output
Link Existent	Real Prediction Error (RPE) $\lvert\text{RSSI}_{\text{MODEL}} - \text{RSSI}_{\text{REAL}}\rvert$	False Neg. Count (FALSE$^-$)	
Non-Existent	False Pos. Count (FALSE$^+$) FALSE$^+$Pred. Error (FPPE) $\lvert\text{RSSI}_{\text{MODEL}} - (\text{RSSI}^{\text{MIN}}_{\text{REAL}} - 1)\rvert$	Match Count (MC)	

graphs require a single connected network without isolated nodes, where degree and uniformity of node connectivity influence accuracy and fitting success. Realising localisation with a lower and heterogeneous connectivity yields a graph with a distinctive shape, easing a matching to the floor plan. Unfortunately, increasing the connectivity removes these distinctive characteristics, ending up in a fully connected circle. At the same time, a higher connectivity reduces the positioning bias as more links disambiguate a node position with respect to its vicinity. Less complex shapes, e.g., FLOCKLAB and SQUARE, allow for a higher connectivity whereas more complex ones, e.g., H-SHAPE, require the opposite. Indeed, the trade-offs between the complexity of the map scenario and the network connectivity need to be carefully analysed. This also opens interesting opportunities to further process the connectivity information based on both the specific positioning goals as well as the properties of the map shape at hand.

8.4 Model Validation

This section evaluates the models part of MOVE, their accuracy and effectiveness in modelling different setups. These models are then utilised by the other contributions either by serving as foundation for FOLLOW and MOTION or providing simulated measurements for artificial scenarios in FLOW.

8.4.1 Quality Estimation

Before turning to the model analysis, the applied methodology of computing the model quality is described first. Depending on the combination of real and predicted existence, several metrics used throughout the evaluation are investigated and computed according to Table 8.8.

The most important metric used in this section for determining model accuracy or more precise the deviation between model prediction and reality is the average RSSI error per link. For this purpose, the RSSI of each possible pair of sending and receiving devices within the network is estimated by the model ($\mathrm{RSSI}_{\text{MODEL}}$) and then compared to the recorded RSSI values in reality ($\mathrm{RSSI}_{\text{REAL}}$). To consider fluctuations and reduce the impact of possible outliers in measurements, the average RSSI of the last 20 recorded packets (if that many available) is taken as $\mathrm{RSSI}_{\text{REAL}}$ for comparison. If a link exists in both, reality and model, the prediction error results from a subtraction of both RSSI values. Same applies to a non-existent predicted link, where non-existence is assumed if $\mathrm{RSSI}_{\text{MODEL}}$ is below the lowest possible receivable RSSI value ($\mathrm{RSSI}_{\text{REAL}}^{\text{MIN}}$) of the radio chip, which has been -95 dBm for the CC2420 of TelosB devices. For existent links in reality, the error between measurement and prediction is referred to as Real Prediction Error (RPE) providing a precise information on the prediction capabilities of a model.

On the other hand, in case a link does not exist in reality, a direct comparison is not possible as the reference value $\mathrm{RSSI}_{\text{REAL}}$ is unknown. Therefore it is refrained from including these links when determining the average RPE to not distort the prediction error by unconfirmed or optimistic assumptions on non-existent links. Instead, to still cover the full set of possible links when evaluating the model quality, non-existent links in reality are considered in separate metrics. A partly exploitable metric yielding a non-binary error is obtainable if the model produces a false positive (FALSE^{+}). To get an estimate of the FALSE^{+}Prediction Error (FPPE) despite lacking the actual theoretical signal strength is to assume it as just below $\mathrm{RSSI}_{\text{REAL}}^{\text{MIN}}$ and thus not getting received. Indeed, this metric is not absolutely precise, but gives an indication of how optimistic a model computes FALSE^{-}. An agreement on the non-existence of a link does not allow to extract any meaningful non-binary information other than counting the number of these as Match Count (MC). In the same way, false positives and false negatives (FALSE^{-}) are counted in addition. Finally, invalid model outputs due to imposed limitations, e.g., no ray reaches the destination within the maximum number of effects to consider, constitute an exception and are not considered when computing the prediction error, but are treated as a predicted non-existent link when counting the secondary metrics.

To obtain a meaningful overall assessment regarding the quality of a model, a Model Quality Estimation (MQE) is defined in Equation 8.7. It comprises accuracy metrics for both, existent and non-existent links, reflecting the distribution of the three occurring constellations of link errors between model and reality by weighting them accordingly. A lower value translates

$$\text{Model Quality Estimation (MQE)} := \frac{\sum_{n=1}^{\#links} p(n)}{\#links} \tag{8.7}$$

$$\text{with } p(n) = \begin{cases} \text{RPE}(n) & \text{if } n \text{ existent in reality} \\ \text{FPPE}(n) & \text{if } n \text{ non-existent in reality, but in model} \\ 0\ dBm & \text{if } n \text{ non-existent in reality and model} \end{cases}$$

Equation 8.1: *Model Quality Estimation (*MQE*).*

into a better model quality yielding more precise estimations. However, it is clear that the MQE itself does not provide the same expressiveness as investigating each individual metric it is composed of, but provides an approach to combine them and assess the model in a simple way[1].

8.4.2 CPU-based 2D Ray Tracing Model

The first part of the evaluation of MOVE analyses the behaviour of the basic two-dimensional physical model PHY (Section 5.3.2) in a stationary environment. In particular, it is focused on accuracy as well as computation time, a metric that is considered as an index of the possibility of using the approach in real-time applications, e.g., for path planning or network monitoring. In order to perform the analysis, a machine with an Intel i7-4550U dual-core processor with four threads, 2.1 GHz per core and 2 GB of RAM running a 64-bit Windows 8.1 was used. To provide comparable results for the computation time, the corresponding tests have been performed in one single thread in a 64-bit Java virtual machine. For determining the model accuracy, RSSI traces gathered from all available scenarios were used. In analysing the accuracy of the models, it is accounted for the fact that the computation of constructive and destructive interference was disabled (even though available in the implementation) due to negative impact imposed by inaccuracies in the model description. Further, it is refrained from integrating a description of the specific radiation patterns of the target wireless devices, whose integration in the required environment description would make the configuration of the ray tracing engines impractical, even more when the information needs to be provided in three dimensions.

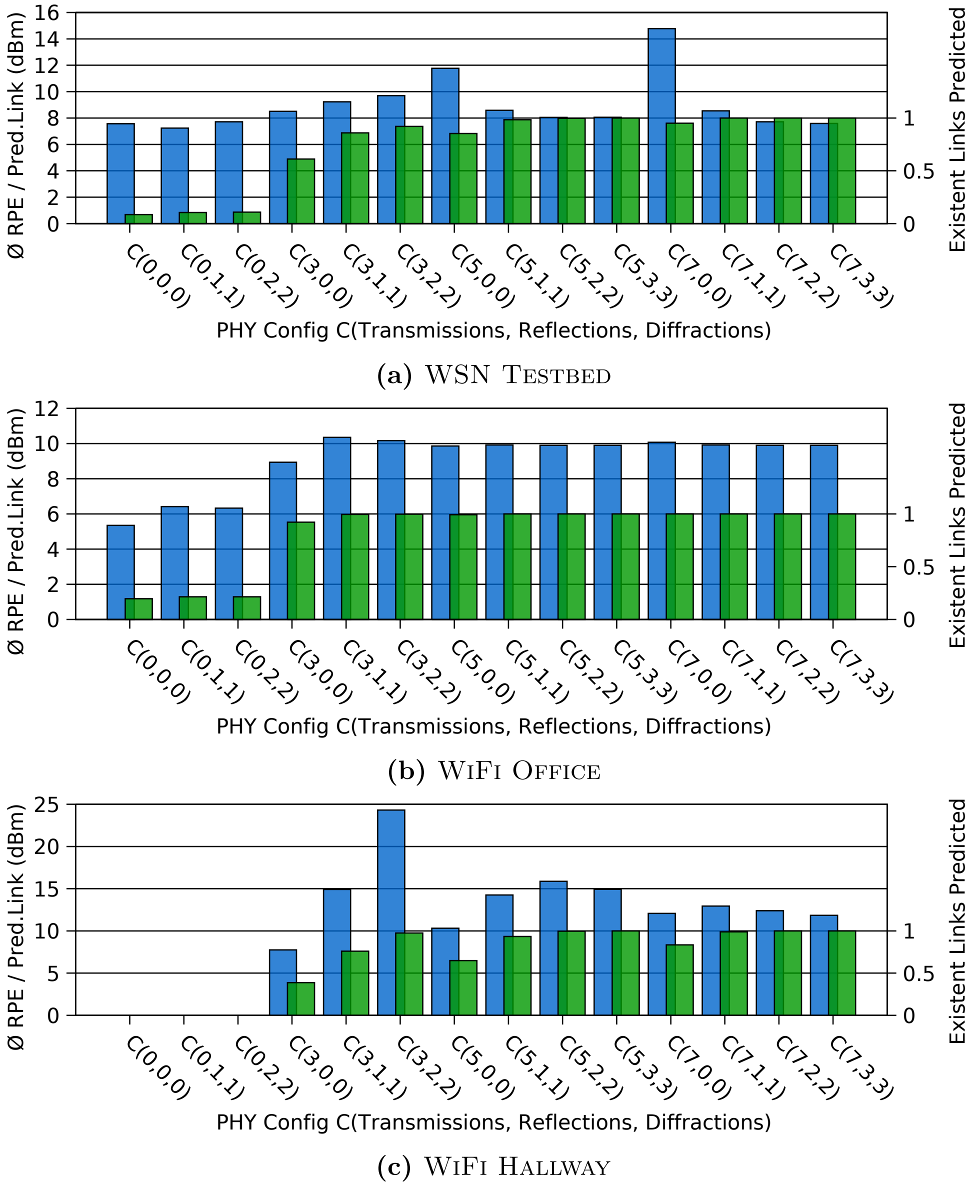
[1] The metrics regarding the prediction quality of the models and their computation, presented in this thesis differ from the ones used in [a3] and [a2]. In the latter, the average RSSI error is computed based on all possible links of a network and thus contains also falsely predicted and non-existent links. A more exact distinction of the possible constellations between reality and prediction (Table 8.8) enables a more differentiated analysis of the models. As a consequence, the presented numbers and graphs regarding the model evaluation deviate with respect to the publications.

Physical Propagation Effects

First, the main parameters involved in the computation of the PHY model are analysed. Beside deployment specific information, e.g., transmission power, noise level, employed radio frequency, the most relevant aspect for both the computation time and the prediction accuracy is the combination of maximum physical effects, i.e., x transmissions, y reflections, z diffractions, considered. This configuration is referred to as $C(x, y, z)$. This analysis serves as a base to understand the trade-offs between computation complexity and accuracy. In this case study, the objective is to find the optimal configuration that minimises both the overall prediction error as well as the individual error on the worst prediction outliers, achieving highly accurate link estimations over a wide area.

The typical assumption is that a higher number of considered effects corresponds to a better accuracy at the cost of longer computation times. In Figure 8.10, the performance of the PHY model is depicted for different combinations of effects considered in the modelling of three scenarios, WSN Testbed, WiFi Office and WiFi Hallway. Using a configuration with too few effects, especially when leaving out transmissions, PHY is barely able to predict the links existing in reality basically limiting estimations to devices of the same room. In WiFi Hallway, no predictions can be made at all, as the access points are located within rooms, suggesting the usage of a sufficiently high transmission count. With an increasing number of existing links covered by the model, the error at first increases as well. Links of larger distance introduce the potentially largest error due to an accumulated error over distance.

Regarding computation speed, the number of considered transmissions in general has a limited impact but provides a substantial contribution for the number of successfully predicted links as well as the achieved accuracy. Necessitating further obstacle collision tests, each additional transmission increases the number of comparisons linearly in dependency of the number of present obstacles: $\mathcal{O}(n \; \#obstacles)$. Since in PHY, the direction of a ray is not affected when passing obstacles, each transmission computation is independent of previous ones. In contrast, the computations of ray paths including reflections or diffractions exhibit a dependency to previous effects on these paths. The recursive path building resulting from the pursued aimed approach requires to compare combinations of multiple obstacles depending on the number of considered reflection or diffraction effects to determine the validity of a path. Therefore reflections and diffractions have an expo-



(a) WSN Testbed

(b) WiFi Office

(c) WiFi Hallway

Figure 8.10: *Accuracy for different* PHY *configurations in* WSN Testbed, WiFi Office *and* WiFi Hallway *scenarios. Green bars indicate the share of existing links,* PHY *is able to compute under a given configuration.*

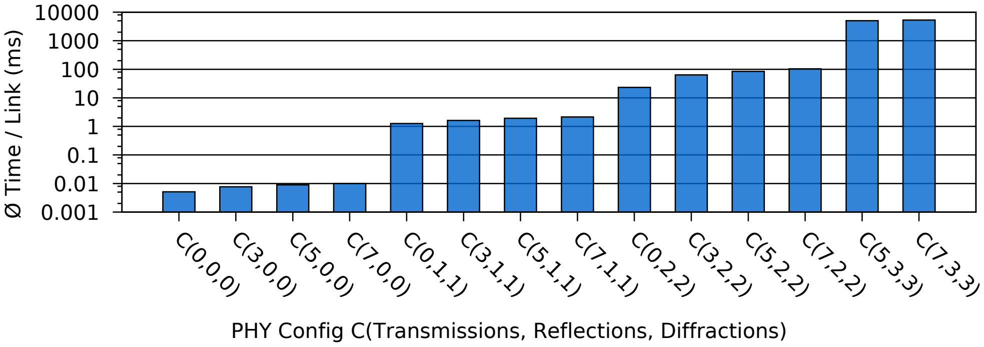


Figure 8.11: *Computation time for different* PHY *configurations in* WSN TESTBED.

nential complexity $\mathcal{O}(\#obstacles^n)$ and affect the computation time much more as shown in Figure 8.11. By considering one more of either effect, the average computation time per link increases by more than one order of magnitude.

Once PHY is able to model all existing links with a certain number of transmissions, including one effect of reflection and diffraction increases the accuracy significantly. The error per link is reduced by more than 3.0 dBm for the RSSI in the WSN TESTBED scenario (Figure 8.10 (a)). In the WIFI HALLWAY scenario (Figure 8.10 (c)), seven transmissions alone do not cover all links due to the complex shape of the building. Only the addition of one reflection and diffraction allows PHY to reach all measurement positions. In case of the WIFI OFFICE scenario (Figure 8.10 (b)), the error remains almost constant when considering more effects as they are conflicting with the body shadowing, distorting the measured RSSI values. Furthermore, more detailed configurations result in declining improvements on the accuracy, which can be ascribed to insignificant influences of weak signals reflected or diffracted multiple times as well as inaccurate and incomplete environmental information. Especially for the WIFI HALLWAY scenario where only rudimentary information about the environment were available, this lack of detail is reflected in a higher overall average RSSI error compared to WSN TESTBED.

As a result, $C(7, 1, 1)$ was chosen, counting seven transmissions, one reflection and one diffraction, which provides a reasonable trade-off between accuracy and computation time for all used scenarios while covering all measured links by the model. However, as it has been experienced when studying the reconfiguration strategies of MOTION, a minimum number of considered transmissions is necessary, especially for achieving precise estimates on the existence of weak links at arbitrary positions. Moreover, these are required to obtain reliable relocation suggestions and avoid non-calculable links.

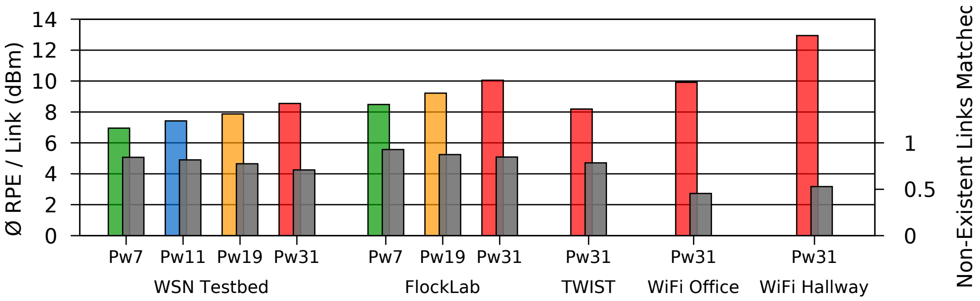


Figure 8.12: *Average absolute error per link in the RSSI prediction (coloured bars) and non-existent link matches (grey bars) performed by the* PHY *model for various power levels using all available scenarios.*

Prediction Accuracy

Second, the accuracy of PHY is evaluated for all given scenarios using the chosen configuration $C(7,1,1)$ and having all main accuracy optimisations enabled, which are material-specific attenuation and the inclusion of available height information. As with incorporating observations in wireless communication modelling, further optimisations become possible. The effect of individual optimisations on the model accuracy is investigated in Section 8.4.4.

In Figure 8.12 the average absolute error per link is shown for each available scenario. RSSI error values are based on all measured existing links since $C(7,1,1)$ allows PHY to compute the entire set of existing links in all cases. In addition, the share of non-existing links is given in which reality and model coincide. Providing further relevant information, Table 8.9 lists the number of links measured in reality and computed by the model including false predictions. Due to limited access or the usage of pre-recorded traces only, not all power levels are available in all scenarios. At higher transmission powers more links become available as stronger signals travel further and thus involve more obstacles on their path. As a consequence, errors resulting from coarse environment descriptions and disregarded propagation effects accumulate increasing the prediction error in general concomitant with a growing number of false predictions. Depending on the size and complexity of each scenario, the method of collecting measurements as well as the amount of details provided, the prediction accuracy remarkably differs.

Focusing on the three most similar scenarios first, WSN Testbed, FlockLab and TWIST, all of them use TelosB devices collecting measurements without direct human involvement and therefore less distorting effects. The WSN Testbed scenario contains the most accurate description of the environment and has average size and complexity compared to the other two

Table 8.9: *Link distribution in reality and model per link for the* PHY *model using different scenarios and power levels.*

Scenario	$\text{LINK}_{\text{REAL}}$	$\text{LINK}_{\text{MODEL}}$	FALSE^{+}	FALSE^{-}	*Share*
WSN TESTBED	1560 possible links in total				
POWER LEVEL 7	473	572	169	70	15.3 %
POWER LEVEL 11	622	713	172	81	16.2 %
POWER LEVEL 19	798	856	171	113	18.2 %
POWER LEVEL 31	984	1027	168	126	18.8 %
FLOCKLAB	702 possible links in total				
POWER LEVEL 7	115	134	42	23	9.3 %
POWER LEVEL 19	204	228	63	39	14.5 %
POWER LEVEL 31	262	268	67	61	18.2 %
TWIST (4^{th} floor)	992 possible links in total				
POWER LEVEL 31	797	728	42	111	15.3 %
WIFI OFFICE	2280 possible links in total				
POWER LEVEL 31	1083	1710	653	26	29.8 %
WIFI HALLWAY	1300 possible links in total				
POWER LEVEL 31	1020	920	132	232	28.0 %

scenarios as reported in Section 8.1 and Table 8.4. In relation to node density and number of links, the PHY model shows the best accuracy results in this scenario. The congruence of non-existent links between reality and model is constantly around 87 %. Considering the FLOCKLAB scenario, the prediction error is around 20 % higher than for the WSN TESTBED scenario as a result of the less detailed provided description of the environment. In contrast, the non-existent match share is higher due to the large area this scenario covers in conjunction with the low density of nodes. Only few nodes are connected having a large number of correctly predicted non-existing links versus a small number of existing links. The opposite occurs in the TWIST scenario having a high node density in a small area. The likewise coarse description of the scenario is compensated by only short-distance link computations yielding less accumulated inaccuracy and thus resulting in an average RSSI error comparable to the WSN TESTBED scenario.

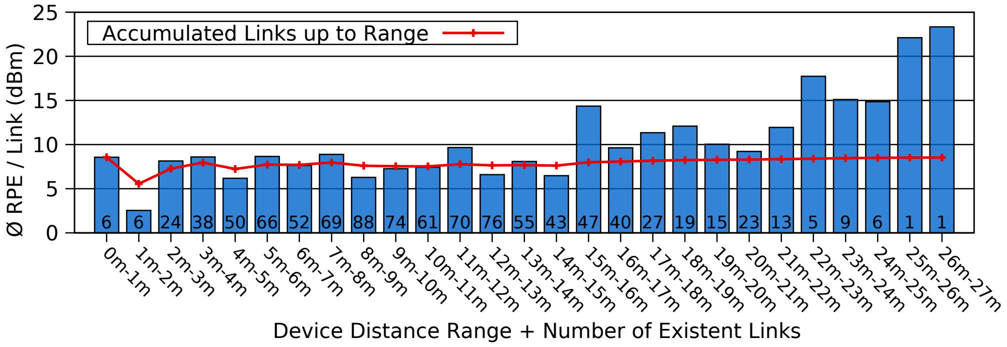


Figure 8.13: *Average absolute error per link in the RSSI prediction based on device distances performed by the* PHY *model for the highest power level in* WSN Testbed. *There are 984 existent links in total.*

Observing PHY behaviour over different power levels, the increasing number of false negative predictions becomes evident as more links at larger distance become available. Accumulated errors from unconsidered paths due to computational limitations provide a reasonable explanation. Another contemplable reason, overestimated attenuation effects of obstacles along the path due to slightly different material properties is disproved as exclusive cause in Section 8.4.3 where the 2D GPU model uses the same attenuation values but produces more accurate results due to more considered effects. In Figure 8.13, links taken in WSN Testbed on the highest power level 31 are grouped by the distances between sending and receiving devices. The average RSSI error computed exclusively for links of each group shows a remarkably increasing prediction error at larger distances starting at around 15 metres but also a nearly constant prediction error before. The red line shows the accumulated average RSSI error up to the given distance interval including all previous links with smaller device distances. Since the majority of links lies within 15 metres, greater errors from more distant links do not significantly contribute to the total average RSSI error.

Turning to the WiFi scenarios, both exhibit special properties which are not modelled by PHY, i.e., occurring body shadowing and constructive interferences, resulting in either too many false positive or negative predictions. The mentioned effect of body shadowing becomes evident in the WiFi Office scenario reflected in a much higher number of links predicted to be existing by the PHY model, but not observed in reality. Therefore, the non-existent match share is the lowest of all scenarios with only 36 %. For the prediction of existing links, the error is comparable to FlockLab although the scenario description is more accurate as the environment materials are known. In the WiFi Hallway scenario, traces were collected with a laptop instead of a smartphone, showing less unexpected dead links in reality, but still the largest overall error. The tube-shaped corridors where

Table 8.10: *Average computation times per link for the* PHY *model used in different scenarios.*

Scenario	∅ CompTime/Link	StdDeviation σ	Obstacle Entries
WSN Testbed	3.21 ms	0.17 ms	214
FlockLab	1.71 ms	0.07 ms	226
TWIST	0.59 ms	0.02 ms	90
WiFi Office	0.50 ms	0.01 ms	59
WiFi Hallway	3.87 ms	0.02 ms	249

measurements were taken create constructive interferences through massive reflections increasing the effective distance of each access point. As interferences cannot be considered properly by PHY when considering a limited number of reflections and using imprecise descriptions of the environment, a more accurate modelling in this scenario is prevented. This assumption is also supported by the large number of false negative link predictions (28 %) compared to the other scenarios. Due to decoupled sender and receiver roles, the number of measured locations in the WiFi scenarios is much higher increasing the potential of generating bad predictions at locations where an only coarse representation of the environment, as present in WiFi Hallway, is particularly disadvantageous.

Scenario Computation Time

Lastly, the average computation time of the PHY model for each link is investigated, shown in Table 8.10. Depending on the complexity of the scenario description, i.e., the number of obstacles to consider, the time varies between a half of a millisecond in the rather small scenarios, TWIST and WiFi Office, and a few milliseconds in the larger scenarios. Although the model representation of FlockLab contains more elements than WSN Testbed, computation times are remarkably lower. Existing obstacles better distribute on the large area of the former scenario, reducing their overall density and therefore resulting in a reduced number of performed collision checks for each link when computing the signal paths. In addition, the squareness of most rooms and missing details, such as pillars or recesses as present in WSN Testbed and WiFi Hallway, leave less edges to consider for computing diffractions in FlockLab and thus save time. On the basis of the average computation times per link, PHY is able to provide signal predictions for entire networks consisting of hundreds of devices located in a medium-sized, well modelled area within a few minutes. This qualifies PHY for being employed in real-time network monitoring and optimisation tools as changes in the position of, e.g., mobile objects, transmitting or passive ones, and the resulting effects on the expected signal qualities can be predicted quickly. Furthermore, the independence of each ray path on the one hand delivers constant computation times allowing for reliable estimations and on the other hand opens the path for further improvements through parallelisation utilising multi-core systems, as examined in the following section.

8.4.3 GPU-based 3D Ray Tracing Model

In this section, differences between the previously evaluated two-dimensional PHY model running on a CPU and the enhanced three-dimensional ray tracing model optimised for GPU execution (Section 5.3.3) are investigated. Furthermore, the prediction characteristics are analysed, relevant for further usage in FOLLOW. To allow for a clear distinction between the approaches, the PHY model is referenced as 2D CPU model within this section. The computation of the ray tracing models was performed on a machine equipped with an Intel i7-7800X with six 3.50 GHz cores and a GeForce 1080 GTX Ti with 3584 cores. The 2D CPU model was configured to account for two reflections instead of one ($C(7, 2, 1)$) as these become relevant when determining possible signal paths involving dynamic obstacles. For the GPU case, each computation cycle started with 50000 rays with a maximum of seven reflections and/or transmissions in arbitrary combinations, resulting in up to 6.4 million rays for each round. Given that each diffraction requires a complete new computation cycle, they were limited to a maximum of two. The reception sphere was set to 0.1 metres.

The evaluation in this section is entirely based on traces collected from WSN TESTBED since it is the only scenario where height information of the environment were available, necessary for meaningful predictions of signal paths in three-dimensional modelling approaches.

2D GPU Model To analyse the differences between the ray tracing engines independently of the added ability to handle full 3D descriptions of the environment, a downsized version of the GPU-based ray tracer has been created handling only 2D scenario descriptions. In this case, the third dimension is fixed and the rays propagate on a plane. This allows to spread much more rays and increase their space density, thus increasing the space accuracy at higher distances at the cost of missing stronger signal paths possibly generated around obstacles detectable only through a third dimension. To make, however, the comparison between the two 2D models fair, the height information is preserved and used a posteriori to compensate for incorrectness in the computed path length.

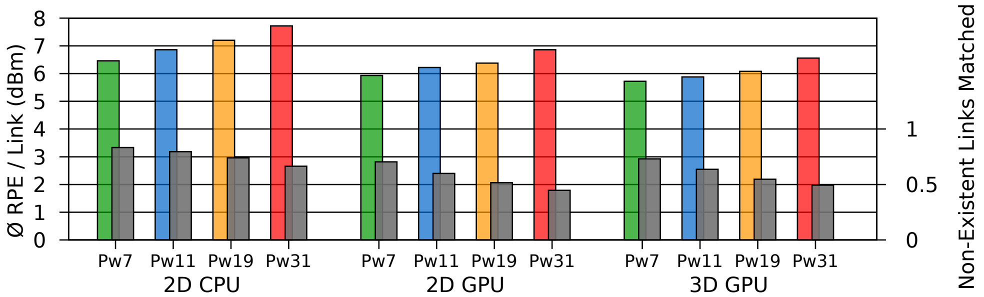
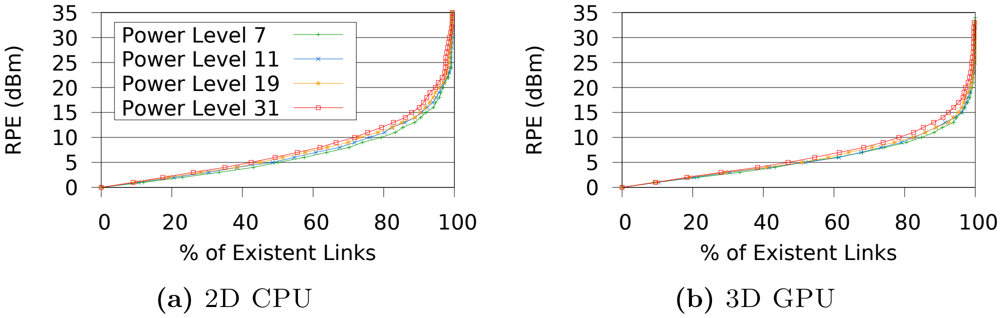


Figure 8.14: *Average* RPE *and non-existent link matches performed by the different CPU and GPU modelling techniques for various power levels using* WSN Testbed.



(a) 2D CPU

(b) 3D GPU

Figure 8.15: *Percentage of existent links predicted within a given RSSI threshold.*

Model Accuracy Comparison

Figure 8.14 shows the accuracy of the models for different transmission power levels. The average error of the GPU models is relatively low if one considers that the map has imprecisions and various real effects are not modelled, like constructive and destructive interferences and radiation patterns. In particular, considering that 2.4 GHz signals are analysed, the misplacement of objects and devices of few centimetres could lead to significant different behaviours. Still, the approach of launching rays into all directions and considering effects at arbitrary combination used by the GPU models provides a noticeable improvement of accuracy. In contrast, although reducing the error even further, the extension of path computations by considering the third dimension shows less distinct improvements than changing the ray computation method. Using 3D GPU, the average RPE reduces between 11.4 % (0.74 dBm) on the lowest power level and 15 % (1.16 dBm) on the highest power level compared to 2D CPU. A further analysis shows that, however, a CPU approach makes bigger errors in predicting the links that exist while performing better for links that do not exist in reality. The opposite applies to the GPU case. This effect is

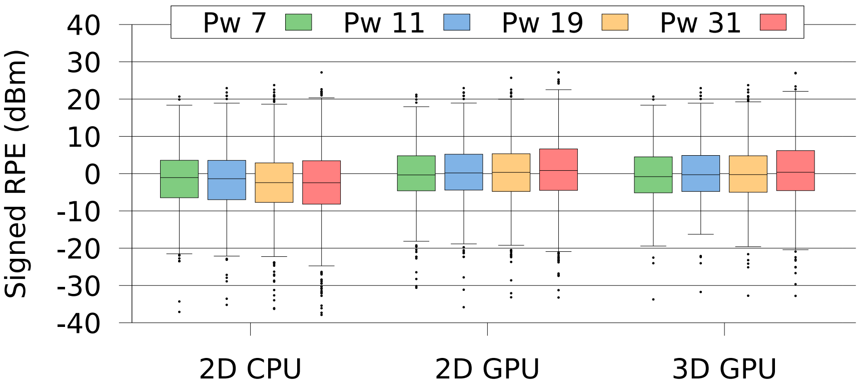


Figure 8.16: *Error per link in the RSSI prediction performed by the different modelling techniques for various power levels.*

analysed in more details later. Figure 8.15 shows a cumulative view of the error distribution for 2D CPU and 3D GPU computations, highlighting how greater errors are restricted to a small percentage of links. In comparison, depending on the transmission power, the 2D CPU model is able to compute around 70 % to 80 % of the existing links with an error below 10 dBm while for the 3D GPU model more than 80 % of the links are below this error. Furthermore, different power levels follow similar trends with a small shift due to an increasing number of existing links for higher transmission powers.

In Figure 8.16, a further view on the accuracy is provided to display the variation inside the set of existent links. Indeed, the 2D CPU model underestimates the received signal strengths of a major part of the links whereas the predictions of both GPU models are almost equally spread. The latter behaviour will turn out to be beneficial during event matching of FOLLOW as more possible ray paths with higher signal strength will be available. More optimistic predictions, on the other hand, manifest in increased FALSE$^+$rates, making it less suitable for reliably predicting links of border regions at unknown positions. Table 8.11 reports the different number of links that each model predicts. Overall, all models foresee more links than what experienced in reality, an expected result caused by the anisotropic antenna of the devices and the absence of a corresponding radiation pattern description in the models. This fact is clear when looking at the number of false positives in comparison to the false negatives. To confirm the conjecture, the number of the former is significantly higher than the latter. Especially the GPU models show a larger gap than 2D CPU, having a

Table 8.11: *Comparison of the number of predicted existing and non-existing links for different power levels in the* WSN Testbed *scenario.*

Used Model	2D CPU	2D GPU	3D GPU
Power Level 7			
$\text{Link}_{\text{real}}$	473		
$\text{Link}_{\text{model}}$	591	765	729
$\text{False}^{+}{}_{(\text{pred}< -90\ dBm)}$	181 (97)	326 (201)	291 (180)
$\text{False}^{-}{}_{(\text{real}< -90\ dBm)}$	63 (33)	34 (11)	35 (17)
Power Level 11			
$\text{Link}_{\text{real}}$	622		
$\text{Link}_{\text{model}}$	739	957	937
$\text{False}^{+}{}_{(\text{pred}< -90\ dBm)}$	191 (107)	375 (200)	341 (188)
$\text{False}^{-}{}_{(\text{real}< -90\ dBm)}$	74 (28)	40 (17)	26 (11)
Power Level 19			
$\text{Link}_{\text{real}}$	798		
$\text{Link}_{\text{model}}$	910	1128	1108
$\text{False}^{+}{}_{(\text{pred}< -90\ dBm)}$	198 (129)	369 (181)	345 (177)
$\text{False}^{-}{}_{(\text{real}< -90\ dBm)}$	86 (33)	39 (13)	35 (17)
Power Level 31			
$\text{Link}_{\text{real}}$	985		
$\text{Link}_{\text{model}}$	1084	1269	1230
$\text{False}^{+}{}_{(\text{pred}< -90\ dBm)}$	193 (106)	318 (120)	291 (119)
$\text{False}^{-}{}_{(\text{real}< -90\ dBm)}$	94 (37)	34 (15)	46 (24)

tendency of overestimating most links. However, for higher power levels, the 2D CPU model evinces the mentioned tendency of underestimating links by producing an increasing number of false negative predictions while the GPU show no remarkable tendency.

Computation Time

Table 8.12 describes the average time taken to compute the model of a wireless link. Indeed, the massive parallelisation capabilities of the graphics card help to complete the three-dimensional computation on a GPU slightly faster than the computation performed by a CPU in two dimensions only. To make the comparison fair, the two-dimensional model running on the

Table 8.12: *Average computation times per link performed by the different CPU and GPU modelling techniques using the* WSN TESTBED *scenario.*

Used Model	2D CPU	2D GPU	3D GPU
∅ COMPTIME / LINK	12.70 ms	18.23 ms	12.29 ms
STDDEVIATION σ	0.40 ms	0.38 ms	0.10 ms

GPU performs the same postprocessing to include the height information that is also computed in the CPU case. Given that such computations can be performed only once the GPU has processed all the rays and, therefore, are performed afterwards by the CPU, the time taken to describe a wireless link in the 2D GPU model is the highest.

8.4.4 Statistical and Hybrid Modelling

The last models of MOVE, the STAT and HYBRID models are jointly investigated in this section, mainly focusing on the latter. MOTION bases on the HYBRID model fusing the description of the wireless signal propagation provided by the STAT and PHY models. For the STAT model the prediction accuracy heavily depends on both, quality and quantity of available measurements.

To provide a comprehensive analysis, three heterogeneous scenarios were chosen characterised by varying physical properties of the environment, types of wireless technology, as well as distribution and amount of available measurements. For all three scenarios, WSN TESTBED, WIFI OFFICE and WIFI HALLWAY (Section 8.1) data was collected or available traces were used in order to analyse the corresponding model accuracy and effectiveness. Relevant characteristics of these scenarios for the HYBRID model are briefly described in the following paragraphs.

WSN Testbed Scenario Peculiar to this scenario is the sparse, reduced number of observed locations (40) with a high number of samples (20 per link). This provides the STAT model with reliable mean values on the one hand and sparse resolution of the observations on the other (1 per 15 m^2). Therefore, WSN TESTBED is representative of scenarios with few locations for calibration, making the HYBRID model rely mostly on the PHY description of the environment.

Table 8.13: *Impact of optimisation features for* PHY *and* HYBRID *models on the accuracy for* WSN TESTBED *measurements taken on power level 19 having a balanced ratio of 798 existing and 762 non-existing links. The* HYBRID *model uses all* PHY *optimisations by default.*

Optimisation	∅RPE	∅FPPE	FALSE$^+$	FALSE$^-$	*Share*	MQE
PHY Base	5.93	7.35	405	16	27.0 %	4.94
PHY Angle Dependency	6.02	7.47	373	18	25.0 %	4.86
PHY 2.5D	5.91	7.65	414	11	27.2 %	5.05
PHY Material Attenuation	7.63	7.23	241	95	21.5 %	5.02
PHY Material + 2.5D	6.73	7.18	256	72	21.0 %	4.62
PHY All Optimisations	7.20	6.85	198	86	18.2 %	4.55
HYBRID Parameter Opt.	7.02	6.66	203	80	18.1 %	4.46
HYBRID Node Adjustment	6.93	6.76	186	82	17.2 %	4.35
HYBRID All Optimisations	6.75	6.69	188	74	16.8 %	4.26

WiFi Office Scenario As this scenario has a high density of observed locations (1 per 1 m^2), the HYBRID model can base mainly on the description provided by the STAT model. In addition, involving observations will aid compensating for body shadowing effects implicitly included in the collected RSSI traces, which the PHY model does not take into consideration.

WiFi Hallway Scenario In this scenario, fewer obstacles are present in the measured area, i.e., the corridor in the middle of the building (400 m^2) with 130 observed locations (1 per 3 m^2), which is also where the accuracy of the HYBRID model can be verified, favouring the STAT model. Moreover, in contrast to the other two chosen scenarios, the representation of the environment was solely based on a given elementary map from which it was not possible to derive any characteristics of the building materials or the presence of smaller but possibly relevant obstacles.

Model Optimisations

Before evaluating the HYBRID model in comparison to other models, the effects of specific optimisations on the accuracy are investigated first. In addition to the PHY model optimisations, the exploitation of the available connectivity information from the deployment enables MOVE to perform further optimisations of PHY within the HYBRID model, as mentioned in Section 5.3.5. Table 8.13 shows the optimisation features and their impact on the model performance in the WSN Testbed scenario, divided into the major metrics as well as the combined share of False$^+$and False$^-$on the entirety of possible links. To demonstrate the benefit of incorporating observations for calibrating the underlying physical model, the reported values of HYBRID base on the exclusive output of PHY, ignoring the predictions of STAT. Goal of the optimisations is to adapt links in an individual way, reproducing the behaviour of the signal within the environment in conformity with reality and hence minimising the values of all listed metrics. Too general adaptations, equally affecting a large portion of links will mostly shift the relation between false positive and negative predictions, but not reducing the overall number.

Initially, the base PHY configuration is investigated, where a different but fixed attenuation coefficient is added for each type of effect, independent of the material type and thickness or the incident angle of a ray. The average RPE is comparably low, but the high number of false predictions, especially false positives is noticeable. These are a direct consequence of too weak attenuations, neglecting the signal path dynamics. Adding an angle dependent attenuation factor slightly increases both prediction errors, but reduces the false positives and thus improving the MQE. In particular, rays being diffracted at edges are attenuated more, the larger the angle of change in direction turns out. Therefore this optimisation rather changes predictions in a more pessimistic way. Having height information of the environment available, a three-dimensional postprocessing is applicable, adjusting the signal path in two ways. First, depending on the height difference of sender and receiver, the length of each signal path and hence the path loss are increased in most cases. The more important second effect is the possibility of avoiding obstacles with limited height which would have been considered otherwise in a pure two-dimensional computation. With potentially less obstacles on their path, predicted signals are more optimistic, reflected in an increased false positive count and a worse MQE when considering the 2.5D computation without individual material attenuation. In fact, adding height information to a model that inappropriately describes the material characteristics further accentuates the starting inaccuracies. Since transmissions have the most significant effect on the model accuracy, an inclusion of material type and thickness adds the aforementioned individual changes to specific links. As result of higher attenuations due to different material types compared to a flat addition of a fixed value, PHY predictions become less optimistic. While increasing the average prediction errors, the total number of false predictions significantly decreases. However, the combination of the latter two approaches complementing each other achieves better results. These are yet increased by adding the angle dependency having all optimisations enabled, further eliminating false predictions yielding the best MQE. Compared to the base version, the optimised version of

PHY has a lower accuracy but a higher precision when predicting links in the border region of reception. This is particularly important when optimising networks to, e.g., explore positions of new devices which are usually not placed in areas where the reception is strong enough anyway.

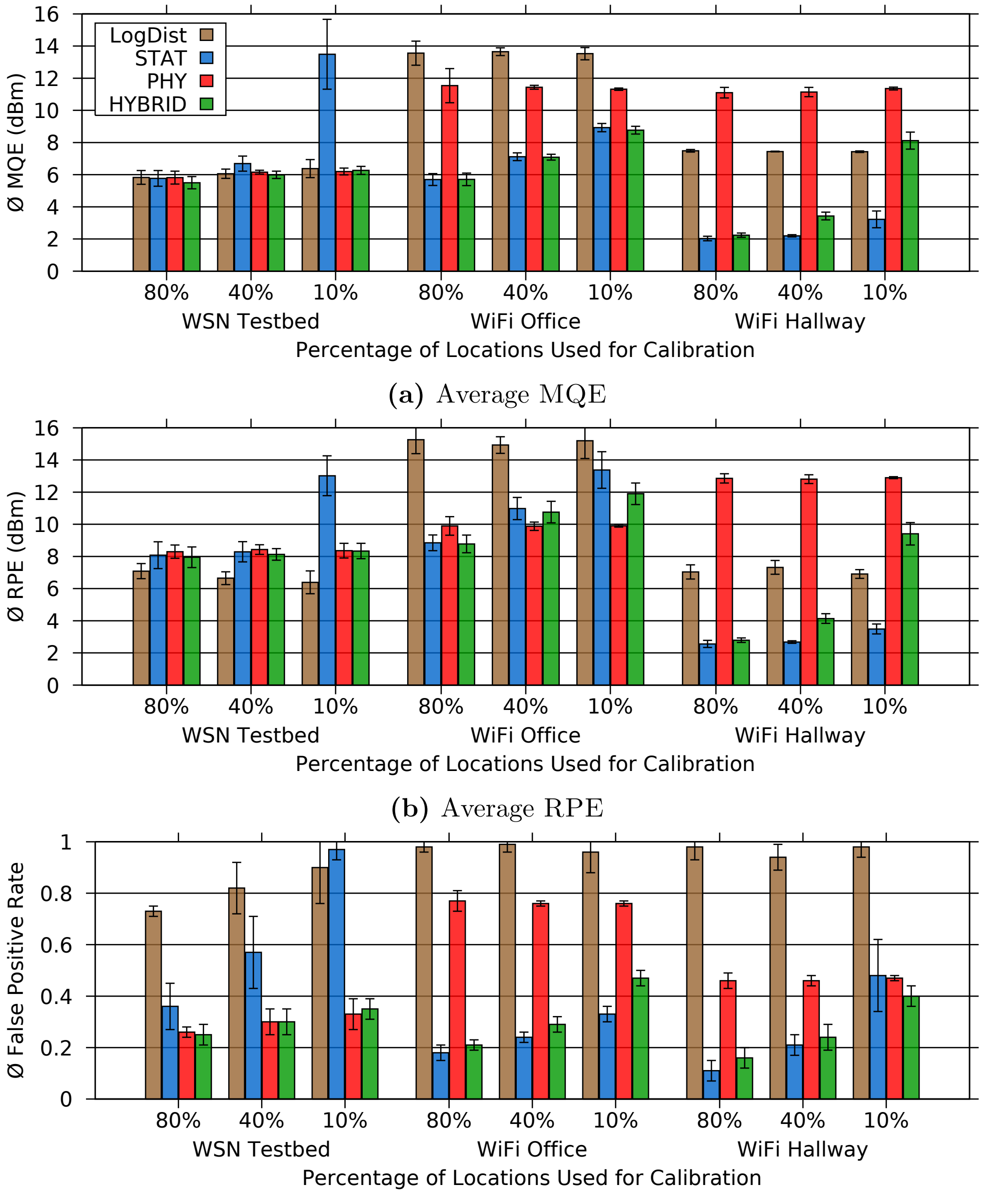
Basing the predictions of the ray tracing component on the most optimised version of PHY, the HYBRID model utilises available measurements of the network connectivity allowing further calibration. The individual sender parameter optimisation slightly improves all of the metrics. Same applies to the device position adjustment reducing the number of outliers and highlighting the impact of a few centimetres on the link quality. Lastly, combining available knowledge from environment and network clearly demonstrates the benefits. Compared to the fully optimised PHY model, all metrics are improved when using HYBRID optimisations, improving the overall model quality even further.

Conclusively, applying the full range of available optimisations achieves improved model predictions closer to reality. Primarily, the number of false link predictions is significantly reduced by more than 10 % in conjunction with a decreasing average FPPE. Although the average RPE increases by roughly 1 dBm on the other hand, the MQE still improves as the false rate reduction has a larger impact. In most cases, reliable predictions regarding the general existence of links in critical border regions or an adherence of thresholds are of more interest for network optimisations than exact predictions, albeit preferable. Furthermore, due to inaccurate scenario descriptions and necessary trade-offs between computation time and details, certain errors are inevitable.

Model Accuracy Comparison

Having chosen the proper configuration for the HYBRID model, the overall accuracy and the computational effectiveness of the resulting modelling are now evaluated in comparison to the individual models. In this section, the same machine setup is used as for the PHY model in Section 8.4.2. As references, the individual STAT and PHY models were used. In addition, the commonly used log-distance path loss model [p123] is also considered, computing the path loss according to Equation 3.1.

The path loss at reference distance d_0 is set to 1 metre and the parameter γ is trained with the same measurements used for the calibration of the STAT and HYBRID models. It is adjusted so that the predictions best match the given measurements with the lowest possible accumulated squared RSSI error. To enable an evaluation of the trace-dependent models, the RSSI traces gathered for each scenario were divided between a certain percentage



(a) Average MQE

(b) Average RPE

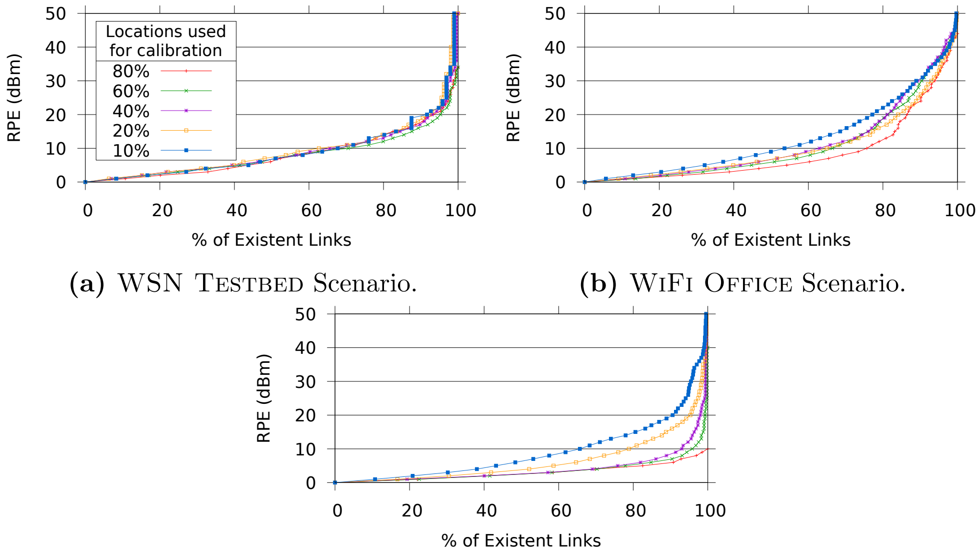
(c) Average FALSE^{+}

Figure 8.17: *Quality metrics and their standard deviation for different calibration sets (5) of the RSSI prediction performed by the different modelling techniques for various amounts of observed locations used for training.*

of observed locations used for the calibration of the models and the remaining locations, whose available measurements serve as ground truth. The accuracy is then computed as MQE, RPE and False$^+$rate for each measured link not used for the calibration and the corresponding model output. To ensure comparability to WiFi Hallway and WiFi Office, only the highest power level 31 has been used in WSN Testbed.

Starting with the log-distance model first, it achieves almost similar MQE results as the models of MOVE in WSN Testbed and even outruns them regarding the RPE, as shown in Figure 8.17. The quality of the log-distance model depends on the availability of measurements to find appropriate coefficients specific to the environment for configuration. If not obtainable, it performs worse as the value has to be estimated based on the type of environment the model is applied to. One reason for the good performance is the equal distribution of devices within the scenario, providing a suitable path loss average over all distances, qualifying the model as a good estimation for existing links. Thus the ability of the model to generalise improves accuracy compared to PHY especially when not having an accurate environment description, as seen in WiFi Hallway. On the contrary when having inconsistent measurements, e.g., resulting from body shadowing as in WiFi Office, it performs worse than the other models of MOVE. However, the biggest disadvantage of the log-distance model becomes obvious when examining the False$^+$share on the number of non-existing links. Except for WSN Testbed with sufficient calibration data, more than 90 % of these links are always incorrectly assumed to be existing, clearly disqualifying the model for exploring alternative positions for devices. As this is an integral part of MOTION, the log-distance model is an inappropriate choice for identifying specific spots in order to optimise a network.

Now focusing on the models of MOVE, the PHY and STAT models behave differently in each scenario. In general, for WSN Testbed, both models provide quite accurate average estimations per link with significantly less False$^+$than the log-distance model. This is particularly true for PHY, which is provided with a more detailed map of the environment. Concerning the other scenarios, STAT clearly outruns PHY. In these cases, in fact, an overall higher density of observed locations corresponds to a better estimate of communication based on actual measurements. As expected, the accuracy of PHY is independent of the number of samples used for calibration, while STAT performs worse for smaller training sets. Finally, reducing the data points has the lowest effect on the error of the STAT model in WiFi Hallway, since the symmetric arrangement of observed locations in the measured areas compensates for the omission of a large number of samples. Only when reducing the sample size to 20 % and below, areas not covered by observations significantly increase where HYBRID relies on PHY predictions, resulting in a worse MQE and revealing the deficiency



(a) WSN Testbed Scenario.

(b) WiFi Office Scenario.

(c) WiFi Hallway Scenario.

Figure 8.18: *Percentage of links predicted by the* HYBRID *model within a given RSSI threshold using different numbers of training samples.*

of the environment description. If the errors of both individual models are compared to the error generated by HYBRID, it can be seen that the latter is always able to closely follow the best prediction performed by either STAT or PHY. This demonstrates its general applicability in heterogeneous scenarios and its independence from the measurement methodology.

In Figure 8.18, the cumulative distribution of the RPE for all existing sender-receiver pairs is reported, for different sizes of training data. Confirming the observations from the previous charts, the training set size has an impact on the error per link, the more HYBRID bases its predictions on measurements. In WiFi Hallway, this dependency becomes most salient. Due to the high number of measurement points, a reduction of training samples down to 40 % does not significantly influence the errors for 80 % of the links. The balanced interplay of STAT and PHY in WiFi Office results in a more steady general degradation of the prediction error when using less samples. For the most erroneous 25 % of links, less training samples (20 % vs. 40 %) even reduce the errors. At that point, PHY takes over most of the predictions, providing better estimations than STAT despite not considering body shadowing effects (cf. Figure 8.17). Lastly, considering the WSN Testbed scenario, the training set size has almost no effect on the distribution of link errors except for insignificant fluctuations due to the random choice of considered devices as the density of observation points with regard to the covered area is low. Using a sufficiently large set

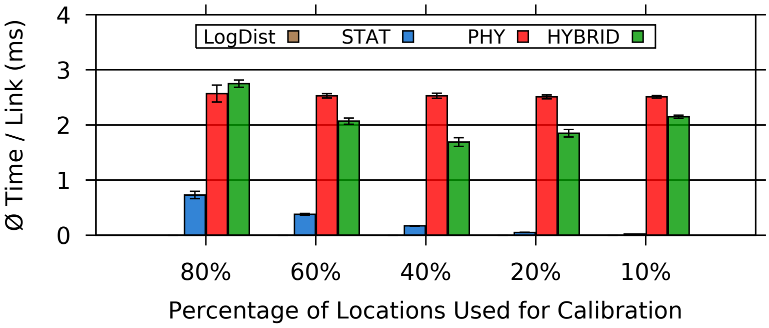


Figure 8.19: *Time required to compute the different models in the* WiFi Hallway *scenario. The log-distance model, barely visible, performs best in terms of computation time*

of training samples, HYBRID is able to predict 45 % to 85 % of the existing links correctly within a range of +/− 5 dBm with 80 % of the training data. Reducing the training data to a minimum of 10 %, the rate is still at 30 % to 45 %.

Computation Time

Finally, the required computation effort is analysed. As shown in Figure 8.19, it is possible to observe the impact of the number of observed locations used for calibration on the computation of STAT. PHY is independent from the training set and the log-distance path loss model is extremely fast (barely visible in the figure) at the cost of a worse False$^+$rate. In the HYBRID approach, the computation time is driven by the different areas in which the estimation is based either on the STAT or the PHY component exclusively or the combination of them. As a result, the computation time decreases with the number of calibration data used, until the reduced observed locations require a bigger part of the environment to be described by the PHY component. In contrast to PHY and STAT model where the setup of the model is performed almost instantly and optimisations are applied during link prediction, the optimisations of HYBRID are applied beforehand requiring an initial pre-computation phase. To find the best configuration, parameter and node position exploration perpetually compare predicted against measured values for different settings. Therefore the necessary time depends on the number of effects, the underlying PHY model considers for performing predictions. Using $C(7, 1, 1)$, the parameter optimisation takes around 30 seconds, the node position optimisation around 90 seconds to finish, scaling in the same order of magnitude as the link computation time (Figure 8.11).

8.4.5 Discussion

The evaluation of MOVE has revealed strengths and limitations of the different approaches which are now discussed. All investigated models have demonstrated the capabilities of predicting wireless signals within few milliseconds and with a reasonable accuracy, scaling with the quality of the respective required inputs.

The accuracy of the ray tracing models depend on the level of details of the provided descriptions of the environment. More complex and cluttered environments will require more time to be described depending on the targeted accuracy also at an additional computational cost, even more in the case of 3D computations. Similarly, the STAT model depends on the distribution and density of provided observations. Measurement campaigns typically require a lot of time. Nonetheless, by sacrificing accuracy, MOVE is able to provide first estimates that can be further refined upon need by monitoring specific areas of the environment.

Being fully adaptable to the requirements, after finding the best tradeoff between prediction accuracy and computation time with respect to the given scenario, PHY is able to provide accurate link descriptions. The aimed approach of PHY provides a reliable way of determining the essential ray paths between two devices. Nevertheless, exponentially increasing computation times for each additionally considered reflection and diffraction effect limit their numbers and require simplifications to ensure a better scaling, especially in larger scenarios with many obstacles. Thus as a consequence, more complex ray paths including multiple effects, which could potentially reduce prediction errors are neglected. Applied optimisations at other stages of computation, like the consideration of height information or material properties help to mitigate or counteract the potential accuracy losses of these restrictions.

By offloading ray tracing computations to the GPU, adding a third dimension becomes realisable without increasing computation times and requiring only minor adjustments to the environment representation. In comparison with the 2D CPU model, the 3D GPU model in general provides more optimistic predictions improving accuracy for existing links. Besides having more possible ray paths to discover due to the launching approach, another reason for this is the increased number of effects taken into account. Accordingly, the availability of special GPU hardware presumed, the 3D GPU model is an appropriate choice for modelling dense networks with many links. However, despite the larger but yet not unlimited number of considered rays, the launched approach bears the risk of completely missing a destination position at larger distance although in reality a path exists. To prevent this possibility, 3D GPU also includes a small set of aimed rays, targeted directly or indirectly via floor or ceiling at the destination. As a further consequence, the richness of discovered paths decreases with distance until being degraded to the basic paths at last. Therefore 3D GPU provides a more exhaustive and resource intensive modelling solution, able to discover further signal paths at close and medium distances, thus showing less prediction errors for existing links than PHY.

Comparing the aimed PHY model against the launched GPU models reveals strengths and weaknesses of both. PHY has a higher chance to discover relevant essential paths with few effects on larger distances than the launched GPU models. In the latter case chances degrade over distance since the probability of initial rays hitting the right obstacles in the correct positions to reach the destination reduces with distance. However, the consistency of path discovery of PHY over distance comes at the cost of considering less effects as well as a higher computation time per ray. Conversely, the GPU models contribute to a lower RPE for existing links at closer distances due to considering more effects and thus discovering more ray paths. Hence the fields of application for aimed and launched approaches differ. The aimed approach of PHY is more suitable for applications where spatially independent reliability is preferable, e.g., for considering possible available links when determining new device positions, to improve connectivity, as done in MOTION (Section 8.7). The increased number of discovered ray paths of a launched approach can rather be exploited to, e.g., analyse the influence of certain obstacles on the quality of links in their vicinity, as required for FOLLOW (Section 8.6). Nevertheless, for both ray tracing models the waiver of centimetre accuracy as well as inaccurate environment descriptions become noticeable in larger scenarios and at high transmission powers when dealing with increasing distances as the prediction error generally increases. Furthermore, the irregular radiation pattern of the antennas also has a considerable impact on the model accuracy as well as the other unconsidered scenario specific characteristics. The severity of the former depends on the network hardware and it was refrained from integrating such feature into the model to maintain its general applicability and simplicity. In Section 8.5.1 the effects of radio antenna and transceiver on the predictions of MOVE are further investigated. Other irregularities related to certain scenarios, such as the distortions of body shadowing in the measurements of WIFI OFFICE are required to be compensated or tolerated at the cost of accuracy.

The evaluation has shown that the HYBRID model is able to fairly accurately predict link qualities when considering nodes at unknown positions. The weighted combination of PHY and STAT models allows to achieve an estimation accuracy close to the most accurate of both at a low computational cost. Provided measurements offer the possibility of compensating local inaccuracies of PHY resulting from coarse environment descriptions, enabling two optional approaches for improving the prediction accuracy of the HYBRID model. This flexibility increases the number of potential application scenarios independent of which information are available. Besides improving the overall MQE, exploiting these synergies of combining environment description and observations, especially the adjustment of node positions, also underlines the importance of a precise positioning of devices and obstacles for the physical models. Minimal adjustments of device positions in the environment model already improve the predictions noticeably and furthermore demonstrate the actual capabilities of PHY.

With four different types of models, MOVE provides suitable solutions for a wide range of scenarios. Available information, either in the form of measurements, taken directly from a network, a provided description of an environment or both, is exploited in an optimal way to enable best possible and accurate predictions of network communication. Utilising these models then facilitates the acquisition of further network knowledge by associating their output with additional sources of information or ultimately optimise network properties by translating into physical reconfigurations.

MOVE models communication based on the information gathered at a specific point in time. As the environment changes, the models require updated physical maps and new measurements. Nevertheless, measurements can be taken during the system operation to detect significant deviations from the model estimations and used to restore the accuracy at least in the surrounding of the observed locations. With only up to few milliseconds per link, all models of MOVE show acceptable computation times allowing to estimate links and re-calibrate the models online.

8.5 Heterogeneous Device Setup

In this section, the impact of different hardware on the network behaviour is investigated with regard to the predictions of MOVE. The other conducted experiments in this evaluation using WSN Testbed are based on TelosB devices [p120] exchanging data wirelessly on the 2.4 GHz frequency using an inverted F antenna printed on the board. To evaluate the applicability of the models using different configurations, comprising hardware as well as transmission frequency, the TelosB devices of WSN Testbed have been replaced with OpenMoteB devices [m12] to perform the experiments presented in this section.

In order to ensure best possible comparability between configurations, number and positions of devices remained unchanged. Unfortunately, the position of a few shelves and lockers has changed over time generating minor differences between both environments which have been compensated by adapting the description accordingly. Similarly to Section 5.3.2, $C(7, 2, 1)$ is used as PHY configuration for the evaluations reported in this section. Because of a defective radio chip, experiments with the OpenMoteB devices have been performed with 39 devices in total. This device has been retrospectively removed in TelosB measurements as well. As the number of links on the same power level remarkably differs depending on the radio, a direct comparison is difficult. Thus, when comparing two measurements from different radios, packets received due to a more sensitive radio were filtered so that both measurements only contain those in the same range of RSSI values. The following sections compare and analyse the influences of two main factors, antenna and frequency on the model prediction accuracy.

The proposed STAT model reproduces and interpolates predictions based on the RSSI measurements, which are used as input. Beyond that, different communication properties of utilised devices, ultimately providing RSSI values, do not necessitate any further adaptations since STAT abstracts from such details and does not require any information related to physical signal propagation. Thus, the investigation of changes in communication properties are only of interest with respect to those models of MOVE, including such information. Accordingly, the following evaluations confine to investigate the effect of changing communication properties on the PHY model.

8.5.1 Radiation Pattern

Having worked with TelosB nodes over several years, certain hardware-specific properties have been identified and investigated. As already indicated in Section 8.7.3, especially the irregular radiation pattern of the printed F antennas [p74] has a significant impact on the prediction quality [p143]. Experience has shown that not modelling this property for the sake of general applicability and simplicity introduced prediction errors affecting the obtained results to the disadvantage of the used approaches. Giving two experienced examples demonstrating the impact of using printed F antennas, the devices 2 and 3 respectively 14 and 15 (cf. Figure 8.2) were unable to communicate directly on lower power levels despite line of sight and a distance of just a few metres. As their antennas lay exactly on the same plane, blind spots prevented communication, whereas models predicted an expected reception. As a consequence, an elimination of these exacerbating factors would expectedly reduce the prediction errors of the underlying

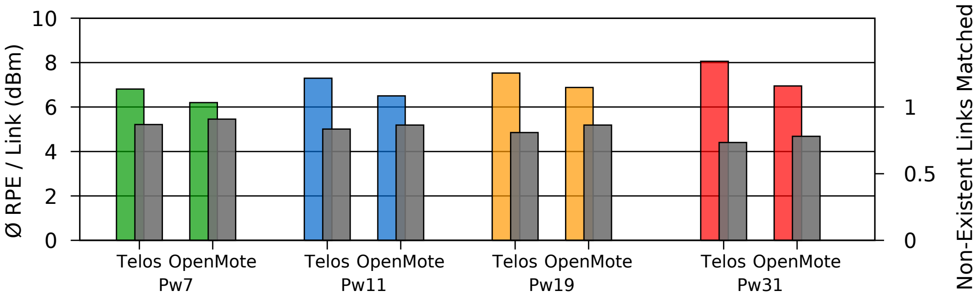
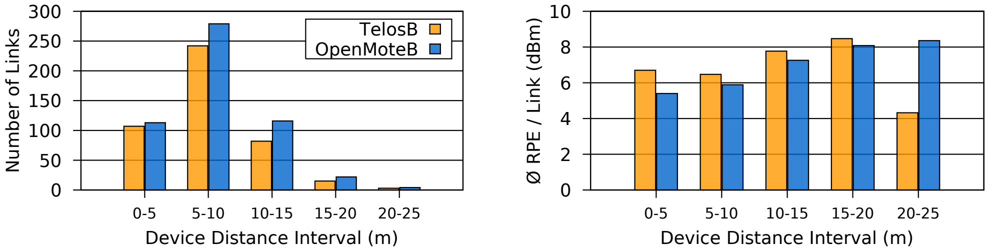


Figure 8.20: *Comparison of* PHY *average RSSI prediction errors for 2.4 GHz* WSN Testbed *measurements on different power levels using both TelosB devices with printed F antennas and OpenMoteB devices with external dipole antennas.*

models and accordingly improve the quality of results. The external dipole antennas of the OpenMoteB devices emit a signal in a more uniform way closing the reception holes of the TelosB and thus leading to a more predictable signal behaviour closer to the model output. Due to similar transceiver characteristics, measurements of OpenMoteB devices on the 2.4 GHz frequency reported in this section have been recorded using the CC2538 radio. In addition, links of the OpenMoteB CC2538 radio evincing RSSI values below the reception capabilities of the TelosB CC2420 radio have been filtered out, ensuring a fair comparison between both device types having the same network size and reception preconditions.

Figure 8.20 compares PHY model predictions against both TelosB and OpenMoteB measurements on four different power levels confirming the change of the latter measurements closer towards the PHY predictions. At all power levels, taking the OpenMoteB measurements as a basis, the prediction error is constantly lower than for the TelosB measurements. Differences in average errors vary between 0.8 dBm and 1.7 dBm, increasing for higher power levels, demonstrating more reliable links with less bias at larger distances when using external antennas. Likewise, non-existent links are matched proportionately more often. As the model predictions for both devices are almost identical except for a few links due to minor differences in the environment description, this match improvement is mostly related to a greater number of existing links, OpenMoteB devices are able to establish. In absolute numbers, there are even less matches for OpenMoteB, but also fewer non-existent links at all.

The influence of aforementioned larger reception distances on the prediction error is investigated in the following. Thereto, recorded existing links are assigned to sets, containing links from devices with their physical distance lying within a certain range. The number of links for each set is shown in Figure 8.21 (a). Especially for lower power levels, the difference in the number of existent links at medium distance become obvious. Within this range,



(a) Measured links **(b)** Average RPE

Figure 8.21: *Comparison of measured links and* PHY *average RSSI prediction errors between TelosB and OpenMoteB devices based on device distances using* WSN Testbed *on power level 7.*

Table 8.14: *Comparison of existent link statistics between TelosB and OpenMoteB devices, recorded on power level 7.*

Links / *Device*	Total	∅ RPE/ Link	Common	∅ RPE/ Link	Exclusive	∅ RPE/ Link
TelosB	449	6.81 dBm	364	6.32 dBm	85	8.92 dBm
OpenMoteB	534	6.20 dBm		5.70 dBm	170	7.26 dBm

OpenMoteB devices are able to establish additional 80 links compared to the TelosB devices. For each of these sets, the prediction error is then computed separately, reported in Figure 8.21 (b). Investigating the sets of links at larger distances, the results likewise show an increasing prediction error, except for the largest distance interval, which however only contains three (TelosB) or four (OpenMoteB) links, respectively. Throughout all distance intervals, the measured link qualities of OpenMoteB devices better match PHY predictions than the ones measured with TelosB devices.

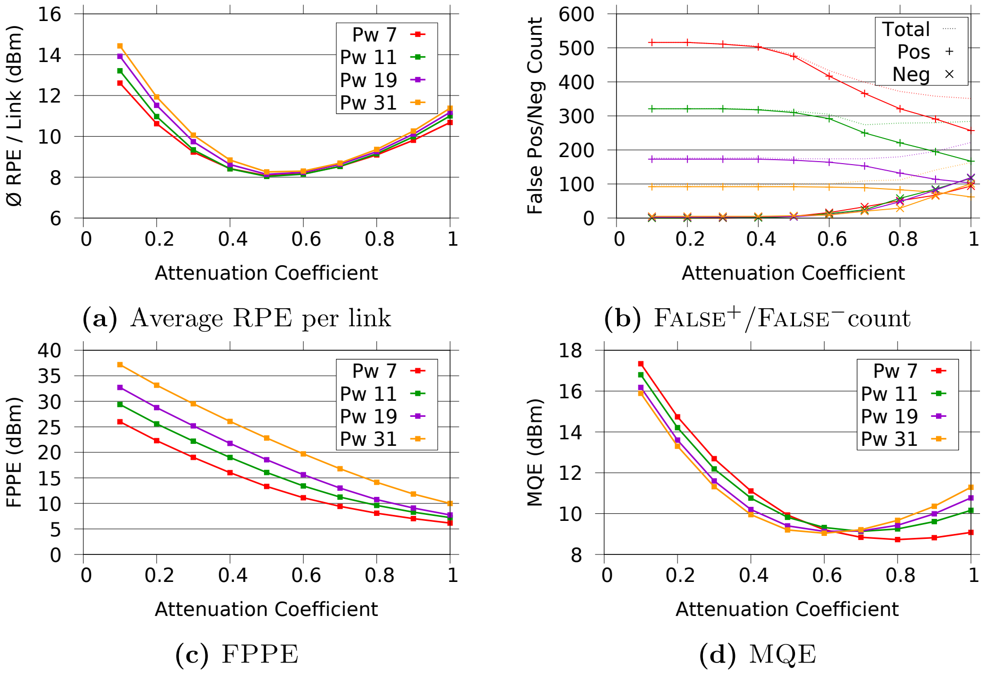
Table 8.14 compares the existing links from both measurements and divides them into a common set of links, which existed in both cases and a set of links exclusive to one of the two measurements. Only more than half (59 %) of the total observations on existing links are shared by both measurements, highlighting the significant variation in receiving signals from surrounding devices, caused by the influence of radiation patterns on the observed links. Common links show better predictability by the model than the disjoint ones which contribute a remarkably greater error to the overall accuracy. As already stated, links at larger distances become less foreseeable, especially with unconsidered antenna properties as reflected in an even greater error for TelosB devices.

8.5.2 Frequency

To satisfy the demand for simultaneous communication of wireless devices, utilising multiple frequencies is one option among others to achieve this goal. Additionally, exploiting their distinct properties in interaction with obstacles, using multiple frequencies can be beneficial for bypassing areas exhibiting strong attenuations to a certain frequency band only. In this section, the capabilities of the PHY ray tracing model to predict signal quality in the Sub-GHz frequency band are evaluated and necessary adaptations shown to achieve acceptable errors. As the AT86RF215 radio has significantly different characteristics than the CC radio family, values reported in this section are based on Sub-GHz (868 MHz) and 2.4 GHz measurements taken from the former radio only to maximise comparability.

As a consequence of using different frequency bands, the model parameters have to be adapted accordingly. Especially the signal attenuation of obstacles is an important factor for precise estimations. In general, lower frequencies penetrate obstacles with less signal loss than higher ones [p144] and therefore also achieve larger transmission distances in obstructed environments at the cost of lower data transmission rates.

Staying in line with the principle of applying plain and general approaches, it is investigated how a common adaptation of the existing material attenuation values for 2.4 GHz to the Sub-GHz frequency affect the prediction accuracy of PHY to analyse the transferability of the modelling approach to other frequencies. Instead of manually adjusting each attenuation factor individually for each material, which would be required when modelling different frequencies, existing values for 2.4 GHz are multiplied by a correction factor depending on the frequency. Moreover, this enables an automated adaptation when measurements on the target frequency are available. As a drawback, implicit errors arising from imprecise attenuation values have to be tolerated. Figure 8.22 shows several metrics of PHY when experimenting with different attenuation coefficients on various power levels using 868 MHz measurements. According to the average RPE of Figure 8.22 (a), an attenuation coefficient of 0.6 leads to best predictions, slightly differing between the four shown power levels. Similar to measurements using the 2.4 GHz frequency, higher power levels show greater prediction errors due to the growing influence of an increasing amount of large distance links which are more difficult to predict accurately. Since the error bases on existent links only, false positive predictions and their possible error do not affect this value. Taking false model predictions into account, as reported in Figure 8.22 (b), albeit decreasing compared to lower attenuation coefficients, the still large number of false positives at a coefficient of 0.6 becomes obvious, especially for the lower power levels. Emphasised by a very low



(a) Average RPE per link

(b) False$^+$/False$^-$count

(c) FPPE

(d) MQE

Figure 8.22: *Quality metrics for Sub-GHz* WSN Testbed *predictions with varying attenuation coefficients on different power levels using* PHY *with* $C(7,2,1)$

amount of false negatives, the model output can be regarded as still too optimistic. This is supported by the FPPE, depicted in Figure 8.22 (c), showing a substantial error for incorrectly assumed existing links per se, naturally decreasing as the assumed material attenuation increases and the false positive count decreases.

Considering the model behaviour from an optimisation perspective, choosing too high obstacle attenuation values result in pessimistic predictions and increase the number of false negatives, i.e., missing valuable links when planning alternative device positions for optimisation. On the other side, relying on too optimistic predictions could lead to worse network performance by depending on unreliable links in reality. Thus the goal is to find an optimal balance, taking all relevant quality metrics into account. Using the MQE, shown in Figure 8.22 (d), this is achieved for all power levels when using a coefficient of 0.7, showing the lowest count of false predictions while still being close to the smallest prediction error among the existing links.

Having chosen the most suitable attenuation coefficient, now the differences between Sub-GHz and 2.4 GHz PHY predictions are analysed. Figure 8.23 compares both measurements of the AT86RF215 radio on different power levels. As the receiver sensitivity is much higher, links are established at larger distances compared to the CC radios. Accordingly, the prediction

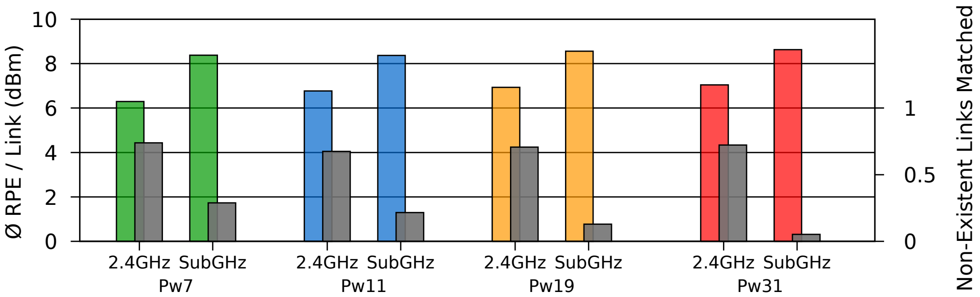
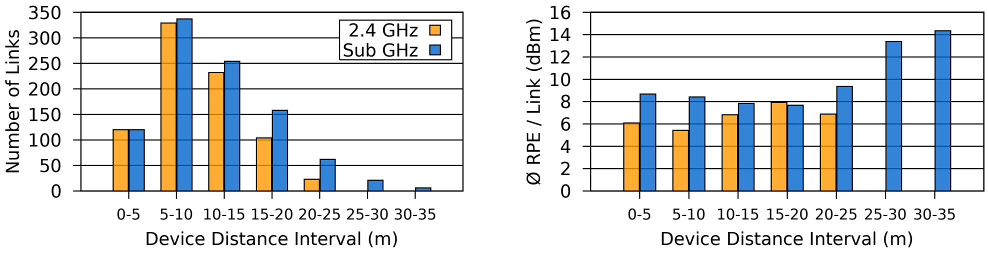


Figure 8.23: *Comparison of* PHY *average RSSI prediction errors for Sub-GHz and 2.4 GHz* WSN TESTBED *measurements on different power levels using OpenMoteB AT86RF215 radios.*

Table 8.15: *Comparison of existent link statistics of OpenMoteB devices, using the AT86RF215 radios sending at 2.4 GHz and Sub-GHz on power level 7.*

Links / *Device*	Total	∅ RPE/ Link	Common	∅ RPE/ Link	Exclusive	∅ RPE/ Link
2.4 GHz	808	6.29 dBm	723	6.22 dBm	85	6.89 dBm
Sub-GHz	958	8.38 dBm		8.08 dBm	235	9.33 dBm

error for 2.4 GHz on lower power levels is already greater than for the other radios. On the highest evaluated power level 31, 85 % of all possible links of WSN TESTBED actually exist when sending at 2.4 GHz and even more, 94 %, when using Sub-GHz communication, almost covering the complete area. Therefore the reported RPE value is an appropriate overall indicator of PHY accuracy for the entire scenario. Comparing both frequencies, there are two obvious possible reasons explaining the increased error on the Sub-GHz band. On the one hand there is a further increase in range through the lower frequency and on the other hand aforementioned errors resulting from not using individual obstacle attenuation values. Figure 8.24 (a) depicts the large increase of links at medium distance as well as the availability of a few links at large distance on the Sub-GHz frequency even on low power levels. These large distance links also evince the greatest error, as shown in Figure 8.24 (b), but only contribute to the overall error in a negligible way due to their small number. However, if restricting the RPE computations exclusively to links established at both frequencies, shown in Table 8.15, the resulting prediction accuracy of PHY decreases by almost 2 dbm (30 %). The greater average error can be observed throughout all links, indicating the undifferentiated adaptation of attenuations as the most probable reason.



(a) Measured links **(b)** Average RPE/ Link

Figure 8.24: *Comparison of measured links and* PHY *average RSSI prediction errors for Sub-GHz and 2.4 GHz* WSN Testbed *measurements based on device distances using OpenMoteB AT86RF215 radios on power level 7.*

The evaluation in this section has shown the capability of PHY modelling different frequencies by utilising existing knowledge from the 2.4 GHz band. Even though certain materials show a completely different attenuation behaviour on lower signal frequencies, a general adaptation of all coefficients is possible, although entailing remarkable losses in the prediction accuracy.

8.5.3 Discussion

This section has investigated the effect of using different devices and configurations on the prediction accuracy of MOVE. By conducting all experiments at the same device locations of WSN Testbed, replacing the devices on the spot, other disruptive factors related to site-specific effects could be minimised, enabling to focus on the comparison of device properties and configurations. Hence measured values of both device setups are compared to predictions of PHY utilising almost identical scenario descriptions and thus sharing the same underlying inaccuracies in both cases.

The investigation has highlighted a significant influence of an irregular radiation pattern on the measurements. Eliminating this distortion by using isotropic external antennas, the observations adhere closer to the model predictions. As the radiation pattern of each antenna type is unique and complex to model, a consideration would inevitably complicate the environment description as well. In addition, the exact antenna direction of each device would be required to achieve best possible prediction accuracy, leaving such sophisticated realisation infeasible. Yet, to stay in line with the goal of simplicity, assuming an isotropic signal propagation is the only reasonable option, even though the introduced pattern-related error has to be accepted in that case.

Furthermore, applying PHY to model communication on different frequencies is possible without providing additional details and utilising existing knowledge instead. However, this comes at the cost of reduced accuracy as the applied knowledge does not necessarily fit the scenario without restrictions. On the other side, it also paves the way for realising an automated approach with minimal user intervention to obtain first estimates of any scenario in a practical time leaving the contribution of additional information as option to refine predictions when required.

8.6 Understanding Changes in the Environment

In this section, it is experimented with Follow in an indoor scenario. The performance of each modelling approach is analysed and the ability in matching wireless changes to their cause.

8.6.1 Methodology

The evaluation of Follow bases on the experience in the group's indoor WSN Testbed made of 40 low-power 2.4 GHz wireless TelosB devices [p120]. As already stated, this platform has well-known radio propagation irregularities that have a significant impact on system behaviour. Given that the models discard the description of such irregular patterns in order to keep the approach practical, this choice poses interesting challenges to this work. The scenario and the location of the devices are shown in Figure 8.2.

Being the area completely under control, it was possible to experiment with different events in a systematic way without interference. In particular, the impact of opening or closing doors and windows could be analysed in a quiet environment without people. During the experiments, the devices were scheduled to perform RSSI measurements according to a transmission scheme avoiding collisions along different transmission powers. Focusing the evaluation on investigating the capabilities of MOVE to identify the links affected by an event and perform the matching against the actual observed changes without distortions, traces were recorded separately before and after performing changes to the environment. Thus, pre and post-event state detections on the link-level are forestalled, which would otherwise become necessary in an online monitoring approach.

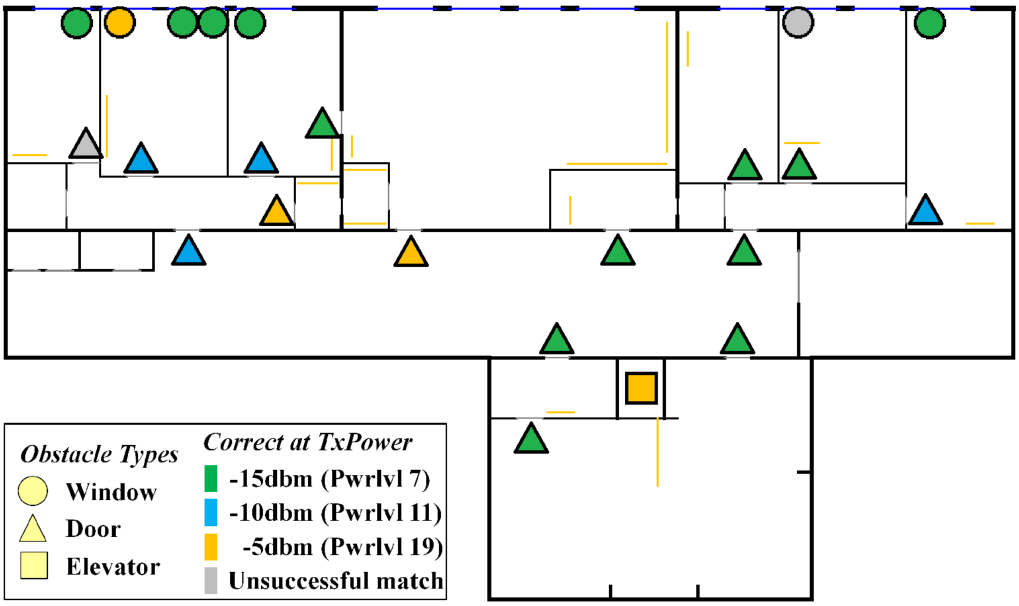


Figure 8.25: *Map of the investigated obstacles (doors, windows and an elevator) and the minimum transmission power of correct assignment to the corresponding measurements using all 40 nodes on varying power levels, as computed through the use of the* 3D GPU *model. The colours denote successful assignments of obstacles beginning at a certain transmission power level as more links become available.*

8.6.2 Matching Changing Wireless Properties

Now it is analysed to what extent the wireless communication descriptions are able to reveal information of real systems. This evaluation offers a specific usage scenario as an example of the more general applicability. It was experimented with 23 different obstacles, whose state could be changed, and different power levels were measured as done in the previous analysis of MOVE. In particular, various doors and windows were considered and their state changed by opening and closing them. Similarly, the impact caused by the elevator was investigated when present or absent at the testbed floor. A map of WSN Testbed including type, position and power level of the first successful matching is depicted in Figure 8.25. Most of the doors and windows have adjacent devices, allowing to study if they are essential for the matching. Furthermore, in the upper left area, a more fine grained set of windows was chosen to evaluate the ability of the solution to distinguish among close events sharing most links.

Table 8.16: *Number of events matched correctly against an obstacle for different models and power levels. On the weakest transmission power, 2 out of 23 events were not reflected in any link change and resulted therefore invisible. In the corresponding matching percentages, these links were excluded.*

Used Model	2D CPU	2D GPU	3D GPU
Power Level 7 (only 21 events visible)			
Correct Matches	12	12	13
Percentage	57 %	57 %	62%
Power Level 11			
Correct Matches	15	14	17
Percentage	65 %	61 %	74 %
Power Level 19			
Correct Matches	16	19	21
Percentage	70 %	83 %	91 %
Power Level 31			
Correct Matches	20	20	21
Percentage	87 %	87 %	91 %

Model Cluster Matching

After taking a base RSSI measurement in an empty environment with all doors and windows closed and the elevator absent, the state of the different obstacles was changed one after the other while recording further RSSI samples. Afterwards, the data was processed as described in Section 6.3 with the aforementioned exception of pre and post-event state detection due to a prior separation of these phases. Through the computation of the different models in the various possible object states, the sets of links that should be influenced by each object are determined. Then, the 23 complete measurement sets corresponding to the different changes are compared against the base measurement set to obtain the reference clusters of relevant links that have manifested a significant variation. These clusters serve as input for the matching algorithm to compare reality against the model predictions.

The results of the matching algorithm are listed in Table 8.16. The first evident result is that an increasing transmission power corresponds to a steady increase in the amount of correct matches. At the lowest transmission power only half of the changes find a correct match through the observations and two events are completely invisible as no variations have

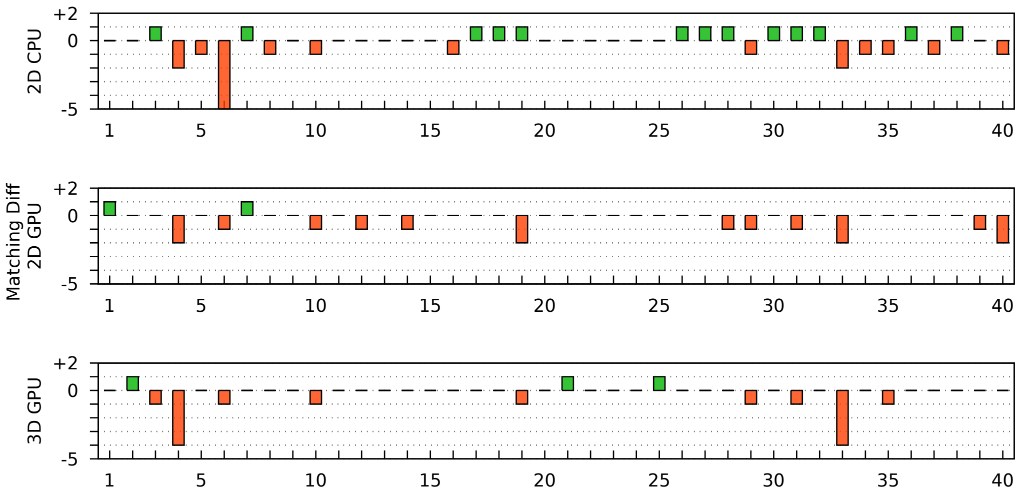


Figure 8.26: *Consequences of removing a single node on the number of correctly matched obstacles for the highest transmission power.*

been observed on any link. This problem disappears for higher transmission powers, since the number of links increases both in reality as well as in the models, effectively offering a richer set of information to exploit. With higher transmission powers, links increase their RSSI values and become stronger, better exposing changes in correspondence to environmental dynamics. In general, more connectivity is beneficial as it reduces the possible holes in the communication map caused by distant nodes in sparse regions. However, it must be taken into account that new links might disturb the matching, deviating the decision towards alternative obstacles. This can happen in particular if a ray to an obstacle should exist according to the model but it is not significantly affected in reality.

Before going into further details of the impact of specific nodes on the matching accuracy, it can be observed that events with a high Jaccard similarity coefficient are more robust to the changes in the number of available links. At the centre of the environment, however, where more links are present, events happening are harder to match correctly. The Jaccard coefficient itself also presents limitations as a link match in a set with more links counts less for the coefficient. Finally, describing the correct communication frequency has a significant impact as well. In this study, moving the model from 2400 MHz to 2481 MHz (corresponding to the actual frequency of the measurements gathered from the testbed) made the 3D GPU model correctly match the changes caused by the elevator.

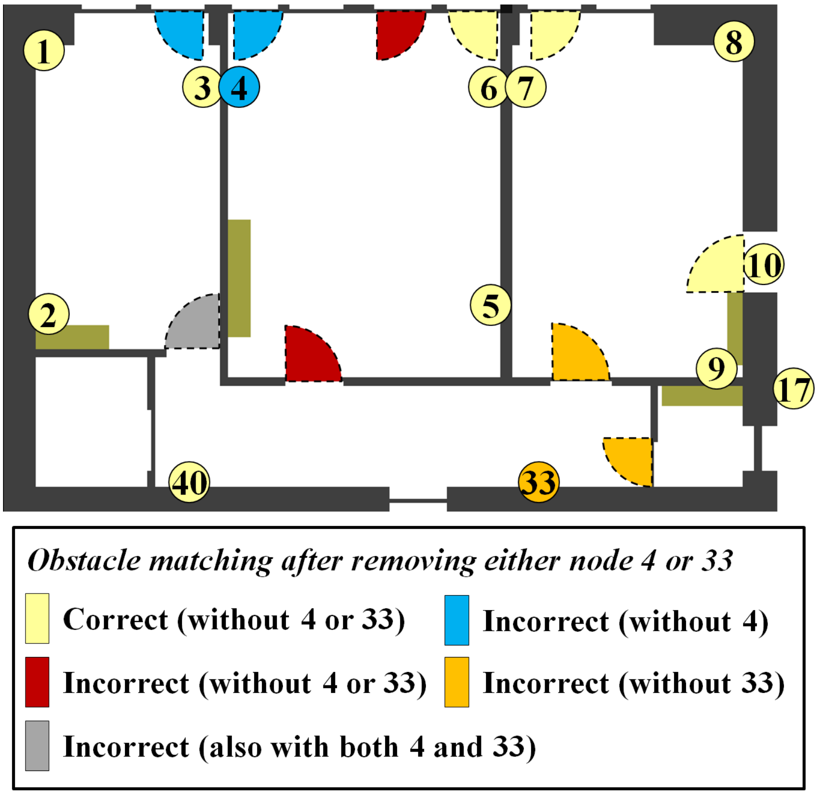

1
3
4
6
7
8
10
2
5
9
17
40
33
Obstacle matching after removing either node 4 or 33
Correct (without 4 or 33)
Incorrect (without 4)
Incorrect (without 4 or 33)
Incorrect (without 33)
Incorrect (also with both 4 and 33)

Figure 8.27: *Consequences of removing either node 4 or 33 from the measurement on the matching using the* 3D GPU *model. For visual clarity, nodes 3,4,6 and 7 have been moved downwards from their original position next to the windows.*

Influence of Individual Nodes

Now the relevance of each individual node in matching the correct events happening in the environment is analysed. In Figure 8.26, the difference in the number of correct matching is reported when one node is removed from the deployment. In general, it is possible to observe that the GPU-based modelling is more robust to the presence or absence of individual nodes. This behaviour can be justified by the fact that the CPU employs an aimed-approach where rays are targeted through obstacles towards a destination (the receiver), only through exclusive reflections or diffractions, without exploring their possible combinations. On the other side, instead, the GPU explores more possibilities by spreading rays equally in all directions and considering the combination of multiple effects together. From these results, it is also apparent, that specific nodes can contribute significantly to the matching, in particular node 4 and 33 in this case.

Figure 8.27 depicts the specific events that node 4 and 33 contribute to detect when present or absent. Node 4 is a crucial node due to the high density of observable events in its surrounding with 5 windows and 2 doors close to it. Moreover, it lies on a line with many other nodes close to the set of windows and is essential to distinguish between different windows. Without node 4, in fact, the other links, e.g., between 3 and 6, would span at least 3 windows, making it harder to differentiate between them. Node 33 presents similar conditions since its links cross multiple adjacent doors. Without such node, indeed, these doors cannot be differentiated by the remaining links that would inevitable cross both.

8.6.3 Discussion

Follow has demonstrated the ability of assigning events to their cause by employing a ray tracing model identifying relevant links based on a description of the environment and interrelate them with observations from reality. With an increasing transmission power and consequently also increasing number of links affected by the obstacles present in the environment, the matching accuracy can be successively improved up to 91 % of correct matches. Furthermore, the differences between aimed and launched ray tracing approaches have been revealed, favouring the latter due to its ability to discover more individual and relevant paths, especially at closer link distances, which are of most importance to identify the obstacles. Achieving the highest matching accuracy, however, requires the appropriate hardware to run the 3D GPU model. Nevertheless, the less computationally intensive, but also less exploitable aimed 2D CPU model, in terms of path quantity, provides comparable results as well, especially at higher transmission powers.

Investigating the complexity of the performed computations and their corresponding scalability, the approach can be divided into two phases. The first, most computationally intensive step in the presented solution involves the identification of the sets of links impacting each obstacle. At this stage, the ray tracing model is employed, whose computational complexity is affected by the details of the environment as these can increase the number of comparisons needed to evaluate the ray paths. Similarly, more nodes result in a quadratic grow of links that need to be investigated.

An appropriate filtering based on the environment properties can speed up these computation times. It is, however, worth noting that these factors affect the complexity of a first step that needs to be computed only at the very beginning. In the case of new or relocated nodes, exclusively the associated links and ray paths need to be recomputed, favouring scalability after the initial setup.

The second phase identifies correlated link changes among links as well as within rooms in comparison to the sets identified in the first stage. The corresponding computational effort increases linearly with each link, room, and comparison set, limiting its impact on scalability for bigger networks or more complex scenarios.

As it is refrained from requiring centimetre accuracy, FOLLOW is limited to reliably predict link changes, but not to quantify their actual change. For this reason, the approach is able to identify that an object has changed its status but not to distinguish between subtle changes in an obstacle state, e.g., the degree at which a door is open. This requires more accurate description of the environments and a supervised placement of nodes in an area. The same applies to the distinction among changes to objects spatially close to each other. Only the use of more detailed maps and a constrained positioning of devices would allow to address such limitations.

8.7 Network Optimisation

In this section, the attention is turned to the implementation and evaluation of MOTION. Using the HYBRID model as basis, the ability of MOTION to support network optimisation is studied. Afterwards, the limitations of this approach are discussed.

The evaluated accuracy of the HYBRID model has an impact on its support for effective network optimisation. Therefore the effect of node relocation strategies is analysed, as introduced in Section 7.3, on network metrics as well as their potential, indirect benefit on the application performance. WSN TESTBED is used as scenario for these tests, where devices could be freely relocated and reprogrammed with two different applications. At first, a monitoring application monitors the network connectivity by scheduling the transmission of beacons to avoid collisions and logging their reception. Together with the corresponding RSSI information, these traces describe the current network state and serve also as training data for the HYBRID model. Based on this information, the reconfiguration strategy is computed and the devices are relocated accordingly. In the specific instantiation of the strategy, nodes were allowed to be relocated only in the same room and at least 1 metre from another device used by the application; a decrease of quality of maximum 60 % of the absolute RSSI difference from the noise floor was allowed and differentiated between good and weak links with a −90 dBm threshold (Table 7.1). Two strategies are evaluated with the goals of either increasing the individual node connectivity or decreasing the network diameter.

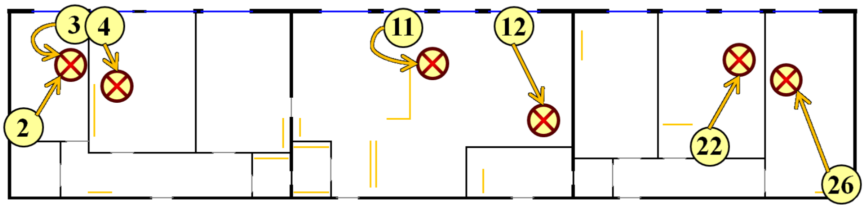


Figure 8.28: *Map of the selected nodes and relocation positions for the node connectivity optimisation analysis. For reasons of clarity, all other nodes of the deployment are not shown.*

8.7.1 Network Metric Improvements

At first, a strategy is computed that ranks the expected improvement of connectivity for each node in the network at the possible relocation positions. Based on this information, the nodes with the highest foreseen benefit were selected for relocation. The chosen positions are depicted in Figure 8.28 and the corresponding results are shown in Table 8.17. Cal indicates the connectivity state of node n in its original position, when the trace used for the calibration of the model was taken. Pred denotes the predicted behaviour of the node relocated in its new position. It is referred to the corresponding, measured real network behaviour with the physical relocation of the device as Rel.

The results show that the HYBRID model is able to provide a correct estimation of good links for the given nodes at the indicated positions and thus allows to increase their connectivity. However, further weak links are also present at the target positions. These are in general hard to predict precisely in the border region of reception due to both the inherent inaccuracy of the model and hardware factors, e.g., the anisotropic antenna of the tested devices. The combination of these factors leads to the occurrence of unforeseen links, especially for nodes located in the center of the network, as depicted for nodes 11 and 12. Accordingly, the real RSSI value is often worse than the predicted one, since the model is not always able to detect all of these weak links for each position. If most of the weak links have been considered and thus avoided when choosing a position for relocation, prediction and reality match closely, as for node 26. Nonetheless, the evaluation demonstrates that MOTION is able to improve the node connectivity, complying with the optimisation goal.

Table 8.17: *Node connectivity in reality before and after relocation and predicted by the* HYBRID *model. Since the model assumes symmetric links and isotropic antennas, the prediction is the same for both directions.*

	Links to Node n			*Links from Node n*		
	∅RSSI	Links^{+}	Links^{-}	∅RSSI	Links^{+}	Links^{-}
Cal n=2	−78.65 dBm	6	0	−78.54 dBm	6	0
Pred	−76.25 dBm	8	0			
Rel	−80.02 dBm	9	0	−80.87 dBm	8	1
Cal n=3	−78.93 dBm	5	0	−76.42 dBm	5	0
Pred	−75.94 dBm	8	0			
Rel	−79.60 dBm	8	1	−78.08 dBm	9	0
Cal n=4	−80.00 dBm	8	1	−79.52 dBm	8	1
Pred	−78.48 dBm	10	0			
Rel	−79.01 dBm	8	2	−76.89 dBm	9	0
Cal n=11	−85.66 dBm	5	5	−84.38 dBm	9	2
Pred	−80.10 dBm	12	0			
Rel	−86.63 dBm	11	6	−84.80 dBm	15	2
Cal n=12	−83.99 dBm	5	4	−81.01 dBm	5	2
Pred	−78.90 dBm	11	0			
Rel	−84.91 dBm	12	5	−83.30 dBm	13	2
Cal n=22	−78.26 dBm	8	2	−79.16 dBm	8	2
Pred	−73.65 dBm	9	0			
Rel	−80.02 dBm	9	1	−80.26 dBm	10	1
Cal n=26	−83.62 dBm	7	1	−82.16 dBm	8	0
Pred	−73.54 dBm	8	0			
Rel	−73.76 dBm	8	0	−75.41 dBm	8	1

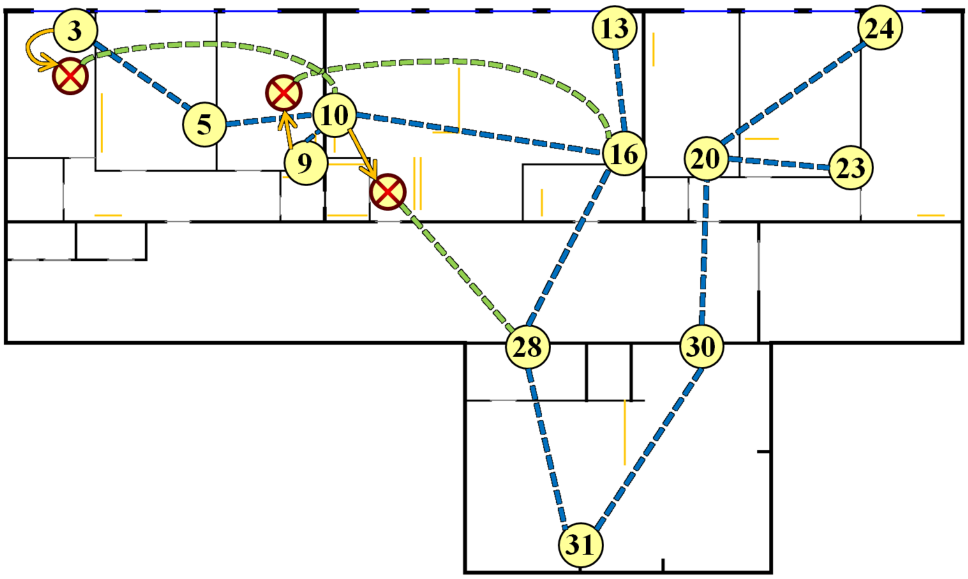
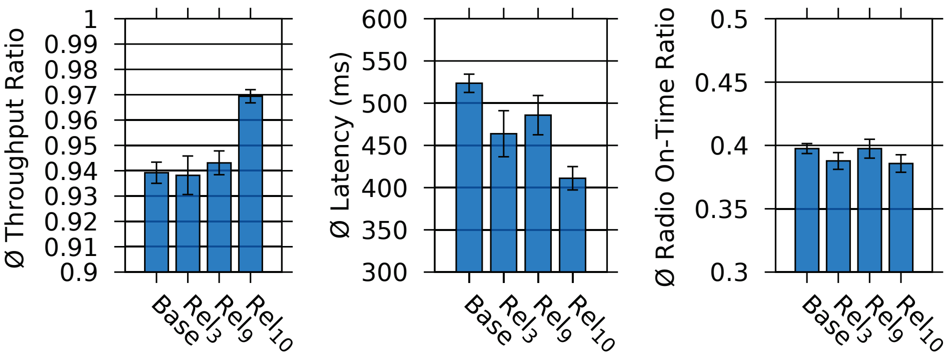


Figure 8.29: *Map of the selected nodes and positions for the optimisation of the network diameter. Blue lines denote the shortest paths from each node to the sink (node 31), green lines indicate newly created links after relocating the correspondent node to the given position.*

8.7.2 Application Metric Improvements

In the second step, it was experimented with the optimisation strategy aiming at reducing the network diameter. Such strategy can be beneficial, in particular, for traditional applications gathering data at a sink. To analyse such impact, a default data collection application based on the Collection Tree Protocol (CTP) [p53] is used. For each node, this protocol determines the path to the sink which has the lowest costs with respect to a certain metric. In [p53], the route with the lowest count of expected transmissions (ETX) to the sink is chosen. By regularly advertising ETX information regarding a node's current optimal path to other nodes in transmission range, the entire routing tree can be established and updated successively in case of topological changes.

After multiple experiments, monitoring the states of the devices and tracking the transmissions of packets, three application metrics were computed, i.e., throughput, latency and radio-on time (as an index of lifetime). This application is executed on a subset of twelve nodes with node 31 as sink, as depicted in Figure 8.29, spanning the routing tree denoted by the blue lines. The setup (named BASE) covers the whole floor with at least one node per office or lab. With REL_n, a setup is denoted in which node n was relocated. The effects of these relocations on the application metrics are shown in Figure 8.30.



(a) Throughput **(b)** Latency **(c)** Radio on-time

Figure 8.30: *CTP application metrics behaviour for different network configurations, reducing the diameter of the network by relocating single nodes.*

Starting with the two leaf nodes 3 and 9 of the CTP tree, their relocation has only minor impact on throughput and radio-on time, but shows a noticeable improvement in latency since these nodes have the longest path to the sink with 5 respectively 4 hops. A shorter path, as shown in Figure 8.29, reduces the average latency either by 11.4 % or 7.2 %. Due to the beneficial impact on several data paths, relocating node 10 shows the most significant improvement for all metrics. The throughput changes from 93.9 % to 96.9 %, the average latency of 523 ms to 411 ms (corresponding to a decrease of 21.4 %) and the radio-on time from 39.8 % to 38.6 %. These results show that MOTION is able to improve the application performance by properly manipulating the underlying network without using dedicated knowledge.

8.7.3 Discussion

The evaluation of MOTION has shown that relocating specific nodes according to the estimations given by the HYBRID model result in an actual improvement of connectivity towards the predicted number of links demonstrating the applicability of MOVE for optimising networks. However, as the irregular radiation pattern of the antennas may have a considerable impact on the model accuracy, this problem is also carried over to deployment or relocation. Nonetheless, it is possible to rotate the device in situ and monitor the resulting links to identify the orientation best matching the model estimations.

In any case, it has been experienced that also small changes in the environment (in this case, few metallic closets and the presence/absence of the elevator) can have a considerable impact on the network topology and the system behaviour. This underlines the relevance and challenge of modelling the environment appropriately in order to design and deploy a system able to match the user needs in the operational environment. When substantial changes are experienced, maintenance procedures can employ MOTION to detect deviations and reconfigure the system accordingly.

Finally, MOTION is able to support network optimisation at an application level and change the physical network topology according to specific metrics. However, the aim to perform optimisation in an application-agnostic manner limited the potential benefits. Indeed, by exploiting direct application metrics and knowledge, it becomes possible to identify relocation strategies tailored to the specific application requirements.

8.8 Summary

The evaluation of this thesis comprises four main parts, related to the contributions presented in Section 1.3. Before going into the details of each contribution, the scenarios are introduced, in which the experiments of this evaluation have been performed. Each scenario has unique characteristics, providing variety in size, structural details and hardware equipment to demonstrate the versatile applicability of the presented approaches. In the following section, the used hardware as well as the organisation of measurements are explained. Two devices used for most of the evaluations, TelosB and OpenMoteB, are compared regarding their relevant radio communication properties, such as the frequency, receiver sensitivity and the utilised antennas. Furthermore, RSSI measurements were performed at four linearly increasing transmission power levels to assess links in varying numbers and at different distances.

FLoW

Initially, FLoW and its capability of localising devices based on sparse amounts of information are evaluated. In addition to three real-world scenarios, three artificial setups with specific features are identified constituting a challenge for FLoW, e.g., large networks or unfavourable floor plans.

Connectivity Information At first, the physical localisation is performed exclusively through connectivity information. Since this approach lacks of any reference points, it results in topologies that are arbitrarily rotated, scaled and mirrored, challenging the direct comparison with a real topology. For this reason, a specific metric has been defined, determining an appropriate scaling factor, transforming the graph of FLoW to a size similar to the real topology. The relative distance error between estimated and given topology is then computed for each pair of nodes. Results show that FLoW is able to outperform the reference approach MDS by exhibiting less relative errors for the majority of the links, yielding a graph closer to the real topology.

Floor Plans The usage of the floor plan in the second step allows to translate relative positions to absolute ones by matching areas and find the most suitable graph configuration. When successfully fitting the graph into the floor plan, FLoW achieves better accuracy for node placement than MDS. The best accuracy could be achieved using a WSN Testbed measurement taken on power level 7, generating a distinctive, edgy graph shape with an average localisation error of 7.3 % of the floor plan diagonal (3.24 m). However, in most scenarios the mere floor plan is insufficient to ensure a reliable matching of the graph. Ambiguities caused by symmetries intrinsic in the map highly affect the result, leaving an incorrectly matched or flipped graph. To compensate for these issues, further information on the positions of some anchor node are indispensable.

Anchors In the last part of this section, it is evaluated how a very limited number of anchor nodes enables FLoW to resolve ambiguous situations. For this purpose both, up to 4 random and manually chosen anchors have been selected. In real scenarios, FLoW manifests errors within an average range of 6 % to 9 % of the floor plan diagonal. Once successfully matched, further anchor nodes do not provide any significant benefit. For the real scenarios, three anchors are sufficient to produce correctly matched results for all of the available data sets. Nevertheless, if chosen randomly, it is possible that an adverse set of anchors still yields an incorrect result. Manually selected anchor nodes, instead, provide good results with an error below the average of a random anchor placement. Regarding computation time, MDS scales better than FLoW especially in scenarios with a high connectivity. For the investigated real scenarios, the most accurate anchor-based matching is performed in less than two seconds in both cases. The differences become obvious when considering the largest and densest artificial scenario, Square with 200 nodes and up to 12k links. Repeated executions of the force-directed algorithm for fitting and optimisation increase computation times of FLoW (5 minutes) by factor ten compared to MDS (30 seconds).

The evaluation has shown that FLoW can localise nodes of a network within a given floor plan using very limited information with a relative error below 9 % for the investigated real scenarios. The main challenge resides in translating the graph produced by the force-directed algorithm to a correct positioning within the floor plan, especially in the case of symmetric shapes. Depending on the scenario, four reasonably placed anchor nodes are sufficient to eliminate ambiguities.

MOVE

In the following section, MOVE is evaluated, validating the four contained models in the first step and then assessing the capabilities of modelling heterogeneous device configurations. In this context, several metrics are introduced to comprehensively measure the quality of the prediction models regarding different aspects. The Real Prediction Error (RPE) is the most meaningful metric since both RSSI values, from measurement and model are present in this case, allowing a precise assessment of the prediction error. To also consider other constellations in which at least one RSSI value is not available, a comprising Model Quality Estimation (MQE) is defined, including also false positives and negatives as well as correctly predicted, non-existent links. Although not as expressive as the particular metrics alone, the MQE provide a meaningful overall assessment of a model quality.

PHY From the four models of MOVE, the basic two-dimensional physical model PHY is analysed first. It uses ray tracing to determine signal propagation paths through a given description of the scenario environment. The prediction accuracy in dependency of the considered propagation effects, i.e., transmission, reflection and diffraction, is investigated at first as these are continuously used in later evaluation steps. The best compromise between computation complexity and accuracy for the available scenarios is given by considering seven transmissions, one reflection and one diffraction ($C(7, 1, 1)$). This configuration allows successful computations for each and every signal path reaching their destination within the range of considered effects in all discussed scenarios. Moreover, each link is computed in a few milliseconds, depending on the detail and size of the given scenario description, allowing to perform signal predictions of entire networks with thousands of links within seconds.

Turning to the prediction accuracy of PHY, further analysis of different scenarios and power levels revealed the strengths and weaknesses of the given approach. As expected, the description of a scenario has significant influence on the accuracy. For the main scenario WSN Testbed, providing the most detailed description of all scenarios, PHY shows an average RPE between 7.0 dBm and 8.6 dBm and a share of falsely predicted links between 15.3 % and 18.8 %. Both, prediction error and false rate increase with the power level. At higher power levels, more links at larger distances become available which are harder to accurately predict due to accumulated errors from unconsidered paths caused by the given computational limitations. An individual consideration of links at specific device distance intervals reveals that closer links within a range of up to 15 metres are predicted with a nearly constant error before the error gradually increases thereafter. Other scenarios using the same devices as WSN Testbed, but having a coarser environment description evince a greater prediction error. For the largest scenario with the sparsest node distribution, FlockLab, PHY achieves an average RPE of up to 10 dBm which is still close to WSN Testbed, considering that the description of the environment has been derived from a coarse floor map without any knowledge of the actual materials and that the distances between the devices are larger.

Disregarded effects on the other hand can considerably affect the prediction accuracy. E.g., measurements taken in the comparably small, but well described WiFi Office scenario include an implicit body shadowing effect and in WiFi Hallway, more reflections are required to model signal propagation in the distinctive long corridors correctly. If unconsidered, the observed measurements consequentially deviate from the predictions of PHY, demonstrating its limitation, attributable to incomplete information. However, as such mentioned effects might not be obvious when employing PHY, MOVE provides a solution to circumvent this deficiency by integrating taken measurements into the prediction computation as well using the subsequently evaluated STAT and HYBRID models. Nevertheless, at the presence of undistorted reference measurements, PHY provides accurate predictions for entire networks within a reasonable amount of time, exploitable for performing further optimisation steps.

3D GPU Making use of the independence of each signal path computation, the two-dimensional approach of PHY is transferred to a three-dimensional modelling approach, 3D GPU, exploiting the massive parallelisation capabilities of modern graphic processing units. As an advantage of having many computation cores available, the previously aimed signal path computation strategy of PHY can be altered to a more explorative strategy, launching rays into all directions from the sender. Albeit computing many irrelevant paths not reaching any receiver position, this strategy allows to discover much more possible signal paths without experiencing increased computation times. To distinguish between the effects of adding a third dimension and the path computation strategy on the prediction error, an additional intermediate 2D GPU model has been created, performing the launching strategy on a two-dimensional plane. Evaluations show that both amendments reduce the average RPE by either 8.2 % to 11.3 % (2D GPU) or 11.5 % to 15.6 % (3D GPU) compared to the base PHY model in the WSN Testbed scenario. Regarding prediction tendencies, the GPU models performing the launched approach provide more balanced outputs than PHY, which has a tendency to underestimate link qualities. The latter is beneficial when relocating devices as predictions are more conservative reducing the probability of false positives. On the other hand when investigating relevant signal paths, as done in Follow, some of these might be missed. Having the computational capabilities of a GPU available, 2D GPU and 3D GPU provide the opportunity to further improve the prediction accuracy of PHY.

STAT/ HYBRID The last models of MOVE, STAT and HYBRID are jointly investigated. While STAT is entirely dependent on quality and quantity of provided measurements to produce accurate predictions, HYBRID combines both approaches, STAT and PHY to benefit from their respective advantages. The availability of verification measurements also enable further optimisations, in addition to the ones of PHY. While the latter optimisations, especially the usage of material-specific obstacle attenuation, are necessary to provide a more balanced model with fewer false predictions, the HYBRID additions then result in further improvements of all relevant metrics. Applied to three heterogeneous scenarios with varying qualities in environment description and measurements, HYBRID demonstrates its strengths. The weighted combination of PHY and STAT models allows to achieve an estimation accuracy close to the most accurate of both at a low computational cost. This flexibility increases the number of potential application scenarios independent of which information are available.

With four different types of models, MOVE provides suitable solutions for a wide range of scenarios. Available information, either in the form of measurements, taken directly from a network, a description of an environment or both, is exploited in an optimal way to enable best possible and accurate predictions of network communication. Utilising these models then facilitates the acquisition of further network knowledge by associating their output with additional sources of information or ultimately optimise network properties by translating into physical reconfigurations.

In the second part of MOVE evaluation, the goal is to analyse its applicability to model different device configurations only utilising the existing knowledge used in the first validation part without adding further information. For this purpose, the TelosB devices in WSN Testbed have been replaced by OpenMoteB devices on the spot to change the communication behaviour without altering the environmental conditions.

Radiation Pattern First subject of investigation is the effect of the radiation pattern caused by the usage of different antennas. While TelosB devices use a printed F antenna, exhibiting a rather irregular radiation pattern, OpenMoteB devices use external dipole antennas with an almost isotropic radiation pattern. Since the physical models of MOVE refrain from considering antenna-specific radiation patterns and assume isotropic radiation to avoid requiring too many details and keep the approach practical, a usage of devices with dipole antennas is expected to reduce prediction errors. This expectation is also confirmed by the evaluation. For PHY, the total prediction error of existing links decreases by 8.8 % to 13.7 % as well as the rate of false positive links by around 4 % compared to the TelosB measurements. By eliminating further distorting factors implicitly reflected in the RSSI measurements, the actual prediction quality is revealed, entirely based on the description of the environment.

Frequency Second, the capabilities of predicting communication at a different frequency are investigated. Focusing on the aspect of transferability, PHY is applied to predict Sub-GHz measurements without providing additional attenuation values for the materials specific to these frequencies. Staying in line with the principle of applying plain and practical approaches, the existing knowledge on materials at 2.4 GHz is adapted instead by multiplying a general correction coefficient to these attenuations. In compliance with the property of lower frequencies to penetrate most materials with fewer losses, a correction factor of 0.7 yields the lowest results on the MQE. However, this general adaptation comes at the cost of reduced accuracy as the applied knowledge does not fit the scenario without restrictions. The average RPE increases by 22 % to 30 % (+2 dBm) to compared to the 2.4 GHz measurements, still providing acceptable predictions.

The evaluations have demonstrated the ability of MOVE to model and adapt to scenarios with heterogeneous hardware and configurations. Consequentially, due to the pursued approaches of simplicity and flexibility, each undifferentiated adaptation step or deliberately disregarded characteristic, e.g., radiation pattern or body shadowing, comes at the cost of losses in the model accuracy. Nonetheless, this paves the way for realising an automated approach with minimal user intervention to obtain first estimates of any scenario in a practical time leaving the contribution of additional information as option to refine predictions when required.

Follow

In the next section, it is experimented with Follow in an indoor scenario. It is analysed to what extent the wireless communication descriptions are able to reveal information of real systems. The state of 23 different obstacles located in WSN Testbed was separately changed by opening and closing doors and windows as well as calling and sending away the elevator while recording RSSI samples. Compared to a base RSSI measurement in an empty environment with all doors and windows closed and the elevator absent, the reference clusters of relevant links that have manifested a significant variation were determined. These clusters serve as input for the matching algorithm to compare reality against the precomputed predictions of the physical models of MOVE.

Model Cluster Matching At the lowest transmission power only half of the changes find a correct match through the observations, whereas at the highest transmission power, up to 21 changes (91 %) could be matched correctly. Between the models, there exist minor differences, with the most detailed 3D GPU performing best and closely followed by the other two, 2D GPU and PHY with matching 20 changes (87 %) successfully. The first evident result is that an increasing transmission power corresponds to a steady increase in the amount of correct matches. With higher transmission powers, links increase both, in numbers and RSSI values, better exposing changes in correspondence to environmental dynamics, effectively offering a richer set of information to exploit. However, it must be taken into account that new links might disturb the matching, deviating the decision towards alternative obstacles.

Influence of Individual Nodes A more detailed investigation on the relevance of each individual node in matching the correct events reveals that certain devices located in areas with a high density of observable events are crucial. Placed at positions adjacent to these events they enable a clear distinction, implying the necessity of a sufficiently high node coverage to improve the matching accuracy in such areas. As it is refrained from requiring centimetre accuracy, Follow is limited to reliably predict link changes, but not to quantify their actual change. For this reason, the approach is able to identify that an object has changed its status but not to distinguish between subtle changes in an obstacle state, e.g., the degree at which a door is open. This requires more accurate description of the environments and a supervised placement of nodes.

Summarising, Follow has demonstrated the ability of assigning events to their cause by employing a ray tracing model identifying relevant links based on a description of the environment and interrelate them with observations from reality.

MOTION

Finally, in the last section of this chapter, it is evaluated, how MOTION is able to support network optimisation by utilising the proposed HYBRID model. Based on monitored network connectivity and the corresponding RSSI information of WSN Testbed, two reconfiguration strategies were computed and the devices relocated accordingly.

Network Metric Improvements The first strategy aims at improving the individual node connectivity, focusing on the network level. By ranking the expected improvement of connectivity for each node in the network at possible relocation positions, nodes with the highest foreseen benefit were selected for relocation. Comparing the links from and to the selected devices before and after relocation, an increase of good quality links is observed in many cases. The results show that the HYBRID model is able to provide a correct estimation of good links for the given nodes at the indicated positions and thus allows to increase their connectivity. However, further weak links are also present at the target positions. These are in general hard to predict precisely in the border region of reception due to both the inherent inaccuracy of the model and hardware factors, e.g., the anisotropic antenna of the tested devices. Nonetheless, the evaluation demonstrates that MOTION is able to improve the node connectivity, complying with the optimisation goal.

Application Metric Improvements For the second strategy, the application layer is targeted for improvements by reducing the diameter of the underlying network. Using the Collection Tree Protocol (CTP) application as reference, which specifies data transfer paths to a sink based on expected costs of each transmission, a reduction of the network diameter is expected to influence these cost calculations and therefore the application metrics. For the evaluation, three metrics were computed, i.e., throughput, latency and radio-on time (as an index of lifetime). By considering a network consisting of a reduced number of devices and therefore less links, on which CTP then establishes a collection tree, the effect on the relocation of single nodes can be investigated more clearly. After identifying relocation positions for three nodes using HYBRID with a predicted decrease of the network diameter while maintaining all previously existing links, the actual relocation then shows the expected effect. Depending on the position of a device in the tree hierarchy and its number of children, the benefits of the relocations differ. In all cases, the average latency of the collection process could significantly be reduced by up to 21.4 % and the throughput noticeably improved from 93.9 % to 96.9 % when relocating the node with the most significant effect.

The evaluation of MOTION has shown that relocating specific nodes according to the estimations given by the HYBRID model results in an actual improvement of connectivity towards the predicted number of links demonstrating the applicability of MOVE for optimising networks. It further shows the ability of MOTION to improve the application performance by properly manipulating the underlying network without using dedicated knowledge.

9 Conclusions and Outlook

9.1 Conclusions

Understanding the environment in which a network is situated as well as its characteristics are crucial for ensuring a proper operation and taking effective measures to ultimately optimise the network behaviour towards specific performance goals. However, each scenario is unique and provides a differing amount of available knowledge about the network with regard to both quality and quantity, requiring adaptable solutions able to deal with such variation and exploit it in the best possible way. This thesis provides approaches for deriving valuable knowledge about the properties of wireless networks on different levels of available information. The process of knowledge gain is separated into four basic stages building on each other requiring an increasing amount of information about the system: (1) identifying basic network properties, (2) modelling wireless communication, (3) understanding reasons for changing network behaviours, (4) improving the network performance. Each of these steps is addressed by a contribution proposing approaches of exploiting the varying amounts of available information of each scenario in a flexible way while maintaining the general applicability.

FLoW Addressing the first stage of identifying relevant basic information of a network, FLoW provides an approach to identify a given network topology and localise its devices using only minimal information. By calibrating the underlying force-directed algorithm with connectivity data, taken from an operating network, FLoW is able to practically infer the relative position of the wireless devices. With the addition of a floor plan outline and up to four anchor node positions, graph placement ambiguities for the evaluated indoor testbeds could be resolved and devices localised with an accuracy of a few metres. Additional more challenging simulated scenarios reveal limitations of the approach, such as more complex building structures and highly connected dense networks, complicating a proper graph expansion and placement. However, such limitations can be addressed by further increasing the number of anchors or dividing the localisation process into multiple segments, thus reducing their complexity. In conclusion, this contribution shows that it is possible to position devices correctly in common real scenarios with only limited user information and intervention, having the network identify the displacement of its own components and supporting further reasoning about system properties.

MOVE With MOVE, a fast and accurate modelling of wireless communication has been realised providing suitable solutions for a wide range of scenarios. Including models of both empiric and deterministic types, requiring orthogonal inputs, MOVE is able to deal with different types of available information and combines the benefits of both types of models. Measurements, taken directly from a network and a description of the environment, individually or jointly available, are exploited in an optimal way, creating synergies to enable best possible and accurate predictions of network communication. Focusing on the more fruitful deterministic physical modelling for the process of knowledge gain due to their ability of taking the environment into account, MOVE includes two ray tracing models, operating in two and three dimensions, respectively. Handling the complexity of physical signal propagation, several optimisations, runtime-adaptable computational parameters as well as the chosen granularity of the environment description enable an apposite adjustment of MOVE to the given scenario with the desired prediction accuracy. With only up to few milliseconds per link, all models show acceptable computation times allowing to estimate the connectivity of medium sized networks within seconds and thus quickly adapt to changes in the network or the environment. The prediction accuracy of MOVE is further proven by the other contributions, utilising the models for the acquisition of further network knowledge by associating their output with additional sources of information. Overall, this contribution realises a flexible and precise modelling of wireless communication, supporting a virtual exploration of signal paths and coverage estimation, sparing efforts of manual measurements and serving as the basis of gaining further insights about the network.

Follow Given sufficient knowledge about the environment of the investigated scenario to model wireless communication, the third stage is addressed by Follow. The implemented ray tracing models of MOVE are utilised to understand how events happening in the environment and causing stable changes to the wireless communication impact signal propagation. In particular, it is possible to exploit the features of the employed models to localise such events and identify the causing obstacles, gaining further insights about the network behaviour. As demonstrated in an indoor testbed, with an adequately dense network and devices placed at strategical positions, FOLLOW is able to even distinguish adjacent obstacles, e.g., windows, and match them to observed changes in the network connectivity. As the approach rather focuses on the existence of signal paths above a specific strength than exact predictions of their quality, certain computations of physical effects can be omitted, reducing the requirements regarding provided information. Thus, FOLLOW can be applied to a large number of scenarios where only coarser descriptions of the environment are available, giving networks the ability to reason about their own communication properties.

MOTION The fourth and last contribution, MOTION, concludes this thesis and constitutes the final step following the process of knowledge acquisition, exploiting it to ultimately improve the network performance. MOTION provides a technique to optimise wireless networks by relocating devices in the environment. Based on the predictions of MOVE, the impact of arbitrary displacements of devices in the target environment can be investigated to devise relocation strategies, aiming at optimising the overall resulting network performance regarding specified goals and constraints. These then recommend the most promising device relocation positions from which one can be chosen to change the physical network topology accordingly. Furthermore, by exploiting application knowledge and identifying relevant key metrics, MOTION is able to support network optimisation at an application level, adapting the network structure to the requirements of the target application. In summary, MOTION exploits the available network knowledge to provide an automatised approach for network monitoring, analysis and optimisation, comprising the expertise to also assist inexperienced users in a guided incremental relocation process.

This thesis has demonstrated the viability of automated general solutions for network analysis and optimisation at different stages of knowledge, managing to realise universal, network-independent applicability on the one side and adaptability to specific systems and environments on the other side. The presented approaches are able to flexibly operate with variable amounts of information, ranging from the necessary minimum up to a large amount of further relevant details, enhancing the quality of results. Allowing for optional inputs beyond the minimal ones addresses the challenge of heterogeneity between and within networks, which may provide a large variance of available information, and considers the network uniqueness as well. The applicability is preserved due to demanding only indispensable information as a requirement, while nonetheless being able to adapt the approaches to characteristics of the given network. Furthermore, as the provision of more detailed information then becomes optional, costs and efforts for collecting can be saved if the already obtainable coarser results are considered as sufficient. To ensure usability for users at different levels of experience, all approaches are kept practical requiring minimal user intervention once configured with the initial inputs. In addition, evaluated on common computer hardware, the approaches achieve acceptable computation times when analysing medium size networks. Especially for the computationally most expensive part of modelling wireless communication, the availability of different models and their configurable complexity allow for an adaptation to any target system. Conclusively, each contribution of this thesis provides an approach to analyse a different aspect of the network and extract valuable knowledge to exploit. Altogether, they cover the complete range of knowledge acquisition to assist users in gaining insights about their network to ultimately improve the performance.

9.2 Outlook

Taking the contributions of this thesis as a basis, there exist several possibilities for further research topics as well as potential application scenarios in other domains which can be pursued. Some of them are discussed in the following.

9.2.1 Three-Dimensional Localisation Including Mobility

Interesting future work is an extension to the three-dimensional space for localisation, which would allow to address networks spanning over multiple floors but would also further exacerbate the relevance of the encountered problems. Difficulties with graph distributions of FLoW on complex floor plan shapes could be reduced by separating the graphs based on a decomposition into simpler segments which are easier to match. However, this would require further information about each segment, i.e., more anchors to ensure the correct alignment with respect to the other segments on the one hand and an algorithm to connect and merge the segments appropriately on the other hand. Finally, employing FLoW to localise mobile devices in stationary environments is worth of investigation, not only to obtain an estimated position of the mobile elements but also to acquire further information about the stationary nodes if their positions were to be estimated, too. In this sense, FLoW is a first step in understanding the possibilities offered by existing, already deployed and not customised infrastructures together with scenario and system information. Positioning information is only one knowledge about a system that FLoW demonstrated practically possible to obtain.

9.2.2 Online Event Detection and Reasoning

Due to fast computation times, an application of the approaches in online monitoring domain is conceivable, especially the event detection of Follow. The presented approach and analysis focus on individual events. However, multiple events can happen simultaneously making the matching of each event challenging due to the possible overlaps in the affected set of links. In addition, persons present in the environment affect links in a volatile manner while moving around. This produces complex fingerprints that need to be preprocessed in order to extract the different, individual components.

Even with a lower detection accuracy due to multiple events possibly happening in parallel, it is still possible to build a fingerprint that can be integrated in an overall library with a corresponding probability assigned to the possible causing events. By continuously observing the behaviour of the system, it is possible to refine such estimations to progressively increase the correctness of the acquired knowledge. Ultimately, this refinement cycle can allow systems to reason about their own behaviour. At the same time, the process can produce factual knowledge about fingerprints that the current models are not yet capable of perceiving, further systematically highlighting specific interesting cases worth investigating.

9.2.3 Exploiting Opportunistic Communication

The Internet of Things is made of devices with heterogeneous communication characteristics. The presented modelling techniques can seamlessly work with different radio frequencies, depending on the specific employed technologies. Considering that different frequencies correspond to different impacts of obstacles and materials on the signal propagation, it becomes interesting to explore how the approaches can exploit such richness of signals and observations. This would allow to better recognise events happening in the environment and identify the specific best radio technology to detect each of them.

In addition, it would offer the possibility to use the model in order to, e.g., position the devices in the environment if signals are not affected by walls and so on. Interestingly enough, this would allow any device visiting or settling in an environment to contribute with active measurements to the understanding of communication in the scenario, progressively refining their situated system and network knowledge. This could provide a significant step forward both in the ability of systems to self-configure and in the validation, diagnosis and debugging of operational systems over time.

9.2.4 Seeing Wireless Signals

To further support the deployment and design of networks with accessible user interfaces, the employed ray tracing techniques could be used to guide, e.g., the placement of devices in an operational environment based on specific application requirements and utility functions. Considering the advancements in virtual and augmented reality, it is conceivable to let the user ”see” actual wireless signals and navigate in an environment to explore the behaviour of a network and modify its configuration. In this respect, system reconfiguration and optimisation tasks could be integrated as well. With machine learning techniques it would be possible to explore the space of possibilities and learn about the impact of alternative configurations on the system performance and robustness against specific events. Considering, however, the complexity of exploring a 3D environment for optimal configurations, dedicated techniques need to be investigated in order to keep the problem tractable.

Acronyms

AoA	Angle of Arrival
CPS	Cyber-Physical Systems
CPU	Central Processing Unit
CTP	Collection Tree Protocol
ETX	Expected Transmission Count
FLoW	Force-directed Localisation of Wireless Devices
Follow	Follow the Rays
FPPE	False Positive Prediction Error
GPU	Graphics Processing Unit
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
MC	Match Count
MDS	Multidimensional Scaling
MLF	Multilevel Force-directed Algorithm
MOTION	Model-based Optimisation of Wireless Networks
MOVE	Models for Network Optimisation in Versatile Environments
MQE	Model Quality Estimation
OpenCL	Open Computing Language
RAM	Random-Access Memory
RPE	Real Prediction Error
RSSI	Received Signal Strength Indicator
SSE	Sum of Squared Error
TDoA	Time Difference of Arrival
ToA	Time of Arrival
WiFi	Wireless Fidelity
WSN	Wireless Sensor Network

Bibliography

Complete List of Author's Contributions

[a1] Sascha Jungen, Matteo Ceriotti and Pedro José Marrón. "Demo: A Model-based Framework for the Situated Design and Deployment of Wireless Embedded Systems". In: *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems (SenSys)*. ACM. 2017. ISBN: 9781450354592.

[a2] Sascha Jungen, Matteo Ceriotti and Pedro José Marrón. "Follow the Rays: Understanding the Interplay between Environment and System through In-Situ Wireless Modelling". In: *Proceedings of the 22nd ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'19)*. ACM. 2019, pp. 153–161. ISBN: 9781450369046.

[a3] Sascha Jungen et al. "Situated Wireless Networks Optimisation through Model-based Relocation of Nodes". In: *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE. 2017, pp. 389–397. ISBN: 9781538623244.

[a4] Sascha Jungen et al. "Where are You? Localising Stationary Nodes with Limited Information". In: *Proceedings of the IEEE 44th Conference on Local Computer Networks (LCN)*. IEEE. 2019, pp. 161–168. ISBN: 9781728110288.

[c1] Falk Brockmann et al. "Accurate event detection and velocity estimation in wireless environments". In: *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE. 2016, pp. 1057–1066. ISBN: 9788360810903.

[c2] Eduardo Ferrera et al. "Marrying Stationary Low-Power Wireless Networks and Mobile Robots in a Hybrid Surveillance System". In: *2018 14th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE. 2018, pp. 131–138. DOI: 10.1109/DCOSS.2018.00029.

[c3] Richard Figura et al. "Demo Abstract: IMAC, Enabling Flexible Configuration and Result Analysis for Diverse Wireless Sensor Network Experiments". In: *EWSN 2012* (2012), p. 6.

[c4] Richard Figura et al. "IRIS: A Flexible and Extensible Experiment Management and Data Analysis Tool for Wireless Sensor Networks". In: *International Conference on Sensor Systems and Software*. Springer. 2013, pp. 94–110. ISBN: 9783319041667.

[c5] Richard Figura et al. "Iris: Efficient visualization, data analysis and experiment management for wireless sensor networks". In: *EAI Endorsed Transactions on Ubiquitous Environments* 1.3 (2014). DOI: `10.4108/ue.1.3.e4`.

[c6] Richard Figura et al. "Poster Abstract: Deployment, Reconfiguration and Adaptation through Modelling and Simulation in DREAMS". In: *EWSN 2015 Poster / Demo Session*. 2015, pp. 5–6.

[c7] Richard Figura et al. "Μο$\rho\phi\varepsilon\upsilon\varsigma$: Simulate Reality for the Orchestration of Deployed Networked Embedded Systems". In: *Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks*. Junction Publishing. 2018, pp. 145–156. ISBN: 9780994988621.

[c8] Robert Sauter et al. "Notos: efficient emulation of wireless sensor networks with binary-to-source translation." In: *SimuTools*. 2015, pp. 165–174. ISBN: 9781631900792.

References to Scientific Publications

[p1] VS Abhayawardhana et al. "Comparison of empirical propagation path loss models for fixed wireless access systems". In: *2005 IEEE 61st Vehicular Technology Conference*. Vol. 1. IEEE. 2005, pp. 73–77.

[p2] Mohammad Abtahi and Homayoun Hashemi. "Simulation of indoor propagation channel at infrared frequencies in furnished office environments". In: *Proceedings of 6th International Symposium on Personal, Indoor and Mobile Radio Communications*. Vol. 1. IEEE. 1995, pp. 306–310.

[p3] Andreas Achtzehn et al. "MoteMaster: A scalable sensor network testbed for rapid protocol performance evaluation". In: *2009 6th IEEE Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*. IEEE. 2009, pp. 1–3.

[p4] Fadel Adib et al. "Capturing the Human Figure Through a Wall". In: *ACM Transactions on Graphics (TOG)* 34.6 (Oct. 2015), 219:1–219:13. ISSN: 0730-0301.

[p5] Michiel Aernouts et al. "TDAoA: A combination of TDoA and AoA localization with LoRaWAN". In: *Internet of Things* 11 (2020), p. 100236.

[p6] F Aguado Agelet, F Perez Fontan and Arno Formella. "Fast ray tracing for microcellular and indoor environments". In: *IEEE transactions on Magnetics* 33.2 (1997), pp. 1484–1487.

[p7] F Aguado Agelet et al. "Efficient ray-tracing acceleration techniques for radio propagation modeling". In: *IEEE transactions on Vehicular Technology* 49.6 (2000), pp. 2089–2104.

[p8] Alaa Alhamoud et al. "Empirical investigation of the effect of the door's state on received signal strength in indoor environments at 2.4 GHz". In: *Proceedings of the 39th Conference on Local Computer Networks Workshops (LCN Workshops)*. 2014, pp. 652–657.

[p9] Abdulaziz AlSayyari, Ivica Kostanic and Carlos E Otero. "An empirical path loss model for wireless sensor network deployment in an artificial turf environment". In: *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*. IEEE. 2014, pp. 637–642.

[p10] Iyad H Alshami et al. "The effect of people presence on WLAN RSS is governed by influence distance". In: *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*. IEEE. 2016, pp. 197–202.

[p11] WS Ament. "Toward a theory of reflection by a rough surface". In: *Proceedings of the IRE* 41.1 (1953), pp. 142–146.

[p12] Cassio Bento Andrade and Roger Pierre Fabris Hoefel. "IEEE 802.11 WLANs: A comparison on indoor coverage models". In: *CCECE 2010*. IEEE. 2010, pp. 1–6.

[p13] Paramasiven Appavoo et al. "Indriya2: A heterogeneous wireless sensor network (wsn) testbed". In: *International Conference on Testbeds and Research Infrastructures*. Springer. 2018, pp. 3–19.

[p14] Anish Arora et al. "Kansei: A high-fidelity sensing testbed". In: *IEEE Internet Computing* 10.2 (2006), pp. 35–47.

[p15] M Ayadi, A Ben Zineb and S Tabbane. "A UHF path loss model using learning machine for heterogeneous networks". In: *IEEE Transactions on Antennas and Propagation* 65.7 (2017), pp. 3675–3683.

[p16] Mohamad Ayadi and A Ben Zineb. "Body shadowing and furniture effects for accuracy improvement of indoor wave propagation models". In: *IEEE Transactions on Wireless Communications* 13.11 (2014), pp. 5999–6006.

[p17] Paramvir Bahl and Venkata N Padmanabhan. "RADAR: An in-building RF-based user location and tracking system". In: *Proceedings IEEE INFOCOM 2000. Conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)*. Vol. 2. Ieee. 2000, pp. 775–784.

[p18] Michael J Bannister et al. "Force-directed graph drawing using social gravity and scaling". In: *International Symposium on Graph Drawing*. Springer. 2012, pp. 414–425.

[p19] John R. Barry et al. "Simulation of multipath impulse response for indoor wireless optical channels". In: *IEEE journal on selected areas in communications* 11.3 (1993), pp. 367–379.

[p20] Marcel Baunach, Reiner Kolla and Clemens Muhlberger. "Beyond theory: development of a real world localization application as low power WSN". In: *32nd IEEE Conference on Local Computer Networks (LCN 2007)*. IEEE. 2007, pp. 872–884.

[p21] Imane Benkhelifa, Nadia Nouali-Taboudjemat and Samira Moussaoui. "Disaster management projects using wireless sensor networks: An overview". In: *2014 28th International Conference on Advanced Information Networking and Applications Workshops*. IEEE. 2014, pp. 605–610.

[p22] Paul Blaer and Peter K Allen. "Topbot: automated network topology detection with a mobile robot". In: *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. Vol. 2. IEEE. 2003, pp. 1582–1587.

[p23] Carlo Alberto Boano et al. "Hot Packets: A Systematic Evaluation of the Effect of Temperature on Low Power Wireless Transceivers". In: *Proceedings of the 5^{th} Extreme Conference on Communication (ExtremeCom)*. Thórsmörk, Iceland, Aug. 2013, pp. 7–12. ISBN: 978-1-4503-2171-6.

[p24] Lucien Boithias and L-J Libols. *Radio wave propagation*. North Oxford acad. London, 1987.

[p25] Falk Brockmann et al. "RSSI Based Passive Detection of Persons for Waiting Lines Using Bluetooth Low Energy". In: *Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks*. EWSN'18. Madrid, Spain, 2018, pp. 102–113. ISBN: 978-0-9949886-2-1.

[p26] Nirupama Bulusu, John Heidemann and Deborah Estrin. "GPS-less low-cost outdoor localization for very small devices". In: *IEEE personal communications* 7.5 (2000), pp. 28–34.

[p27] Marco Cattani, Carlo Alberto Boano and Kay Römer. “An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Communication”. In: *Journal of Sensor and Actuator Networks* 6.2 (2017). ISSN: 2224-2708.

[p28] Matteo Ceriotti et al. “Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment”. In: *2009 International Conference on Information Processing in Sensor Networks.* IEEE. 2009, pp. 277–288.

[p29] Matteo Ceriotti et al. “Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels”. In: *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks.* IEEE. 2011, pp. 187–198.

[p30] Zhongqiang Chen, Henry L Bertoni and Alex Delis. “Progressive and approximate techniques in ray-tracing-based radio wave propagation prediction models”. In: *IEEE transactions on Antennas and Propagation* 52.1 (2004), pp. 240–251.

[p31] Yu-Fang Chung and Chia-Hui Liu. “Design of a wireless sensor network platform for tele-homecare”. In: *Sensors* 13.12 (2013), pp. 17156–17175.

[p32] Albert Cohen et al. “Interdisciplinary research challenges in computer systems for the 2020s”. In: *USA, Tech. Rep.* (2018).

[p33] David D Coleman and David A Westcott. *Cwna: certified wireless network administrator official study guide: exam Pw0-105.* John Wiley & Sons, 2012.

[p34] Sylvain Collonge, Gheorghe Zaharia and Ghais El Zein. “Influence of the furniture on 60 GHz radio propagation in a residential environment”. In: *Signals, Circuits and Systems, 2003. SCS 2003. International Symposium on.* Vol. 2. IEEE. 2003, pp. 413–416.

[p35] Electronic Communications Committee et al. “The European table of frequency allocations and applications in the frequency range 8.3 kHz to 3000 GHz (ECA table)”. In: *Proceedings of European Conference of Postal and Telecommunications Administrations; Electronic Communications Committee: Copenhagen, Denmark.* 2013.

[p36] Eraldo Damosso, Luis M Correia et al. “Digital mobile radio towards future generation systems”. In: *COST 231 final report* (1999), p. 474.

[p37] Silvia Demetri, Gian Pietro Picco and Lorenzo Bruzzone. “Estimating low-power radio signal attenuation in forests: A LiDAR-based approach”. In: *2015 International Conference on Distributed Computing in Sensor Systems.* IEEE. 2015, pp. 71–80.

[p38] Jacques Deygout. “Multiple knife-edge diffraction of microwaves”. In: *IEEE Transactions on Antennas and Propagation* 14.4 (1966), pp. 480–489.

[p39] Edsger W Dijkstra. “A note on two problems in connexion with graphs”. In: *Edsger Wybe Dijkstra: His Life, Work, and Legacy.* 2022, pp. 287–290.

[p40] Manjunath Doddavenkatappa, Mun Choon Chan and Akkihebbal L Ananda. “Indriya: A low-cost, 3D wireless sensor network testbed”. In: *International conference on testbeds and research infrastructures.* Springer. 2011, pp. 302–316.

[p41] Nguyen Vo Quang Dong, Chung-Ta King and Cheng-Chih Liu. “On deploying IoT indoor: An incremental approach”. In: *2013 International Conference on Parallel and Distributed Systems.* IEEE. 2013, pp. 552–557.

[p42] Wei Dong et al. “Dynamic packet length control in wireless sensor networks”. In: *IEEE Trans. on wireless communications* 13.3 (2014), pp. 1172–1181.

[p43] Xavier Emery. “Simple and ordinary multigaussian kriging for estimating recoverable reserves”. In: *Mathematical Geology* 37.3 (2005), pp. 295–319.

[p44] Jess Epstein and Donald W Peterson. “An experimental study of wave propagation at 850 Mc”. In: *Proceedings of the IRE* 41.5 (1953), pp. 595–611.

[p45] Matthias Faessler et al. “Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle”. In: *Journal of Field Robotics* 33.4 (June 2016), pp. 431–450. ISSN: 1556-4959.

[p46] Ngewi Fet, Marcus Handte and Pedro José Marrón. “A Model for WLAN Signal Attenuation of the Human Body”. In: *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing.* UbiComp ’13. Zurich, Switzerland, 2013, pp. 499–508. ISBN: 978-1-4503-1770-2.

[p47] Steven Fortune. “A beam-tracing algorithm for prediction of indoor radio propagation”. In: *Workshop on Applied Computational Geometry.* Springer. 1996, pp. 157–166.

[p48] Steven Fortune. “Efficient algorithms for prediction of indoor radio propagation”. In: *VTC’98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No. 98CH-36151).* Vol. 1. IEEE. 1998, pp. 572–576.

[p49] Augustin Fresnel. *Oeuvres complètes d'augustin Fresnel.* Imprimerie impériale, 1868.

[p50] Harald T Friis. "A note on a simple transmission formula". In: *Proceedings of the IRE* 34.5 (1946), pp. 254–256.

[p51] Songwei Fu et al. "An Approach to Detect Anomalous Degradation in Signal Strength of IEEE 802.15. 4 Links". In: *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2018, pp. 1–9.

[p52] Zhang Fu et al. "Managing your trees: Insights from a metropolitan-scale low-power wireless network". In: *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2014, pp. 664–669.

[p53] Omprakash Gnawali et al. "CTP: An Efficient, Robust, and Reliable Collection Tree Protocol for Wireless Sensor Networks". In: *ACM Transactions on Sensor Networks (TOSN)* 10.1 (2013), 16:1–16:49. ISSN: 1550-4859. DOI: 10.1145/2529988. URL: http://doi.acm.org/10.1145/2529988.

[p54] Álvaro González. "Measurement of areas on a sphere using Fibonacci and latitude–longitude lattices". In: *Mathematical Geosciences* 42.1 (2010), p. 49.

[p55] Jean-Marie Gorce, Katia Jaffres-Runser and Guillaume De La Roche. "Deterministic approach for fast simulations of indoor radio wave propagation". In: *IEEE transactions on Antennas and Propagation* 55.3 (2007), pp. 938–948.

[p56] Jean-Marie Gorce, Katia Jaffrès-Runser and Guillaume de La Roche. "The Adaptive Multi-Resolution Frequency-Domain ParFlow (MR-FDPF) Method for Indoor Radio Wave Propagation Simulation. Part I: Theory and Algorithms". In: (2005).

[p57] S Gronemeyer and A McBride. "MSK and offset QPSK modulation". In: *IEEE transactions on communications* 24.8 (1976), pp. 809–820.

[p58] Vlado Handziski et al. "Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks". In: *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality.* 2006, pp. 63–70.

[p59] KVS Hari et al. "Channel models for fixed wireless applications". In: *IEEE 802.16 Broadband wireless access working group* (2003).

[p60] Masaharu Hata. "Empirical formula for propagation loss in land mobile radio services". In: *IEEE transactions on Vehicular Technology* 29.3 (1980), pp. 317–325.

[p61] Tian He et al. “Range-free localization schemes for large scale sensor networks”. In: *Proceedings of the 9th International conference on Mobile computing and networking*. 2003, pp. 81–95.

[p62] E. Hecht and A. Zajac. *Optics*. Addison-Wesley world student series edition. Addison-Wesley Publishing Company, 1987. ISBN: 978-0201116090.

[p63] Salaheddin Hosseinzadeh et al. “Empirical propagation performance evaluation of LoRa for indoor environment”. In: *2017 IEEE 15th international conference on industrial informatics (INDIN)*. IEEE. 2017, pp. 26–31.

[p64] Andrew Howard, Maja J Mataric and Gaurav Sukhatme. “Relaxation on a mesh: a formalism for generalized localization”. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. Vol. 2. IEEE. 2001, pp. 1055–1060.

[p65] Jim Hurley. “Ray Tracing Goes Mainstream.” In: *Intel Technology Journal* 9.2 (2005).

[p66] Christiaan Huygens. *Traité de la lumière:...* chez Pierre vander Aa, marchand libraire, 1920.

[p67] Frank K Hwang and Dana S Richards. “Steiner tree problems”. In: *Networks* 22.1 (1992), pp. 55–89.

[p68] *Letter symbols to be used in electrical technology - Part 3: Logarithmic and related quantities, and their units*. Standard. Geneva, CH: International Electrotechnical Commission, July 2002.

[p69] “IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”. In: *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)* (2016), pp. 1–3534.

[p70] “IEEE Standard for Low-Rate Wireless Networks”. In: *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)* (2020), pp. 1–800.

[p71] Fumio Ikegami et al. “Propagation factors controlling mean field strength on urban streets”. In: *IEEE Transactions on Antennas and Propagation* 32.8 (1984), pp. 822–829.

[p72] Tetsuro Imai and Teruya Fujii. "Fast algorithm for indoor microcell area prediction system using ray-tracing method". In: *Electronics and Communications in Japan (Part I: Communications)* 85.6 (2002), pp. 41–52.

[p73] ZigBee Alliance Inc. "ZigBee Specification". In: *ZigBee Document 05-3474-21* (Aug. 2015). URL: `https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf` (visited on 02/09/2020).

[p74] Texas Instruments Incorporated. *2.4-GHz Inverted F Antenna.* 2007. URL: `http://www.ti.com/lit/an/swru120d/swru120d.pdf` (visited on 22/01/2020).

[p75] Timofei Istomin et al. "Trident: In-field connectivity assessment for wireless sensor networks". In: *Proceedings of the 6th extreme conference on communication and computing (ExtremeCom)*. 2014.

[p76] Paul Jaccard. "Lois de distribution florale dans la zone alpine". In: *Bull Soc Vaudoise Sci Nat* 38 (1902), pp. 69–130.

[p77] Zhong Ji et al. "Efficient ray-tracing methods for propagation prediction for indoor wireless communications". In: *IEEE Antennas and Propagation Magazine* 43.2 (2001), pp. 41–49.

[p78] Ossi Kaltiokallio et al. "Non-invasive Respiration Rate Monitoring Using a Single COTS TX-RX Pair". In: *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IPSN '14. Berlin, Germany, 2014, pp. 59–70. ISBN: 978-1-4799-3146-0.

[p79] Kaustav Kar et al. "Motley Keenan model of in-building coverage analysis of IEEE 802.11 n WLAN signal in electronics and communication engineering department of National Institute of Technology Durgapur". In: *2016 International Conference on Microelectronics, Computing and Communications (MicroCom)*. IEEE. 2016, pp. 1–6.

[p80] Tom Karygiannis and Les Owens. "Wireless network security". In: *NIST special publication* 800 (2002), p. 48.

[p81] Janice M Keenan and Andrew J Motley. "Radio coverage in buildings". In: *British telecom technology Journal* 8.1 (1990), pp. 19–24.

[p82] Michael Kintner-Meyer and Michael R Brambley. "Pros & cons of wireless". In: *ASHRAE journal* 44.11 (2002), pp. 54–56.

[p83] Dmitrii Kirov, Roberto Passerone and Massimo Donelli. "Statistical characterization of the 2.4 GHz radio channel for WSN in indoor office environments". In: *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE. 2016, pp. 1–9.

[p84] Naoki Kita et al. "Experimental study of propagation characteristics for wireless communications in high-speed train cars". In: *2009 3rd European Conference on Antennas and Propagation*. IEEE. 2009, pp. 897–901.

[p85] JeongGil Ko et al. "Wireless sensor networks for healthcare". In: *Proceedings of the IEEE* 98.11 (2010), pp. 1947–1960.

[p86] Stephen G Kobourov. "Spring embedders and force directed graph drawing algorithms". In: *arXiv preprint arXiv:1201.3011* (2012).

[p87] Danie G. Krige. "A statistical approach to some basic mine valuation problems on the Witwatersrand". In: *Journal of Chemical, Metallurgical, and Mining Society of South Africa* 52.6 (1951), pp. 119–139.

[p88] Alexander Kröller et al. "Deterministic boundary recognition and topology extraction for large sensor networks". In: *Proceedings of the ACM-SIAM symposium on Discrete algorithm*. 2006, pp. 1000–1009.

[p89] Thomas Kurner and Alexander Meier. "Prediction of outdoor and outdoor-to-indoor coverage in urban areas at 1.8 GHz". In: *IEEE Journal on Selected Areas in Communications* 20.3 (2002), pp. 496–506.

[p90] J-F Lafortune and Michel Lecours. "Measurement and modeling of propagation losses in a building at 900 MHz". In: *IEEE Transactions on Vehicular Technology* 39.2 (1990), pp. 101–108.

[p91] Christos Laoudias et al. "A Survey of Enabling Technologies for Network Localization, Tracking, and Navigation". In: *IEEE Communications Surveys & Tutorials* 20.4 (2018), pp. 3607–3644.

[p92] Viet-Duc Le et al. "Distributed localization in wireless sensor networks based on force-vectors". In: *2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. IEEE. 2008, pp. 31–36.

[p93] William CY Lee. "A new propagation path-loss prediction model for military mobile access". In: *MILCOM 1985-IEEE Military Communications Conference*. Vol. 2. IEEE. 1985, pp. 359–368.

[p94] Philip Levis et al. "TinyOS: An operating system for sensor networks". In: *Ambient intelligence*. Springer, 2005, pp. 115–148.

[p95] Roman Lim et al. “FlockLab: A Testbed for Distributed, Synchronized Tracing and Profiling of Wireless Embedded Systems”. In: *Proceedings of the International Conference on Information Processing in Sensor Networks*. 2013, pp. 153–166.

[p96] André GM Lima and Luiz F Menezes. “Motley-Keenan model adjusted to the thickness of the wall”. In: *SBMO/IEEE MTT-S International Conference on Microwave and Optoelectronics, 2005.* IEEE. 2005, pp. 180–182.

[p97] Jaime Lloret et al. “A fast design model for indoor radio coverage in the 2.4 GHz wireless LAN”. In: *1st International Symposium onWireless Communication Systems, 2004.* IEEE. 2004, pp. 408–412.

[p98] Ranjan Kumar Mahapatra and NSV Shet. “Experimental analysis of RSSI-based distance estimation for wireless sensor networks”. In: *2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER).* IEEE. 2016, pp. 211–215.

[p99] Sathiya Kumaran Mani et al. “An architecture for IoT clock synchronization”. In: *Proceedings of the 8th International Conference on the Internet of Things*. 2018, pp. 1–8.

[p100] Xiao Hong Mao, Yee Hui Lee and Boon Chong Ng. “Propagation modes and temporal variations along a lift shaft in UHF band”. In: *IEEE Transactions on Antennas and propagation* 58.8 (2010), pp. 2700–2709.

[p101] Ramona Marfievici. “Measuring, Understanding, and Estimating the Influence of the Environment on low-power Wireless Networks”. PhD thesis. University of Trento, 2015.

[p102] Ramona Marfievici et al. “How Environmental Factors Impact Outdoor Wireless Sensor Networks: A Case Study”. In: *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems.* Oct. 2013, pp. 565–573.

[p103] David W Matolak and Ruoyu Sun. “Channel characteristics for elevator shafts at 5 GHz”. In: *2013 IEEE Global Communications Conference (GLOBECOM).* IEEE. 2013, pp. 3931–3935.

[p104] Fatiha Mekelleche and Hafid Haffaf. “Classification and comparison of range-based localization techniques in wireless sensor networks”. In: *Journal of Communications* 12.4 (2017), pp. 221–227.

[p105] Ivan Minakov et al. “A Comparative Study of Recent Wireless Sensor Network Simulators”. In: *ACM Transactions on Sensor Networks (TOSN)* 12.3 (2016), 20:1–20:39. ISSN: 1550-4859. DOI: `10.1145/2903144`. URL: `http://doi.acm.org/10.1145/2903144`.

[p106] Daniel Minder, Marcus Handte and Pedro José Marrón. “Tinyadapt: An adaptation framework for sensor networks”. In: *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. IEEE. 2010, pp. 253–256.

[p107] Eugenia Montiel, Alberto S Aguado and François X Sillion. “A radiance model for predicting radio wave propagation in irregular dense urban areas”. In: *IEEE Transactions on Antennas and Propagation* 51.11 (2003), pp. 3097–3108.

[p108] Luca Mottola et al. “Not All Wireless Sensor Networks Are Created Equal: A Comparative Study on Tunnels”. In: *ACM Transactions on Sensor Networks (TOSN)* 7.2 (2010), 15:1–15:33. ISSN: 1550-4859. DOI: 10.1145/1824766.1824771. URL: http://doi.acm.org/10.1145/1824766.1824771.

[p109] Aaftab Munshi. “The opencl specification”. In: *2009 IEEE Hot Chips 21 Symposium (HCS)*. IEEE. 2009, pp. 1–314.

[p110] Dumitru-Iulian Nastac et al. “Automatic data acquisition with robots for indoor fingerprinting”. In: *2018 International Conference on Communications (COMM)*. IEEE. 2018, pp. 321–326.

[p111] Randall K Nichols, Panos Lekkas and Panos C Lekkas. *Wireless security*. McGraw-Hill Professional Publishing, 2001.

[p112] Dragos Niculescu and Badri Nath. “Ad hoc positioning system (APS) using AOA”. In: *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies*. Vol. 3. 2003, pp. 1734–1743.

[p113] Dragoş Niculescu and Badri Nath. “DV based positioning in ad hoc networks”. In: *Telecommunication Systems* 22.1-4 (2003), pp. 267–280.

[p114] OS Oguejiofor et al. “Trilateration based localization algorithm for wireless sensor network”. In: *International Journal of Science and Modern Engineering* 1.10 (2013), pp. 2319–6386.

[p115] Felix Jonathan Oppermann et al. “Automatic protocol configuration for dependable internet of things applications”. In: *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)*. IEEE. 2015, pp. 742–750.

[p116] Fredrik Osterlind et al. “Cross-level sensor network simulation with cooja”. In: *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*. IEEE. 2006, pp. 641–648.

[p117] Juan Pascual-Garcia et al. "Using tuned diffuse scattering parameters in ray tracing channel modeling". In: *2015 9th European Conference on Antennas and Propagation (EuCAP)*. IEEE. 2015, pp. 1–4.

[p118] Juan Pascual-Garcia et al. "On the importance of diffuse scattering model parameterization in indoor wireless channels at mm-wave frequencies". In: *Ieee Access* 4 (2016), pp. 688–701.

[p119] Gilbert E Perez, Abdulaziz Alsayyari and Ivica Kostanic. "Comparison of the propagation loss of a real-life wireless sensor network and its complimentary simulation model". In: *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*. IEEE. 2015, pp. 1832–1837.

[p120] Joseph Polastre, Robert Szewczyk and David Culler. "Telos: Enabling Ultra-low Power Wireless Research". In: *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*. IPSN '05. Los Angeles, California, 2005. ISBN: 0-7803-9202-7.

[p121] E.M. Purcell and D.J. Morin. *Electricity and Magnetism*. Electricity and Magnetism. Cambridge University Press, 2013. ISBN: 9781107014022. URL: `https://books.google.de/books?id=A2rS5vlSFq0C`.

[p122] Alok Ranjan, Prasant Misra and HB Sahu. "Experimental measurements and channel modeling for wireless communication networks in underground mine environments". In: *2017 11th European Conference on Antennas and Propagation (EUCAP)*. IEEE. 2017, pp. 1345–1349.

[p123] Theodore S Rappaport et al. *Wireless communications: principles and practice*. Vol. 2. 1996.

[p124] Terhi Rautiainen, Reiner Hoppe and Gerd Wolfle. "Measurements and 3D ray tracing propagation predictions of channel characteristics in indoor environments". In: *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE. 2007, pp. 1–5.

[p125] Jose Carlos Reyes-Guerrero, Gordan Sisul and Luis A Mariscal. "Measuring and estimating the propagation path loss and shadowing effects for marine wireless sensor networks at 5.8 GHz". In: *2012 20th Telecommunications Forum (TELFOR)*. IEEE. 2012, pp. 323–226.

[p126] Mónica Ribero et al. “Deep Learning Propagation Models over Irregular Terrain”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE. 2019, pp. 4519–4523.

[p127] Josiane C Rodrigues et al. “People and furniture effects on the transmitter coverage area”. In: *2006 IEEE Ninth International Symposium on Spread Spectrum Techniques and Applications.* IEEE. 2006, pp. 391–395.

[p128] Dennis M Rose and Thomas Kürner. “Outdoor-to-indoor propagation—Accurate measuring and modelling of indoor environments at 900 and 1800 MHz”. In: *2012 6th European Conference on Antennas and Propagation (EUCAP).* IEEE. 2012, pp. 1440–1444.

[p129] Dennis M Rose et al. “Impact of realistic indoor mobility modelling in the context of propagation modelling on the user and network experience”. In: *2013 7th European Conference on Antennas and Propagation (EuCAP).* IEEE. 2013, pp. 3979–3983.

[p130] Balqies Sadoun and Omar Al-Bayari. “Location based services using geographical information systems”. In: *Computer Communications* 30.16 (2007), pp. 3154–3160.

[p131] Adel AM Saleh and Reinaldo Valenzuela. “A statistical model for indoor multipath propagation”. In: *IEEE Journal on selected areas in communications* 5.2 (1987), pp. 128–137.

[p132] Majdi Salem, Mahamod Ismail and Norbahiah Misran. “Validation of three-dimensional ray-tracing algorithm for indoor wireless propagations”. In: *International Scholarly Research Notices* 2011 (2011).

[p133] Ricardo Santos et al. “Crowdsourcing-based fingerprinting for indoor location in multi-storey buildings”. In: *IEEE Access* 9 (2021), pp. 31143–31160.

[p134] Kurt R Schaubach, Nathaniel J Davis and Theodore S Rappaport. “A ray tracing method for predicting path loss and delay spread in microcellular environments”. In: *[1992 Proceedings] Vehicular Technology Society 42nd VTS Conference-Frontiers of Technology.* IEEE. 1992, pp. 932–935.

[p135] P Series. “Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 900 MHz to 100 GHz”. In: *Recommendation ITU-R* (2012), pp. 1238–7.

[p136] Yi Shang et al. “Localization from mere connectivity”. In: *Proceedings of the ACM international symposium on Mobile ad hoc networking & computing.* 2003, pp. 201–212.

[p137] Muhammad Usman Sheikh, Kimmo Hiltunen and Jukka Lempiainen. “Enhanced outdoor to indoor propagation models and impact of different ray tracing approaches at higher frequencies”. In: *Adv. Sci. Technol. Eng. Syst. J.* 3.2 (2018), pp. 58–68.

[p138] Guowei Shen, Rudolf Zetik and Reiner S Thoma. “Performance comparison of TOA and TDOA based location estimation algorithms in LOS environment”. In: *2008 5th Workshop on Positioning, Navigation and Communication.* IEEE. 2008, pp. 71–78.

[p139] Santar Pal Singh and SC Sharma. “Range free localization techniques in wireless sensor networks: A review”. In: *Procedia Computer Science* 57 (2015), pp. 7–16.

[p140] Nicolas Sklavos and Xinmiao Zhang. *Wireless security and cryptography: specifications and implementations.* CRC Press, 2017.

[p141] YF Solahuddin and R Mardeni. “Indoor empirical path loss prediction model for 2.4 GHz 802.11 n network”. In: *2011 IEEE International Conference on Control System, Computing and Engineering.* IEEE. 2011, pp. 12–17.

[p142] Kannan Srinivasan et al. “An Empirical Study of Low-power Wireless”. In: *ACM Transactions on Sensor Networks (TOSN)* 6.2 (Mar. 2010), 16:1–16:49. ISSN: 1550-4859.

[p143] John C Stein. “Indoor radio WLAN performance part II: Range performance in a dense office environment”. In: *Intersil Corporation* 2401 (1998).

[p144] William C Stone. *Electromagnetic signal attenuation in construction materials.* Tech. rep. 1997.

[p145] Moritz Strübe et al. “DrySim: simulation-aided deployment-specific tailoring of mote-class WSN software”. In: *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems.* 2014, pp. 3–11.

[p146] Shu Sun, George R MacCartney and Theodore S Rappaport. “Millimeter-wave distance-dependent large-scale propagation measurements and path loss models for outdoor and indoor 5G systems”. In: *2016 10th European Conference on Antennas and Propagation (EuCAP).* IEEE. 2016, pp. 1–5.

[p147] Satoshi Takahashi et al. “An evaluation point culling algorithm for radio propagation simulation based on the imaging method”. In: *Wireless personal communications.* Springer, 1996, pp. 111–121.

[p148] “The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition”. In: *IEEE Std 100-2000* (2000), pp. 1–1362.

[p149] *The International System of Units (SI)*. 8th ed. International Bureau of Weights and Measures, 2019. ISBN: 978-92-822-2272-0.

[p150] Inc. Thread Group. *Thread 1.2 Base Features*. Tech. rep. June 2019. URL: `https://www.threadgroup.org/Portals/0/documents/support/Thread%201.2%20Base%20Features.pdf` (visited on 02/09/2020).

[p151] Ankit Tiwari, Prasanna Ballal and Frank L Lewis. "Energy-efficient wireless sensor network design and implementation for condition-based maintenance". In: *ACM Transactions on Sensor Networks (TOSN)* 3.1 (2007), 1–es.

[p152] Gilman Tolle et al. "A macroscope in the redwoods". In: *Proceedings of the 3rd international conference on Embedded networked sensor systems*. 2005, pp. 51–63.

[p153] Sameer Babu TP and Sunil Kumar. "Underwater communications". In: *2015 IEEE Underwater Technology (UT)*. IEEE. 2015, pp. 1–3.

[p154] Ubaid Ullah et al. "Outdoor-to-Indoor and Indoor-to-Indoor Propagation Path Loss Modeling Using Smart 3D Ray Tracing Algorithm at 28 GHz mmWave". In: *Arabian Journal for Science and Engineering* (2020), pp. 1–10.

[p155] Muhammad Umer et al. "Spatial interpolation in wireless sensor networks: localized algorithms for variogram modeling and Kriging". In: *Geoinformatica* 14.1 (2010), pp. 101–134.

[p156] Reinaldo A Valenzuela. "Ray tracing prediction of indoor radio propagation". In: *5th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Wireless Networks-Catching the Mobile Future*. Vol. 1. IEEE. 1994, pp. 140–144.

[p157] Anitha Varghese and Deepaknath Tandur. "Wireless requirements and challenges in Industry 4.0". In: *2014 International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE. 2014, pp. 634–638.

[p158] Abhishek Verma et al. "Relay node placement in constrained environment". In: *Proceeding of International Conference on Communications and Computing Systems (ICCCS 2016)*. 2017, pp. 417–420.

[p159] Arthur R Von Hippel, ed. *Dielectric materials and applications: papers by 22 contributors*. Technology Press of MIT and Wiley, 1954.

[p160] Rene Wahl, Oliver Staebler and Marian Jurado Gallardo. "Requirements for indoor building databases to increase the accuracy of the propagation results". In: *2007 16th IST Mobile and Wireless Communications Summit.* IEEE. 2007, pp. 1–4.

[p161] Joram Walfisch and Henry L Bertoni. "A theoretical model of UHF propagation in urban environments". In: *IEEE Transactions on antennas and propagation* 36.12 (1988), pp. 1788–1796.

[p162] Chris Walshaw. "A multilevel algorithm for force-directed graph-drawing". In: *Journal of Graph Algorithms and Applications* 7.3 (2006), pp. 253–285.

[p163] Geoff Werner-Allen et al. "Fidelity and yield in a volcano monitoring sensor network". In: *Proceedings of the 7th symposium on Operating systems design and implementation.* 2006, pp. 381–396.

[p164] Patrick Michael Whelan and Michael John Hodgson. *Essential Principles Of Physics.* London, 1987.

[p165] Robert Wilson. "Propagation losses through common building materials 2.4 GHz vs 5 GHz". In: *E10589 Magis Networks, Inc.: San Diego, CA, USA* (2002).

[p166] Gerd Wölfle et al. "Dominant path prediction model for indoor scenarios". In: *German Microwave Conference (GeMIC).* Vol. 27. Ulm. 2005.

[p167] Jiachen Yang et al. "A real-time monitoring system of industry carbon monoxide based on wireless sensor networks". In: *Sensors* 15.11 (2015), pp. 29535–29546.

[p168] Zheng Yang, Chenshu Wu and Yunhao Liu. "Locating in fingerprint space: Wireless indoor localization with little human intervention". In: *Proceedings of the 18th annual international conference on Mobile computing and networking.* 2012, pp. 269–280.

[p169] Ali Yassine et al. "Hybrid positioning data fusion in heterogeneous networks with critical hearability". In: *EURASIP Journal on Wireless Communications and Networking* 2014.1 (2014), pp. 1–16.

[p170] Misun Yu, Jun-Keun Song and Pyeongsoo Mah. "RNIndoor: A relay node deployment method for disconnected wireless sensor networks in indoor environments". In: *2011 Third International Conference on Ubiquitous and Future Networks (ICUFN).* IEEE. 2011, pp. 19–24.

[p171] Zhengqing Yun and Magdy F. Iskander. "Ray Tracing for Radio Propagation Modeling: Principles and Applications". In: *IEEE Access* 3 (2015), pp. 1089–1100. ISSN: 2169-3536.

[p172] Wei Zhang et al. "Physiological data acquisition system for education assessment using wireless sensor network". In: *2008 International Conference on Information Technology and Applications in Biomedicine*. IEEE. 2008, pp. 471–473.

[p173] Xi Zheng et al. "Perceptions on the State of the Art in Verification and Validation in Cyber-Physical Systems". In: *IEEE Systems Journal* 11.4 (Dec. 2015), pp. 2614–2627. ISSN: 1932-8184.

[p174] Gang Zhou et al. "Models and Solutions for Radio Irregularity in Wireless Sensor Networks". In: *ACM Transactions on Sensor Networks (TOSN)* 2.2 (May 2006), pp. 221–262. ISSN: 1550-4859.

[p175] Marco Zimmerling et al. "pTunes: Runtime parameter adaptation for low-power MAC protocols". In: *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. 2012, pp. 173–184.

Miscellaneous References

[m1] University of Konstanz Algorithmics Group. *MDSJ: Java Library for Multidimensional Scaling (Version 0.2)*. 2009. URL: `http://www.inf.uni-konstanz.de/algo/software/mdsj/` (visited on 08/03/2019).

[m2] NGMN Alliance. *5G White Paper, February 2015*. 2015. URL: `https://www.ngmn.org/wp-content/uploads/NGMN_5G_White_Paper_V1_0.pdf` (visited on 01/10/2021).

[m3] Andrea Antonello, Silvia Franceschi and Rigon Riccardo. *The Horton Machine - Spatial tools for environmental Modelling*. 2022. DOI: `10.5281/zenodo.5881777`. URL: `https://thehortonmachine.github.io/hortonmachine/` (visited on 01/02/2023).

[m4] Aparapi. *Open-source framework for executing native Java code on the GPU*. 2018. URL: `http://aparapi.com` (visited on 28/04/2020).

[m5] Atmel. *AT86RF215 /AT86RF215IQ /AT86RF215M Datasheet*. May 2016. URL: `http://ww1.microchip.com/downloads/en/devicedoc/atmel-42415-wireless-at86rf215_datasheet.pdf` (visited on 20/10/2020).

[m6] Borb. *Inverse Square Law*. CC BY-SA 3.0 (https://-creativecommons.org/licenses/by-sa/3.0). 2008. URL: `https://commons.wikimedia.org/wiki/File:Inverse_square_law.svg` (visited on 28/04/2020).

[m7] Bundesnetzagentur. *Allgemeinzuteilung von Frequenzen für die Nutzung in lokalen Netzwerken; Wireless Local Area Networks (WLAN-Funkanwendungen).* In: Vfg 10/2013, geändert mit Vfg 64/2018. 2018. URL: `https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/Allgemeinzuteilungen/2013_10_WLAN_2,4GHz_pdf` (visited on 04/05/2020).

[m8] Ming Hsieh Department of Electrical Engineering at the University of Southern California (USC). *Tutornet: A Tiered Wireless Sensor Network Testbed.* URL: `https://anrg.usc.edu/www/tutornet/` (visited on 10/11/2020).

[m9] Texas Instruments. *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver.* Mar. 2013. URL: `https://www.ti.com/lit/ds/symlink/cc2420.pdf` (visited on 20/10/2020).

[m10] Texas Instruments. *CC2538 Powerful Wireless Microcontroller System-On-Chip for 2.4-GHz IEEE 802.15.4, 6LoWPAN, and ZigBee® Applications.* Apr. 2015. URL: `https://www.ti.com/lit/ds/symlink/cc2538.pdf` (visited on 20/10/2020).

[m11] Thomas King et al. *CRAWDAD dataset mannheim/compass (v. 2008-04-11).* URL: `http://crawdad.org/mannheim/compass/20080411` (visited on 01/10/2021).

[m12] Industrial Shields. *Open Mote B User Guide.* 2019. URL: `https://www.industrialshields.com/web/content?model=ir.attachment&field=datas&id=111227&` (visited on 22/01/2020).

[m13] SuperManu. *Electromagnetic Wave.* CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0). 2007. URL: `https://commons.wikimedia.org/wiki/File:Onde_electromagnetique.svg` (visited on 28/04/2020).

[m14] Gilles Tran. *Glasses, pitcher, ashtray and dice (POV-Ray).* 2006. URL: `http://www.oyonale.com/modeles.php?lang=en&page=40` (visited on 28/04/2020).

[m15] Gilles Tran. *Real-time raytracing (POV-Ray).* 2006. URL: `http://www.oyonale.com/modeles.php?lang=en&page=54` (visited on 28/04/2020).

Appendix

The appendix of this thesis is provided in digital form. The enclosed DVD contains the following files:

- Java source code including implemented and used algorithms;
- Performed measurements used for the evaluation.