Carl von Ossietzky Universität Oldenburg

Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

Improving End-to-end Quality of Service in Low-power Wireless Sensor Networks

Dissertation zur Erlangung des Grades eines
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorlegt von

M.Sc. Peilin Zhang

Gutachter:

Prof. Dr.-Ing. Oliver Theel
Prof. Dr. Oliver Kramer
Prof. Dr. Olaf Landsiedel

Tag der Disputation:
05.06.2020

Abstract

Nowadays, smart homes, smart cities, and intelligent transportation are infrastructure systems connecting human beings and increasingly changing our daily life. Such systems are commonly defined as the Internet of Things (IoT) or Cyber-Physical Systems (CPS), where the entire physical world is closely associated with sensors, machines, and networked embedded devices. In such a sophisticated dynamic system, devices are interconnected to sense measurements, to process valuable information, and to exchange data in distributed networks. A wireless sensor network (WSN) is a network that comprises a large number of sensor nodes. Each node is equipped with sensors to detect physical phenomena such as light, heat, pressure, and humidity to name but a few. Sensor nodes communicate with each other wirelessly, thus, WSNs feature easier deployment and better flexibility of devices compared with wired solutions. Owing to the recent advances in electronics, networking, and information processing, WSNs have risen as a promising technology for IoT and CPS. Over the past decade, a wide range of WSN-based applications have been proposed and implemented, such as environmental monitoring systems, forecasting systems and healthcare systems.

In most of these applications, low power, inexpensive and tiny sensor nodes cooperate as a network. In particular, such networks have to be energy-efficient and must be able to provide a sufficient level of quality of services (QoS), such as reliability, timeliness, energy efficiency, and security. However, QoS provision in WSNs is an extremely challenging task, since these QoS metrics are typically contradicting to each other. This is because of, for instance, bounded resources of the deployed devices with respect to computation capability, memory capacity, energy budget, multihop communication over lossy low-power wireless channels, and unpredictable and dynamic changes in (often adverse) environments.

Recently, there has been growing interest in Industry 4.0. Industry 4.0, commonly known as the 4th industrial revolution, refers to the current trend of automation and data exchange in manufacturing technologies aiming to foster the so-called "smart factory" concept, which originates from high-technology strategies of the German government. Within modular structured smart factories, IoT and CPS monitor physical processes, communicate and cooperate with each other and with humans in real-time. For a range of WSN-based applications, especially mission-critical applications under adverse conditions, for example, in smart buildings, industrial monitoring and control, cooperative driving and so forth, maintaining a consistent QoS guarantee throughout the network lifetime is highly required. That is, any performance degradation over time in WSN-based applications should be avoided as much as possible. In this thesis, the main goal is to facilitate the designs of WSN applications and protocols for the realizations of IoT and CPS in Industry 4.0. Specifically, we provide a number of solutions to improve end-to-end QoS in WSN communications. Namely, this thesis presents four novel techniques for WSNs, their analytical studies, practical implementations, as well as real-world eval-

uations and corresponding analyses. Particularly, we make four main contributions in this thesis:

- **Multichannel Opportunistic Routing**. WSNs technically share the 2.4 GHz ISM band with a number of wireless technologies, such as WiFi and Bluetooth. This and external interference from electrical devices, such as, for example, microwaves, deteriorate the reliability of many routing protocols in WSNs. Multichannel communication strategies allow routing protocols to provide reliability in presence of interference. We propose robust, reliable, and energy-efficient *Multichannel Opportunistic Routing* (MOR) for WSNs. MOR employs both, opportunistic routing and multichannel hopping strategies, to improve the robustness of the network to interference. The combination of both, opportunistic routing and opportunistic multichannel hopping, empowers MOR to take advantage of not only the spatial and temporal diversities as traditional opportunistic routing in WSNs does, but also of frequency diversity. We implement MOR in Contiki and conduct extensive experiments in the FlockLab testbed. Under interference MOR provides an end-to-end packet delivery ratio (PDR) of more than 98%, while other protocols such as, for example, opportunistic IPv6 Routing Protocol for Low-Power and Lossy Networks (ORPL), obtain a PDR of merely 25%. Additionally, our duty cycle stays below 2% for these settings and latency is less than two seconds. In interference-free scenarios, MOR achieves a performance similar to our baseline protocol ORPL, with only an approximately 0.3% increment of the duty cycle.

- **Machine Learning-based Flooding**. Concurrent transmission, a novel communication paradigm, has been shown to effectively accomplish a reliable and energy-efficient flooding in wireless networks. With multiple nodes exploiting a receive-and-forward scheme in the network, this technique inevitably introduces communication redundancy and consequently raises the energy consumption of the nodes. We propose *Less is More* (LiM), an energy-efficient flooding protocol for wireless sensor networks. LiM builds on concurrent transmissions, exploiting constructive interference and the capture effect to achieve high reliability and low latency. Moreover, LiM equips itself with a machine learning capability to progressively reduce redundancy while maintaining high reliability. As a result, LiM is able to significantly reduce the radio-on time and therefore energy consumption. We compare LiM with our baseline protocol Glossy by extensive experiments in the 30-node testbed FlockLab. Experimental results show that LiM highly reduces the broadcast redundancy in flooding. It outperforms the baseline protocol in terms of radio-on time while attaining a high reliability of over 99.50%, and an average end-to-end latency around two milliseconds in all experimental scenarios.

- **Concurrent Transmission-based Collection**. Concurrent transmission is able to effectively accomplish a reliable and energy-efficient flooding in low-power wireless networks. With multiple nodes exploiting a receive-and-forward scheme, this technique works effectively in flooding-based network, i.e., in one-to-many scenarios. However, for data collection in WSNs, application-level scheduling has to be introduced. We propose *Packet-in-Packet* (PiP), an energy-efficient paradigm requiring no application-level scheduling for timely data collections in low-power

WSNs. PiP builds on concurrent transmissions, exploiting constructive interference and the capture effect to achieve high reliability and low latency. Moreover, PiP equips a packet concatenation capability to gather single-hop information in a best-effort manner. As a result, PiP reduces significantly the collection duration and thereby the energy consumption. We further compare PiP with a state-of-the-art protocol by extensive experiments in FlockLab. Experimental results show PiP highly reduces collection time (in terms of number of rounds) and achieves a good performance in terms of high reliability of approximately 98.7% and high energy efficiency in all experimental scenarios in the real-world testbed.

- **Application-oriented Adaptation**. We propose a novel strategy, referred to as the *Lifetime Planning* (LP) for achieving best-effort QoS in WSNs, while reaching an adequate lifetime required to complete the assigned task simultaneously. The core idea is to sidestep lifetime maximization strategies where sensor nodes continue functioning even after their fulfillment of the required tasks. We deliberately bound the operational lifetime to the expected task lifetime. As a result, residual energy can be spent throughout the entire task lifetime for enhancing performance. An analytical QoS model is engineered to validate the trade-offs among various application-level metrics. Lifetime planning is based on design-time knowledge, and thus, estimates boundaries of different metrics. During run-time, the controllable low-level parameters are tuned by a proactive adaptation mechanism to further adjust to different environmental conditions. To demonstrate the effectiveness of our design, we conduct an intensive simulation-based evaluation using an office monitoring scenario as a case study. The scenario is designed within the simulator Cooja in Contiki OS. Furthermore, we examine the profit of our strategy and compare it to two state-of-the-art protocols. Experimental results show that lifetime planning is able to achieve an expected network lifetime but improves reliability and reduces latency.

Overall, the guarantee and optimization of end-to-end QoS is the key to the adoption of WSNs in real-world applications in order to accomplish IoT and CPS within Industry 4.0. This thesis explains that opportunistic routing, machine learning, concurrent transmission, and application-oriented adaptation provide enabling technologies for this purpose. The evaluations based on real-world testbeds as well as simulations validate the excellent performances of the dedicated designs of the WSN protocols. This anticipates a bright future for the full employment of WSNs in the forthcoming industries.

Zusammenfassung

Smart Homes, Smart Cities und intelligente Verkehrsmittel sind heute Infrastruktursysteme, die Menschen verbinden und unseren Alltag zunehmend verändern. Solche Systeme werden gemeinhin als das Internet of Things (IoT) oder Cyber-Physical Systems (CPS) definiert, bei denen die gesamte physikalische Welt eng mit Sensoren, Maschinen und vernetzten eingebetteten Geräten verbunden ist. In so einem komplexen dynamischen System sind Geräte miteinander verbunden, um Messungen zu erfassen, wertvolle Informationen zu verarbeiten und Daten in verteilten Netzwerken auszutauschen. Ein drahtloses Sensornetzwerk (WSN) ist ein Netzwerk, das eine große Anzahl von Sensorknoten umfasst. Jeder Knoten ist mit Sensoren ausgestattet, um physikalische Größen wie Licht, Wärme, Druck und Feuchtigkeit zu erfassen, um hier nur einige aufzuführen. Sensorknoten kommunizieren drahtlos miteinander, so dass WSNs im Vergleich zu drahtgebundenen Lösungen eine einfachere Bereitstellung und bessere Flexibilität der Geräte bieten. Aufgrund der jüngsten Fortschritte in den Bereichen Elektronik, Vernetzung und Informationsverarbeitung haben sich WSNs zu einer vielversprechenden Technologie für IoT und CPS entwickelt. Im letzten Jahrzehnt wurde eine breite Palette von WSN-basierten Anwendungen konzipiert und implementiert, wie z.B. Umweltüberwachungs-, Vorhersage- und Gesundheitssysteme.

In den meisten dieser Anwendungen arbeiten kleine, kostengünstige Niederspannungssensoren als ein Netzwerk zusammen. Insbesondere müssen solche Netze aufgrund begrenzter Ressourcen energieeffizient sein und ein ausreichendes Quality of Services (QoS), wie Zuverlässigkeit, Datenaktualität, Energieeffizienz und Sicherheit, gewährleisten können. Die QoS-Bereitstellung in WSNs ist jedoch eine äußerst anspruchsvolle Aufgabe, da diese QoS-Anforderungen in der Regel im Gegensatz zueinanderstehen. Dies liegt zum Beispiel an den begrenzten Ressourcen der eingesetzten Geräte in Bezug auf Rechenleistung, Speicherkapazität, Energiebudget, Multihop-Kommunikation über verlustbehaftete drahtlose Kanäle und unvorhersehbare und dynamische Veränderungen in (oft widrigen) Umgebungen.

In jüngster Zeit ist das Interesse an Industrie 4.0 gestiegen. Allgemein bekannt als die vierte industrielle Revolution, bezieht sich dieser Begriff auf den derzeitigen Trend der Automatisierung und des Datenaustauschs in der Produktionstechnik, der auf die Förderung des so genannten “Smart Factory”-Konzepts abzielt, welches aus den Hightech-Strategie der Bundesregierung stammt. In modularen Smart Factories überwachen IoT und CPS physikalische Prozesse, kommunizieren und kooperieren miteinander und mit Menschen in Echtzeit. Für eine Reihe von WSN-basierten Anwendungen, insbesondere für einsatzkritische Anwendungen unter widrigen Bedingungen, z.B. in intelligent vernetzten Gebäuden, überwachung und Steuerung in der Industrie, kooperativem Fahren usw., ist die Aufrechterhaltung einer durchgängigen QoS-Güte über die gesamte Lebensdauer des Netzwerks dringend erforderlich. Das heißt, jede Beeinträchtigung der Leistung im Zeitverlauf in WSN-basierten Anwendungen sollte soweit wie möglich ver-

mieden werden. Diese Dissertation stellt sich zur Aufgabe, die Gestaltung von WSN-Anwendungen und Protokollen für die Realisierung von IoT und CPS in Industrie 4.0 zu erleichtern. Insbesondere bieten wir eine Reihe von Lösungen zur Verbesserung der End-to-End-QoS in der WSN-Kommunikation. Mit dieser Dissertation werden vier neue Techniken für WSNs, ihre analytischen Studien, praktische Implementierungen, sowie reale Bewertungen und entsprechende Analysen vorgestellt. Insbesondere machen wir vier Hauptbeiträge in dieser Arbeit:

- **Multichannel Opportunistic Routing**. WSNs teilen sich das 2.4-GHz-ISM-Band in der Praxis mit einer Reihe von kabellosen übertragungsverfahren, wie WiFi und Bluetooth. Dieser Umstand und externe Störungen durch elektrische Geräte, wie z.B. Mikrowellen, beeinträchtigen die Zuverlässigkeit vieler Routing-Protokolle in WSNs. Mehrkanalige Kommunikationsstrategien ermöglichen Routing-Protokolle, die bei Störungen zuverlässig sind. Wir bieten ein robustes, zuverlässiges und energieeffizientes Multichannel Opportunistic Routing (MOR) für WSNs. MOR verwendet sowohl Opportunistic Routing als auch Opportunistic Multichannel-Hopping-Strategien, um die Robustheit des Netzwerks gegenüber Störungen zu verbessern. Die Kombination von Opportunistic Routing und Multichannel Hopping erlaubt es MOR, nicht nur die räumlichen und zeitlichen Unterschiede, wie beim traditionellen Opportunistic Routing in WSNs zu nutzen, sondern auch die verschiedenen Frequenzen. Wir implementieren MOR in Contiki und führen umfangreiche Experimente im FlockLab-Testbed durch. Unter Störeinflüssen bietet MOR ein End-to-End packet delivery ratio (PDR) von mehr als 98%, während andere Protokolle wie beispielsweise das Opportunistic IPv6-Routingprotokoll für Low-Power- und Lossy-Netzwerke (ORPL) ein PDR von nur 25% erreicht. Zusätzlich bleibt die Auslastungsrate unter 2% und die Latenzzeit beträgt weniger als zwei Sekunden. In störungsfreien Szenarien erreicht MOR eine Leistung ähnlich unserem Baseline-Protokoll ORPL, mit nur ca. 0,3% Erhöhung der Auslastungsrate.

- **Machine Learning-based Flooding**. Concurrent Transmission, ein neuartiges Kommunikationsschema, hat sich bewährt, um eine zuverlässige und energieeffiziente Flooding in drahtlosen Netzwerken zu erreichen. Da mehrere Knoten ein Empfangs- und Weiterleitungsschema im Netzwerk nutzen, führt diese Technik zwangsläufig zu einer Kommunikationsredundanz und damit zu einem erhöhten Energieverbrauch der Knoten. Wir schlagen vor, dass Less is More (LiM), ein energieeffizientes Floodingprotokoll für drahtlose Sensornetze, verwendet wird. LiM baut auf gleichzeitigen übertragungen auf und nutzt konstruktive Interferenzen und den Capture-Effekt, um eine hohe Zuverlässigkeit und geringe Latenz zu erreichen. Darüber hinaus stattet sich LiM mit einer maschinellen Lernfähigkeit aus, um Redundanzen zunehmend zu reduzieren und gleichzeitig eine hohe Zuverlässigkeit aufrechtzuerhalten. Dadurch kann LiM die Einschaltzeit und damit den Energieverbrauch signifikant reduzieren. Wir vergleichen LiM mit unserem Baseline-Protokoll Glossy durch umfangreiche Experimente im 30-Knoten FlockLab-Testbed. Experimentelle Ergebnisse zeigen, dass LiM die Broadcast-Redundanz bei Flooding stark reduziert. Es übertrifft das Baseline-Protokoll in

Bezug auf die Radio-On-Zeit und erreicht gleichzeitig eine hohe Zuverlässigkeit von über $99,50\%$ und eine durchschnittliche End-to-End-Latenz von rund zwei Millisekunden in allen experimentellen Szenarien.

- **Concurrent Transmission-based Collection**. Concurrent Transmission ist in der Lage, ein zuverlässiges und energieeffizientes Flooding in kabellosen Low-Power Netzwerken effektiv durchzuführen. Da mehrere Knoten ein Empfangs- und Weiterleitungsschema nutzen, funktioniert diese Technik effektiv in Flooding-basierten Netzwerken, d.h. in One-to-Many-Szenarien. Für die Datenerfassung in WSNs muss jedoch die Planung auf Anwendungsebene eingeführt werden. Wir schlagen Packet-in-Packet (PiP) vor, ein energieeffizientes Schema, welches keine Planung auf Anwendungsebene für zeitnahe Datenerhebungen in Low-Power-WSNs erfordert. PiP baut auf Concurrent Transmission auf und nutzt konstruktive Interferenzen und den Capture-Effekt, um eine hohe Zuverlässigkeit und geringe Latenz zu erreichen. Darüber hinaus ist PiP in der Lage Pakete zu verbinden, um Single-Hop-Informationen auf einfachste Weise zu sammeln. Dadurch reduziert PiP die Dauer der Datenerfassung und damit den Energieverbrauch deutlich. Ein Vergleich von PIP mit einem hochmodernen Protokoll durch umfangreiche Experimente in FlockLab zeigt, dass PiP die Erfassungszeit (in Bezug auf die Anzahl der Runden) stark verkürzt und eine gute Leistung in Bezug auf hohe Zuverlässigkeit von ca. $98,7\%$ und hohe Energieeffizienz in allen experimentellen Szenarien im realen Testbed erreicht.

- **Application-oriented Adaptation**. Um die für die gleichzeitige Erfüllung der gestellten Aufgabe erforderliche bestmögliche QoS in WSNs und gleichzeitig eine angemessene Lebensdauer zu erreichen, schlagen wir eine neuartige Strategie vor, die als Lifetime Planning (LP) bezeichnet wird. Die Kernidee ist es, Strategien zur Maximierung der Lebenserwartung zu umgehen, bei denen Sensorknoten auch nach Erfüllung der erforderlichen Aufgaben weiter funktionieren. Die Betriebsdauer wird bewusst an die erwartete Lebensdauer der Aufgabe gebunden. Dadurch kann die Restenergie über die gesamte Lebensdauer der Aufgabe zur Leistungssteigerung genutzt werden. Ein analytisches QoS-Modell wird entwickelt, um die Trade-offs zwischen verschiedenen Metriken auf Anwendungsebene zu validieren. Die Planung der Lebenserwartung basiert auf dem Wissen zur Entwicklungszeit und schätzt so die Grenzen verschiedener Kennzahlen. Während der Laufzeit werden die steuerbaren Low-Level-Parameter durch einen proaktiven Anpassungsmechanismus abgestimmt. Dadurch können sie sich weiter an unterschiedliche Umgebungsbedingungen anpassen. Um die Effektivität dieses Designs zu demonstrieren, wird eine intensive simulationsbasierte Auswertung anhand eines Office Monitoring Szenarios als Fallstudie durchgeführt. Das Szenario ist innerhalb des Simulators Cooja in Contiki OS konzipiert. Darüber hinaus untersuchen wir den Nutzen unserer Strategie und vergleichen ihn mit zwei modernen Protokollen. Experimentelle Ergebnisse zeigen, dass Lifetime Planning eine erwartete Netzwerklebensdauer erreichen kann und gleichzeitig die Zuverlässigkeit verbessert und die Latenzzeit reduziert.

Insgesamt ist die Sicherstellung und Optimierung von End-to-End-QoS der Schlüssel zur Einführung von WSNs in realen Anwendungen, um IoT und CPS innerhalb von

Industrie 4.0 zu erreichen. Diese Arbeit erklärt, dass Opportunistic Routing, Machine Learning, Concurrent Transmission und anwendungsorientierte Anpassung grundlegende Technologien für diesen Zweck bereitstellen. Die auf realen Testbeds basierenden Auswertungen sowie Simulationen bestätigen die hervorragenden Leistungen der dedizierten Designs der WSN-Protokolle. Dies lässt eine vielversprechende Zukunft für die vollständige Integration von WSNs in den kommenden Industrien erwarten.

Acknowledgments

It is a wonderful journal.
It seems to finally end here;
Yet, there is no finale on the way of exploration.

It is not only a great challenge, but also a great pleasure to thank all those who have given me the opportunity, support, and time to work on this thesis.

Firstly, I would like to thank my supervisor Prof. Dr.-Ing Oliver Theel for consistently giving me the biggest support, motivation, and advice during my doctoral research. From the first day in his research group on, he has been a great source of inspiration while providing me with independence, patience, and motivation for my research interests. It has been a great honor for me to work together with him and to learn from his approach and experience in the research. Moreover, I thank Prof. Dr. Oliver Kramer for being my second supervisor and for his encouraging comments for my work, especially in the field of machine learning. I would give the biggest thanks to Prof. Dr. Ernst-Rüdiger Olderog for the solid support and motivation of my research since the first day I have started working in SCARE project. I, hereby, acknowledge the great thanks to the German Research Foundation (DFG GRK 1765) for funding our Research Training Group. I also thank all the professors in SCARE project for numerous fruitful discussions.

Besides, this work is based on the results of collaborations with several excellent researchers. It has been a great honor to work with Prof. Dr. Olaf Landsiedel. With his strong support, I had the chance to conduct a research visit within his group at Chalmers Institute of Technology, Sweden. I have benefited a lot from plentiful discussions and an inspiring working atmosphere in his group. I am grateful to Alex Yuan Gao for many inspiring Skype discussions, paper reviewing, and his invaluable advice in the field of machine learning. I am profoundly thankful to Xiaoyuan Ma for the great collaboration that we have conducted: 1st place and 3rd place respectively in EWSN Dependability Competition 2018 and 2019. I can never forget the countless late nights in which we have strived for conducting ideas, coding, and experiments.

I would like to take the opportunity to thank all current and past colleagues of the System Software and Distributed System Group and the Research Training Group SCARE for providing me with such a creative and friendly atmosphere that has made my study one of the most unforgettable experience. Additionally, many thanks to Ira Wemper, Meike Burke, and Katrin Müller, who have helped me with many practical arrangements.

Last but not least, I would like to give the greatest thanks to my family, to whom I dedicate this thesis, for their continuous support and priceless love.

Dedicated to my family...

From
Peilin Zhang
Oldenburg, Germany
15.10.2019

Contents

1 Introduction

This chapter provides a brief introduction to this thesis. Section 1.1 gives the background of key research topics in this work: Wireless Sensor Network (WSN) and Quality of Service (QoS). Next, Section 1.2 demonstrates the current challenges in terms of QoS provision in WSNs. After that, Section 1.3 and Section 1.4 present the research goals and the general contributions of this work, respectively. Finally, Section 1.5 demonstrates the outline of this thesis.

1.1. Background

This section gives the background of wireless sensor networks and quality of service, respectively.

1.1.1. Wireless Sensor Networks

In the 1950s, the Sound Surveillance System (SOSUS) [Whi05] was developed by the United States military to detect and track Soviet submarines. SOSUS consisted of thousands of underwater acoustic sensors widely distributed in the Atlantic and Pacific ocean. The sensing technology is still available, albeit serving more peaceful functions of monitoring undersea wildlife and volcanic activity. SOSUS was the first wireless network that bore an resemblance to a modern WSN. From then on, the WSN technology began to be rapidly developing and applied in an increasing number of applications such as air quality monitoring, forest fire detection, natural disaster prevention, weather stations and structural monitoring to name but a few. Promoted by the technology giants of that time in the world, such as IBM and Bell Labs, the WSN technology started to be used in industrial applications such as power distribution, waste treatment, specialized factory automation and so forth. Over the past decades, along with the rapid developments of

Integrated Circuits (ICs), Micro-Electro-Mechanical Systems (MEMSs) and low-power wireless communication technology, WSN technology was increasingly progressing and bore tremendous potential thanks to the vision of ubiquitous computing [Wei93]. This technology performed as a promising role in a large number of cutting-edge real-world applications, ranging from healthcare, industry, agriculture, to military.

Today, as Moore's law [Sch97] continues, connected things are everywhere — people have become so used to being always connected to the Internet. Sensor nodes now have become not only tiny and cheap but also low-powered and addressable devices, that also boosts the development of the Internet of Things (IoT) [Ash09] and Cyber-Physical Systems (CPS) [Lee08]. Industrial giants such as Ericsson AB and Cisco Systems, Inc. predict a growing connectivity and project 29 billions devices to be connected by the year of 2022 [AB15]. If this trend lives up to the predictions, then a large number of device is being connected to each other and to the Internet.

In smart city scenarios, hundreds or thousands of distributed battery-powered sensor nodes are connected and they communicate wirelessly with each other. Industry 4.0 [KHHW13] is commonly known as the 4th industrial revolution, refers to the current trend of automation and data exchange in manufacturing technologies, that aims to foster the so-called "smart factory" concept, which originates from high-technology strategies of the German government. Endorsed by industrial companies such as Robert Bosch GmbH, Siemens AG, and SAP SE, it aims to enhance the automation of factories with WSNs and connections to cloud services, so as to, for example, improve logistics management, predict failures, trigger predictive maintenance procedures automatically [Gil16].

In general, a WSN is a wireless network that is built of a number of (wireless) devices, namely sensor nodes. Generally, each sensor node is made of several parts: a radio transceiver with an antenna, a micro-controller, an electronic circuit to interface with various sensors such as temperature, humidity and pressure sensors, and an energy source i.e., a battery or an energy harvesting source. Due the limited energy source, a sensor node is often designed as a "low-power" (consumption) device so as to achieve a longer lifetime for WSNs.

Normally, all the communication in WSN is taken place between *source* and *destination* via multiple hops. In a data collection scenario, one or multiple sensor nodes act as a sink node (sometimes also referred to as "gateway") and other nodes generate, transmit, and reply the information to the sink node. Moreover, in a data dissemination scenario, one or multiple sensor nodes work as a initiator, which normally triggers an information propagation, and other nodes receive the information from the initiator and reply it to the destination. All these nodes cooperate to fulfill various tasks so as to provide an acceptable level of QoS to the applications. However, the tiny size and low cost of sensor nodes result in corresponding resource constraints such as limited energy, memory, computational speed, communication range, and communication bandwidth. These resource constraints further result in the characteristics of WSNs such as low-power of the sensor nodes, multihop communications, and lossy radio links. Therefore, owing to the resource-constrained nature of sensor nodes, providing satisfactory application-level QoS is extremely challenging, especially in such low-power, multihop, and lossy wireless networks.

1.1.2. Quality of Service

QoS is the description of the overall performance of a service such as a computer network or a telephone network. Defined by the International Telecommunication Union (ITU), it explains the totality of characteristics of a telecommunication service that bear on its ability to satisfy stated and implied needs of the user of the service [Gro08]. In order to quantitatively measure and describe QoS, a number of related aspects of the network service are considered as indicators of the service, for instance, reliability, timeliness, energy efficiency, and security to name but a few. Traditional QoS metrics, such as reliability, timeliness, and energy efficiency, also apply to WSNs, but their importance diverges from legacy communication networks. For example, the reliability is particularly significant in WSNs. A data packet in computer networks is routed via highly reliable wire, while a data packet in WSNs is forwarded via lossy wireless links by single or multiple hops. On each wireless link, the Packet Error Rate (PER), for instance, can vary a lot, thereby decreasing the end-to-end reliability greatly.

Achieving a best-effort performance is the main goal of the design of WSN protocols. There are a number of QoS metrics (shown in Figure 1.1) in WSNs, which are most critical for the performance of WSNs, such as reliability, timeliness, energy efficiency, security, et cetera [ABK$^+$09]. Moreover, QoS metrics are used to measure the performance of WSNs and presents a practical expression of QoS trade-offs in WSN protocols. Therefore, a selected set of QoS metrics are explained in detail in the following paragraphs.

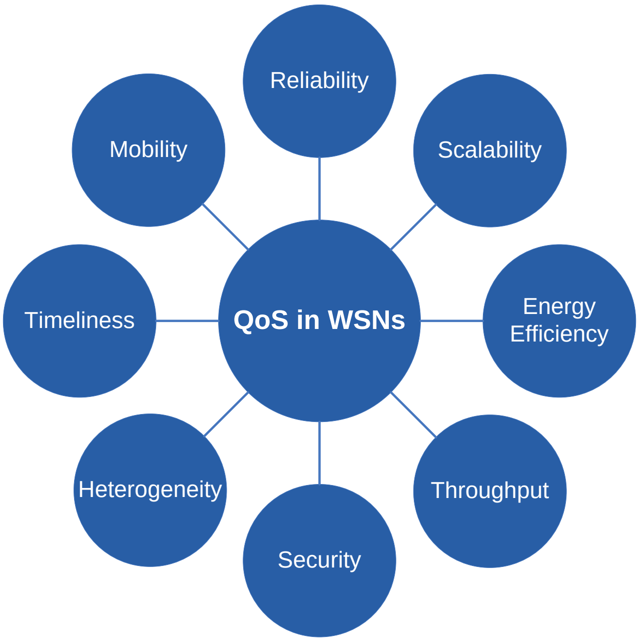


Figure 1.1.: QoS performance metrics in WSNs.

- **Reliability**: Reliability refers to the ability of a system or a component that performs its required functions under predefined conditions for a specified period of time. Since WSNs may be extremely difficult or even impossible to be physically accessed again once they are deployed [För16], this performance metric is especially important. Usually in WSNs, sensor nodes are expected to be alive as long as possible. However, on the one hand, WSN hardware (e.g., sensors) should perform well under the potential harsh environmental conditions, such as vibration, mechanical impacts, temperature, humidity, and moisture, as well as the interference from Radio Frequency (RF) and Cross-Technology Interference (CTI) [GAKS11]. On the other hand, the resources of the sensor node limit the capability of the applications in WSNs. Even worse, the multihop nature of the communication even deteriorates this situation. As a consequence, considering reliability becomes a must in the design process of WSNs in order to overcome the difficulties of these adverse conditions, thus mitigating maintenance actions and improving application usability.

- **Timeliness**: Timeliness represents the timing behavior of an application in terms of computations and communications. It encompasses the timing issues such as message transmission delay, task execution time. A number of WSN applications might impose to finish specific tasks within a certain time limit (i.e., deadline) which are usually referred to as "real-time" applications requiring real-time computations and communications. For instance, in some mission-critical WSN applications, there might be a task to monitor and detect a certain event (e.g., power outage in smart grids), and to transmit warning information to a remote sink within at most several seconds. Such type of time-critical CPS requires an adaptation in the normal computing and networking concepts [Lee06], and given that the computing entities closely interact with their environment, timeliness is of increasing importance [SAL+03].

- **Energy Efficiency**: Energy efficiency can be defined as the ratio of workload done to the amount of energy consumed. From the advent of the WSN paradigm on, it has been a major focus of research in the community since most of WSN nodes rely on small batteries with a restricted energy budget. Consequently, a large number of techniques for WSNs have been proposed to improve the efficiency so as to maximize the lifetime of battery-powered sensor nodes. These techniques aim at energy conservation, that reduces energy consumption through a reduction in the workload while still fulfilling the application task. In addition, energy harvesting is technique that extracts energy from the ambient environment and converts it into consumable electrical energy. Note, that in this thesis we mainly focus on the battery-powered WSN applications, and the energy harvesting techniques are out of the scope in this thesis.

- **Mobility**: Basically, mobility refers to the geographical change of the entity in time, such as the physical movement of sensor nodes, while logical mobility refers to the dynamic changes in the network topology due to adding or removing entities [ABK+09]. In WSNs, if some nodes are likely to move physically or logically relatively to each other, then mobility can be a crucial issue. Supporting mobility in WSNs significantly increases the capabilities of the network, i.e., to repair or extend the network connectivity [LBD+05], to balance energy consumption [JSS05], to adapt to dynamic stimulus changes, or to improve the lifetime of WSNs with mobile sink node [YX10]. Therefore, mobility support in WSNs is also a rather heterogeneous and challenging aspect.

- **Security**: Security is one of the key aspects for WSNs' acceptance outside the research community, given the interactive and pervasive nature of WSNs [ABK$^+$09]. As a standalone research topic, security in WSNs is a more difficult long-term problem than it is today in desktop and enterprise computing. Generally, tiny sensor nodes have many limitations in terms of energy consumption, computation, storage, and communication capabilities, that lead to severe constraints on security solutions that can be applied in WSNs. Furthermore, a WSN uses a wireless medium to communicate with each other. Therefore, an adversary with a simple radio transceiver can easily eavesdrop, manipulate, inject as well as modify data packets in a wireless network. Meanwhile, security in WSNs is getting increasingly crucial with the rapid increase of the software content of embedded wireless systems and networks. Note, that security in WSNs, as a research topic on it own, is out of interest in this thesis.

Generally, in this thesis, we consider three most fundamental QoS metrics, i.e., reliability, timeliness, and energy efficiency, while other metrics are out of scope of this thesis. Specifically, a set of relevant performance metrics in WSNs are used to measure the degree of satisfaction of the QoS, namely, the end-to-end Packet Delivery Ratio (PDR) for reliability, the end-to-end latency for timeliness, and the duty cycle or radio-on time for energy efficiency.

- **Packet Delivery Ratio**: PDR refers to the ratio of the number of packets that are successfully delivered to a destination over the number of packets that have been sent by the sender in an end-to-end communication. In most cases, PDR is used as a performance metric of a WSN to indicate the reliability of the communication protocol.

- **Latency**: Latency is the time elapsed from the application on the source node handing the packet to the Medium Access Control (MAC) layer until the packet arrives at the destination's application. It represents the timeliness of a packet generated and sent by the source (e.g., a sensor node) and received by the dedicated destination (e.g. a sink node). In general, latency indicates the end-to-end timeliness on the application level. Minimizing end-to-end latency in random access networks is one of the key goals of protocol design, especially for mission-critical WSN applications.

- **Duty Cycle**: In WSNs, low-power wireless sensor node must maintain strict power budgets to achieve years of lifetime. The wireless transceiver (i.e., radio) often has the highest power consumption among all components on a low-power wireless sensor node [Dun11]. Therefore, it is common to address the energy consumption of a radio with respect to the energy efficiency of a sensor node. The duty cycle is the portion of radio-on time over the total time. It is a hardware-independent indicator of power consumption and thus describes the energy efficiency of a protocol.

- **Radio-on Time**: Similar to the duty cycle, radio-on time is the time duration that the radio is active in one single duty cycle, including the time for listening, receiving, and transmitting. It expresses the power consumption of a radio straightforwardly and implies the energy consumption of a protocol.

1.2. Challenges

Particularly, the characteristics of WSNs differ from other traditional computer networks. Such a WSN either requires to sense environmental data from the surroundings and to forward the sensed data towards a more powerful node (i.e., a sink node), or obligates to distribute information from a single node to all the other nodes in the network. QoS provision in WSNs is an extremely challenging task due to the facts that WSN resources such as power source, processing power, and memory are highly constrained and QoS metrics are typically contradicting. Therefore, in the following, a number of selected significant challenges are discussed in detail.

- **Limited Resources**: Due to the characteristics of the cheap, tiny, and battery-powered WSN sensor nodes, there are inevitably a number of significant resource constraints in WSNs, namely, energy budget, communication bandwidth, computation capability, and transmission range.

- **Unreliable Communication Medium**: Generally, wireless sensor nodes communicate wirelessly with each other via low-power radio. Thus, this wireless medium is inherently less reliable. Meanwhile, the wireless links are lossy and occasionally affected by various environmental factors such as noise and interference (e.g., Cross-Technology Interference (CTI).

- **Redundant Data**: Since WSN sensor nodes are normally deployed densely in an area of interest, sensor nodes therefore generate a large amount of redundant data. While the redundancy benefits reliability and fault tolerance of the WSNs, they also cause a significant amount of energy wastage, thus, degrading the energy efficiency of the network.

- **Dynamic Network Topology**: Mostly, the topology of WSNs can get influenced dynamically due to lots of aspects, such as the physical movement of sensor nodes, link failures, and hardware defects. Self reorganizing of WSNs and being adaptable to such changes is a challenging issue. For instance, during run-time, new sensor nodes may be added and some may even die due to empty batteries. All of these factors may potentially dynamically change the network topology of WSNs. However, QoS should not be affected drastically due to the mobility, failure, addition or removal of sensor nodes. Managing dynamics requires QoS adaptation mechanisms to work in unpredictable conditions. That is, WSNs must be adaptive and flexible at run-time to all the dynamic changes.

- **Mixed Application and Traffic Patterns**: In the real world, diverse applications may need to share the same deployed WSNs, such as many-to-one (data collection), one-to-many (data dissemination), and many-to-many applications.

Besides, data can be both, periodic and aperiodic data generated by various source nodes. With respect to data flow, data from a large number of sensor nodes are collected by the sink node, while in some cases (e.g., data dissemination), data from the initiator are also required to flow to all other sensor nodes. Moreover, some sensor nodes may be used to create the measurements of physical variables in a periodic manner for the purpose of monitoring and control, e.g., temperature, humidity, pressure, vibration. Meanwhile, others may be deployed to detect critical events.

- **Conflicting QoS Metrics**: Additionally, QoS metrics in WSNs usually contradict with each other [HBT^{+}09]. For instance, improving the end-to-end reliability might affect the end-to-end timeliness and energy efficiency in WSNs. Also, there is a trade-off between energy and timeliness [SG09]: Optimizing latency may introduce a lot of overhead, such as more radio-on time, thus, increasing the energy consumption. Optimizing QoS trade-offs and mitigating QoS conflicts in WSNs are also challenging topics determining the performance of WSN applications.

1.3. Research Goals

The focus of the thesis is to provide, to guarantee, and to optimize QoS in low-power multihop WSNs with respect to achieving high reliability, high timeliness, and high energy efficiency in WSN communications while satisfying the application-level requirements. Thereby, the results of this thesis contribute to the realization of WSN applications for IoT and CPS in real world and then further bring the accomplishment of more advanced WSN applications closer towards the Industry 4.0.

The main goals of the thesis are to: (i) propose and implement novel communication algorithms and protocols to optimize multi-objective QoS in multihop low-power WSNs; (ii) validate and evaluate the effectiveness of the proposed algorithms and protocols in terms of several key QoS metrics, i.e., reliability, timeliness, and energy efficiency, through simulations and real-world experiments; and (iii) compare the performance of solutions to the existing state-of-the-art solutions and then analyze performance based on the QoS trade-offs.

1.4. Contributions

The contributions of the thesis are mainly in improving performance while satisfying the QoS requirements in low-power multihop WSNs, which is essential for the increasing adoptions of WSNs in IoT, CPS, and other mission-critical applications. Particularly, the contributions can be divided into four aspects according to the WSN protocol stack [KW07]: improving QoS by (i) exploiting multichannel opportunistic routing to resist the harsh conditions, (ii) using machine learning techniques to mitigate communication redundancy, (iii) applying concurrent transmission for reliable and timely data collection, and (iv) utilizing application-level adaptations on run-time dynamics together with design-time knowledge.

1.4.1. Multichannel Opportunistic Routing

We make the following contributions to mitigate the problem of harsh CTI in WSNs:

- We propose Multichannel Opportunistic Routing (MOR) [ZLT17] for duty-cycled multihop WSNs. By opportunistically exploiting temporal, spatial, and frequency diversities, MOR achieves good performance in both, interference-free and interfered conditions.
- As a basis for MOR, we introduce a lightweight channel-hopping strategy for asynchronous Low-Power Listening (LPL)-based MAC protocols. It guarantees a fast rendezvous between sensor nodes, where the sender and the receiver both opportunistically perform fast channel hopping in each active duty cycle.
- We implemented MOR in Contiki OS [DGV04], and evaluated the performance of the protocol in terms of end-to-end reliability, latency, and radio duty cycle in the FlockLab testbed [LFZ$^+$13b].

1.4.2. Machine Learning-based Flooding

We make the following contributions to the problem of data redundancy in flooding:

- We propose Less is More (LiM) [ZGT17, ZGT18], an energy-efficient flooding protocol with progressive learning ability for low-power multihop WSNs.
- As a basis for LiM, we explore and implemented a light-weight bandit-learning scheme to determine the number of broadcasts in each node. It guarantees a correct exploration of the “redundant” nodes and further conducts a progressive learning of the other nodes to greatly reduce broadcast redundancy.
- We implemented LiM in Contiki OS and conducted extensive experiments with various configurations in a 30-node real-world testbed (FlockLab), and evaluated the performance of LiM and compared to the baseline protocol in terms of end-to-end reliability, radio-on time, and latency.

1.4.3. Concurrent Transmission-based Collection

The main contributions in this part of the thesis are as follows so as to realize reliable and timely data collection:

- We propose a communication scheme — Packet-in-Packet (PiP) [ZMTW18a,ZMTW18b, MZTW20] — for reliable and timely data collection in low-power WSNs.
- We introduce a novel hardware operation — the so-called Power Amplifier (PA) operation to realize concurrent packet concatenation in PiP. It guarantees a natural scheduling in the MAC layer between single-hop neighboring nodes, where the senders opportunistically perform their in-packet concatenations in each active transmission round.
- We implemented PiP in Contiki OS and carry out extensive real-world experiments respectively in single-hop and multihop scenarios, and evaluated the performance of PiP and compare to the state-of-the-art in terms of end-to-end reliability, collection duration (time), and duty cycle in FlockLab.

1.4.4. Application-oriented Adaptation

In order to solve the QoS contradiction problem and to further realize application-oriented adaptations in WSNs, we make the following contributions:

- We propose a novel strategy named Lifetime Planning (LP) [AZT15, ZAT18] to improve performance of low-power WSNs.
- We introduce a light-weight hierarchical framework for self-adaptation, namely monitoring, analysis, planning, and execution.
- We investigated a QoS analytic model that maps low-level controllable parameters to application-level performance metrics, and we validated the resulting model by performing simulations in Contiki OS.
- We compared the impact of LP with other strategies, i.e., static heuristics and unplanned adaptations, using the network simulator Cooja [ÖDE+06].

Specifically, this work was a joint work with Dr.-Ing. Mohamed Abdelaal, which has been partly published in the doctoral dissertation [Abd16]. In this research, my main focus is the investigation of the analytic model and the practical implementation of the adaptive system in Contiki OS [ZAT18].

1.5. Outline

For easily understanding the proposed WSN protocols and algorithms in this thesis, the structure of the thesis is outlined in Figure 1.2. Detailed in the perspectives of WSN protocol stack [KW07], this thesis develops through a series of layers: physical (PHY) layer, MAC layer, network layer, and ends in application layer.

Chapter 2 focuses on the network layer and MAC layer. It introduces a Multichannel Opportunistic Routing scheme — MOR, that significantly improves the reliability and robustness against the cross-technology interference with only slight overhead of energy consumption. Following the state-of-the-art standard IEEE 802.15.4 [ISA11], MOR further equips opportunistic routing with a multichannel hopping scheme, thus, benefiting from both.

Chapter 3 details a machine learning strategy, called Less is More (LiM), for one-to-many communication (i.e. data dissemination) in WSNs, aiming at the MAC layer and the physical layer. LiM applies a multi-armed bandit algorithm to lessen data redundancy in concurrent transmissions. Therefore, it further improves energy efficiency in flooding, while maintaining a high reliability and a low latency.

Chapter 4 specifies a concurrent packet concatenation pattern, named Packet-in-Packet (PiP), for reliable and timely many-to-one communication (i.e. data collection) in WSNs, addressing on the MAC layer and the physical layer. PiP concatenates the packets on the receiver from different concurrent senders in the air by manipulating the power-amplifier operation in the radio. PiP greatly decreases the data collection time in the network, yet keeping up a high reliability and a low latency.

Chapter 5 concentrates on an application-oriented adaptation strategy — Life Planning (LP). In general, LP is based on both, design-time knowledge and run-time adaptation, in order to achieve a best-effort QoS performance while satisfying the expected

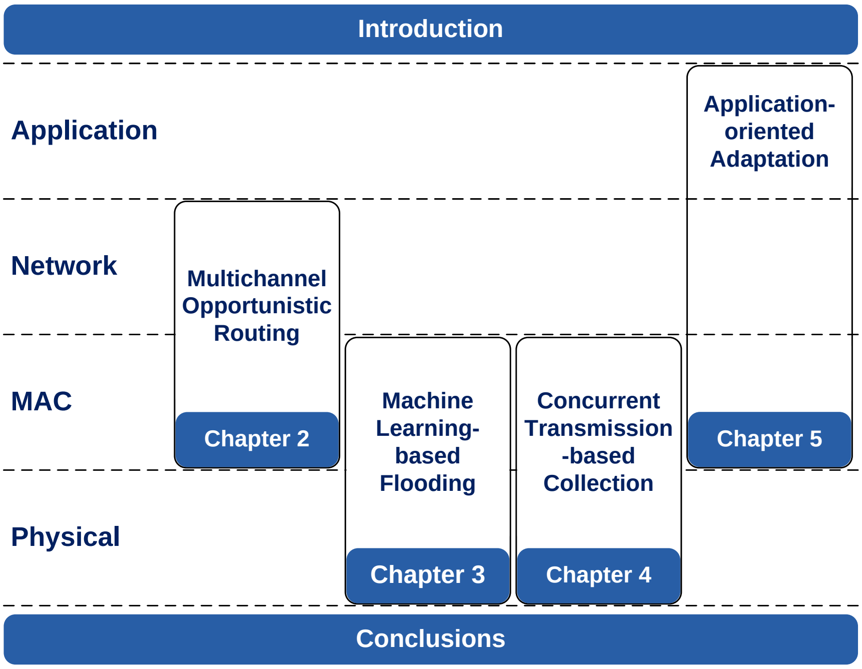

Introduction
Application
Network
MAC
Physical
Multichannel Opportunistic Routing
Chapter 2
Machine Learning-based Flooding
Chapter 3
Concurrent Transmission -based Collection
Chapter 4
Application-oriented Adaptation
Chapter 5
Conclusions

Figure 1.2.: Thesis outline based on WSN protocol stack.

task time. In this case, LP connects the application layer to the MAC layer so as to realize a cross-layer performance optimization.

Chapter 6 concludes the thesis by a summary of the contributions, prospects, and opportunities for the future.

Appendix provides more details in the regard of practical designs and implementations in competitions. In order to further evaluate and investigate our designed protocols in real-world cases, we have participated twice the Dependability Competition in the international conference on Embedded Wireless Systems and Networks (EWSN), respectively in 2018 and 2019. Basically, we have combined various designs, such as the channel hopping from MOR [ZLT17] and the concurrent transmission used in LiM [ZGT18] and PiP [ZMTW18b] . In the end, among all the teams from both academia and industry, we have successfully achieved the third place and the first place respectively in 2018 and 2019.

2
Multichannel Opportunistic Routing

2.1. Introduction

Over the past decades, WSNs began to play a significant role as an enabling technology in a large number of applications, including healthcare, industry and agriculture. Due to the limited number of radio channels in WSNs, sensor nodes share the 2.4 GHz Industrial, Scientific and Medical (ISM) band with each other, as well as with other wireless technologies such as WiFi, Bluetooth, and microwaves to name but a few. As a result, there exists not only internal interference within the network but also plenty of CTI [GAKS11]. For instance, under interference from WiFi devices and microwaves, the performance of the X-MAC protocol can degrade by over 50% [BVN$^+$11, BYAH06], resulting in high network latency and reduced reliability. Under these adverse conditions, the communication in WSNs, however, is obligated to maintain strong robustness and resilience to both, internal and external interference.

Multichannel hopping schemes in WSNs efficiently mitigate the interference, as shown by a number of existing approaches [SGJ08, KSC08, WMP09, BSL10, TSGJ11, IVHJH11, ANDIV14, MGC16]. By exploiting the frequency diversity, these approaches are able to improve reliability and robustness against internal interference within the network as well as external Cross-Technology Interference (CTI), e.g., caused by WiFi, Bluetooth, and microwave. Figure 2.1 demonstrates the basic idea of multichannel hopping in time-slotted WSNs, i.e. Time Slotted Channel Hopping (TSCH) [WPG15]. As shown in the figure, nodes in the network use three channels, namely, channel A, B, and C. In the first time slot, node 1 transmits a message to node 2 in channel C. Then in the second time slot, it continues to send a message to node 3 using channel A. Similar processes also apply to the other nodes. In this case, multichannel hopping scheme improve the channel utilization so as to improve overall dependability against interference.

Meanwhile, a number of challenges arise: As more channels are involved in the communication, the power consumption increases accordingly, e.g., due to channel allocation and switching. Furthermore, the Time-To-Rendezvous (TTR) between sender and receiver is another crucial factor when utilizing multiple channels in duty-cycled WSNs, that indirectly determines the end-to-end latency of the whole network. We argue in this chapter, that most state-of-the-art multichannel protocols for WSNs fail to provide the best-effort balance among reliability, latency, and power consumption.

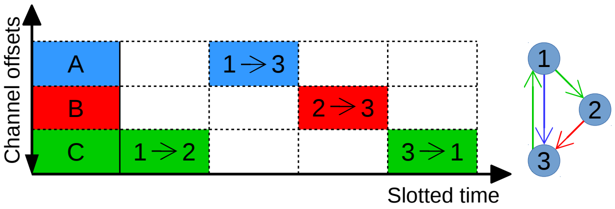


Figure 2.1.: Time-slotted multichannel hopping. Note, that this multichannel hopping is based on the synchronous MAC, where nodes are highly time-synchronized.

Opportunistic routing has drawn much attention from research communities because of its capability to improve the performance of wireless networks, for instance, in [BM04, RSMQ09, MTX$^+$11, LGDJ12, DLV13, GHG$^+$14]. Figure 2.2 depicts the general idea of opportunistic routing. A source node uses not only the reliable links but also the unreliable link, in order to deliver a packet to the destination. Opportunistic routing exploits the broadcast nature of the wireless channel and selects multiple potential candidates as next hop to forward data packets. Instead of relying on one "good" single path, opportunistic routing utilizes multiple paths to route data from source to destination. Consequently, it effectively improves reliability, reduces delay as well as power consumption,and highly increases resilience to wireless link dynamics. However, most approaches to opportunistic routing in low-power duty-cycled WSNs are limited to a single channel. As a result, their performance strongly deteriorates in presence of interference.

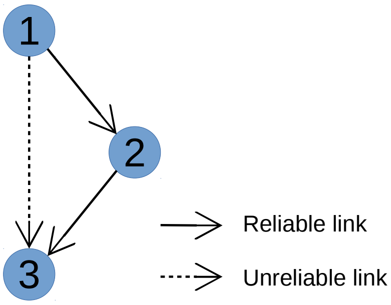


Figure 2.2.: Opportunistic routing. Node 1 reaches node 3 via node 2 on reliable links or direct on an unreliable link.

In this chapter, we propose *MOR*, a Multichannel Opportunistic Routing scheme for low-power duty-cycled multihop WSNs. Incorporated with opportunistic routing, MOR is able to effectively increase the end-to-end reliability and to reduce the end-to-end latency as well as the power consumption. Moreover, MOR empowers opportunistic routing *more* opportunistically on multiple channels. It fully takes advantage of frequency diversity to provide a satisfactory level of QoS (i.e., reliability and timeliness) and to maintain a best-effort resilience to dynamic interference in real-world scenarios. MOR trades a slight portion of energy for low-power-listening (LPL) on multiple channels, while improving reliability, minimizing latency and power consumption.

We implement MOR in Contiki OS [DGV04] and conduct extensive experiments in the 30-node testbed FlockLab [LFZ$^+$13b]. We compare MOR with selected state-of-the-art single-channel and multichannel protocols. Our evaluation shows that MOR effectively limits the impact of interference: Under interference, MOR provides an end-to-end PDR of more than 98%, while other protocols such as Opportunistic RPL (ORPL) [DLV13] achieve a PDR of merely 25%. Moreover, MOR's duty cycle settles below 2% for these settings and the average latency is less than two seconds. In interference-free scenarios, MOR achieves a performance similar to our baseline protocol ORPL with only an approximately 0.3% increment of duty cycle.

The remainder of this chapter is organized as follows. Section 2.2 discusses related work, with a focus on multichannel MAC and routing protocols in WSNs. Section 2.3 explains the basis of our proposed protocol and provides a brief overview of it. Section 2.4 details the design of MOR, followed by the performance evaluation elaborated in Section 2.5. Section 2.6 provides concluding remarks.

2.2. Related Work

Multichannel communication is essential to provide reliable communication under interference and is part of many standards such as Bluetooth and WirelessHART [Fun06]. In the domain of WSNs, multichannel communication helps to, for example, improve reliability, resilience to interference, throughput, and reduce latency [SGJ08, KSC08, WMP09, BSL10, TSGJ11, IVHJH11, ANDIV14, MGC16]. These approaches take advantage of location-specific knowledge of the wireless channel: its diversities in frequency, time, and space. As a result, these protocols ensure reliable, and robust co-existent wireless communication.

In the following, we group approaches to multichannel routing into two classes, according to the MAC layer they are based on: multichannel routing for (i) synchronous, and (ii) asynchronous protocols. In synchronous MAC protocols, sensor nodes maintain a tight time synchronization and the wake-ups of each node are commonly scheduled to when neighboring nodes wake up. Asynchronous MAC protocols, on the other hand, establish communication between two nodes that are on different active/sleep schedules.

2.2.1. Synchronous MAC Protocols

Y-MAC [KSC08] is an energy-efficient multichannel MAC protocol for WSNs. It is a Time Division Multiple Access (TDMA)-based MAC protocol, thus, requiring accurate time synchronization. In Y-MAC, sensor nodes exchange the remaining time in the current time slot to synchronize their starting points for the next slot. A light-weight channel hopping mechanism is implemented in Y-MAC that enables multiple nodes to communicate simultaneously on multiple channels. This mechanism increases network throughput and reduces latency. Experimental results demonstrate that Y-MAC is able to achieve a low duty cycle under light traffic conditions and ensures an energy-efficient transmission of bursty messages under high traffic conditions.

MC-LMAC [IVHJH11] is a multichannel MAC protocol, designed to maximize the throughput of WSNs by coordinating transmissions over multiple channels. In MC-LMAC, time is slotted and each node is assigned the control over a time slot to transmit on a particular channel. Hence, MC-LMAC takes advantage of both, scheduled and multichannel communication, which can minimize communication collisions. Therefore, it overcomes the increased contention and interference on the limited bandwidth and improves the channel utilization. Simulation results show that MC-LMAC obtains significant bandwidth utilization and high throughput while ensuring an energy-efficient operation.

Moreover, in RPL networks, Orchestra [DANLW15] provides a TSCH [WPG15]. In Orchestra, nodes autonomously compute their own local schedules and maintain the schedules allocated to a particular traffic plane, i.e., application, routing, and MAC. Nodes update their local schedules automatically as the topology evolves. Orchestra (re)computes local schedules without signaling overhead. Instead, it only relies on the existing network stack information to maintain the schedules. This scheme allows Orchestra to handle non-deterministic network traffic while exploiting the robustness of TSCH. Extensive evaluations in simulation and in two different testbeds demonstrate the practicality of Orchestra and its ability to consistently achieve a very high delivery ratio in the order of 99.99%, while obtaining a balance between latency and energy consumption.

2.2.2. Asynchronous MAC Protocols

MuChMAC [BSL10] is a low-overhead multichannel MAC protocol, which combines TDMA with asynchronous MAC techniques and requires no coordination or tight synchronization between nodes. MuChMAC is a receiver-initiated multichannel MAC protocol. In every time slot, each node switches its radio channel according to a pre-defined channel assignment, which is based on the parallel rendezvous principle [SWM07]. The channel is calculated based on a node's ID and the current slot number following a pseudo-random hopping sequence. Experiments in a testbed demonstrate the applicability of MuChMAC and show that it can efficiently operate multichannel communication without coordination or synchronization overhead.

Chrysso [IWL11] is a multichannel protocol for data collection. In Chrysso, sensor nodes are organized in parent-children groups, where each parent-children group uses two channels: one for packet transmissions and one for receptions. When a node in Chrysso detects interference on one channel, both, parent nodes and child nodes, switch to another channel based on a channel hopping policy. The authors of Chrysso show its reliability under severe WiFi interference and jamming.

Efficient Multichannel MAC (EM-MAC) [TSGJ11] introduces mechanisms for adaptive receiver-initiated multichannel rendezvous and predictive wake-up scheduling. To achieve high energy efficiency, EM-MAC enables a sender to predict both, the receiver's transmission channel and wake-up time. In EM-MAC, a node is able to select channels dynamically based on the channel conditions it senses. In this matter, it avoids utilizing channels that are heavily loaded or are undesirable because of interference or jamming.

In their evaluation, the authors show that it can achieve a low duty cycle, low latency, and high PDR under interference.

MiCMAC [ANDIV14] is a multichannel extension of ContikiMAC based on LPL. MiCMAC performs a sender-initiated channel hopping. Namely, in every wake-up period, the channel is determined by the sender according to a pseudo-random sequence. Similar to the phase-lock mechanism in ContikiMAC, a channel-lock mechanism is integrated in MiCMAC to shorten the rendezvous time between the sender and the receiver on various communication channels. Experiments show that MiCMAC improves the performance of the network in terms of reliability, latency, duty cycle, and resilience to external interference.

Oppcast [MGC16] is a multichannel LPP-based data collection protocol. It opportunistically utilizes both, broadcast and unicast transmissions, to maintain good network performance in the presence of interference. Oppcast selects and uses three good channels, i.e., channel 15, 25, and 26 out of all 16 ZigBee channels. In Oppcast, both, receivers and senders, simultaneously perform channel hopping with a round-robin principle. Based on opportunistic routing, Oppcast takes advantage of the spatial diversity. It utilizes the hop count as a routing metric to optimize performance. Experiments in a large-scale testbed show that Oppcast consistently maintains high reliability, low latency, and low duty cycle in several urban scenarios.

2.2.3. Summary

Multichannel routing is essential for reliable communication under interference and it has received significant attention in the recent years. Nonetheless, most approaches focus on traditional unicast routing. In this chapter, we argue that opportunistic routing, such as ORPL and ORW, opens new design options for reliable, multichannel communication. Thus, in MOR, we extend the concept of opportunistic routing to the frequency domain: The first node that (i) wakes up on the rendezvous channel, (ii) successfully receives the packet, and (iii) provides routing progress, acknowledges and acts as a forwarder. We show in our experimental evaluation that MOR significantly improves robustness in presence of interference when compared to other state-of-the-art protocols.

2.3. Overview

In this section, we provide the required background on both, channel hopping and opportunistic routing in low-power WSNs. Next, we introduce the basic concepts of MOR.

2.3.1. Channel Hopping Strategies in WSNs

Regarding the selection of channels, channel hopping strategies fall into two categories: “whitelisting” and “blind hopping” [WMP09]. In whitelisting, neighboring nodes agree on which channels to use at what point in time for their communication. In blind channel hopping, nodes do not know which channels their neighboring nodes use at what point in time. To establish communication, nodes uniformly hop over all utilized channels, i.e., up to 16 radio channels in IEEE 802.15.4 [ISA11].

Practically, there are two types of channel allocations in multichannel communication, i.e., static channel allocation and dynamic channel allocation [HXS+13]. Depending on the scenarios, dynamic channel allocation can be more effective if the interference condition is changing dynamically over time. It, however, often performs complex rendezvous algorithms, thus, resulting in non-trivial communication computing overhead. To balance the performance and the computing overhead of the sensor node, MOR chooses to use static channel allocation.

The main goal of any channel hopping scheme is to increase robustness towards interference. We observe three approaches of channel hopping strategies in wireless communication: fast channel hopping, slow channel hopping, and hybrid channel hopping [HXS+13]. Fast channel hopping switches to a new channel in each time slot. Fast channel hopping is used in a number of applications and standards in order to improve secrecy and to make the system more robust against jamming or interference. For example, Bluetooth and WirelessHART [Fun06] employ fast channel hopping. Meanwhile, this approach increases the overhead for a packet transmission, i.e., frequent channel switching makes a device consume energy faster than others. Slow channel hopping stays for multiple continuous time slots on a single channel before switching. Compared to fast channel hopping, slow channel hopping generates less latency when two devices need to rendezvous on a common channel. Hybrid channel hopping combines both fast and slow channel hopping, where fast channel hopping improves the robustness towards interference and slow hopping allows for fast rendezvous.

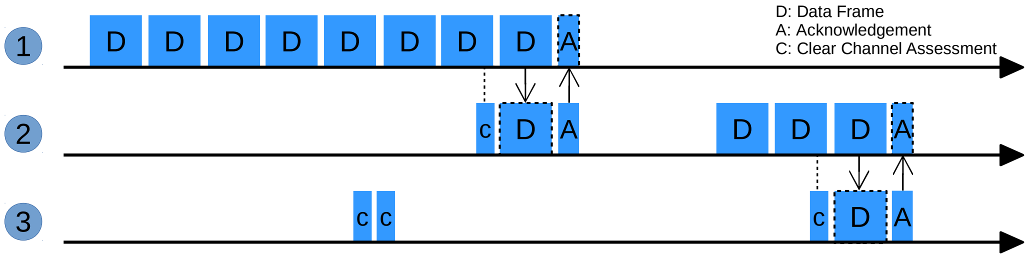
Generally, MOR exploits hybrid channel hopping scheme. Duty-cycled sensor nodes perform fast channel hopping to ensure robustness towards the interference. That is, they switch to a new channel in a short time slot so as to avoid keeping using a interfered channel for rather long time. An always-on node (i.e., the sink), which does not go to sleep mode at all, employs the slow channel hopping scheme to guarantee the fast rendezvous of the last-hop neighbors. In this case, whenever there comes a packet from last-hop neighbors to the sink, the sink can capture and receive it in at least one "good" channel, simply because that the last-hop neighbors hop to a new channel more frequently than the sink does.

2.3.2. Opportunistic Routing in WSNs

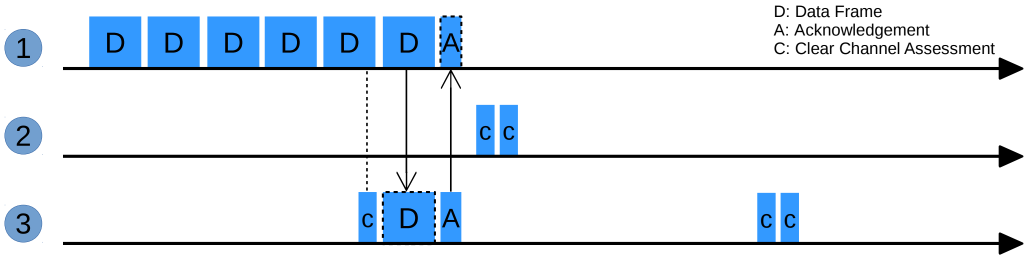
Approaches to opportunistic routing in duty-cycled WSNs differ from traditional unicast, where packets are addressed to one specific neighbor. In traditional unicast, as shown in Figure 2.3(a), if node 1 has a data frame to send, then it keeps sending a data frame via a reliable link. A receiver, on the other hand, wakes up and detects the data by a Clear Channel Assessment (CCA). In IEEE 802.15.4, the MAC layer employs the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. CCA is used in the physical layer to determine the channel occupancy [ISA11]. Generally, a CCA performs Energy Detection (ED), or Carrier Sense (CS), or a combination of both. CCA aims to report a busy channel upon detecting any energy above a preset ED threshold. Afterwards, the receiver sends an acknowledgment back to the sender, i.e. node 1. Node 2 then sends the data frame to the destination, i.e., node 3. Node 3 wakes up, detects the data by a CCA, receives the data frame, and sends back an acknowledgment. In this case, the routing set is built based on the link quality. That means one node selects its next-hop forwarder from the neighboring nodes based on the link quality.

Opportunistic Routing for Wireless sensor networks (ORW) [LGDJ12] is an opportunistic routing scheme for duty-cycled WSNs. ORW uses *anycast* addressing a one-to-any-one scenario where data packets are routed to any single member of a group of potential receivers. Consequently, data packets in ORW are forwarded by one of the neighboring nodes which (i) wakes up first, (ii) successfully receives the packet, and (iii) provides routing progress. As shown in Figure 2.3(b), in LPL-anycast, node 1 repeats sending the data frame regardless of the link quality. The next-hop node, who wakes up earlier, detects the data frame using a CCA, receives the data, and acknowledges the sender.

ORW is able to sufficiently reduce delay and energy consumption and improves the resilience to wireless link dynamics. Furthermore, Opportunistic RPL (ORPL) integrates the concepts of opportunistic routing with RPL [Win12], the standard protocol for low-power and lossy Internet Protocol version 6 (IPv6)-based networks. ORPL provides any-to-any and on-demand traffic. Both ORW and ORPL utilize the Expected Duty Cycles (EDCs) [LGDJ12] as the routing metric. When a node is selecting its next-hop forwarder from its neighboring nodes, EDCs of the neighboring nodes are used as a metric to compare. This allows the node to select the set of neighboring nodes in different hops that provide sufficient routing progress. Experimental results from testbeds show that ORW and ORPL outperform the state-of-the-art solutions including RPL and the Collection Tree Protocol (CTP) [GSC09] in terms of latency, power consumption, robustness, and scalability.



(a) LPL-based unicast. One sender unicasts the data packet over a single channel to a neighbor based on a routing metric, e.g., link quality.



(b) LPL-based anycast. One sender anycasts the data packet over a single channel to the neighbor who wakes up earliest thus, reducing the delay.

Figure 2.3.: Low-power listening-based unicast and anycast using the same topology as the one in Figure 2.2.

2.3.3. MOR in a Nutshell

MOR extends opportunistic routing with multichannel hopping to combine their key advantages: low latency and high energy efficiency of opportunistic routing with strong robustness to interference of multichannel hopping. Thus, MOR inherits the spatial and temporal diversities of opportunistic routing and additionally exploits the frequency diversity of multichannel routing.

MOR builds on ORPL: It employs the EDC routing metric [LGDJ12] and the integration with RPL [Win12]. Additionally, unlike a number of synchronous MAC protocols for WSNs, e.g., [KSC08] and [IVHJH11], MOR is based on asynchronous Low-Power Listening (LPL). It, thus, does not lead to additional synchronization overhead within the network and efficiently operates its channel hopping without coordination overhead. Moreover, MOR does not only transmit opportunistically, but also selects channels opportunistically: For each listening and (re)transmission of the underlying MAC layer, MOR utilizes a new channel. For example, while in ORPL it takes multiple transmissions of the MAC on a single channel until one neighboring node wakes up and successfully receives the packet, MOR does each of these (re)transmissions on a different channel.

Overall, MOR extends the concept of opportunistic routing to the frequency domain. That is, in MOR, the first node that wakes up on the rendezvous channel and successfully receives the packet, acts as a forwarder and, thus, provides the routing progress. We show in our experimental evaluations that MOR significantly improves robustness in the presence of interference when compared to other state-of-the-art protocols. In addition, we show that the duty cycle of MOR is only approximately 0.3% higher when compared to our baseline protocol ORPL in interference-free scenarios.

2.4. Design of MOR

In this section, we detail the design of MOR. We discuss the allocation of channels, opportunistic channel rendezvous of senders and receivers, and implementation aspects for integrating multichannel hopping scheme into opportunistic routing.

2.4.1. Channel Allocation

In MOR, we utilize a subset of the 16 IEEE 802.15.4 channels. To determine this subset of channels, we execute a number of sets of experiments in FlockLab [LFZ$^+$13b], respectively on 16 individual ZigBee channels. We use the standard protocol ContikiMAC/RPL in Contiki OS [DGV04]. Note, that these experiments aim to help evaluating the diversity of each channel, instead of the performance of the protocol.

Figure 2.4 reveals the link qualities of the 16 channels in FlockLab. As shown in the figure, there are only eight "good" channels with an end-to-end PDR higher than 50%: channel 26, 25, 20, 15, 21, 22, 19, 14 (sorted in order, with best quality first).

Employing all these eight channel shown in Figure 2.4 might not be advantageous, while considering the trade-off between the number of channels utilized and the computing overhead: The more channels are utilized, the more time is required for the nodes to rendezvous, because every receiving node needs to scan the "whitelist" of channels.

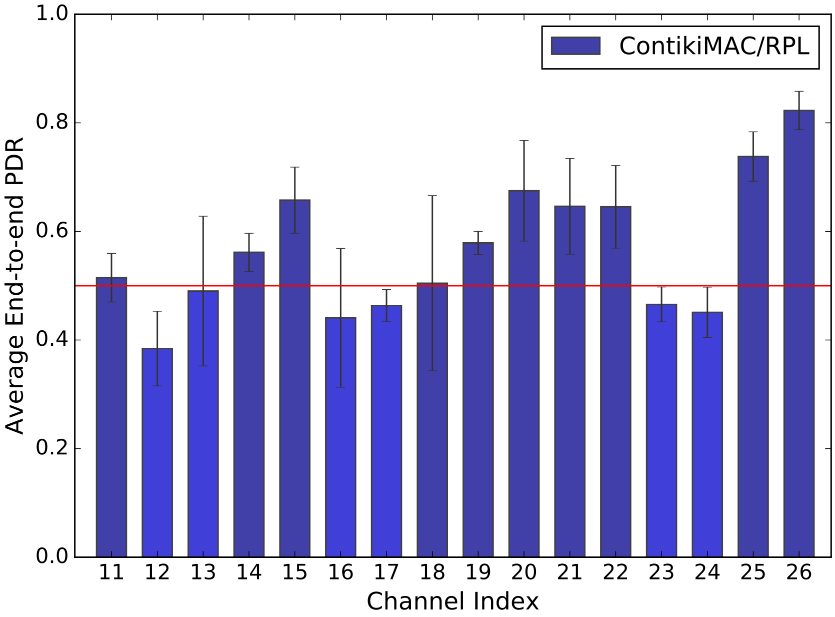


Figure 2.4.: Average end-to-end reliability of 16 ZigBee channels evaluated with ContikiMAC in FlockLab. The red line indicates the PDR of 50%.

Besides, since low-power listening is required on each channel to exploit frequency diversity, the total energy consumption for channel sensing increases correspondingly as the number of channel increases. Finally, using some "bad" channels does not help to improve the reliability but leads to high latency and energy consumption.

Therefore, MOR chooses to assign *three* "best" ZigBee channels by default, for instance, channel 15, 25, and 26. These channels are orthogonal to WiFi channels in most scenarios: Even with a fully deployed WiFi network (IEEE 802.11 channel 1, 6, and 11), there are still a few channels that are free from the interference (i.e., channel 15, 20, 25, and 26) in [ANDIV14] and [MGC16]. Furthermore, the effectiveness of the number of channels will be discussed in the following subsection and evaluated by experiments in Section 2.5.

2.4.2. Channel Rendezvous

In this section, we discuss the rendezvous scheme of MOR. MOR operates without synchronization: A node does not know when and on which channel a neighboring node wakes up. As a result, a node opportunistically transmits repeatedly and on different channels until its packet has been received. The sender and the receiver rendezvous until they select the same channel at the same point in time. As a result, it provides an upper bound of the TTR. Formally, the *rotation closure property* in an asynchronous channel hopping system can be defined as follows by Bian et al. [BP11]:

$$\forall k,\ l \in [0,\ T-1],\ C(rotate(\mu, k),\ rotate(\nu, l)) \geq m, \tag{2.1}$$

where the positive natural number m is the degree of channel overlaps in the system and T represents the number of time slots. μ and ν denote two different channel hopping sequences, respectively. $C(\mu, \nu)$ denotes the number of rendezvous channels between two channel hopping sequences μ and ν. A channel hopping sequence in T time slots (μ of T) can be represented as a set of channel: $\mu = \{\mu_0, \mu_1, ..., \mu_{T-1}\}$. Furthermore, $rotate(\mu, k)$ denotes a cyclic rotation of channel hopping sequence μ by k time slots, i.e.,

$$rotate(\mu, k) = \nu_j \mid \nu_j = \mu_{j+k} \bmod T,\ j \in [0,\ T-1], \tag{2.2}$$

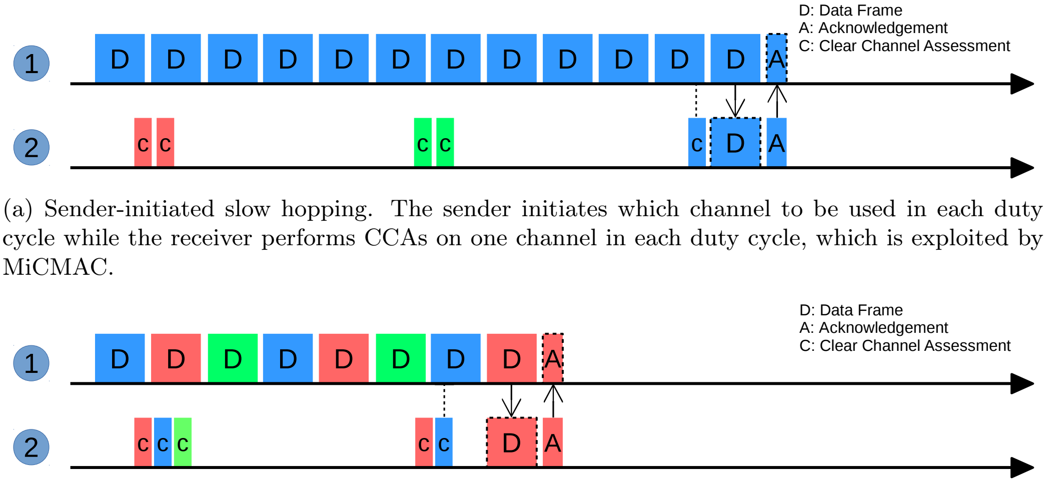
where j, k are non-negative integers. Generally, if two channel hopping sequences satisfy Equation 2.1, then two nodes with these two sequences μ and ν can rendezvous on at least m distinct channels. For instance, given $T = 3$, $\mu = \{1, 2, 3\}$ and $\nu = \{3, 2, 1\}$, there exist $k = 0$ and $l = 0$ satisfying Equation 2.1, i.e., $C(rotate(\mu, 0),\ rotate(\nu, 0)) \geq 1$. Specifically, it means that by using these two sequences μ and ν, two nodes can rendezvous at least on one channel, i.e., in the second time slot on channel 2 in this example.

If $T = 4$, $\mu = \{1, 2, 3, 4\}$ and $\nu = \{4, 3, 2, 1\}$, it renders $C(rotate(\mu, 0),\ rotate(\nu, 0)) \geq 1$ being false. Therefore, $T = 3$ guarantees that $C(\mu, \nu) \geq 1$ is always true regardless of the value of non-negative integers k and l. That means, these two sequences rendezvous at least on one channel no matter how each individual sequence rotates. For the detailed proof, we refer the reader to the paper of Lin et al. [LLCL11]. In this case, utilizing three hopping channels in MOR is appropriate for maximizing the probability of fast rendezvous. Consequently, an upper bound of the rendezvous time can be provided as five (i.e., $2T - 1$) time slots by these channel hopping sequences. Based on this, we construct our channel rendezvous sequences in a *round-robin* fashion, as $\mu = \{15, 25,26\}$ and $\nu = \{15,26, 25\}$ for transmitting and receiving, respectively.

2.4.3. Fast Channel Hopping

Generally, there are two types of sender-initiated channel hopping: slow hopping and fast hopping as shown in Figure 2.5. Sender-initiated communication means that the communications are initiated by the sender: Whenever sender has a data frame to send, it starts keeping transmitting the packet until an Acknowledgment (ACK) has been received or a time-out occurs. Before each reception, the receiver performs two consecutive CCAs to check whether the medium is occupied. If the receiver detects a busy channel, then it prepares to receive the data frame and sends back an ACK. Otherwise, it switches to a different channel and repeats the CCA process. Figure 2.5(a) shows the idea of slow hoping. That is, the sender stays on a channel for the whole communication period. The receiver switches to another channel every time it wakes up and keeps on this channel for two consecutive CCAs. Differently, in fast hopping, the sender switches its channel for every single data frame and so does the receiver for different CCAs, as depicted in Figure 2.5(b)

MOR employs fast channel hopping for both senders and receivers as shown in Figure 2.5(b). Thus, for each (re)transmission in the MAC layer, senders in MOR transmit on a different channel, allowing it to quickly iterate over the channels in use. Receivers, upon duty-cycled wake-up, sense the multiple channels. As a result, MOR ensures that senders and receivers rendezvous quickly with this iteration.



(a) Sender-initiated slow hopping. The sender initiates which channel to be used in each duty cycle while the receiver performs CCAs on one channel in each duty cycle, which is exploited by MiCMAC.

(b) Sender-initiated and receiver-initiated fast hopping. The sender and the receiver both perform fast channel hopping in each time slot instead of the whole duty cycle, thus, significantly reducing the TTR.

Figure 2.5.: Multichannel hopping schemes: slow hopping and fast hopping.

The transmission sequence of a sender consists of three steps: (i) sending, (ii) waiting for an ACK, if there is an ACK received, the transmission is completed, otherwise (iii) the sender switches to the next channel based on the rendezvous sequence. As soon as it gets an ACK from the receiver, the sender enters to a low-power mode. Alternatively, it keeps the transmission of the packet until a time-out occurs. If not successfully transmitted, this packet will be re-transmitted in the next active period after a pre-defined time. In a word, the sender performs fast channel hopping in each individual time slot, and channels are chosen according to the rendezvous sequence.

Generally, when a receiver wakes up, it first senses the channel activity using a CCA, and then hops to the next channel if it does not detect anything. If the receiver detects a data packet on a particular channel, it prepares itself with the correct channel ready for the next time slot and then goes into a fast-sleep mode. Furthermore, when a receiving node wakes up in each duty cycle, it performs LPL by a number of consecutive channel sensing, one per channel used by MOR. We set the number of CCAs M to the number of channels N plus 1, i.e., $M = N + 1$. In MOR, the default number of channels is $N = 3$, thus M is set to 4 by default. This increases the probability of early rendezvous and the randomness of the channel selection: Receivers use a different starting index of the hopping sequence to sense the channel every time they wake up. It exploits the frequency diversity more opportunistically.

Overall, MOR trades more energy consumption of channel sensing and switching than slow channel hopping strategies and single-channel approaches. However, this portion of extra energy highly improves the robustness to interference by exploiting frequency diversity, thus, enabling MOR to attain a better network performance. We illustrate this further in Section 2.5.

2.4.4. Implementation Aspects

In this section, we discuss the implementation details of MOR.

2.4.4.1. MAC and Routing Layer

We implement the multichannel extensions of MOR in the MAC layer. While MOR is tailored to opportunistic routing and ORPL in particular, our modifications are transparent to any routing protocol. To ensure a fair comparison, we retain the modifications of the ContikiMAC [Dun11] version of ORPL [DLV13], i.e., 63 milliseconds (ms) guard time for phase locking. This also includes five retransmission attempts with exponential backoffs of the MAC layer, i.e., on top of the LPL of ContikiMAC.

Moreover, we choose EDC as the routing metric of MOR. Since ORPL uses only one channel for communication, EDC calculation in ORPL is not appropriate for MOR. Thus, we disable the minimal "penalty" function for updating the EDC value in the MAC layer. Because in MOR, if one particular channel is busy, the sending node simply switches to another one, which shall not have a negative effect for the routing metric EDC.

2.4.4.2. Hit and Hop: Carrier Sense

With the fast multichannel hopping strategies of MOR, a node always switches to another channel after a time slot, i.e., the time period to complete one packet transmission in MOR. It is, as a result, more challenging for a receiver to not only rendezvous on the same channel with the sender, but also detect the data packet. Thus, it requires the protocol to ensure that the receiver can firstly rendezvous on the same channel with the sender and secondly detects the data packet and successfully receives it.

As shown in Figure 2.5(b), every time it rendezvous with the sender on a particular channel, the receiver prepares itself for the reception of data on the next-hop channel that is determined by the sender's round-robin hopping sequence. Basically, the receiver follows the process of "*Hit* → *Hop* → *Sleep* → *Listen*", whereas (i) "*Hit*" stands for the receiver successfully rendezvousing with the sender on a common channel, (ii) "*Hop*" means that it then hops to the next channel and enters a fast "*Sleep*" mode, (iii) afterwards, before the start of next time slot in the sender, the receiver wakes up again, *listens* to the channel and receives the packet. Therefore, while waking up after "*Sleep*" in this procedure, the node can always receive the packet on the assigned channel. Throughout this work, we assume that two consecutive channels in the hopping sequence are not always interfered at exactly the same time point. This is assumed, because the channels utilized by MOR are not influenced exactly by WiFi at the same time in most real-world cases.

2.4.4.3. Slow Hopping of the Sink

By default, in both, unicast and opportunistic routing protocols including CTP, RPL, ORW, and ORPL, the sink node is not duty-cycled. MOR reflects this always-on nature of the sink in its channel-hopping strategy. Thus, the sink executes the slow hopping strategy. For example, if MOR uses three channels, then the sink node hops to another channel every 1/3 of the duration of the duty cycle. As a result, MOR helps the last-hop nodes to attain a rendezvous with the sink within the duration of one duty cycle. In addition, this shortens the strobing time and effectively reduces the power consumption of the last-hop nodes.

2.4.5. Summary

In summary, MOR effectively extends the concept of opportunistic routing to the frequency domain: In MOR, the first node that wakes up on the rendezvous channel and successfully receives the packet, acknowledges and acts as a forwarder. In next section, we show that in our experiments MOR drastically improves robustness in presence of interference when compared to other state-of-the-art protocols. Further, while in interference-free scenarios, MOR trades only a small portion of the duty cycle to achieve a similar performance as our baseline protocol ORPL.

2.5. Performance Evaluation

In this section, we perform an extensive experimental evaluation of MOR. We compare MOR to the state-of-the-art, including ContikiMAC/RPL, ORPL, MiCMAC/RPL [ANDIV14], and Oppcast [MGC16], respectively, in scenarios with and without interference. We focus on three key metrics to evaluate the performance of the protocols: reliability, latency, and energy efficiency. To calculate PDR, we log both, the packets sent by each node and the ones received by the sink. We measure latency based on the time-stamps of the serial port outputs from the source nodes and the sink node. For energy efficiency, we measure duty cycle by using the software-based energy profiler [DOTH07] of Contiki OS.

2.5.1. Methodology

We use the FlockLab testbed [LFZ^{+}13b] for our experimental evaluation. FlockLab is a second generation WSN testbed developed and run by the Computer Engineering and Networks Laboratory at the Swiss Federal Institute of Technology ETH Zürich in Switzerland. It features a co-located and tightly coupled observer platform together with every sensor node. This enables creating testing scenarios which are much more realistic and hence more complex than with standard bus-based testbed architectures. The testbed consists of 30 observers and one server which are spread across one level of the ETZ-building at ETH Zürich and the surrounding rooftops, thus, supporting joint indoor and outdoor testing of WSN applications. For more details, we refer readers to the paper [LFZ^{+}13b] and website[1] of FlockLab. The topology of the testbed is shown in Figure 2.6. We use node 16, a node on the edge of the network, as the network sink to expand the network diameter. In all the experiments, we use the maximum transmission power of the CC2420 radio chip [Tex06], i.e., 0 dBm. We run a periodic data collection application, where each sensor node transmits a 64-byte payload as User Datagram Protocol (UDP) datagram over IPv6 over Low power Wireless Personal Area

[1] `https://gitlab.ethz.ch/tec/public/flocklab/wikis/home`

Networks (6LoWPAN) to the sink node with an average interval of two minutes. The default wake-up frequency of all protocols is 2 Hz. We use JamLab [BVN$^+$11] to generate interference in a deterministic and reproducible manner.

For each experimental setting, we perform five independent runs and each run lasts 60 minutes. Experiments with three interfering nodes are executed for 90 minutes. All the results are averaged over these five runs, and the standard deviations are shown by error bars. Following recent trends, such as the 2016 EWSN Dependability Competition[2], we include the whole experimental run in our evaluation, including the starting phase of network initialization. The phase of initialization is more energy-consuming since senor nodes have to communicate more frequently so as to discover their neighbors and build their own lists of neighbors. However, considering this phase is realistic with respect to the protocol performance in real-world scenarios. This is also justified by the short initialization time of ORPL and other opportunistic routing protocols, as we show in our evaluation.

2.5.2. Protocols

We compare MOR to RPL, ORPL, MiCMAC, and Oppcast [MGC16], four state-of-the-art routing protocols, which are all implemented in Contiki OS [DGV04].

- ContikiMAC/RPL: RPL is a unicast, tree-based data collection protocol. It uses the Expected Transmission Count (ETX) routing metric (by default) and operates over a single radio channel. We run RPL on ContikiMAC [DEFT11], a default power-saving MAC in Contiki OS. It is duty-cycled and employs LPL with optional phase lock. In our experiments, we choose channel 26 to limit external interference and obtain predictable performance.

- ContikiMAC/ORPL: ORPL is an extension of RPL and employs opportunistic routing over a single channel. As for ORPL, we use channel 26 if not noted elsewise. ORPL utilizes EDC as routing metric.

- MiCMAC/RPL: To exploit the frequency diversity, MiCMAC extends ContikiMAC with a multichannel hopping scheme. By default, MiCMAC runs RPL as routing protocol.

- Oppcast: Oppcast is an opportunistic, multichannel data collection protocol based on Low-Power Probing (LPP). It applies a combination of spatial and frequency diversities. It selects three best ZigBee channels out of 16 for channel hopping and it considers the hop count as the routing metric. MOR is different from Oppcast in utilizing different MAC-layer techniques and different routing metrics.

2.5.3. Cost of Multichannel Routing

As first step, we compare MOR to single-channel routing protocols, ContikiMAC/RPL and ContikiMAC/ORPL. Our goal is to measure the overhead of MOR compared to the traditional, single-channel routing protocols in scenarios without interference. In this

[2] `ewsn2016.tugraz.at`

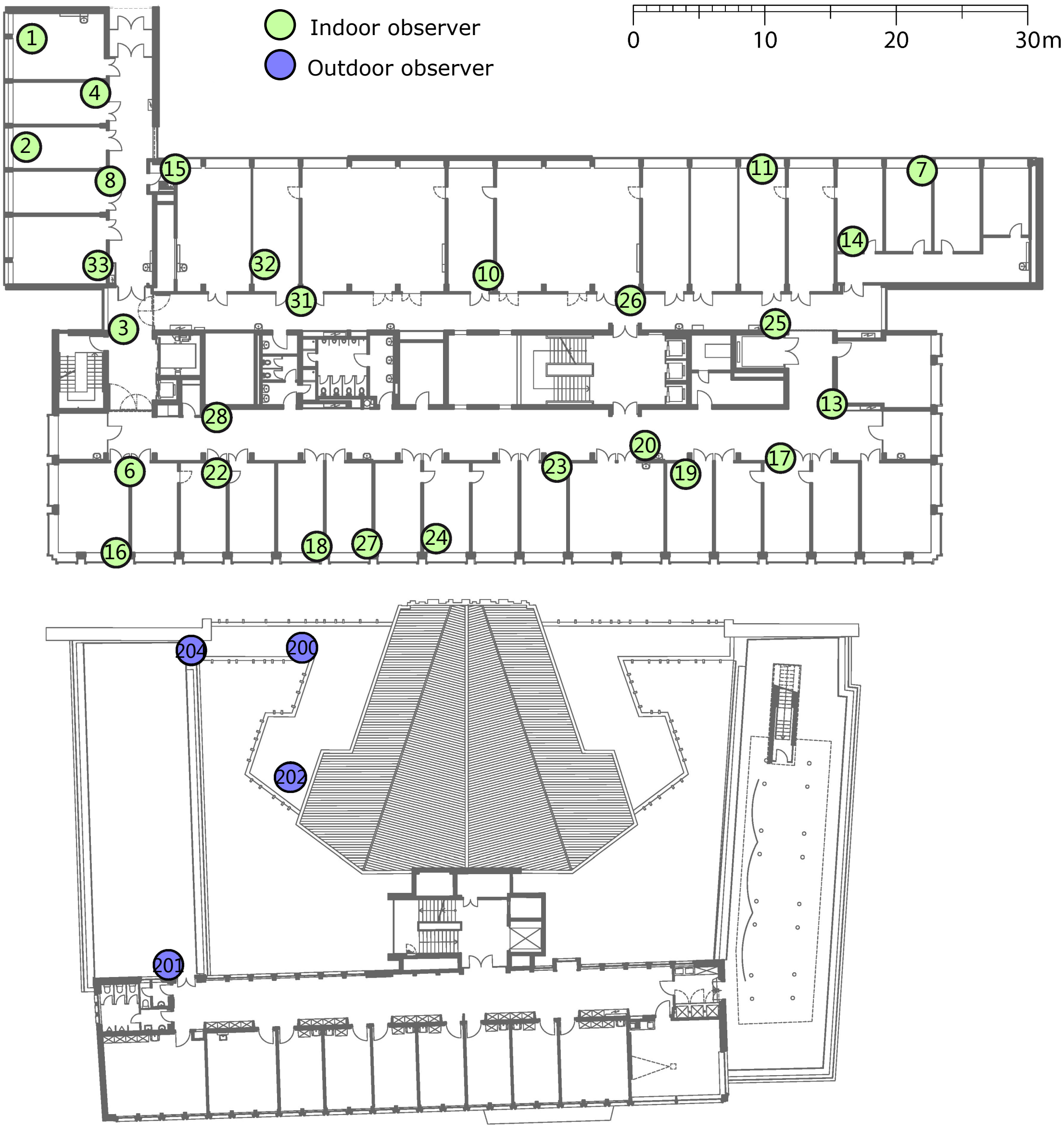
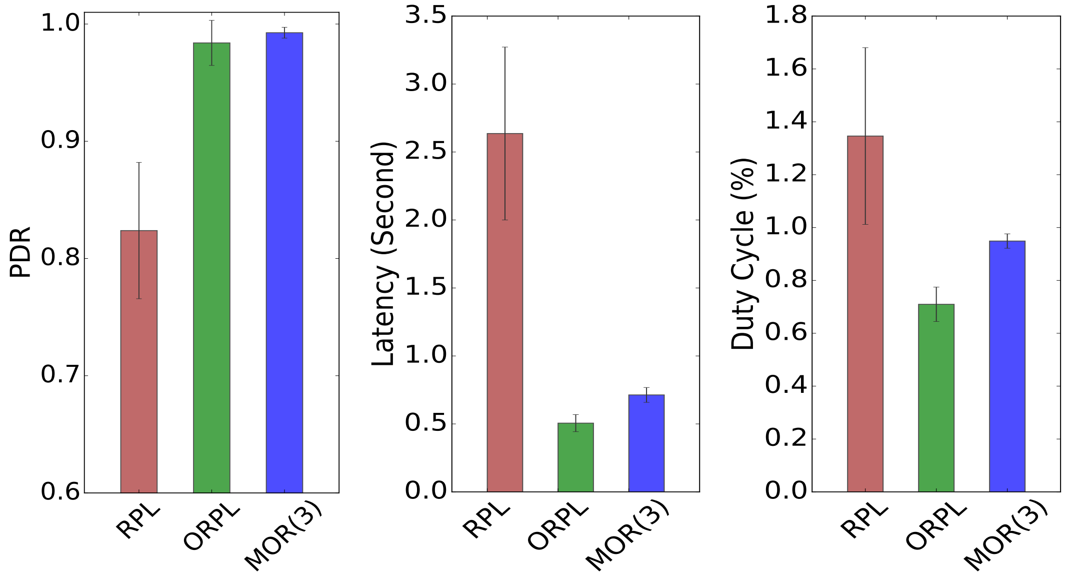


Figure 2.6.: The deployment of sensor nodes in FlockLab.

scenario, RPL and ORPL operate on channel 26 and MOR utilizes three channels: 26, 25, and 15. We show that the multichannel operation of MOR leads to a reasonable overhead when compared to both RPL and ORPL. Figure 2.7 presents the results of these three protocols, with respect to PDR, latency, and duty cycle.

Taking latency and duty cycle into account, ORPL outperforms both MOR and RPL protocols, with MOR outperforming RPL. MOR inherits key advantages of ORPL such as the high PDR. Namely, MOR achieves an average PDR of 99.26%, slightly better than the one of ORPL (98.41%).

On the other hand, utilizing more communication channels, MOR inevitably suffers an approximate 0.21 second longer average end-to-end latency than ORPL. Also, its duty



(a) PDR

(b) Latency

(c) Duty Cycle

Figure 2.7.: Effectiveness of opportunistic routing. In interference-free scenarios, ORPL performs slightly better than MOR in terms of latency and duty cycle, since it operates over a single channel, leading to less overhead than in MOR. In terms of PDR, MOR achieves similar performance to ORPL. ORPL and MOR outperform RPL on all three metrics.

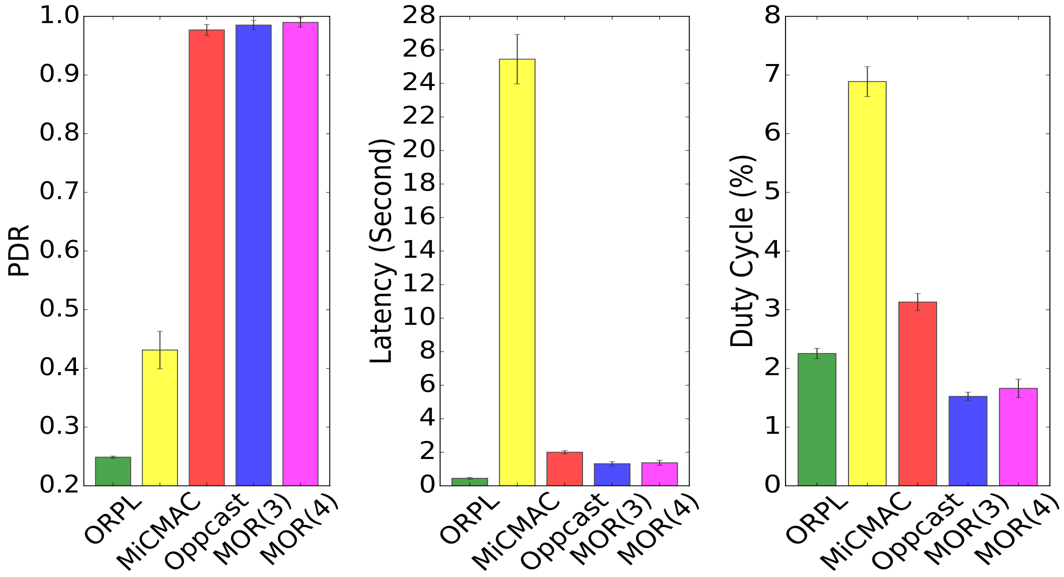
cycle increases from roughly 0.70% to 0.95% when compared to ORPL. These results show that multichannel routing in MOR does not come for free, but with a — as we argue — reasonable overhead. Later, we show that this overhead becomes negligible, once we add interference or switch away from channel 26.

2.5.4. Benefits of Multichannel Routing

In this section, we evaluate the performance of MOR and related approaches under interference. We compare MOR with our baseline protocol ORPL and two other state-of-the-art multichannel protocols for WSNs, namely MiCMAC/RPL and Oppcast. By default, MiCMAC/RPL utilizes four channels. In contrast, Oppcast utilizes three channels — at least the version provided by the authors to us. Thus, to ensure a fair comparison, we depict results for two configurations of MOR: with three and four channels, denoted as MOR(3) and MOR(4). We use channels 26, 25, and 15 and channels 26, 25, 20, and 15, respectively, in MOR(3) and MOR(4).

We use JamLab [BVN$^+$11] to introduce external interference in the testbed. In this setup, a JamLab node acts as a jamming (or an interference) node. For the experiments in this section, we select node 22 in FlockLab as the jamming node, which is close to the sink node 16. We hence expect node 22 to strongly influence the performance of the different protocols. To ensure fairness, we switch on the source of interference after the start of each experiment so that each protocol has time to complete the initial setup of its routing tables. We use a transmit power of 0 dBm on channel 26 and keep it on until the test is completed. Our goal is to illustrate the impact that a single interfering node has on the performance of each individual protocol. Later, we extend to dynamic interference scenarios with multiple jamming nodes.

Figure 2.8 shows the key metrics of the above-mentioned protocols under an augmented interference. As a single-channel protocol, ORPL suffers the most: When compared to the scenario without interference, its PDR drops to approximately 25% while its radio duty cycle rises to above 2%. MiCMAC/RPL shows a PDR of about 43% with very high latency and duty cycle. Oppcast, a recent state-of-the-art multichannel protocol, performs fairly well in terms of PDR and latency. However, in terms of duty cycle, Oppcast performs worse than ORPL and both configurations of MOR. Oppcast has a higher duty cycle: ORPL has roughly 2.2%, Oppcast 3.1%, and MOR 1.5% and 1.7%, for three and four different channels, respectively.



(a) PDR

(b) Latency

(c) Duty Cycle

Figure 2.8.: Multichannel routing under interference. MOR is superior to other protocols in the light of PDR, latency, and duty cycle. As a single-channel protocol, interference has the strongest impact on ORPL. MiCMAC and Oppcast improve over ORPL, but MOR outperforms them with its fast hopping strategy.

In contrast, MOR outperforms other protocols under interference: Both configurations of MOR are able to obtain a high reliability of over 98.5%. Similarly, MOR achieves a lower latency and the lowest duty cycle: MOR shows a less than 1.4 seconds latency and less than 1.7% duty cycle with both configurations.

Figure 2.9 summarizes the key metrics of every individual sensor node under two conditions accordingly: with and without emulated interference, respectively in ContikiMAC/ORPL, MiCMAC/RPL, Oppcast, MOR(3) and MOR(4). Please note, that we use log-10 scale for the y-axis of latency in Figure 2.9, in Figure 2.10, as well as in Figure 2.11. Overall, MOR accomplishes a duty cycle that is roughly half the one of Oppcast, the second best protocol in this setting, while — in addition — achieving improved reliability and latency.

2.5.5. Resilience to Interference

In this section, we evaluate MOR, ORPL, MiCMAC/RPL, and Oppcast, respectively, under (static) single-channel and (dynamic) multichannel interference: Three jamming nodes are dynamically enabled on a single channel and on multiple channels throughout the experiment. We are interested to evaluate the performance of the selected protocols under different levels of interference on a single channel, as well as the robustness under multichannel interference.

2.5.5.1. Resilience to Single-channel Interference

We evaluate MOR, ORPL, MiCMAC/RPL, and Oppcast under relatively static interference with different levels. Three jamming nodes are enabled throughout a 90-minute experiment. We select node 15, 19, and 22 of FlockLab as jamming nodes. These nodes are well distributed over the testbed.

In this experiment, we change the communication channel from channel 26 to channel 15 so as to address the frequency diversity. In this case, ORPL uses channel 15 as its single channel. Also, we interfere channel 15. We divide each 90-minute test into two periods of 45 minutes. Each 45-minute period consists of three phases of a 15-minute run: (i) No jamming node is enabled; (ii) One jamming node (node 15) is enabled; And (iii) three jamming nodes are enabled. Our aim is to evaluate the impact of different levels of interference (on a single channel) on the performance and the ability of the protocols to recover after interference.

Figure 2.10 illustrates our metrics over time: Both, MOR and Oppcast, bear a strong capability to withstand the interference, maintaining higher PDR, lower latency and duty cycle, independent of the interference levels. These protocols benefit from frequency diversity, that is, while there is interference on a certain channel, the other channels can be effectively utilized opportunistically. Nonetheless, the higher radio duty cycle of Oppcast becomes apparent: It is constantly roughly twice as high as MOR(3), independent of whether there is interference or not. Furthermore, under severe interfered conditions, MiCMAC/RPL earns a better performance than ContikiMAC/ORPL with respect to the average end-to-end PDR and latency. However, MiCMAC/RPL has to pay a high duty cycle of approximately 8%.

On the contrary, the performance of the single-channel ContikiMAC/ORPL degrades along with the aggressiveness of the interference, i.e., the more aggressive the interference is, the lower reliability ORPL gains. It is interesting to observe how the performance of ContikiMAC/ORPL recovers once interference ends. Overall, the results underline that MOR obtains a robust performance even under strong adverse conditions, outperforming the state-of-the-art protocols.

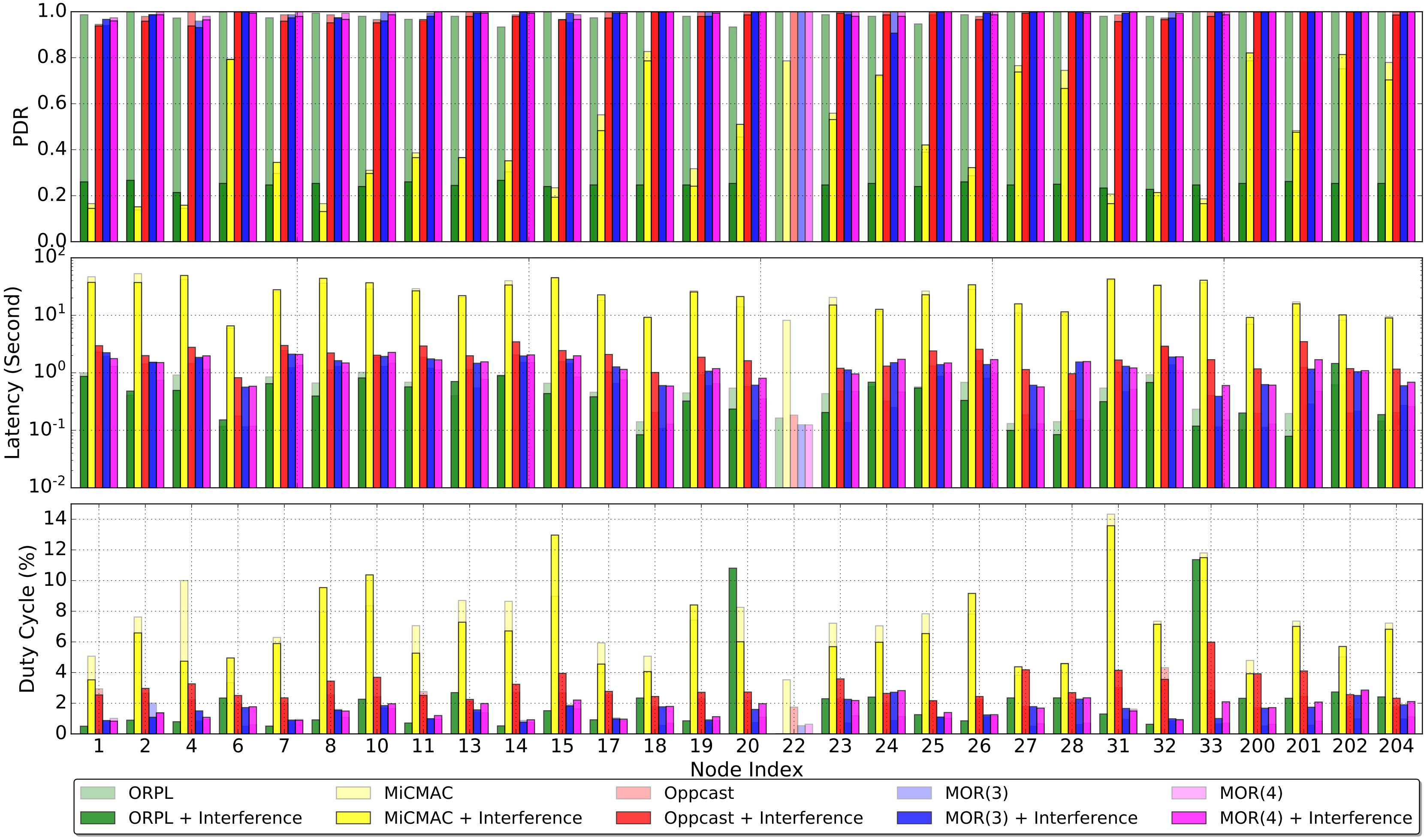


Figure 2.9.: Summary of key metrics with node indices in interference-free and interfered scenarios. The lighter colors represent the nodes performance in the interference-free scenarios, while the darker colors depict the nodes in the emulated-interfered scenarios. In the emulated-interfered scenarios, node 22 acts as the jamming node. Note, that the latency values are shown in log-10 scale.

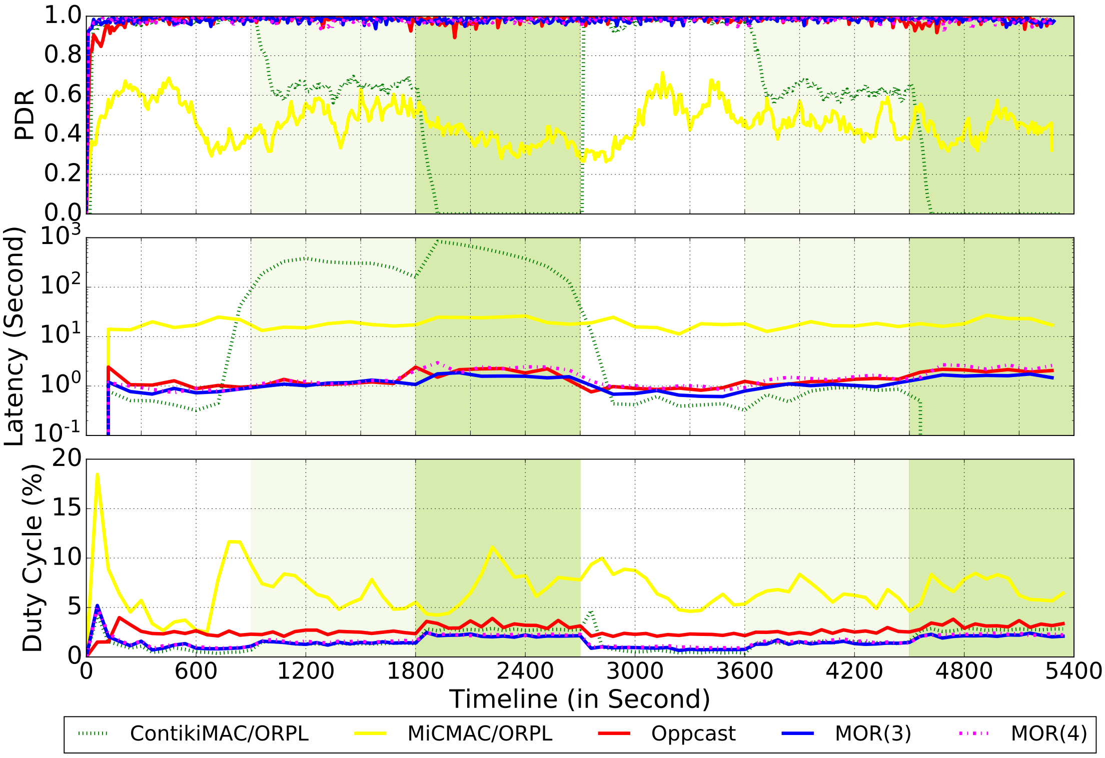


Figure 2.10.: Impact of single-channel interference. The areas filled with white color represent the interference-free conditions, while the areas filled with light green and dark green indicate the interfered conditions with one jamming node and three jamming nodes, respectively. Note, that the latency values are shown in log-10 scale.

2.5.5.2. Resilience to Multichannel Interference

Furthermore, we evaluate MOR, ORPL, MiCMAC/RPL, and Oppcast under dynamic interference. Similarly, three jamming nodes (node 19, 22, and 32) are enabled throughout a 60-minutes experiment.

To introduce dynamic interference, we enable jamming nodes interfere in various channels, i.e., 16 IEEE 802.15.4 channels in this experiment. That means, a jamming node is able to hop to a different channel every second, spanning from channel 11 to 26. Therefore, in this case, it cannot be assumed that some IEEE 802.15.4 channels are constantly interference-free throughout the testbed.

We divide each 60-minutes test into four periods of 15 minutes: First quarter, no jamming node is enabled; Second quarter, one multichannel jamming node (node 32) is enabled; Third quarter, two multichannel jamming nodes (node 19 and 32) are enabled; And last quarter, all three multichannel jamming nodes (node 19, 22, and 32) are enabled. Our aim is to evaluate the impact of dynamics of interference (on multiple channels) on the performance of the protocols.

Figure 2.11 reveals the performance metrics over time: Both, MOR and Oppcast, bear a strong capability to withstand the interference, maintaining higher PDR, lower latency, and lower duty cycle, independent of interference levels, benefiting from frequency diversity. Both, MOR(3) and MOR(4), achieve a higher reliability, a lower latency, and a lower duty cycle than the other protocols. Nonetheless, Oppcast still maintains a higher duty cycle than MOR. The duty cycle of Oppcast is roughly twice as high as the one of MOR(4), independent of whether there is interference or not. These protocols benefit from frequency diversity, that is, while there is interference existing on a certain channel, the other "good" channels can be effectively utilized.

Furthermore, thanks to the opportunistic routing, ORPL performs better than MiCMAC/RPL with respect to average end-to-end PDR and duty cycle, however, with a higher latency. It is also interesting to observe how ORPL performs, once interference is more dynamic. Overall, the results underline that MOR obtains a robust performance even under more realistic dynamic interference, thereby outperforming the state-of-the-art protocols.

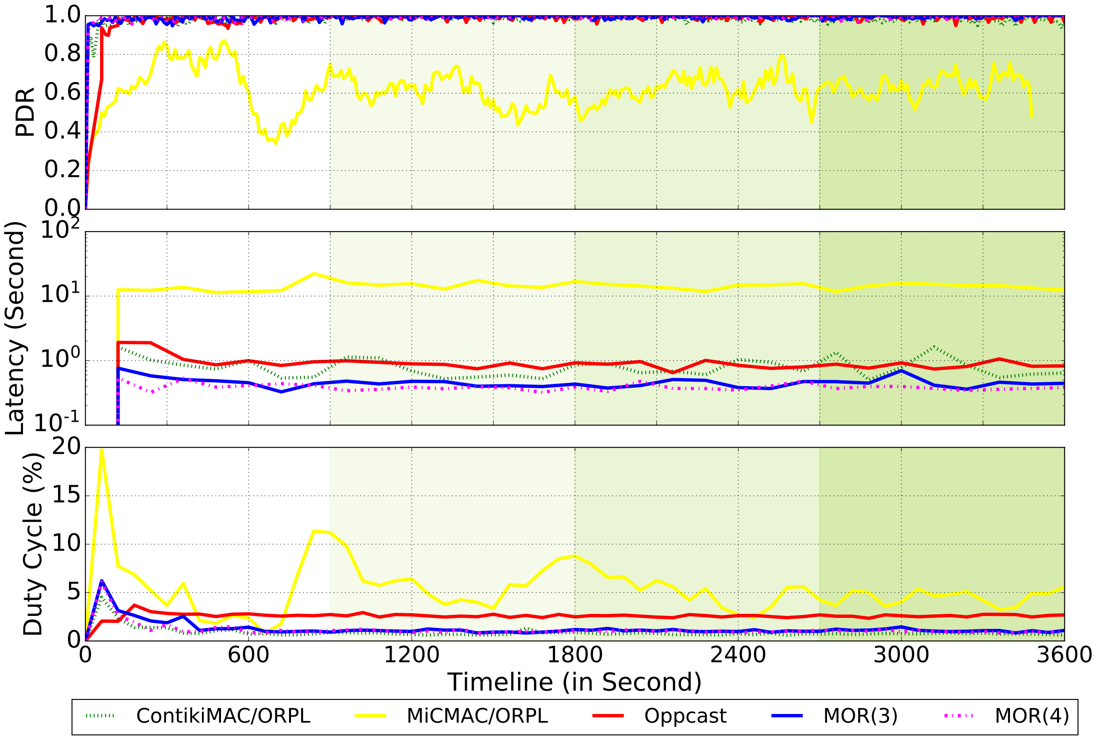


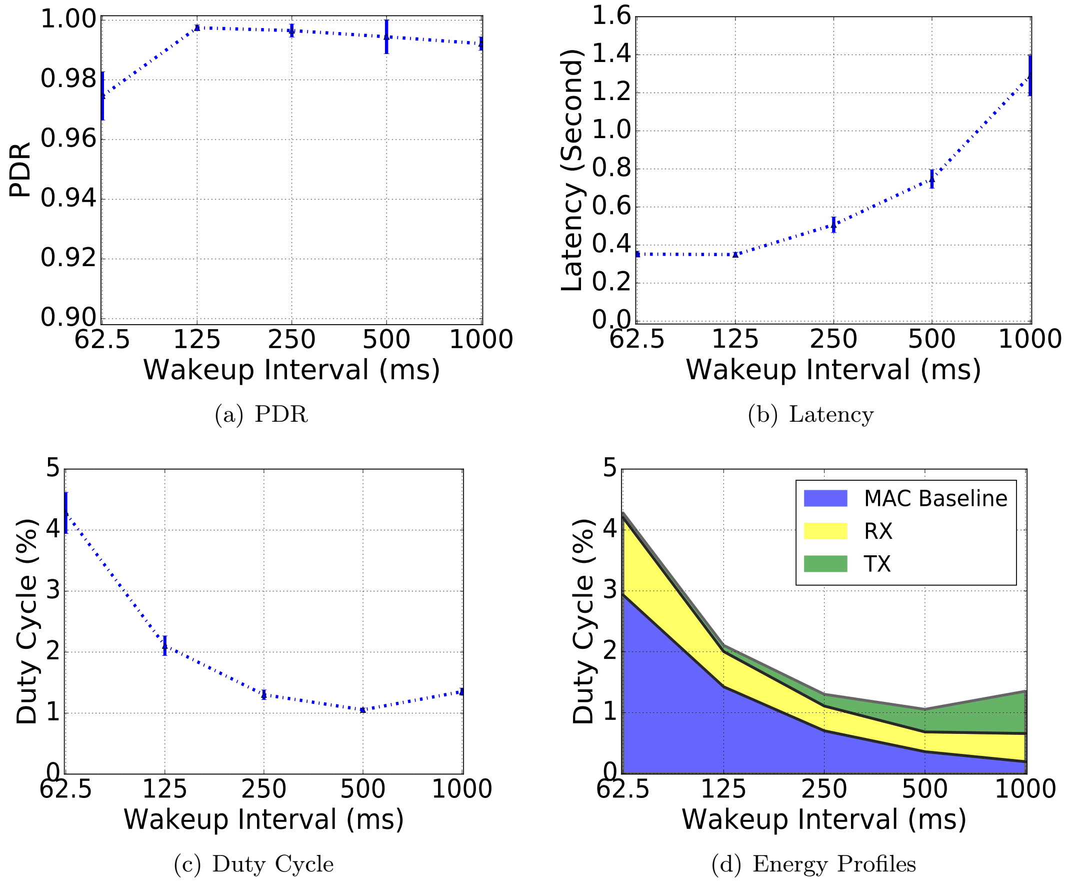
Figure 2.11.: Impact of dynamic multichannel interference. The areas in white represent the interference-free conditions, while the areas in light green, medium green, and dark green indicate interfered conditions, with one, two, and three multichannel jamming nodes, correspondingly. Note, that the latency values are shown in log-10 scale. ORPL has the ability to resist this type of (short-term) dynamic interference, but trades a portion of latency. MOR and Oppcast, both, perform better than the others in terms of PDR, latency, and duty cycle. Oppcast, however, introduces a higher duty cycle compared to MOR.

2.5.6. Impact of Low-level Parameters

In this section, we provide a set of low-level benchmarks to further evaluate the key parameters, for example, the number of assigned channels and the wake-up rate in the MAC layer.

2.5.6.1. Wake-up Interval

At first, we investigate how the wake-up interval of sensor nodes in MOR affects our metrics of reliability, latency, and duty cycle. We preserve the same settings as before, e.g., the number of nodes is 30, and we generate one data packet per node every two minutes. In this experiment, we configure wake-up intervals of 62.5, 125, 250, 500, and 1000 ms, representing channel check rates of 16, 8, 4, 2, and 1 Hz, respectively.



(a) PDR

(b) Latency

(c) Duty Cycle

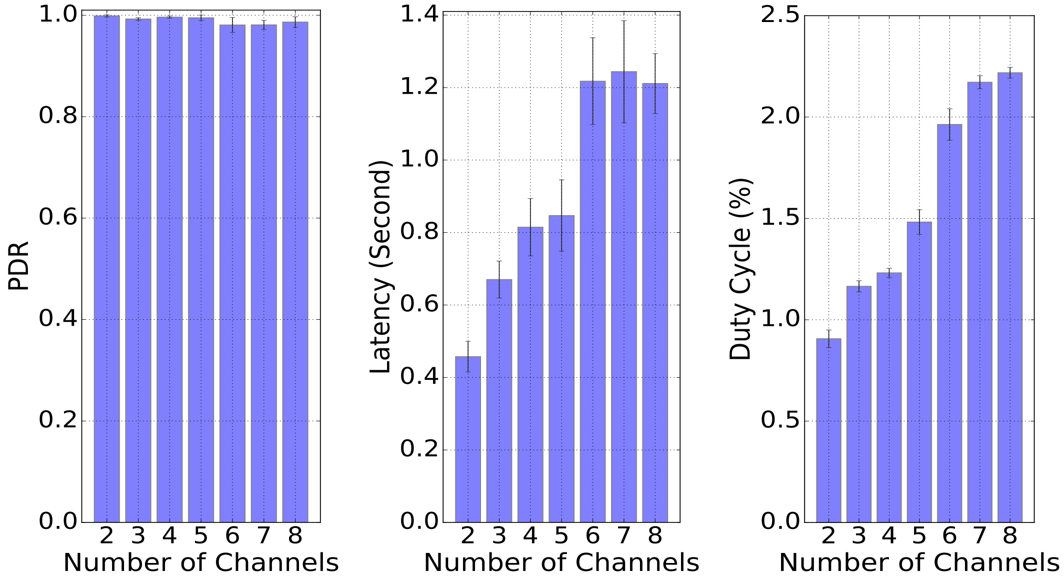
(d) Energy Profiles

Figure 2.12.: Effectiveness of the channel check rate in MOR. Performance metrics vary along with utilizing different channel check rates.

Figure 2.12 depicts the impact of the different channel check rates in the MAC layer. The results underline that configurations with channel check rates of 2 Hz efficiently balance end-to-end reliability, latency, and power consumption. More specifically, the detailed energy profiles are illustrated in Figure 2.12(d). Basically, the energy cost of the MAC baseline decreases when increasing the wake-up interval: A larger channel check rate results in more channel listening and, thus, increases power consumption. Additionally, when the wake-up interval increases, the energy spent by each transmission is also increasing: When the channel check rate decreases, then the strobing time of a packet is also increased until a rendezvous with a receiver on the same channel happens.

2.5.6.2. Number of Channels

Next, we evaluate the impact of the number of channels MOR utilizes. We expect that the power consumption increases when the number of utilized channels increases. Using more channels inherently increases the time until rendezvous and adds channel switching overhead, LPL overhead on each individual channel. Figure 2.4 indicates that there are only eight "good" channels in FlockLab, i.e, channels with more than 50% end-to-end PDR: channel 26, 25, 20, 15, 21, 22, 19, and 14 (sorted in order of decreasing quality). To quantify the impact of the number of channels in detail, we run experiments of MOR in FlockLab using from two to eight of these channels. In this experiment, we do not add additional interference next to the interference that is already present in the testbed, e.g., from WiFi or Bluetooth.



(a) PDR (b) Latency (c) Duty Cycle

Figure 2.13.: Cost of the number of channels used in MOR. While PDR stays high, latency and duty cycle increase for each channel added.

Figure 2.13 demonstrates how the performance metrics, i.e., PDR, latency, and duty cycle, change when using more channels in MOR. PDR stays high, while both, latency and radio duty cycle, increase — as expected — when increasing the number of channels used. The latency here, however, does not increases linearly with the number of channels. One possible reason for this can be that, as the number of channels increases, several "not-so-good" channels are also in use, which produces a negative effect on latency. Therefore, we argue that it is sufficient to choose the number of channels to reflect the amount of interference expected. Additionally, we have shown in previous sections that even under strong interference, three channels are sufficient to maintain good performance.

2.5.7. Discussion

Table 2.1 summarizes our experimental results in four scenarios: (i) an interference-free scenario; (ii) scenario with only one interference source near the sink; (iii) scenario with three interference sources in one single channel across the network; And (iv) scenario with three multichannel interference sources across the network.

Our experimental results reveal that multichannel routing in MOR comes at a cost: In an interference-free environment MOR sacrifices latency and radio duty cycles when compared to ORPL. However, under interference MOR outperforms other state-of-the-art protocols, including Oppcast and MiCMAC. MOR attains approximately half the duty cycle than Oppcast, the protocol with the second best results and also improves in terms of reliability and latency over the state-of-the-art. Moreover, MOR shows these results independently of the level and the dynamic of interference.

Meanwhile, there are only few limitations in MOR: Practically, the channel rendezvous sequence becomes longer and more complex as the number of channels increases. As a result, the probability of rendezvous in a short period of time cannot always be guaranteed to be 100%. The rendezvous time can vary strongly due to the design of the hopping frequency sequence.

In addition, in scenarios with aggressive interference, i.e., simultaneously on many channels, MOR can only keep its robust performance when at least one channel is available at each point in time. However, this situation might not be practical because the interference does not exist on all the channels simultaneously and for a long period of time; otherwise all the communication protocols on these channels would easily fail. Moreover, the link quality of channels changes over time and can be different in different parts of the network.

To sum up, in this section, we demonstrated the performance of different protocols respectively in different scenarios: interference-free, interfered, and dynamic interfered scenarios. Our experimental results reveal that in the interference-free scenario, MOR effectively inherits the benefits from opportunistic routing. It achieves the best performance in interfered scenarios in terms of PDR, latency, and duty cycle compared to other state-of-the-art protocols. Besides, MOR is able to maintain the robust performance even under adverse conditions of dynamic interfered scenario.

2.6. Conclusion

This chapter introduced MOR, a multichannel opportunistic routing protocol for low-power duty-cycled WSNs. MOR applies multichannel hopping strategies in opportunistic routing, thus, exploiting spatial, temporal, and frequency diversities in WSNs. The opportunistic nature of the packet forwarding in MOR is essential for its performance: In contrast to traditional approaches to unicast routing, e.g., RPL or CTP, MOR does not have to ensure rendezvous with one particular parent. MOR only needs a rendezvous with one of the typically many potential forwarders. Thereby, MOR benefits from both,

Table 2.1.: Summary of experimental results. MOR maintains a best-effort end-to-end PDR regardless of interference. Under adverse interfered conditions, MOR improves the end-to-end latency while preserving higher energy efficiency. Oppcast is also a robust protocol with respect to PDR, latency, and resilience to interference, but it consumes more energy.

Scenario	Network Settings	Protocol	PDR (%)	Latency (s)	Duty Cycle (%)
(i)	Application: Many-to-one Data Collection Experiment Duration: 60 minutes TX Power: 0 dBm Channel Check Rate: 2 Hz Average Packet Generation Interval: 2 minutes	ContikiMAC/RPL	82.38 (max: 91.56 min: 76.69)	2.64 (max: 3.17 min: 1.94)	1.35 (max: 1.67 min: 0.87)
		ContikiMAC/ORPL	98.41 (max: 99.88 min: 95.05)	0.50 (max: 0.63 min: 0.32)	0.65 (max: 0.77 min: 0.55)
		MOR(3)	99.26 (max: 99.77 min: 98.62)	0.71 (max: 0.79 min: 0.65)	0.95 (max: 0.99 min: 0.92)
(ii)	Application: Many-to-one Data Collection Experiment Duration: 60 minutes TX Power: 0 dBm Channel Check Rate: 2 Hz Average Packet Generation Interval: 2 minutes Jamming Channel: Channel 26 Jamming Node (node 22): 15 minutes OFF + 45 minutes ON	ContikiMAC/ORPL	24.87 (max: 25.12 min: 24.60)	0.45 (max: 0.53 min: 0.38)	2.25 (max: 2.32 min: 2.10)
		MiCMAC/RPL	43.13 (max: 47.45 min: 39.65)	25.45 (max: 26.93 min: 23.88)	6.89 (max: 7.21 min: 6.68)
		Oppcast	97.68 (max: 99.13 min: 96.79)	2.00 (max: 2.10 min: 1.84)	3.13 (max: 3.33 min: 2.93)
		MOR(3)	98.51 (max: 99.52 min: 97.48)	1.31 (max: 1.51 min: 1.18)	1.52 (max: 1.62 min: 1.45)
		MOR(4)	98.97 (max: 99.64 min: 97.59)	1.37 (max: 1.57 min: 1.24)	1.66 (max: 1.87 min: 1.52)
(iii)	Application: Many-to-one Data Collection Experiment Duration: 90 minutes TX Power: 0 dBm Channel Check Rate: 2 Hz Average Packet Generation Interval: 2 minutes Jamming Channel: Channel 15 Jamming Node (node 15): (15 minutes OFF + 30 minutes ON) x 2 Jamming Node (node 19): (30 minutes OFF + 15 minutes ON) x 2 Jamming Node (node 22): (30 minutes OFF + 15 minutes ON) x 2	ContikiMAC/ORPL	64.5 (max: 66.20 min: 62.34)	140.13 (max: 163.20 min: 104.40)	1.69 (max: 1.77 min: 1.60)
		MiCMAC/RPL	45.72 (max: 50.00 min: 41.52)	25.87 (max: 29.29 min: 21.23)	6.76 (max: 7.10 min: 6.22)
		Oppcast	99.21 (max: 99.56 min: 98.60)	1.43 (max: 1.49 min: 1.39)	2.70 (max: 2.74 min: 2.66)
		MOR(3)	99.35 (max: 99.66 min: 98.98)	1.17 (max: 1.46 min: 1.07)	1.56 (max: 1.77 min: 1.49)
		MOR(4)	99.33 (max: 99.66 min: 98.88)	1.56 (max: 1.73 min: 1.39)	1.68 (max: 1.79 min: 1.63)
(iv)	Application: Many-to-one Data Collection Experiment Duration: 60 minutes TX Power: 0 dBm Channel Check Rate: 2 Hz Average Packet Generation Interval: 2 minutes Jamming Channel: Channel 11 - 26 Jamming Node (node 15): 15 minutes OFF + 45 minutes ON Jamming Node (node 19): 30 minutes OFF + 30 minutes ON Jamming Node (node 22): 45 minutes OFF + 15 minutes ON	ContikiMAC/ORPL	98.98 (max: 99.87 min: 97.31)	0.83 (max: 1.46 min: 0.45)	0.89 (max: 1.05 min: 0.71)
		MiCMAC/RPL	62.43 (max: 69.05 min: 55.59)	15.63 (max: 18.05 min: 14.32)	5.43 (max: 6.26 min: 4.56)
		Oppcast	99.67 (max: 100.0 min: 99.17)	0.95 (max: 1.13 min: 0.83)	2.27 (max: 2.38 min: 2.15)
		MOR(3)	99.71 (max: 100.0 min: 99.49)	0.47 (max: 0.51 min: 0.43)	1.25 (max: 1.37 min: 1.14)
		MOR(4)	99.90 (max: 100.0 min: 99.87)	0.39 (max: 0.45 min: 0.32)	1.15 (max: 1.22 min: 1.08)

spatial and frequency diversities: If one neighbor is not available on a particular channel, then it either utilizes a different forwarder or a different channel.

We implemented our protocol in Contiki OS and evaluated it with extensive experiments in FlockLab. With trading only a slight portion of power consumption, MOR achieves higher than 98.50% average end-to-end reliability and less than 1.60 seconds average end-to-end latency, in both, interference-free and severely interfered environments. Furthermore, MOR maintains a more robust resilience to highly dynamic interference with less duty cycle while compared to other protocols. To sum up, MOR outperforms the state-of-the-art protocols in the light of end-to-end reliability, latency, and power consumption.

3 Machine Learning-based Flooding

3.1. Introduction

Network flooding is a protocol that delivers messages from a source node to all other nodes in a connected network. For instance, the Open Shortest Path First (OSPF) protocol [Moy97] in Internet Protocol (IP) networks uses flooding to update router information in a network. In addition, as an essential operation for WSNs, flooding is widely used for information dissemination, bulk data transfer, code update, time synchronization, and network configuration. In the last few years, flooding in WSNs has been experimentally proven to be fast, reliable, and energy-efficient [LW09, FZTS11]. Recently Concurrent Transmission (CT)-based flooding has been introduced in low-power wireless networks as a promising technique for data dissemination. For example, Flash Flooding [LW09] and the Glossy protocol [FZTS11] (referred to as "Glossy" in the following) exploit constructive interference and the capture effect to achieve highly reliable data flooding in multihop WSNs. These protocols significantly increase network throughput, enhance packet transmission reliability, and reduce flooding latency.

However, these protocols have to introduce high communication redundancy in order to attain high reliability. That is, to obtain a fast and reliable coverage of the whole network, each sensor node has to broadcast the received packet until every node in the network has been covered. Consequently, there exists a large degree of transmission redundancy, i.e., many of these broadcast transmissions are not necessary. Figure 3.1 demonstrates a network with three nodes: node I, A, and B. As shown in Figure 3.1(a), at first node I broadcasts a packet. After both, node A and B receive the packet, they rebroadcast the packet to each other and to node I, shown in Figure 3.1(b). Apparently, the last two transmissions are unnecessary in the case of delivering the information to every node in a network. Such transmissions are defined as broadcast redundancy. In this case, sensor nodes consume much more energy than expected. This type of aggressive flooding, generally referred to as *blind flooding* [TNCS02], is not energy-efficient. Moreover, redundant transmissions can cause a more serious broadcast storm problem [NTCS99], lead to overwhelmed packet contentions and collisions in a network.

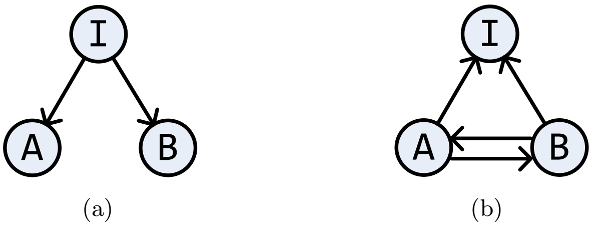


Figure 3.1.: Redundant transmissions by flooding. In (a), node I broadcasts a packet to node A and B. In (b), both node A and B then rebroadcast the received packet to the others.

Moreover, CT-based flooding also suffers from a scalability problem with respect to the temporal misalignment [NPPS$^+$15], since concurrent transmissions highly depend on tight time synchronizations between transmitters. Namely, the packet reception rate degrades as the node density or the size of the network increases. As discussed by the authors in [LFZ13a, RKP$^+$16], the probability of receiving a packet due to the capture effect drops notably as the number of synchronous transmitters increases. To overcome these problems, Chaos [LFZ13a] exploits in-network processing together with concurrent transmissions: While each node receives a packet, it spends a fixed period of time (processing time) to process the data and then makes a decision whether it is necessary to forward the received packet. In this case, it is able to appropriately decrease the number of concurrent transmitters and maintain a best-effort performance even in high-density WSNs. Furthermore, CXFS [CCT$^+$13] concentrates on one-to-one data transmission and builds a forwarder-selection scheme on CT-based flooding. CXFS aims to reduce wasteful transmissions, thus to improve energy efficiency and throughput, while providing a similar reliability. While Chaos and CXFS are based on the Glossy protocol, they still belong to blind flooding after all. To achieve high reliability, these flooding protocols are required to repeat the transmission for a fixed number of times. For instance, Glossy sets the maximal number of transmission to five by default to accomplish high reliability.

In order to avoid blind flooding but at the same time maintain high reliability, each sensor node should be able to decide whether or not it is essential to forward the received packet based on the current environmental conditions. Decisions are adaptive to maintain a good performance of the network while minimizing transmission redundancy. Therefore, this decision-making adaptation can be converted to an optimization problem.

Generally, reinforcement learning [Sut84, SB98] techniques are effectively applied to solve these types of optimization problems. Reinforcement learning is a class of learning algorithms that attempts to maximize the cumulative reward by taking a specific action in a given state and following a predefined policy thereafter. Additionally, among all the reinforcement learning techniques, a set of so-called *multi-armed bandit* algorithms is particularly suitable for the optimization of CT-based WSNs. That is because the number of transmissions in each sensor node can be furthermore modeled as a multi-armed bandit problem, originally described by Robins [Rob85].

A multi-armed bandit, also called K-armed bandit, is similar to a traditional slot machine but generally with more than one lever. As further explained by Auer et al. [ACBFS95], in a multi-armed bandit problem, a gambler need to choose which of slot machines to play. At each time step, the gambler pulls the arm of one of the machines. After that, he receives a reward or a payoff. The purpose is to maximize the total reward over a sequence of trials. Therefore, the goal is to find the arm with the best expected return as early as possible, and then to keep gambling using that arm, since each arm is assumed to have a different distribution of rewards. Similarly, this type of multi-armed bandit algorithm investigates the selection of the "best" action for the dynamic situations in low-power and lossy WSNs as well.

We propose *Less is More (LiM)*, a machine learning-based data dissemination protocol for low-power multihop WSNs. In designing LiM, we utilize a reinforcement learning technique to reduce redundant broadcast transmissions. We model the optimization of the transmission times in each sensor node as a multi-armed bandit problem. Besides, we exploit an exponential-weight algorithm for exploration and exploitation (called *Exp3*), for bandit learning in each sensor node of the WSN. Incorporated with concurrent transmissions, LiM is able to effectively achieve high end-to-end reliability and low end-to-end latency. Moreover, LiM empowers sensor nodes with a learning capability to reduce the redundancy of the flooding step by step, thereby significantly lowering power consumption. We implement LiM in Contiki [DGV04] and conduct extensive experiments in a 30-node testbed — FlockLab [LFZ$^+$13b]. Furthermore, we compare LiM to the baseline protocol Glossy [FZTS11] focusing on reducing communication redundancy in flooding.

Our evaluation shows that LiM is able to effectively limit the number of transmissions of the sensor nodes while still preserving high reliability and energy efficiency, as well as low latency: Sensor nodes, which do not belong to the backbone of the network, stay only in receiving and sleeping mode. The others execute the decision-making based on their obtained experience from the learning phase. With various levels of transmission (TX) power and different topologies, LiM obtains an average reliability of over 99.70% and an average end-to-end latency of less than 2.4 ms in all experimental scenarios. Moreover, LiM reduces the radio-on time by at least 30% compared to the default configuration of Glossy.

The remainder of this chapter is organized as follows. Section 3.2 discusses related work with two foci: on Glossy-based flooding protocols and on bandit-learning strategies applied in WSNs. Section 3.3 explains the basis of LiM and provides a brief overview. Section 3.4 details the design perspectives of LiM, followed by performance evaluations elaborated in Section 3.5. Section 3.6 concludes our work and leads to an outlook for future work.

3.2. Related Work

In this section, we review a number of existing research work in terms of CT flooding protocols in WSNs, multi-armed bandit algorithms, and the bandit learning in WSNs.

3.2.1. CT-based Flooding Protocols

Network flooding is one of the most fundamental services in WSNs. It forms the basis for a wide range of applications and network operations. Glossy [FZTS11] provides a fast and efficient network flooding service by using concurrent transmissions in WSNs. By exploiting constructive interference and the capture effect on the physical layer, Glossy is able to get an average packet delivery ratio of 99.99% in real testbeds. Afterwards, Ferrari et al. adds an application-level scheduler to construct a so-called Low-power Wireless Bus (LWB) [FZMT12]. LWB centrally schedules the data communication to support one-to-many, many-to-one, and many-to-many traffic patterns in WSNs. On the contrary, Chaos [LFZ13a] builds on Glossy to achieve fast all-to-all data sharing in a distributed manner. Chaos further combines programmable in-network processing with concurrent transmissions in WSNs.

Splash [DCL13] builds a tree pipeline [RCBG10] on Glossy, thereby improving channel utilization. Furthermore, Pando [DLZL15] integrates fountain code [Mac05] together with CT and disseminates packets as data pipelines so as to overcome the long-tail problem of Splash. While Glossy disseminates one packet in each communication round, Splash and Pando are designed to deliver large data objects to all nodes in a network, e.g., an up-to-date image for the purpose of reprogramming WSN-based applications. Ripple [YH15] also relies on Splash and network coding techniques to further improve particularly in terms of network throughput.

Carlson et al. propose CXFS [CCT$^+$13], a forwarder selection mechanism for concurrent transmissions. In CXFS, sensor nodes use a hop count in each packet to get their relative distance to each other. CXFS builds on Glossy and supports point-to-point transmissions while achieving high reliability, high energy efficiency, and high throughput. Moreover, Sparkle [YRH14] selects subsets of nodes that participate in Glossy-based flooding. It also supports one-to-one communication. Similarly, LaneFlood [BLS16] is built on Glossy and further integrates the forwarder selection scheme of CXFS with application-level network protocols in WSNs. LaneFlood thus supports one-to-one traffic, forwarder selection, and standard protocols in IoT such as TCP/UDP and the Constrained Application Protocol (CoAP). RTF [ZRHK15] further extends Sparkle and exploits TDMA for data scheduling to improve reliability and energy efficiency in point-to-point traffic. RFT identifies reliable relay nodes to limit the number of concurrently active neighbors to save more energy.

3.2.2. Multi-armed Bandit Algorithms

Many real-word problems require decisions to be made for maximizing the expected reward. Over the last two decades, a number of algorithms have been designed for this purpose.

One simple strategy, called ϵ-greedy, was first investigated by Watkins [Wat89]. This method introduces an ϵ-frequency, which is configured by the users, to decide the probability of uniformly trying an action. Otherwise, the algorithm executes the action with the highest mean value μ. The Exp3 algorithm [ACBFS95], first introduced by Auer, considers using a modified *softmax* function to decide the possibility of trying different actions.

Besides, several more strategies were proposed after ϵ-greedy and Exp3. For instance, in 1998, Cesa-Bianchi et al. introduced SOFTMIX [CBF98]. In 2005, Vermorel et al. introduced the POKER algorithm [VM05]. These algorithms are claimed to perform better than ϵ-greedy and Exp3. However, compared to the others, Exp3 is simple, widely used, and easy to be implemented on an embedded sensor device, especially running on Contiki OS. In this work, we select the Exp3 algorithm as a candidate (i) to prove the feasibility of reinforcement learning algorithms in resource-constrained sensor nodes, and (ii) to evaluate the usability and adaptability of this kinds of learning algorithms in WSNs.

3.2.3. Bandit Learning in WSNs

In [MB07], Motamedi et al. propose a distributed multi-channel MAC protocol for wireless networks. It formulates the dynamic channel selection in wireless networks as a multi-armed bandit problem and derives optimal channel selection rules. The authors investigate the effectiveness of their protocol by using simulations only. Thus, the effectiveness in real-world scenarios is therefore not clear.

Similarly, another multi-channel access scheme is proposed in [ZSJS16] to schedule the access for cognitive users, in order to maximize the throughput in cognitive radio-based WSNs. By their simulation results, the authors claim that the proposed scheme can effectively improve the utilization of the idle spectrum and guarantees the fairness of selecting channels between cognitive users.

In [KF14], Kadono et al. propose a budget-limited multi-armed bandit algorithm, which is suitable for resource-constrained WSNs. It can limit sources to be retrieved when a relatively hard budget limitation has been applied. By conducting simulations, the authors claim that the proposed protocol outperforms the state-of-the-art.

Authors in [dPAP12] present a duty cycle learning algorithm (DCLA) for IEEE 802.15.4 WSNs. DCLA automatically adapts the duty cycle during run-time to minimize power consumption and to balance the packet delivery ratio and delay constraints of the application. It estimates the incoming traffic by collecting network information during each active phase and then uses a reinforcement learning framework to learn the best duty cycle at each beacon interval. Simulations demonstrate that the proposed scheme achieves the best overall performance for both, constant and event-based traffic, compared to existing IEEE 802.15.4 duty-cycled adaptation schemes.

In [TTRJ12], the authors study the long-term information collection in the WSN domain. They propose a multi-armed bandit-based approach for the energy management problem in WSNs. They also describe a multi-armed bandit algorithm — Exp3 — that can be used to efficiently deal with the energy management problem. They show through simulations that their approaches improve the performance of the network by up to 120%.

In [VRP12], Villaverde et al. present a route selection algorithm (InRout), which shares local information among neighboring nodes to enable efficient, distributed route selection. They model it as a as a multi-armed bandit problem and use Q-learning techniques to obtain the best routes based on current network conditions and application settings. The authors compare InRout with existing approaches by simulations. Their results demonstrate that InRout provides gains ranging from 4% to 60% in the number of successfully delivered packets compared to current approaches while having a lower control overhead.

3.2.4. Summary

Concurrent transmissions — a promising technique in this field — allow highly energy-efficient, low-power communication in WSNs. The technique has been developed and integrated with different standards and techniques. None of the state-of-the-art protocols, however, makes a great effort to apply an adaptive machine-learning scheme to concurrent transmissions. On the other hand, the bandit-learning scheme has been exploited in the field of WSNs for smart duty cycling, long-term energy management, and route selection. Most of the work has been investigated by using simulations only. As a consequence, their effectiveness in real-world scenarios has not been shown yet.

LiM incorporates concurrent transmission with a bandit-learning scheme in order to take advantage of both techniques. Meanwhile, LiM proves the feasibility of applying relatively light-weight machine-learning techniques to concurrent transmission for low-power wireless networks in real-world applications. To the best of our knowledge, LiM is the first primitive that integrates a machine-learning scheme with concurrent transmissions, especially for low-power multihop WSNs. We believe that LiM is able to be further developed to robustly resist more adverse conditions in reality, e.g., with a channel hopping scheme in [ZLT17], and to satisfy the requirements of the various applications.

3.3. Overview

In this section, we explain the basis of LiM in two dimensions: (i) reliable flooding and (ii) machine leaning. Then, we provide a brief overview of the protocol.

3.3.1. Reliable Flooding

Proposed in 2011, Glossy [FZTS11] is one of the most representative CT-based flooding protocols in the WSN community. Basically, Glossy exploits both, constructive interference to superimpose (identical) packets and the capture effect to ensure that a receiver is actually able to correctly demodulate a received packet. These two mechanisms empower Glossy to manage a highly reliable flooding and an accurate time synchronization. In Glossy, nodes concurrently transmit packets in a receive-and-forward scheme, which means that nodes receive a valid packet and forward it immediately. Messages are propagated through the network without contention of the wireless medium. Therefore, the latency of flooding could approximately reach the theoretical upper bound, as studied in [FZTS11]. Moreover, nodes in the network get clock-synchronized to the initiator in order to estimate the time to wake up, thereby saving much energy.

Figure 3.2 demonstrates one Glossy flooding example. The network consists three nodes, which are highly time-synchronized among each other. Node I can reach node A but not B, and node B can reach node A but not I, as shown in the figure. Nodes are configured with the number of transmissions equals to two, i.e. $N = 2$, which means each node only transmits the packet twice in total. At first, node I starts to send a packet to node A in time slot 0, as a normal transmission. After node A has successfully received the packet, it transmits the packet to both, node I and B in time slot 1. In time slot 2, as node I and B have received the packet, both of them synchronously send the packet to node A, as concurrent transmissions. At the end of this time slot, node I finishes two transmissions and goes into a sleep mode. In time slot 3, node A sends the packet to node B and then goes to sleep. Node B receives the packet in time slot 4. Afterwards, it sends the received packet again and goes to sleep. At the end, one data packet from node I has been reliably disseminated to each node in the network for several times.

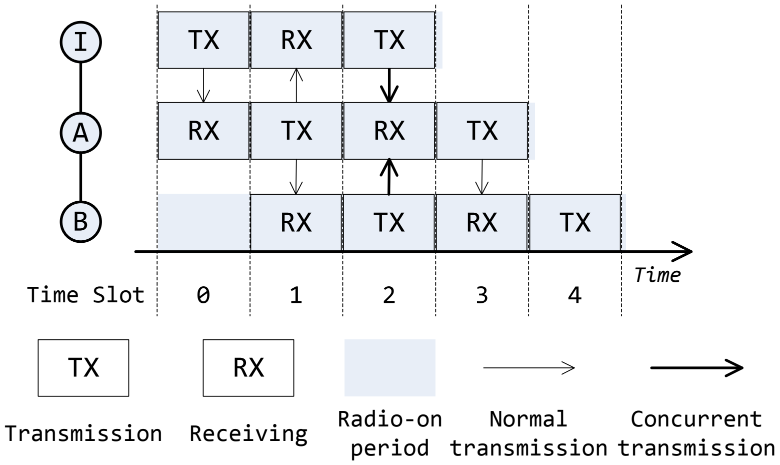


Figure 3.2.: Example of a Glossy flooding round with $N = 2$ in a topology of three nodes. Black lines connecting the nodes imply the communication links. Each node transmits only *twice* in total. Nodes always concurrently transmit packets once they have received them. Thin arrow lines stand for a normal transmission, where CT does not exist to the receiver. Thick arrow lines refer to a CT, where occur the constructive interference and the capture effect to the receiver.

3.3.1.1. Constructive Interference

In physics, interference is a phenomenon in which two or more waveforms superpose to form a resultant wave, which is either reinforced or canceled. As shown in Figure 3.3(a), constructive interference is generated from the two identical waves 1 and 2, which leads to a wave of greater amplitude. Otherwise, destructive interference is created when the two waves cancel each other, thus resulting a wave of weaker amplitude, as shown in Figure 3.3(b).

Specifically, in WSNs, constructive interference occurs only when two or more nodes transmit *identical* packets. Besides, it requires a highly tight time synchronizations among radio transmitters. For instance, with IEEE 802.15.4 radios operating in the 2.4 GHz ISM band, these identical packets from various transmitters are required to overlap within 0.5 μs [DDHC+10, FZTS11] in order to make the packets appropriately superposed.

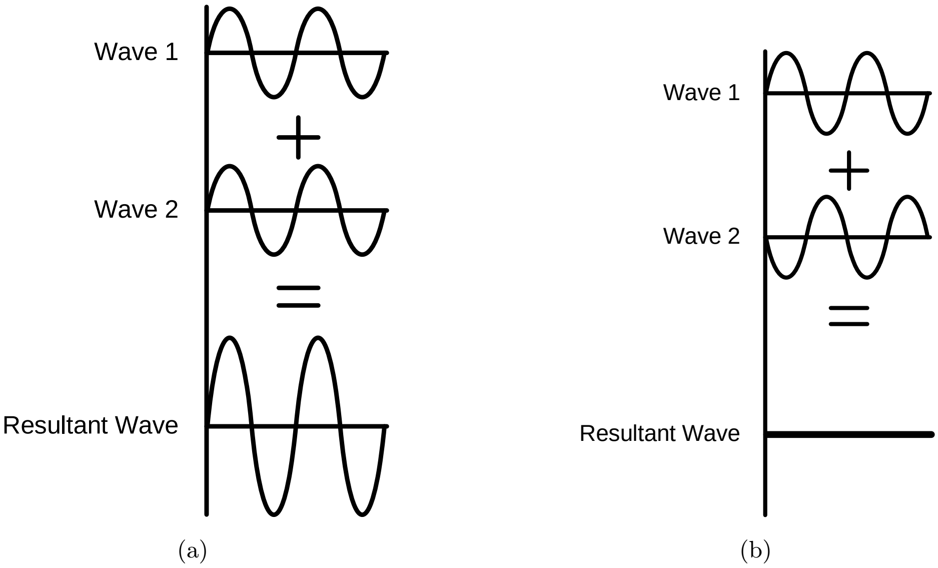


Figure 3.3.: Constructive interference and destructive interference resulted from two waves.

3.3.1.2. Capture Effect

The capture effect, also referred to as co-channel interference tolerance, is a phenomenon where a certain radio correctly receives a strong signal from one transmitter despite significant interference from other transmitters [LF76, WWJ+05]. In IEEE 802.15.4 wireless networks, if a received signal is approximately 3 − 4 dB stronger than the sum of all the other received signals, then the receiver is able to lock on to and correctly demodulate the signal [LF76,WWJ+05,FZTS11]. Additionally, the strongest signal must arrive no later than 160 μs after the weaker signals in order to be properly captured and decoded by the receiver [LFZ13a]. In other words, within a time period of 160 μs, if there are multiple signals transmitting to a receiver, then the strongest signal wins the chance to be correctly received. Accordingly, the others are considered as interference.

3.3.2. Machine Learning

Machine learning is a sub-field of artificial intelligence that is concerned with the question of "how to construct computer programs that automatically improve from experience" [Mit97]. This property makes the family of machine learning-based algorithms attractive for reliable and efficient communications in WSNs.

3.3.2.1. Reinforcement Learning

Reinforcement learning is one of the machine-learning techniques, in which the learning agent earns knowledge from the interaction with the environment. Technically, reinforcement learning is used by a class of algorithms that seeks to maximize the cumulative reward by executing different actions in a task. In this case, different configurations of a system can be modeled as the corresponding actions to maximize the reward in order to optimize the performance of the system.

3.3.2.2. Multi-armed Bandit Problem

The multi-armed bandit problem was originally proposed by Robbins [Rob85] in the year of 1985. A gambler, firstly, chooses K slot machines to play. At each time step, the gambler pulls one arm of one machine (out of K) and then receives a positive, zero, or negative reward. The purpose is to maximize the total reward over a sequence of trials. Assuming each arm in a slot machine has a different distribution of rewards, the goal is to find out the arm with the best expected return as early as possible and then to keep using that specific arm.

The problem is a classical example of the trade-off between exploration and exploitation [ACBFS95]: On the one hand, if the gambler plays exclusively on the machine which the gambler supposes to be the best one ("exploitation"), then the gambler may fail to discover that one of the other arms, in fact, has a higher average return. On the other hand, if the gambler spends too much time trying out all K machines and then makes a decision based on the gathered statistics ("exploration"), then the gambler may fail to play the best arm for long enough a period of time to get a high total return.

To solve the multi-armed bandit problem, the Exponential-weight algorithm for Exploration and Exploitation (Exp3) was proposed by Auer et al. [ACBFS02] in the year of 2002. Exp3 is based on a reinforcement learning scheme and it solves the following problem:

> "*If there are many available actions with uncertain outcomes in a system, how should the system act to maximize the quality of the results over many trials?*"

We provide the details of Exp3 and the related implementation issues later in Section 3.4.

3.3.3. LiM in a Nutshell

LiM builds on Glossy and it is able to effectively inherit the advantages of the CT-based flooding protocols, i.e., high reliability, low latency, and low radio duty cycle. In this case, LiM exploits both, constructive interference and the capture effect to guarantee a good performance of the network. Meanwhile, it is challenging to integrate LiM with Glossy, since Glossy requires a highly tight deterministic software delay and the identical content of the packet for concurrent transmission.

The feedback from the neighboring nodes should be renewed according to the dynamic network conditions, leading various packets within the network. Therefore, the requirement of identical content of the packets cannot be satisfied. However, the packets in LiM are not necessarily identical, since LiM opportunistically uses the capture effect to effectively receive the packet with the strongest signal level. Additionally, LiM models the redundancy optimization problem as a multi-armed bandit problem and maps a number of configurations to the corresponding actions in each sensor node. LiM employs a bandit-learning scheme — Exp3 — in order to progressively optimize the efficiency of the network. This learning scheme investigates the selection of the “best” action for the dynamic environment, dramatically minimizing the redundancy of the communications while still maintaining a high reliability.

Generally, LiM comprises two main phases: a greedy *exploration* phase and a bandit *learning* phase. The former one is an exploration process where the “redundant” nodes in the network could be discovered. In this phase, LiM randomly selects one node to act as a exploring node in the network. This exploring node triggers some unexpected events in the network, e.g., after receiving a packet from its “upside” parent nodes, it stops forwarding for some time slots. As a result, a number of communication links are disabled during those time slots. Some neighboring nodes might suffer a packet loss for this period of time and accordingly they give negative feedback to the exploring node. Otherwise, if no one suffers a packet loss, then the exploring only receives positive feedback and then LiM regards this exploring node as a “redundent” node. This type of “redundant” nodes acts as a concurrent transmitter in CT-based protocols. LiM attempts to seek these nodes and then keeps them staying in either receiving mode or sleeping mode in the network. Therefore, it is able to reduce redundancy and to improve the energy efficiency of the network.

The latter phase is a reinforcement learning process. Similar to the exploration phase, in the learning phase, LiM attempts to grant each node (except the previous found “redundant” node) to select a corresponding action (i.e., the number of transmission (N)) based on the network dynamics. This selection depends on the result of the reinforcement learning in each node — choosing the action with the highest probability. Moreover, the learning phase has no conflict with data dissemination, which means, in this phase, LiM concurrently floods the information while progressively learning from the dynamics. In the following section, we explain these two phases of LiM in more detail.

3.4. Design of LiM

In this section, we detail the design aspects of LiM. We discuss the basis of LiM: concurrent transmissions and the reinforcement learning scheme, respectively. Besides, we depict two phases of LiM in detail.

3.4.1. Concurrent Transmissions

As derived from Glossy, LiM is based on CT, i.e., constructive interference and the capture effect. LiM adds an action scheme together with a feedback scheme to progressively learn the dynamics of the network. The feedback scheme is based on one specific byte in each packet updated by the neighboring nodes of a node. In the following, We specify the design of the frame structure in LiM. As a consequence, the content of a packet cannot be guaranteed to be identical all the time. By exploiting the capture effect, however, the receiver is able to correctly receive a packet with the stronger signal strength. In other cases, LiM similarly works with constructive interference as Glossy.

Figure 3.4 shows the protocol stack of LiM. LiM operates IEEE 802.15.4 radios (i.e., CC2420) on the physical layer which is integrated with Glossy. On the MAC layer, LiM incorporates a machine learning module on the top of Glossy. That is, the learning module helps to update the feedback based on CT. Meanwhile, the parameters in CT (i.e. Glossy) are configured based on the actions determined by the learning results. As a result, LiM can be considered as an extension of Glossy, namely, it builds the learning scheme consisting of feedback and action selection on a higher layer of Glossy. The application layer can be further developed to meet the users' requirements, for instance, data dissemination. Later in this section, we explain the action and the feedback scheme in more detail.

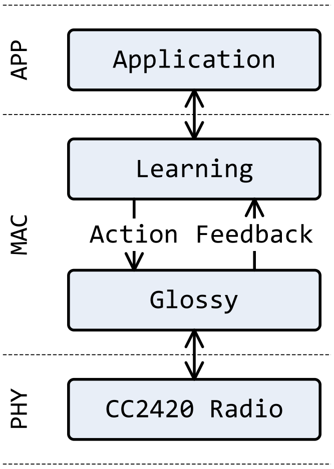


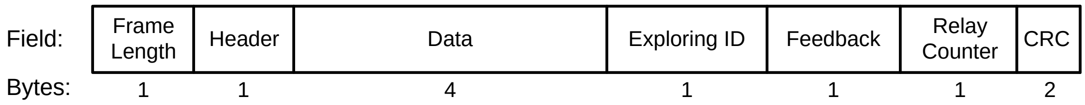
Figure 3.4.: Protocol stack of LiM. LiM builds on CC2420 radio as a physical layer and integrates Glossy on the MAC layer. It exploits an iterative reinforcement learning scheme to select an action based on the feedback. Arrows in the figure refer to the interactions between different layers.

3.4.1.1. Number of Transmissions

By design, LiM maps four configurations of transmission times to four actions respectively: *Action* 0 stands for a node staying only in receiving (i.e., LPL) or sleeping mode, i.e., $N = 0$; *Action* N ($N = 1,\ 2,\ 3$) means that a node works normally except setting the maximal transmission times to N, i.e., transmitting the packet N times. In general, nodes in LiM exploit one of the above-mentioned four actions to effectively reduce the broadcast times in order to improve energy efficiency. Besides, the initiator in LiM is exempted from the action selection. Namely, the initiator maintains the default maximal number of transmission (i.e., $N = 5$, the same as Glossy), and does not execute neither the greedy exploration phase nor the bandit learning phase.

3.4.1.2. Frame Structure

To support the feedback scheme, LiM extends the frame structure in Glossy by adding one byte for an exploring ID field and one byte for a feedback field, respectively. One example of a frame structure is illustrated in Figure 3.5.



Field:	Frame Length	Header	Data	Exploring ID	Feedback	Relay Counter	CRC
Bytes:	1	1	4	1	1	1	2

Figure 3.5.: Application-level frame structure in LiM. By design, the length of the data field (payload) is set to eight bytes in LiM. The exploring ID field is to notify the nodes in the network to proceed to different phases. The feedback field is to carry a response for the learning process.

The frame length stands for the length of the whole frame in bytes. The header is a constant value, e.g., 0xA0 in LiM. The length of the data (i.e., payload) in LiM can be adjusted according to the needs of different applications. By default, LiM sets the payload length to eight bytes. The exploring ID field is to disseminate the ID of the current exploring node: A node that receives the packet is able to notice whether it is the right time for itself to explore or to learn. While a node experiences a packet loss, the feedback field is in use and is updated to a negative feedback value. If not, the feedback is not updated after the data packet has been received from the upper-level nodes, and hence, remains a positive feedback value. The relay counter is inherited from Glossy for concurrent transmissions and time synchronization. A Cyclic Redundancy Check (CRC) is an error-detection field to discover accidental changes to the raw data while transmission in the air.

3.4.1.3. Feedback Scheme

The feedback scheme is one key feature of LiM. Namely, only with the accurate feedback, nodes can make the correct decision of choosing the appropriate action according to the network dynamics.

As mentioned in Section 3.3, LiM consists of two main phases: an exploration phase and a learning phase. The former one is an exploration process where the "redundant" nodes can be discovered so as to reduce the redundant broadcasts. LiM attempts to seek these nodes and then keeps them staying in either receiving mode or sleeping mode in the network. The second phase is a reinforcement learning process. In this phase, a node in LiM floods the information and updates the corresponding actions based on the feedback received from the network, thereby, progressively learning from the dynamics.

To achieve the accurate feedback from neighboring nodes, a node in LiM assigns two types of feedback: a positive feedback (0x01) and a negative feedback (0x00). When a node is in the exploration phase, then it explores whether it is a "redundant" node or not, for instance, node A in Figure 3.2. At first, it stops forwarding the received packets in the current transmission round. As a result, the neighboring nodes, i.e., child node node B and parent node I in Figure 3.2, suffer a packet loss in the current round and accordingly update the feedback byte to a negative one (0x00). In the following round, as a new flooding round comes from node I, the exploring node recovers to receive and forward packets. In this case, it receives the packet with the negative feedback byte written by the neighboring nodes, which have lost a packet in the previous round. Correspondingly, the exploring node, i.e. node A, reads the feedback byte in the packet and then (iii) makes a corresponding decision base on the feedback.

When in the bandit learning phase, the criterion is similar to the one in the exploration phase. If a node is in a learning phase, then it attempts to use different configurations of number of transmissions (N) for each round. Besides, if the neighboring nodes of this node receive at least one packet, then the neighboring nodes update the feedback byte as a positive one (0x01) in the packet. Otherwise, they renew the feedback byte to a negative one (i.e. 0x00). Afterwards, they continue to forward the packet with the up-to-date feedback byte. The learning node subsequently receives the packet and then uses learning algorithm to compute the probability distribution of each action based on the received feedback. After that, the node chooses the corresponding action with the highest probability.

In a few cases, the capture effect could fail to work correctly due to the density of the nodes [LFZ13a]. Another reason is, that the feedback byte from different node is different. For example, when in learning phase, if a positive feedback from "upside" parent node and a negative one from "downside" child node are concurrently transmitted to this learning node, then this could cause an invalid packet so that the feedback is not accurate any more. To overcome this situation, LiM extends one extra listening time slot particularly for the packets from child nodes to obtain a correct feedback, e.g., node B, as shown in Figure 3.6. Figure 3.6 reveals the timeline of one single round in the flooding protocol. In time slot 6, with $N = 3$ node B should have gone to sleep mode because it had already transmitted *three* times in this flooding round. However, since it is in the leaning phase in this round, it keeps listening in time slot 6 and receives an extra packet from its child node, i.e., node C. In next sections, we explain the exploration and learning phases in more detail.

3.4.2. Greedy Exploration

In general, there are two main phases in LiM: (i) a greedy *exploration* phase and (ii) a bandit *learning* phase. In this section, we detail the exploration phase in LiM.

In LiM, after a node is powered on, the first step is, that it keeps its radio on and listening to the communication channel. Meanwhile, a so-called "*initiator*", considered as the information source of a flooding, starts broadcasting packet periodically, similar to Glossy. As a result, the other nodes, which are currently listening, are able to receive the packet from the initiator hop-by-hop. Afterwards, based on the time-stamp of the reception, a node accomplishes time synchronization with the node (its parent node) from which the packet comes. When each node in the network is synchronized, the initialization of the network is then achieved. This is guaranteed by the time synchronization scheme derived from Glossy [FZTS11].

Next, LiM comes to an exploration phase. In this phase, LiM aims to explore the nodes that are not essential for transmitting or forwarding the received packets in the network, i.e., so-called "redundant" nodes. We define this type of nodes as "*absorbing nodes*" that can always stay in either receiving mode or sleeping mode, but not in transmitting mode. Due to the special characteristics of these nodes, the number of absorbing nodes significantly affects the performance of the CT-based flooding protocols, e.g., flooding redundancy.

Determining the trade-off between the number of absorbing nodes and the network

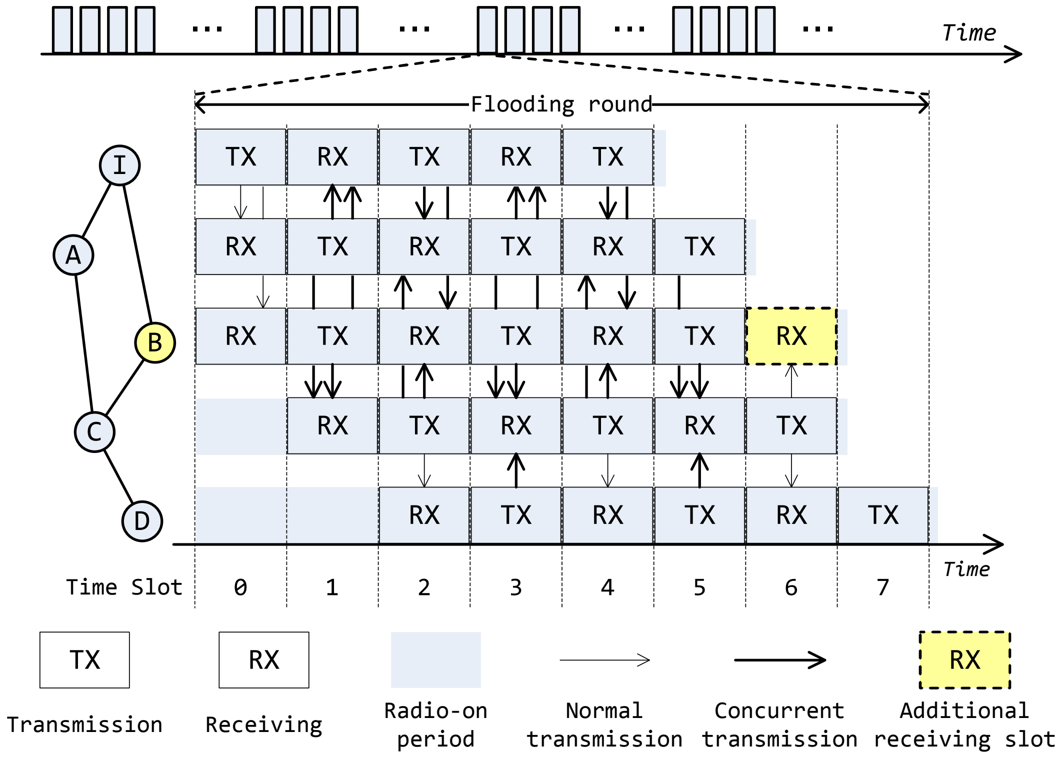


Figure 3.6.: Example of a LiM flooding round with a configuration of $N = 3$ in a topology of five nodes. A data packet is generated in and flooded from node I to all the others. In this round, node B is in a learning phase. LiM compels node B to extend one extra listening time slot for the exploring feedback, particularly for receiving the feedback from child node (i.e., node C).

reliability is quite critical. On the one hand, increasing the number of absorbing nodes decreases the number of concurrent transmitters, consequently strengthening the concurrent transmissions in the network, according to the results in [LFZ13a]: The reliability (i.e., PDR) degrades greatly with the increasing number of concurrent transmitters. On the other hand, maintaining too many absorbing nodes might lead to a fragile network with a higher probability that nodes get disconnected while the environment dynamically changes or the interference suddenly gets harsh.

LiM appropriately explores the absorbing nodes by considering the dynamical environment in the exploration phase, as depicted in Algorithm 1. At first, the initiator generates an exploring list L containing all the node IDs of the whole network, except the initiator itself. Then, the initiator selects one node ID from the exploring list and writes it into the "Exploring ID" field of the data packet (see Figure 3.5). Next, the initiator disseminates this packet for a number of consecutive flooding rounds, e.g., by default with a configuration of $f = 10$ for each single ID in LiM. Then, the initiator removes the ID of the exploring node from the list L. If the exploring list is empty, then the initiator learns that the phase of exploration is complete.

When a node is assigned to be an exploring node by the initiator, it receives a packet that contains its own node ID in the "Exploring ID" field. Thereby, in the current flooding round, it only receives the packets from the neighboring nodes but does not forward any packet, i.e., it acts as an absorbing node. Afterwards, in the following flooding round, the node recovers as a normal node, i.e., it continues to forward the received packets and meanwhile receives the feedback from neighboring nodes — whether they have lost any packet or not. Correspondingly, the node checks the feedback byte of the received data packet and then makes a decision based on the received feedback: If it is negative, then the node considers itself as an essential node in the network; otherwise, the node considers itself as an absorbing node.

Generally, if neighboring nodes of the exploring node lose any packet in the current flooding round, then they update the feedback byte to a negative one (i.e. 0x00) in their received packet and transmit it in the following flooding round. In this case, the exploring node is able to make a decision in the next flooding round based on the feedback from its neighboring nodes.

Algorithm 1 Exploration

```
procedure INITIALIZATION
    initialize a node ID list: $L$
end procedure
procedure ITERATION
    for $L$ is not empty do
        select a node ID $n_i \in L$
        update the "Exploring ID" byte in the packet with $n_i$
        trigger $f$ consecutive rounds of flooding
        remove node ID $n_i$ from $L$
    end for
end procedure
```

Figure 3.7 demonstrates an example after greedy exploration in LiM. Compared to the previous topology shown in Figure 3.6, node B is in the first hop from the initiator, acting as as the child node of node I and the parent node of node C. While in the exploration phase, node B finds out that it is not necessary for itself to forward the received packet to the neighboring nodes. Since with or without it, the neighboring nodes do not lose any packet. Node B decides to act as an "absorbing node", i.e., staying only in receiving mode or sleeping mode, thereby, the link between node B and node C is removed by node B, as shown in Figure 3.7.

Specifically, in LiM, if a node decides to be an absorbing node, then it extends one more slot for LPL on the occasion that it misses the packet in the first slot. This is because, that the time synchronization inherited from Glossy is not fine-grained — it is highly dependent on the clock in CC2420 radio [FZTS11]. Beside, after the exploration phase, a node, once being considered as an absorbing node, can only listen to the channel and receive a packet in the first time slot. In this case, if the clock drifts in an absorbing node, this node might lose the chance to receive any packet, since the time synchronization is not accurate any more and accordingly the node misses the first time slot — its only chance to receive. To overcome this problem, LiM extends one more time slot for listening so as to add a period of guard time, e.g., node B shown in Figure 3.7. It guarantees, that node B has enough time slot to receive a packet regardless of the clock drift problem.

Please note, that in the exploration phase, the initiator in LiM by default does not transmit any real data in the payload except the node ID, since the probability of packet loss is relatively high. However, if users can tolerate the loss, then the initiator can be set to transmit real data also in this exploration phase. With respect to this point, we show the reliability of LiM particularly in the exploration phase later in Section 3.5.

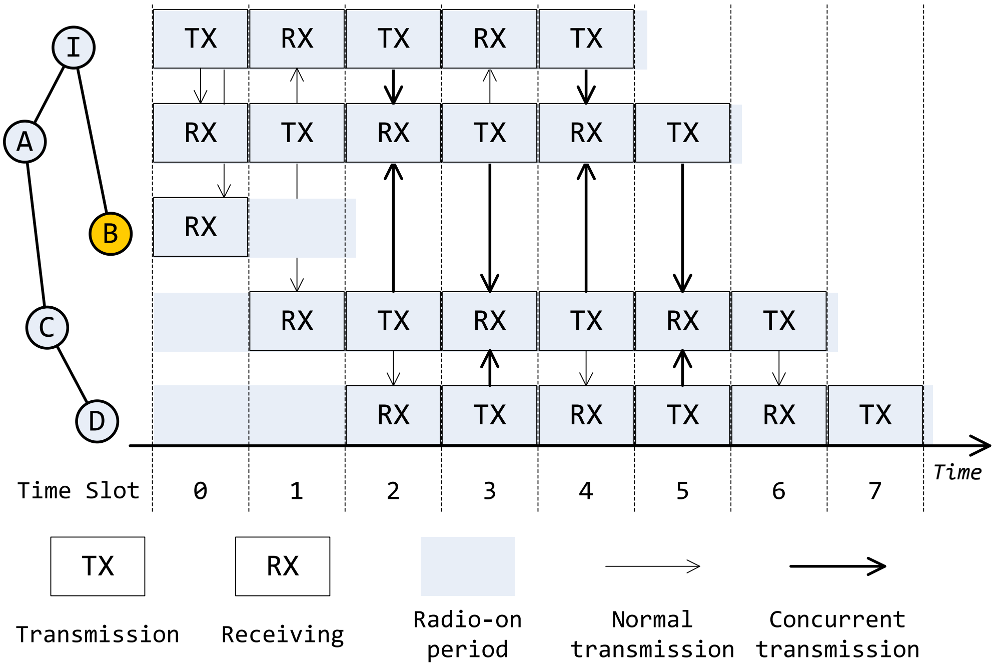


Figure 3.7.: Example of a LiM flooding round with a configuration of $N = 3$ in a topology of five nodes. Nodes self-prune the connection links during the exploration phase. Node B (in yellow) then acts as an absorbing node and stays in receiving mode in a flooding round. In this example, node B attempts to extend the listening time for one more time slot, in case that it misses the packet in slot 0. As a result, node B still saves the energy consumed in four slots compared to the other nodes.

3.4.3. Multi-armed Bandit Learning

Similar to the exploration phase, in the learning phase, LiM attempts to grant each node (except the previous found “redundant” node) to select a corresponding action, i.e., the number of transmission (N). The selection is based on the result of the reinforcement learning in each node — choosing the action with the highest probability. Moreover, the learning phase has no conflict with data dissemination, which means, in this phase, LiM concurrently floods the information while progressively learning from the dynamics. In this section, we detail the learning phase in LiM.

Firstly, we explain the main bandit learning algorithm. As different configurations are mapped to responding actions, we model the optimization problem as a multi-armed bandit problem. In order to overcome this problem, we use one algorithm from the set of multi-armed bandit learning algorithms: *Exp3*. In our case, the goal of the algorithm is to optimize the energy efficiency with reliability based on the policy of selecting transmission times for each sensor node.

Considering a process with K different actions, the *Exp3* algorithm functions as shown in Algorithm 2, where γ is the so-called exploration factor and w_i is the weight of each *action i*. $p_i(t)$ is the probability of selecting *action i* in flooding round t, and $x_i(t) \in [0, 1]$ is the reward of *action i* on flooding round t, while T means the total number of iterations.

At the beginning, the algorithm initializes the exploration parameter γ. This parameter adjusts the possibility that the algorithm attempts to explore other actions while a certain action has already achieved the highest probability, i.e., trying new actions regardless the one with converged probability. Next, the algorithm associates a weight with each action in order to give each action a probability to form a probability distribution $\mathcal{P}$ over all actions.

Algorithm 2 Exp3

```
procedure INITIALIZATION
    initialize $\gamma \in [0, 1]$
    initialize the weights $w_i(1) = 1, \forall i \in \{1, \dots, K\}$
    set $p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}, \forall i \in \{1, \dots, K\}$ for the distribution $\mathcal{P}$
end procedure
procedure ITERATION
    for $t < T$ do
        draw the next action $i_t$ randomly according to the distribution $\mathcal{P}$
        observe the reward $x_{i_t}(t)$
        define the estimated reward $\hat{x}_{i_t}(t) = x_{i_t}(t)/p_{i_t}(t)$
        set the weight of action $i_t$: $w_{i_t}(t+1) = w_{i_t}(t)e^{\gamma \hat{x}_{i_t}(t)/K}$
        set all other weights: $w_j(t+1) = w_j(t), \forall j \neq i_t \text{ and } j \in \{1, \dots, K\}$
        update $\mathcal{P}$: $p_{i_t}(t+1) = (1-\gamma)\frac{w_{i_t}(t+1)}{\sum_{j=1}^{K} w_j(t+1)} + \frac{\gamma}{K}, \forall i_t \in \{1, \dots, K\}$
    end for
end procedure
```

After the exploration phase, the algorithm iterates T times the learning procedure in order to learn from the environment and to generate an accurate probability distribution to receive more accumulative rewards, so that the probabilities of various actions can have enough time to be converged. In the iterative learning procedure, at first, the algorithm randomly selects an *action* i_t as the next executive action based on the distribution $\mathcal{P}$. Then, it executes the *action* i_t, and receives a reward $x_{i_t}(t)$ from the environment. Specifically, hereby in LiM, the reward from the environment implies the feedback from neighboring nodes in the network. Thereafter, an estimated reward $\hat{x}_{i_t}(t)$ is calculated as $x_{i_t}(t)/p_{i_t}(t)$ to further include the influence of the probability on the reward. In the end, the weight of the sampled action is updated, while the weights of other actions

(w_j, $\forall\ j \neq i_t,\ j \in \{1, \ldots, K\}$) remain unchanged. While the algorithm converges, the eventual probability distribution $\mathcal{P}$ over different actions is considered to be the guidance to select the best action in order to maximize the reward.

To integrate Exp3 in LiM, each action in this algorithm is associated with a corresponding configuration in each node, which is mentioned in Section 3.4.1. In each iteration, the probability of selecting a certain action is calculated based on the feedback from the neighboring nodes. For instance, there are three actions ($K = 3$) in the learning procedure of LiM by design, i.e., *action* 1, 2, and 3. Respectively, *action* 1, 2, and 3 are mapped to three different configurations, where nodes transmit the packet *once*, *twice* or *three times*, respectively. That is, if the randomly sampled *action i* is 1, then the node only transmits *once* in the current round. After the node receives the feedback, the weight of the corresponding action (i.e., *action* 1) is updated as shown in Line 11 of the algorithm. The weights of other actions (i.e., *action* 2 and 3) stay the same (in Line 12). In the final step, the distribution $\mathcal{P}$ is updated to prepare for the next iterative flooding round according to the formula in Line 13 of the algorithm.

At this point of time, one learning iteration has been performed. This iteration phase continues until the number of flooding rounds reaches T. By design, LiM sets this value to $T = 200$, i.e., a fixed learning period for each node in LiM. However, due to the dynamic environmental changes, a fixed period might cause a case that the probability of choosing an action does not converge. We detail and evaluate this case later in Section 3.4.4.2.

Similar to the greedy exploration phase depicted in Algorithm 1, the initiator maintains an exploring list L containing all IDs of the nodes in the whole network, except the initiator itself and the absorbing nodes. Here in the learning phase, f equals to T, i.e., 200 consecutive flooding rounds. After the learning phase of one particular node (when T reaches 200), the initiator randomly assigns another node to learn by exploiting the algorithm. This is achieved by the “Exploring ID” field of the data packet: The initiator floods the packet containing the “Exploring ID” for 200 consecutive flooding rounds and then removes the ID of the exploring node from the list L. If the exploring list L is empty, then it means that the phase of learning is done.

Correspondingly, when a node receives the exploring ID information in a data packet, it knows wet her this flooding round is its turn to start learning or not. In the end, after all nodes (except the initiator and the absorbing nodes) have completed their learning phase, the learning procedure finishes and then all the nodes mainly focus on data flooding using their own actions.

Figure 3.8 demonstrates the timeline of an example where all nodes have determined their own actions. After the two main phases of greedy exploration and bandit learning, nodes maintain their own actions accordingly. For instance, node B and D consider themselves as the absorbing nodes, while node A and C choose the *action* 3, i.e., transmitting only three times in one flooding round. As the figure depicts, LiM reduces eight time slots in total (four from node B and four from node D) compared to our baseline Glossy, thereby, improving energy efficiency of the network.

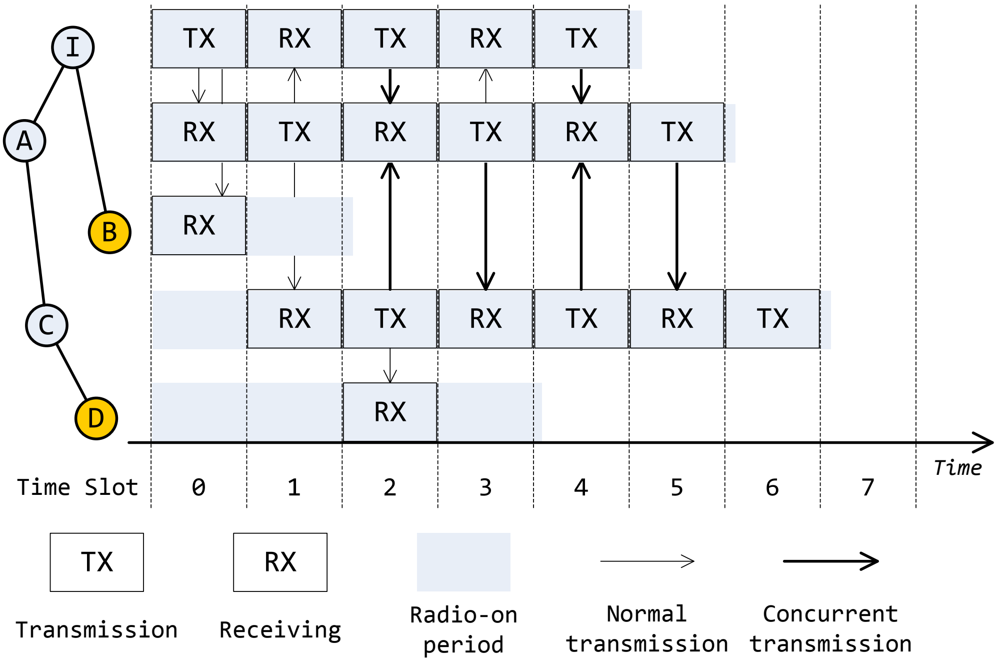


Figure 3.8.: Example of a LiM flooding round with a configuration of $N = 3$ in a topology of five nodes. Nodes self-determine the actions based on the results of their learning phase. This example shows the final state after all sensor nodes have completed their learning phase. Node A and C choose action 3 ($N = 3$) to ensure the reliability, while node B and D (in yellow) act as absorbing nodes. Node D hears nothing in time slot 3, since there is no neighboring nodes on a lower level (as child nodes of node D).

3.4.4. Implementation Aspects

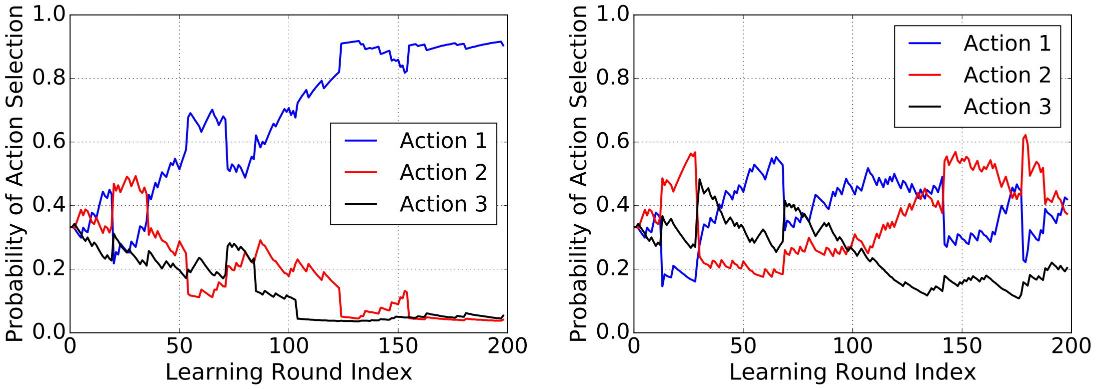
In this part, we give several additional implementation aspects of LiM: destructive action and non-converging case.

3.4.4.1. Destructive Action

In the bandit learning phase, the nodes — except the initiator and the absorbing ones — learn to make a decision based on the feedback they receive. By trying *action* 1, where a node only transmits the received packet once, receivers might miss the packet so that the reliability of the whole network degrades. In reality, this is sometimes harmful to the reliability of the whole network. Because of the dynamics in the environment, e.g., interference, this packet has a higher possibility of getting lost since it is only transmitted once. Consequently, the nodes which are far away from the initiator would suffer a packet loss with a relatively high probability. To avoid this negative effect, LiM conservatively learns to select *action* 1. Namely, if a node in LiM (i) gets a negative feedback of the exploring action in the previous round and (ii) this specific action is *action* 1, then the node abandons selecting *action* 1, i.e., it stops exploring *action* 1. The mechanism leads LiM to make a relatively conservative decision of choosing *action* 1.

3.4.4.2. Non-converging Case

Practically, the learning procedure in a node may not always converge: The learning duration might not be long enough for the node to clearly distinguish the difference of the probabilities on selecting different actions. That means, at the end of the learning phase, the derivation between each probability might not be large enough. Figure 3.9 demonstrates two cases of the convergence of different probabilities of selecting corresponding actions in the learning algorithm. In Figure 3.9(a), the probability of selecting *action* 1 converges and the node chooses *action* 1 as its final decision. Comparatively, as shown in Figure 3.9(b), the probability of choosing *action* 1 and *action* 2 is almost the same — both of which are higher than the one of choosing *action* 3. That means, the node might take a wrong decision according to the final probability distribution, since the probabilities of different actions are not converged. In case of this exception, to be more conservative, LiM selects *action* 3 with a maximal transmission of $N = 3$ in order to maintain a good reliability instead of aggressively reducing the energy consumption further. Note, that this is an example of a special case in FlockLab topology. By default, LiM defines the learning round to be $T = 200$ in order to avoid most non-converging cases.



(a) Converging case.

(b) Non-converging case.

Figure 3.9.: Two convergence cases of a learning phase in LiM. In (a), action 1 dominates at the end of the learning phase, while in (b), action 1 and action 2 still compete with each other in the end.

3.5. Performance Evaluation

In this section, we provide an evaluation of LiM based on a number of experiments in a real-world testbed — the FlockLab testbed [LFZ+13b].

3.5.1. Methodology

Similarly, in this work, we use the FlockLab testbed [LFZ+13b] for our experimental evaluation. The topology of the testbed is shown in Figure 2.6 in Chapter 2. We use

28 sensor nodes (observers) out of 30 existing ones (except node 201 and 202) in the testbed, since they had not been constantly available during our experiments.

To fairly evaluate the performance of the protocols, we use different nodes (i.e., node 1, 16, and 20) as the initiator, respectively, in different scenarios. Besides, we vary the transmission (TX) power level as −7, −3, and 0 dBm in different scenarios. Correspondingly, various levels of TX power result in different network sizes, i.e., different hops. For instance, using a 0 dBm TX power leads to a network with three hops, which means, that all the nodes in the network can be reached within three hops. The default wake-up frequency of all protocols is set to 4 Hz. That means, a packet with eight bytes payload is generated and transmitted by the initiator every 250 ms. We perform three independent runs for each experimental configuration throughout this chapter. Each run lasts 45 minutes, in which there are over 10000 packets generated by the initiator and flooded through the whole network. All the experimental results are averaged over these three runs and the standard deviations are revealed by error bars. The evaluation setting is summarized in Table 3.1.

Table 3.1.: Evaluation settings in the FlockLab testbed.

Initiator IDs	**TX Power (in dBm)**	**Network Size (in hops)**
1, 16, 20	−7, −3, 0	from 8 to 3

A number of the state-of-the-art protocols integrate a specific application layer with Glossy. It might not be fair to compare LiM to them since LiM is not application-specific. Therefore, in this the work, we only compare LiM to the our baseline Glossy, in various scenarios. However, LiM can be easily integrated with a specific application, e.g., a data dissemination application in the application layer.

We focus on three key metrics to evaluate the performance and to draw a comparison, i.e., PDR, radio-on time, and latency. Those are also used by Glossy [FZTS11]) as performance metrics to evaluate the performance.

3.5.2. Impact of Number of Transmissions

In this part, we analyze how the performance metrics are affected by the number of transmissions (N) of a node during network flooding. Firstly, we run the experiments of our baseline, Glossy, and vary N as 1, 3, and 5, respectively.

As LiM flexibly tunes N according to the learning experience, LiM starts with $N = 5$. Glossy constantly sets N to 5 by default. In this part, for both protocols, we set node 1 — a node on the edge of the network — as the initiator, and use various transmission powers of −7, −3, and 0 dBm, respectively, resulting in a WSN with different diameters.

Figure 3.10 revels the performance comparison between Glossy and LiM. For reliability, Glossy achieves extremely high PDRs even with various N values. LiM is able to maintain this advantage of Glossy: Regardless of the different levels of TX power, LiM it achieves a PDR of over 99.85%. However, in Glossy, the PDR changes with the level of TX power: With a higher level of TX power, Glossy achieves a higher PDR. With the configurations of TX power equals to −3 and 0 dBm, LiM achieves a PDR of 99.80% similar to the one of Glossy with $N = 1$ in both cases.

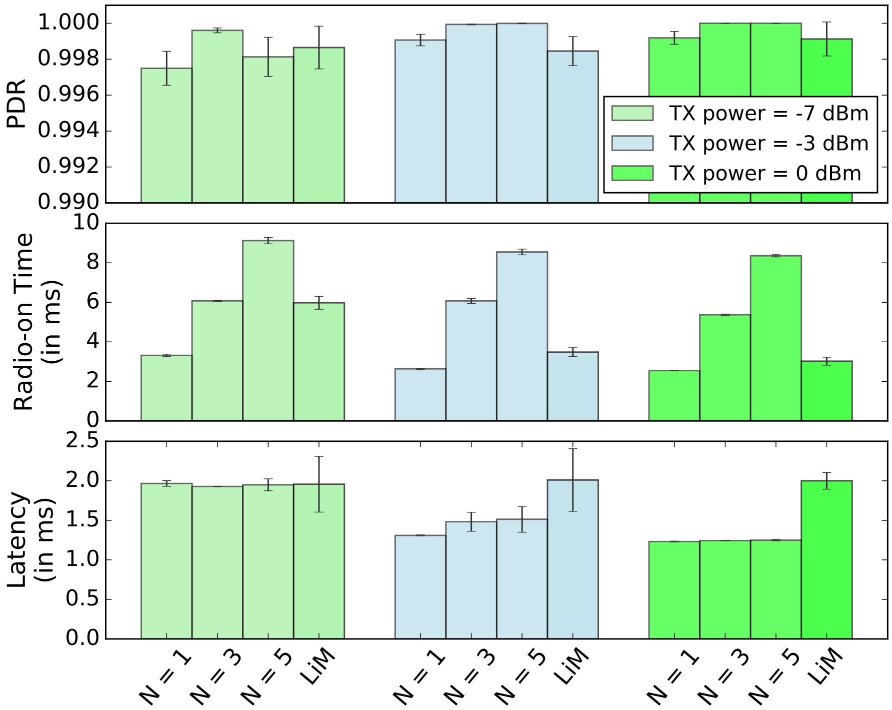


Figure 3.10.: Performance metrics of Glossy with various N values and of LiM, respectively. Both protocols set node 1 as the initiator and use the transmission power of -7, -3, and 0 dBm. LiM inherits the advantages from Glossy in terms of high reliability with various levels of transmission power, while effectively reducing the radio-on time. The longer latency in LiM stems from the overall processing time in the bandit learning phase.

Moreover, LiM succeeds in reducing unnecessary broadcast redundancy, resulting in a notable decreased portion of radio-on time, compared to Glossy with $N = 3$ and 5. The radio-on time is even close to the one of Glossy with $N = 1$ by using -3 or 0 dBm as transmission power level. Even with a transmission power of -7 dBm, LiM is able to carry out a radio-on time similar to the one of Glossy with $N = 3$. With this configuration, the network might consist of 8 hops of nodes. As a side effect, the communication links in the network are more vulnerable because of a lower TX power level, e.g., interference in the environment might influence the communications. In this scenario, according to the radio-on time shown in the figure, we confirm, that LiM actively chooses the most conservative action — *action* 3 ($N = 3$) to maintain a reliable performance.

For the flooding latency, LiM does not perform better than Glossy in all three scenarios. This is because, that LiM has to spend more time for data processing, decision making, and probability calculation, consequently leading to a latency of approximately 2 ms in all cases. However, even with an end-to-end network latency of 2 ms, this could still meet a lot of mission-critical applications' requirements.

Please note, that in reality, Glossy with $N = 1$ may have a bootstrap problem and experience a highly fragile network, according to our experience from the experiments that we carried out. We argue that LiM aims to interactively learn from the environment and, thus, makes a decision of N to progressively reduce the broadcast redundancy while maintaining acceptable levels of reliability, energy efficiency, and latency. In Section 3.5.4, we take a closer look at how LiM determines the number of transmissions (N).

3.5.3. Impact of Topology

In this section, we evaluate LiM with different positions of the initiator. We change the position of the initiator (i.e., node 1, 16, and 20, respectively) to alter the flooding diameter of the network, making the topology different in each set of experiments. While the position of the initiator changes in the testbed, the network topology varies as well: Since the data source of a flooding changes, the absolute hop distance between a node and the source also changes. The idea of changing the position of the initiator is to evaluate whether a protocol is dependent on a specific topology or not. Besides, we exploit the transmission power of −7 dBm to result in a network with approximately eight hops.

Figure 3.11 illustrates the results of LiM with various positions of the initiator (I), i.e., positions of I are node 1, 16, and 20, respectively. For the reliability, LiM achieves a PDR of over 99.70% in all scenarios. The radio-on time and latency change slightly along with the position of the initiator. However, LiM maintains a less than 6 ms radio-on time and a less than 2 ms end-to-end latency only with a transmission power of −7 dBm regardless of the initiator's position.

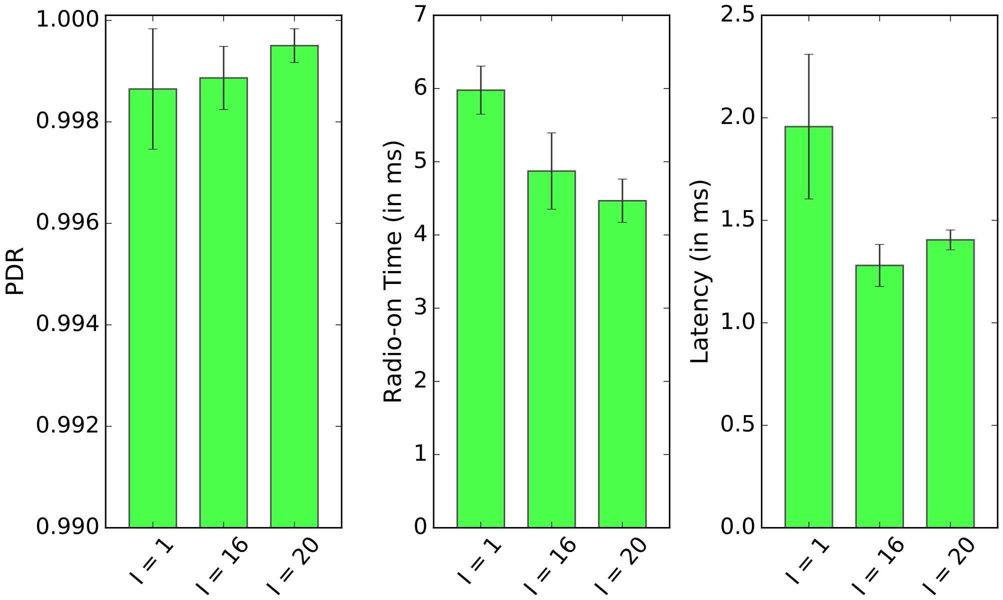


Figure 3.11.: Performance metrics of LiM with various initiator positions using a TX power of −7 dBm. Even with relatively weak link connections, the network can still provide high reliability, low radio-on time, and low latency, regardless of the initiator's position.

3.5.4. Impact of Exploration Phase

In this part, we discuss the influences of the absorbing nodes and the reliability drop in the exploration phase, respectively.

3.5.4.1. Absorbing Nodes

In general, the main part of redundancy reduction is contributed by the exploration phase in LiM, where the absorbing nodes are discovered. These absorbing nodes stay in a receiving mode and do not forward a received packet after waking up from the sleeping mode. Figure 3.12 shows the average number of each node in LiM from one experiment with node 1 as the initiator and 0 dBm transmission power. As the figures shows, the absorbing nodes are successfully discovered in this case. The nodes that have no transmission are actually the absorbing nodes, as the other nodes are the set of backbone nodes of the network.

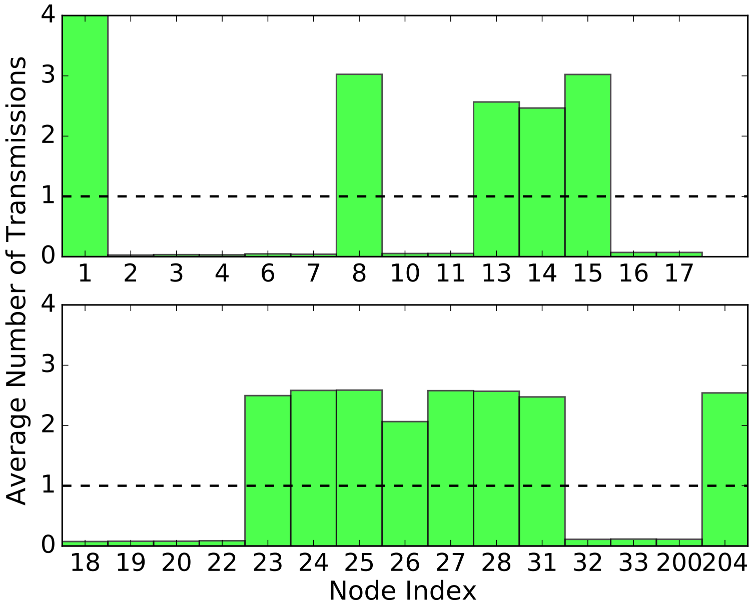


Figure 3.12.: Average number of transmissions in each node with node 1 as the initiator and 0 dBm TX power. The dotted line shows the overall average number of transmissions. A node in LiM starts with N equal to 5 and progressively determines N: The average value of N in each node is determined by itself. After the learning phase, LiM successfully finds all absorbing nodes and eventually obtains an average N equal to 1 as the dotted line shows.

3.5.4.2. Reliability Drop

A node in the exploration phase exploits a temporary "log-out" strategy, where it does not transmit a received packet for a flooding round, so as to review whether it is an absorbing node. In this case, the network reliability cannot be guaranteed to be 100%. However, by default, LiM only reserves ten flooding rounds for each node to explore its role. Additionally, during these rounds, a node only logs out of the network for one particular round (out of 10), where the node only stays in receiving mode and does not transmit. Consequently, assuming there are 30 nodes in a WSN and that nodes are well synchronized, then a node can lose 29 packets out of 300 (i.e., 10×30) in the worst case, i.e., PDR equals 90.33% (i.e., $(300 - 29)/300$). Figure 3.13 illustrates the dynamically changing PDRs of all the nodes in the testbed along with the running time. As shown, even though several nodes suffer a packet loss during the exploration phase, they are still able to maintain a high reliability afterwards.

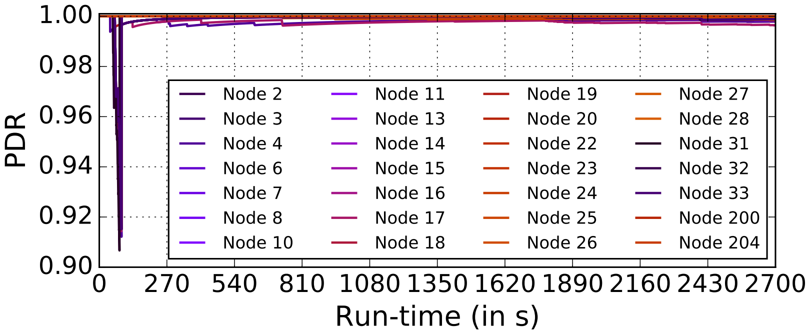


Figure 3.13.: PDRs of all nodes in FlockLab changing over run-time. PDR drops during the greedy exploration phase while finding all the absorbing nodes, but it is still higher than 90%. In this case, users can decide whether to put important or dummy data in the payload during the exploration phase according to application-level requirements.

3.5.5. Discussion

To summarize, in this section, we demonstrated the performance of two different protocols: LiM and the baseline protocol Glossy, respectively, in various evaluation scenarios. The experimental results revealed that LiM effectively inherits the benefits from concurrent transmission. It delivers a high end-to-end reliability of over 99.70% with an average end-to-end latency of less than 2.4 ms in all cases. More importantly, LiM is able to decrease the radio-on time to less than 6 ms step-by-step, and significantly reduces broadcast redundancy. Even with different topologies, LiM is able to manage a high reliability with low end-to-end latency, while reducing unnecessary communication redundancy. Table 3.2 shows the memory usage, i.e., Random-Access Memory (RAM) and Read-Only Memory (ROM), of LiM and Glossy, respectively, in Contiki OS 2.7. Even Equipped with a reinforcement learning scheme, LiM does not increase a lot of the memory usage compared to Glossy.

Table 3.2.: Memory usage of different protocols in Contiki OS 2.7.

Protocol	**Code Footprint (Kb)**	
	ROM	RAM
LiM	22.554	0.526
Glossy	17.900	0.432

Finally, Table 3.3 summarizes all the experimental results of different scenarios in this chapter. Please note, that we do not include the experiments of tuning the initiator's position for Glossy, since we focus on the self performance comparison of LiM in this chapter. For the interest of this part of Glossy, we refer the reader to the Glossy paper [FZTS11] and LWB paper [FZMT12].

3.6. Conclusion

This chapter introduces LiM, a machine learning-based flooding protocol for low-power duty-cycled WSNs. LiM applies a multi-armed bandit learning scheme in CT-based flooding, thereby benefiting from both. Concurrent transmissions ensure LiM a highly reliable communication with low end-to-end latency and low energy cost. Machine learning brings the adaptation ability to deal with the dynamics of the environment, thereby further improving energy efficiency. We implement our protocol in Contiki OS and evaluate it with extensive experiments in a real-world testbed FlockLab. Our experimental evaluation shows that LiM achieves less radio-on time, and — as a consequence — it greatly improves energy efficiency of the network. Meanwhile, LiM manages a more than 99.70% average end-to-end reliability and a less than 2.4 ms average end-to-end latency in all experiments in the testbed. Furthermore, with its learning ability, LiM maintains a flexible adaptation to the dynamics of the network, when compared to the baseline protocol Glossy. To sum up, LiM inherits the benefits from concurrent transmissions and a machine-learning scheme, outperforming the baseline protocol Glossy in the light of energy efficiency while maintaining a high end-to-end reliability and low latency.

Table 3.3.: Summary of experimental results. In all experimental scenarios, LiM maintains a best-effort end-to-end PDR and latency, taking advantage of concurrent transmissions. Besides, LiM decreases approximately 5 ms radio-on time compared to the default configuration of Glossy. The maximal value of radio-on time in Glossy is $N = 5$, while the minimum value is $N = 1$. LiM reduces the radio-on time by at least 3 ms compared to the default setting of Glossy (i.e., $N = 5$).

Protocol	**Initiator ID**	**TX Power**	**PDR (%)**	**Radio-on Time (ms)**	**Latency (ms)**
Glossy	Node 1	0 dBm	99.97 (max: 100.00 min: 99.88)	5.42 (max: 8.41 min: 2.53)	1.24 (max: 1.26 min: 1.23)
		-3 dBm	99.97 (max: 100.00 min: 99.88)	5.61 (max: 8.71 min: 2.61)	1.44 (max: 1.69 min: 1.30)
		-7 dBm	99.84 (max: 99.97 min: 99.66)	6.17 (max: 9.26 min: 3.25)	1.95 (max: 2.03 min: 1.89)
LiM	Node 1	0 dBm	99.91 (max: 99.98 min: 99.80)	3.02 (max: 3.22 min: 2.82)	2.00 (max: 2.12 min: 1.92)
		-3 dBm	99.84 (max: 99.92 min: 99.76)	3.48 (max: 3.69 min: 3.25)	2.01 (max: 2.34 min: 1.44)
		-7 dBm	99.86 (max: 99.95 min: 99.73)	6.00 (max: 6.33 min: 5.68)	1.95 (max: 2.33 min: 1.62)
	Node 16	0 dBm	99.97 (max: 99.99 min: 99.94)	3.89 (max: 4.70 min: 3.14)	0.91 (max: 1.06 min: 0.80)
		-3 dBm	99.84 (max: 99.91 min: 99.76)	3.64 (max: 4.07 min: 3.27)	1.53 (max: 1.58 min: 1.48)
		-7 dBm	99.89 (max: 99.96 min: 99.84)	4.87 (max: 5.38 min: 4.34)	1.28 (max: 1.38 min: 1.18)
	Node 20	0 dBm	99.91 (max: 100.00 min: 99.79)	2.96 (max: 3.11 min: 2.83)	0.94 (max: 1.12 min: 0.87)
		-3 dBm	99.93 (max: 99.94 min: 99.92)	3.11 (max: 3.20 min: 2.98)	1.18 (max: 1.23 min: 1.07)
		-7 dBm	99.95 (max: 99.98 min: 99.92)	4.47 (max: 4.77 min: 4.18)	1.40 (max: 1.44 min: 1.35)

4 Concurrent Transmission-based Collection

4.1. Introduction

In smart city scenarios, hundreds or thousands of distributed battery-powered sensor nodes communicate wirelessly over single or multiple hops. These nodes cooperate together in order to fulfill various tasks and to provide an acceptable level of QoS to the users. However, owing to the resource-constrained nature of sensor nodes, providing satisfactory application-level QoS in WSNs is extremely challenging, especially in multihop, low-power, and lossy networks. For instance, mission-critical applications — such as data collection in smart grids — usually require not only high energy efficiency and high reliability but also ultra low latency.

Rising as a novel communication paradigm, Concurrent Transmission (CT) is introduced by Ferrari et al. [FZTS11]. As the first CT-based flooding primitive, Glossy [FZTS11] has caught much attention in the community. By exploiting a receive-and-forward scheme, sensor nodes transmit an identical packet at the same moment (e.g., within 0.5 μs) to achieve a constructive interference and the capture effect [FZTS11]. It has proven, that CT allows messages to be propagated in the whole network in several milliseconds without any severe collisions. Glossy achieves energy-efficient flooding without causing broadcast storm problems and, thus, does not require any routing information in the network. Basically, Glossy is based on an accurate time synchronization to realize CT. In this respect, it differs from the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)-based IEEE 802.15.4 standard, e.g., used by IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [Win12].

Data collection, as a many-to-one scenario, is one of the most important WSN-based applications. This scenario requires that sensor nodes (also referred to as source node), where the data are generated, e.g., temperature, humidity, vibration, and rotation, transmit the data packets to a so-called sink (node). Acting as a central base station, the sink collects and processes all the data packets from the sensor nodes throughout the network, so as to fulfill the data collection tasks. The state-of-the-art collection protocols, such as CTP [GFJ$^+$09], RPL [Win12], Opportunistic Routing for Wireless sensor networks (ORW) [LGDJ12], Opportunistic RPL (ORPL) [DLV13], and MOR [ZLT17] stick to CSMA/CA to deal with collisions. Inevitably, the back-off scheme of the CSMA/CA approach introduces longer latency and the re-transmission introduces higher overhead. In contrast, CT-based data collection has the potential of effectively providing high end-to-end reliability and low latency. This is, because CT-based protocols do not require MAC-layer collision avoidance and network-layer routing, as presented in the LWB [FZMT12] and Chaos [LFZ13a], since CT replies on constructive interference and the capture effect. In other words, CT counts on collisions but the constructive ones, which come from the tight time synchronization and the identical packet through the whole network in a flooding round.

Nevertheless, current CT-based protocols used for data collection have to deal with two main issues: (i) a flooding-based communication primitive and (ii) network-level scheduling. This type of communication primitive requires that the packet must be identical in one flooding period within the whole network, so that constructive interference can function. Besides, the capture effect is exploited to correctly receive a packet with the stronger signal strength, when multiple senders are concurrently transmitting [LFZ13a]. As a consequence, certain source nodes can never successfully transmit their packets to the receiver if they are not well scheduled globally. Even worse, when too many nodes concurrently transmit different packets, then the destructive collision occurs more often. That leads to a decreasing PDR, a broken flooding process, and clock desynchronizations.

To address this problem, Ferrari et al. [FZMT12] designed an application-level scheduling mechanism for multiple source nodes by a dedicated host in the network. The host centrally computes a schedule and periodically distributes the schedule to all nodes in the network to coordinate the traffic. This technique, however, inevitably introduces a collision period in the beginning of network initialization to construct a global scheduler in a many-to-one communication pattern. Moreover, LWB [FZMT12] triggers periodical synchronization with identical packets in the network to accurately achieve global synchronization. Consequently, the schedule-based many-to-one communication approach increases the complexity of the CT-based network. We argue that there could be a technique to achieve CT-based data collection without any scheduler. We propose a new communication primitive — *Packet-in-Packet (PiP)* — for timely and reliable data collection in multihop low-power WSNs.

PiP uses concurrent transmissions, i.e., constructive interference and the capture effect, to inherit the advantages: high reliability, low latency, and low energy consumption. PiP is able to grant accurate clock synchronization even with different packets in the network. Moreover, PiP exploits packet concatenation to achieve concurrent data collection from multiple neighboring nodes in a single Transmit (TX) slot. By nature, PiP concatenates multiple packets from neighboring nodes in the air. Therefore, it requires neither a global scheduler nor a dedicated setup period. From the network perspective, PiP coordinates the concatenated packet that contains multiple packets from various source nodes as a “higher-priority” packet to the sink. As a result, PiP reduces data collection time and maintains high reliability and high energy efficiency compared to the state-of-the-art.

The remainder of this chapter is organized as follows. Section 4.2 discusses related work, with foci on CT-based protocols and packets in a packet in low-power wireless networks. Section 4.3 explains the basics of our proposed protocol and provides a brief overview. Section 4.4 details the design of PiP, followed by a performance evaluation in Section 4.5. Section 4.6 provides concluding remarks.

4.2. Related Work

In this section, we review a number of existing research work in terms of CT-based protocols in WSNs and the concept of “packets in a packet” in WSNs. Next, we summarize the related work.

4.2.1. CT-based Protocols

Network flooding is one of the most fundamental services in WSNs. It forms the basis for a wide range of applications and network operations. Glossy [FZTS11] provides a fast and efficient network flooding service by using concurrent transmissions in WSNs. By exploiting constructive interference and the capture effect on the physical layer, Glossy is able to get an average packet delivery ratio of 99.99% and ultra-low latency in real testbeds.

Splash [DCL13] builds a tree pipeline [RCBG10] by exploiting Glossy, thereby improving channel utilization. Furthermore, Pando [DLZL15] integrates fountain code [Mac05] (also known as rateless erasure codes) with pipelining to further improve the reliability of Splash. While Glossy disseminates one packet in each communication round, Splash and Pando are designed to deliver large data objects to all nodes in a WSN, e.g., for the purpose of reprogramming or firmware update the WSN-based applications. Ripple [YH15] also relies on Splash and network coding techniques to improve particularly in terms of network throughput.

However, Glossy aims to provide highly reliable flooding for one-to-many applications. Thus, it is not applicable for many-to-one applications such as data collection. To realize the design of many-to-one applications with Glossy, Ferrari et al. added an application-level scheduler to construct a so-called Low-power Wireless Bus (LWB) [FZMT12]. LWB centrally schedules the data communication to support one-to-many, many-to-one, and many-to-many traffic patterns in WSNs. Moreover, Chaos [LFZ13a] builds on Glossy to achieve fast all-to-all data sharing in a distributed manner. Chaos further combines programmable in-network processing with concurrent transmissions in WSNs. Meanwhile, Suzuki et al. proposed a reliable data collection protocol based on Glossy for many-to-one applications named Choco [SYM13]. By their testbed evaluations, they argued that Choco can provide more energy-efficient collections than the state-of-the-art and achieves high end-to-end reliability.

Apart from aforementioned protocols, there are a number of efforts to construct one-to-one application on Glossy. For instance, Carlson et al. propose CXFS [CCT$^+$13], a forwarder selection mechanism for concurrent transmissions. In CXFS, sensor nodes use a hop count in each packet to get their relative distance to each other. CXFS builds on Glossy and supports point-to-point transmissions while achieving high reliability, high energy efficiency, and high throughput. Moreover, Sparkle [YRH14] selects subsets of nodes that participate in Glossy-based flooding. It also supports one-to-one communication. Similarly, RTF [ZRHK15] further extends Sparkle and exploits TDMA for data scheduling to improve reliability and energy efficiency in point-to-point traffic. RFT identifies reliable relay nodes to limit the number of concurrently active neighbors to save more energy. More recently, LaneFlood [BLS16] builds on Glossy and further integrates the forwarder selection scheme of CXFS with application-level network protocols in WSNs. LaneFlood thus supports one-to-one traffic, forwarder selection, and standard protocols in IoT such as TCP/UDP and the constrained application protocol (CoAP) — a specialized Internet Application Protocol for constrained devices, as defined in RFC7252 [She14].

4.2.2. Packets in a Packet

Santhapuri et al. investigated link-layer opportunities and challenges towards harnessing the feature of a so-called message-in-message (MIM) in [SCM+08]. They proposed an MIM-aware framework that reorders transmissions to enable concurrent communications. By continuously scanning for sync-words, their technique allows a transceiver to disengage from the reception of an ongoing transmission and to lock onto a newly-started stronger one.

The usage of in-band signaling to denote the start of a packet is almost universal in digital radios. Authors in [GBM+11] demonstrate that by exploiting this this property, in-band radio signaling mechanisms can be abused to inject raw digital frames. From the perspective of security, they provide a few tested examples of raw frame injection for digital radios such as IEEE 802.15.4 radio. Besides, they show that an attacker can inject frames into a packet, if he/she is able to predict the on-air pattern produced by encapsulated data.

More recently, König et al. [KW16] proposed a class of wireless transmission schemes that decouple synchronization headers from payloads in order to create new transmission primitives of a second sender. By only transmitting a synchronization header, a node is able to let the neighboring nodes receive fragments of a packet. Besides, portions of the payload of longer ongoing packets can be overwritten by exploiting the capture effect. The authors then explored two scenarios potentially benefiting from such schemes: crossing a network “chasm” (i.e., a virtual or physical gap between two parts of the network where communication links are very poor) and insertion of high-priority packets. Further, they examined the technique in these two scenarios. Results in testbeds showed that the packet-received ratio can be increased from 5% to up to 30% in the case of network chasm. In the second scenario, the successful decoding of the injected packet in up to 70% of the cases could be reached.

4.2.3. Summary

Over the years, CT has been proven to be an efficient approach to improve the timeliness and reliability, because packets can positively overlap rather than collide in the air. CT is able to achieve good performance for one-to-many scenarios due to the natural property of flooding. For many-to-one scenarios — e.g., data collection, the most popular applications of a WSN — CT can be only implemented with additional scheduling or content compression on the application layer. However, for the applications requiring original data in a dynamic network, scheduling in advance and content compression sometimes hardly work. PiP, on the contrary, is suitable for this type of scenarios. Original data collection is supported by PiP without any scheduling or compression. So far, to the best of our knowledge, there is no application working in the manner as PiP does in practice.

4.3. Overview

In this section, we explain the prerequisites of concurrent transmission in low-power WSNs — constructive interference and the capture effect. Then, we briefly introduce the concept and background of LiM. Last, we give a general review of PiP.

4.3.1. Concurrent Transmission

Glossy [FZTS11] has boosted much new research interest in wireless network communities, as one of the most representative CT-based flooding protocols in the community. Generally, it exploits constructive interference and the capture effect: It uses *constructive interference* to superimpose (identical) packets; and it exploits the *capture effect* to ensure, that a receiver successfully receives a packet and correctly demodulates the packet while multiple senders simultaneously transmitting. Using these two mechanisms, Glossy is able to manage a highly reliable flooding and an accurate time synchronization [FZTS11].

In Glossy, nodes in the network get time-synchronized to a so-called "initiator" — from which the data flooding is initiated — in order to estimate the time to wake up and listen to the channel. Moreover, nodes concurrently transmit packets in a "receive-and-forward" scheme, which means whenever a node receives a valid packet, then it immediately forwards the packet to all the neighboring nodes. In this way, messages are propagated through the network without contention of the wireless medium. To some extend, Glossy makes use of contention due to the nature of constructive interference and the capture effect. Ferrari et al. [FZTS11] argue, that the latency of flooding could approximately reach the theoretical upper bound since every node in the network is in a receive-and-forward manner to spread the messages. The manner introduces no extra overhead in terms of time as the traditional CSMA/CA does, e.g., the back-off time for contention avoidance.

Constructive interference occurs only when two or more nodes transmit *identical* packets. With IEEE 802.15.4 radios operating in the 2.4 GHz ISM band, these identical packets are required to overlap within 0.5 μs [FZTS11, DDHC$^+$10]. This enables makes the packets appropriately superimposed, thereby, can be correctly received and demodulated by the receiver. The capture effect is a phenomenon, where the receiver can lock onto and correctly demodulate a packet, when the signal of the packet is approximately 3 – 4 dB stronger than the sum of all the other signals [LF76,FZTS11]. Besides, in IEEE 802.15.4 radios, the strongest signal must arrive no later than 160 μs after the weaker signals [LFZ13a], in order to be properly recognized and decoded by the receiver. For more details of CT, i.e. constructive interference and the capture effect, we refer the reader to Section 3.3.1 in Chapter 3.

4.3.2. Packets in a Packet

Technically, most wireless receivers — including IEEE 802.15.4 radios — start to record a packet in the air once a synchronization header (i.e. preamble is detected. They stop the recording based on the frame-length byte in the packet [Tex06]. However, if a collision occurs during the reception of synchronization header, then nothing can be received by the radio. In other words, if a synchronization header of a data frame (i.e. packet) is captured by a radio, then the radio can be synchronized with the data frame. It then records for the time of a frame with the length of the value written in the frame-length byte. Some related work concentrates on empowering the radios to avoid medium access collisions and to receive more valid bytes. However, König et al. [KW16] recently propose a class of wireless transmission primitives by investigating on collisions, specifically, the capture effect. This primitive decouples synchronization headers from payloads in order to create new transmission primitives of a second sender. By using the capture effect, different neighboring senders can overwrite portions of the payload in ongoing packets in the air.

4.3.3. PiP in a Nutshell

Motivated by König et al. [KW16], we propose a link-layer communication primitive — PiP— that by nature manages the "in-packet" time slots for various neighbors in a single transmission. On the one hand, PiP is similar to TDMA scheme: It builds on CT and divides a transmission into several time slots. In PiP, this kind of time slot is defined as "in-packet" slot, which is further explained later in Section 4.4.2.2. On the other hand, different from other power control-based approaches, PiP relies on a hardware-based PA operation. Namely, a PA is an electronic amplifier that converts a low-power RF signal into a higher-power RF signal. This means, if being switched off, a PA can be then used to lower the power level of the output RF signals of the radio. PiP relies on this type of operation during transmission: While multiple neighboring senders are concurrently transmitting to a single receiver, one or more of them can give away the opportunities to others that are simultaneously transmitting. As a result, PiP is able to successfully "control" the capture effect in presence of multiple senders, so that every sender can fairly transmit the packet to the receiver instead of being negatively "captured" (i.e., cause a collision). Moreover, PiP requires no global scheduling and causes no extra overhead of routing. Meanwhile, it achieves accurate time synchronization inherited from Glossy [FZMT12], even when the content of packets is different the network. In a word, by realizing packet-in-packet, PiP is able to obtain timely, reliable, and energy-efficient data collection in multihop low-power WSNs.

4.4. Design of PiP

In this section, we detail the design aspects of PiP. We discuss the basis of PiP: concurrent transmission and the packet concatenation scheme, respectively. Moreover, we depict the network-wide data collection with PiP and additional implementation aspects in detail.

4.4.1. Concurrent Transmission-based Collection

PiP is based on concurrent transmissions, i.e., constructive interference and the capture effect, as derived from the essence of Glossy [FZTS11]. In general, PiP is a sink-initiated data collection protocol, since CT-based data collections are always triggered by the sink. In a data collection phase, there are typically various packets generated by different sensor nodes throughout the network. The content of these packets are most often different in a collection round. For instance, in a machine data collection scenario, various sensors can measure different values from the physical world, e.g., vibrations from different machines. Correspondingly, constructive interference may not function effectively. The capture effect, in this case, however, enables the receiver to correctly receive a packet with the stronger received signal strength if all the senders are highly synchronized (i.e., the time difference is less than 160 μs).

Generally, to well handle data collection, CT protocols exploit either global scheduling such as LWB [FZMT12] or real-time in-network processing such as Chaos [LFZ13a]. The key foundation of CT protocols is one-to-all data flooding, which makes all-to-one data collection rather challenging, because the data flows are totally different. Moreover, CTs require highly accurate time synchronization between senders and the identical content of packets throughout the network.

PiP manages time synchronization with network flooding (similar to Glossy), while employing a number of in-packet slots to collect the data from the single-hop neighbors in a single transmission round. In other words, PiP uses one-to-all data flooding to maintain accurate time synchronization and utilizes various minor slots within a packet to achieve all-to-one data collection. When dealing with the capture effect, PiP does not manipulate power control as other approaches do, e.g., tuning the TX power level of the radio. Instead, PiP relies on a type of hardware-based operation — a so-called PA operation.

Figure 4.1 reveals the basic idea of the packet concatenation with a network of five nodes, where node H is the sink node that ought to collect the data packets from all the neighboring nodes. In time slot 0, node H broadcasts a packet to all the neighboring nodes, so as to maintain the accurate synchronizations (similar to Glossy). After the neighboring nodes have received the packet, they concurrently transmit their own packets immediately in the following time slot, i.e., time slot 1 in Figure 4.1. This receive-and-forward pattern is inherited from CTs, which helps in keeping accurate time synchronization throughout the network. In time slot 1, every transmitter has its own corresponding time period (i.e., defined as in-packet slot in PiP) to transmit its own packet (i.e., defined as in-packet packet in PiP).

A single transmitter only switches its PA on when it is its turn to transmit. For instance, node A switches PA off after it finishes transmitting its in-packet packet in the corresponding in-packet slot. Then, node B switches its PA on and starts to transmit its own in-packet packet. As a result, using the PA operation, multiple senders in PiP are able to transmit their own in-packet packets in different (minor) slots of one major time slot, as shown in the figure. At the receiver (node H), different in-packet packets are naturally concatenated in the air and formed as a longer packet afterwards. Please note, that PA operation is different from radio-on and -off operations. That is, switching PA on and off does not mean powering the radio on and off; Instead, PA operation can only be done when the RF radio is powered on. In the following, we explain the transmission process and the PA operation based on one of the IEEE 802.15.4 radios — CC2420 [Ins07], respectively.

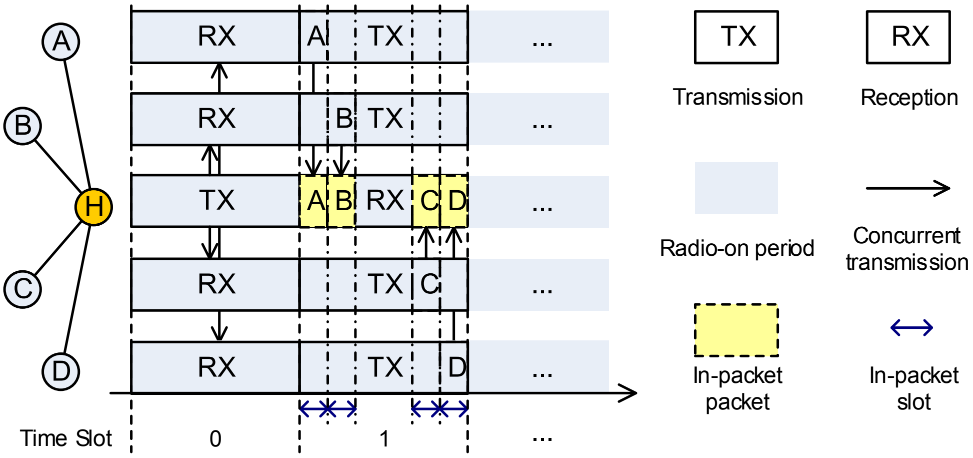


Figure 4.1.: Single-hop packet concatenation in PiP. Node H collects data packets from neighboring nodes in a single transmission round. Node H manages time synchronization by flooding and collects various data packets in different minor time slots of a major receiving time slot.

Furthermore, in the extent of a network, basically, the whole process of data collection in PiP can be defined as a *PiP period*. Each *PiP period* consists of a number of *PiP rounds*. Subsequently, each *PiP round* is made up of several *CT slots*. One CT slot is for the transmission or the reception of one single *PiP packet*. We provide more details in Section 4.4.3: How PiP works in the scope of a multihop low-power WSN.

4.4.1.1. Radio Frequency Transmission

A radio frequency power amplifier (PA) is a type of electronic amplifier that enhances the power of radio-frequency signal. For a better explanation, the RF transmission of CC2420 radio is briefly explained in this section.

On a CC2420 radio chip [Ins07], each byte to be transmitted in TX First In First Out (FIFO) buffer is divided into two symbols, of four bits each. Each symbol is spread by the IEEE 802.15.4 spreading sequence to 32 chips. Then, the chips are converted by the Digital-to-Analog Converter (DAC). Afterwards, the low pass filter and the quadrature up-conversion mixers convert the modulated analog baseband signals (I and Q) to RF signals on the given channel. Finally, the RF signals are amplified by PA based on the

PA_LEVEL in the TXCTRL register [Ins07]. Then, they are fed to the antenna ports (RF Input/Output (I/O)). For more details of CC2420 transmission procedure, we refer the readers to the manual of CC2420 from Texas Instruments [Ins07]. Note, that RF signals could be fed to the RF I/O directly, without PA amplification, but the power of electromagnetic radiation would be quite limited. Consequently, if the signals are fed to the RF I/O directly (without PA amplification), then even a receiver that is extremely close to a sender may detect nothing in the air. In this work, we exploit this phenomenon to enable packet concatenation in PiP without tuning TX power level.

4.4.1.2. Power Amplifier Operation

Generally, a RF PA is a type of electronic amplifier that converts a low-power RF signal into a higher-power signal, as shown in Figure 4.2 [Tex06]. Typically, the RF signal is amplified in the PA, fed to the antenna and then transmitted in the air. In our research, we explore that it is feasible to manipulate PA when a packet is being transmitted. PA can be switched on and off via the registers MANAND and MANOR in CC2420 radio [Ins07] during the transmission. Hence, a packet in PiP can be virtually considered as a number of PA-on and PA-off in-packet packets. Later in this section, we detail the frame structure in PiP. The sender in PiP switches its PA on if it intends to transmit a own packet, and switches PA off while there is nothing to transmit. Besides, the receiver receives the concatenated packet in the air by nature without any PA operation.

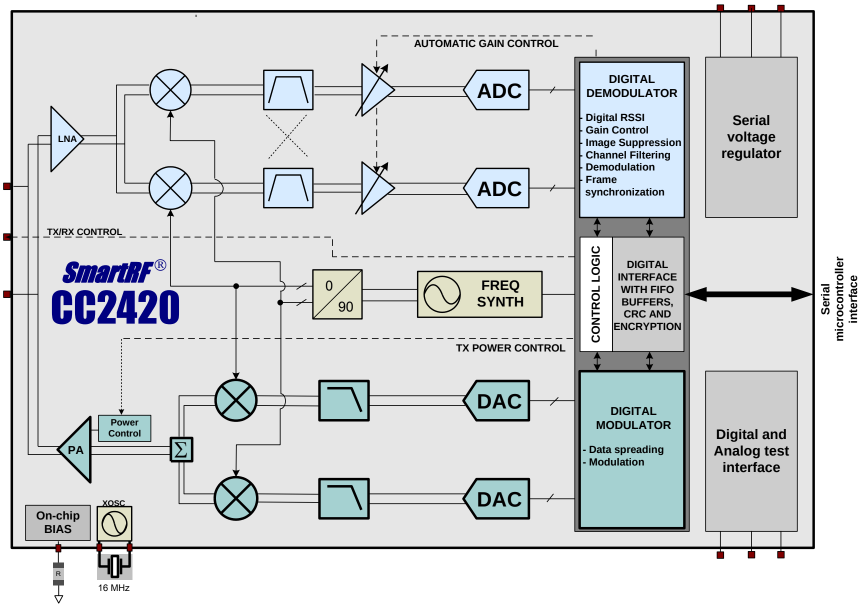


Figure 4.2.: Functional diagram of IEEE 802.15.4 radio CC2420 on page 8 of the technical datasheet [Tex06]. Power amplifier (PA) is located in the lower left side in the figure.

Moreover, the PA operation in PiP also offers another benefit: less energy overhead. For low-power WSNs, the energy consumption is always one of the most critical issues. In PiP, when a node switches PA off, the energy consumption can be significantly reduced.

Figure 4.3 illustrates one specific case of a TelosB sky mote with 0 dBm TX power. With the setting of 0 dBm TX power, the radio normally consumes around 28 mA current for transmission, as the figure shows. With PA operation, specifically switching PA off, the radio saves approximately 10 mA current for almost half a period. When switching the PA on, the radio costs the same current consumption as the one without any PA operation.

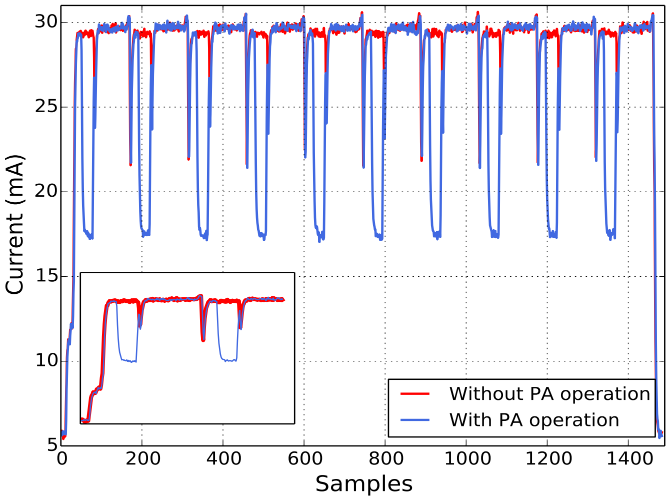


Figure 4.3.: Energy cost of IEEE 802.15.4 radio (CC2420) with and without PA operation at TX power level of 0 dBm.

Please note, that the current consumption in Figure 4.3 includes all overheads such as the energy consumed by the micro-controller, light and temperature sensors, and radio transceiver. In this case, this leads to a higher current level than the one in the CC2420 datasheet [Tex06], i.e. 17.4 mA at 0 dBm TX power. As the zoomed-in sub-figure shows, the current drops deeper when the node switches PA off (in blue) compared to the one without PA operation (in red). This implies, that PiP can benefit from the PA operation in terms of energy consumption. According to our experimental measurements, we calculate that the current consumption is approximately 6.1 mA when PA is switched off. For the sake of simplicity, we calculate the energy consumption of the CC2420 radio according to Equation 4.1.

$$E_{TX_{withPA}} \propto (E_{PA\_ON} + \frac{E_{PA\_OFF}}{\omega}) \tag{4.1}$$

where $E_{TX_{withPA}}$ is the energy consumption of a transmission with PA operation. E_{PA_ON} and E_{PA_OFF} are the energy consumption when PA is turned on and off, respectively. For a fair comparison to other protocols, we use ω, a factor converting the current consumption with PA operation to an overall one provided in the datasheet [Tex06], in order to compute the average energy consumption of PiP. Furthermore, the corresponding current consumption [Tex06] with their factors at different TX power levels are summarized in Table 4.1.

Table 4.1.: Current consumption mappings to the corresponding TX power levels [Tex06].

TX Power (dBm)	Current Consumption (mA)	ω
0	17.4	2.85
-1	16.5	2.70
-3	15.2	2.49
-5	13.9	2.28
-7	12.5	2.05
-10	11.2	1.84
-15	9.9	1.62
-25	8.5	1.39

4.4.1.3. Concurrent Transmission with Power Amplifier Operation

We develop PiP with PA operation through (i) extending the packet length and (ii) dividing the extended packet into a number of in-packet slots. As a consequence, diverse information from multiple single-hop nodes can be received successfully as long as the various in-packet parts do not overlap in the concatenated packet. Based on CT with PA operations, PiP can realize the single-hop packet concatenations.

4.4.2. Packet Concatenation

In this section, we address the following three key aspects to make PiP working. First, the frame structure in PiP is elaborately re-designed since the received packet cannot be validated automatically by the hardware. Besides, the flooding process with respect to transmission and synchronization should continue even when the in-packet packets collide with each other. Moreover, the strategy to avoid overlapping is required to ensure that most of the in-packet packets are tightly aligned and received correctly.

4.4.2.1. Frame Structure

To support the concatenation of packets in a packet, PiP extends the frame structure of Glossy by adding a two-byte checksum field, a 10-byte bitmap for ACK and one dummy byte, as illustrated in Figure 4.4. The checksum is used to check the correctness of the parts, including the bytes of frame length, header, sequence number, relay counter, and bitmap of the concatenated packet. Besides, the dummy byte is considered as a reserved period of time for parsing the received packet at the end of the transmission. The details of the PiP frame structure are shown in Figure 4.4. The PiP payload is divided by a guard byte into various in-packet slots. The guard byte distinguishes the different in-packet slots and reserves the time for operating the PA. In this case, this byte ensures that the PA operation does not affect the completeness of receiving each individual in-packet packet. Each in-packet slot (or packet) is assigned for each neighboring node correspondingly. Namely, an in-packet packet includes a one-byte header, a seven-byte payload, and a slot checksum. The slot checksum ensures the correctness of the corresponding in-packet packet, so that every in-packet packet is independent of each other, even if an error occurs in the overall frame.

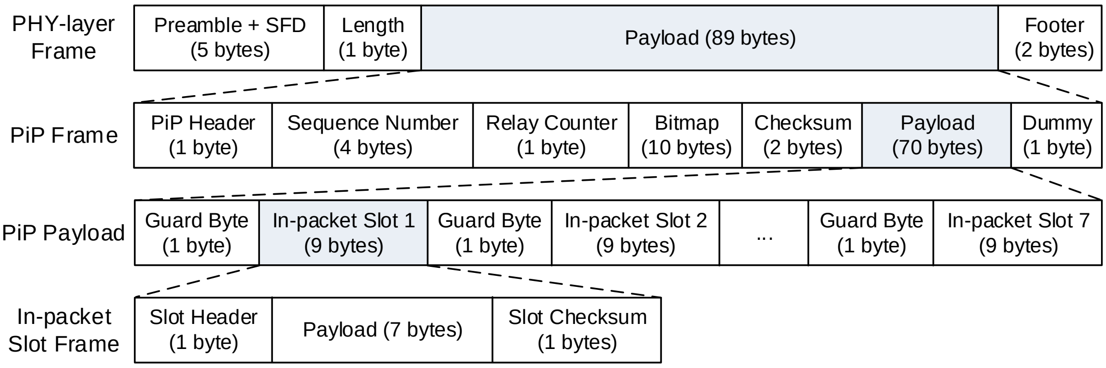


Figure 4.4.: Hierarchical frame structure in PiP. PHY-layer frame refers to the physical-layer frame structure in IEEE 802.15.4 standard.

4.4.2.2. Concatenation in a Packet

In PiP, we divide one packet into a certain number of in-packet packets (e.g., into seven in-packet packets) with fixed length. Correspondingly, the sender switches PA on when there is an in-packet packet to transmit and switches it off when there is nothing to transmit. The receiver detects the (identical) overlapped headers (i.e., the preamble and the Start of Frame Delimiter (SFD)) from multiple senders because of CTs. As long as there is one sender, which turns PA on during the in-packet slot, the receiver can receive the corresponding PA-on parts successfully.

As shown in Figure 4.5, two senders concurrently transmit their own information. The receiver starts to record symbols in the air once a header is detected and stops based on the frame-length byte, i.e., the first byte after the header. In a normal case of CT, when the content of two packets is identical, the receiver can correctly receive the packet due to constructive interference. In this case, the first six bytes (i.e., the header and the length) are received based on constructive interference. Otherwise, when the content is not the same, the receiver can receive the packet with the higher power because of the capture effect. In the case of PiP, the capture effect still happens if multiple senders switch PA on at the same time, e.g., in in-packet slot 4.

However, by switching the PA off, a sender in PiP is able to give away the opportunity of being received to the other senders. For instance, the receiver in Figure 4.5 receives the in-packet packet from sender 1 since sender 2 turns PA off in in-packet slot 1. When it comes to a special case (e.g., in in-packet slot 5), where two senders, both, turn PA off, the receiver does not detect anything in the air. However, in this case, the receiver still makes a soft decision (see the CC2420 manual [Tex06]), even though there is nothing in the air. As a result, a random sequence is placed as the content of the PA-off part at the receiver, as illustrated in in-packet 5.

Moreover, when the PA-off part passes and a following PA-on part arrives, then the PA-on part can be received and demodulated correctly because of the individual checksum in each in-packet slot (see in-packet slot frame in Figure 4.4). In this case, the received CRC in the footer is meaningless for the receiver since a random sequence might exist in the packet. As a result, a received packet in PiP is validated by a software-based check at the receiver instead of a hardware-enabled CRC.

Generally, PiP maintains two schemes to concatenate in-packet packets (also referred to as packet injection): (i) *packet injection with pre-reservation* and (ii) *random packet injection*. On the one hand, packet injection with pre-reservation refers that senders have their pre-defined transmission sequence for their own in-packet packet — pre-resevered in-packet time slot. Therefore, the "collision" — where the capture effect occurs, e.g., in in-packet slot 4 shown in Figure 4.5 — can be avoided. Because this in-packet time slot can either be assigned to sender 1 or sender 2.

In the single-hop scenario shown in Figure 4.1, neighbors of node H concurrently transmit the packet once they have been initiated by node H. Practically, there is almost no chance to pre-assign the in-packet slots for all nodes in a WSN. Compared to the aforementioned relatively "static" packet concatenation scheme, random (packet) injection, on the other hand, is more flexible that requires no knowledge of the network in advance, e.g., topology information. It enables senders to determine whether to transmit their in-packet packet randomly in one in-packet slot without any pre-assigned transmission sequence. Therefore, the node is allowed to transmit its in-packet packet in a randomly chosen in-packet slot, i.e., randomly injecting in-packet packet in the current in-packet slot.

To some extent, this randomness introduces the probability of collisions between senders, e.g., in in-packet slot 4 depicted in Figure 4.5. The capture effect would happen only when multiple neighbors inject their different in-packet packets exactly into the same in-packet slot. In this case, due to the capture effect, the receiver could only receive the packet from one specific sender and would not receive the packets from other neighboring nodes. In Section 4.5.2.2 and Section 4.5.2.3, we evaluate the performance of both, PiP with pre-reservation and random injection, in terms of single-hop and multihop scenarios, respectively.

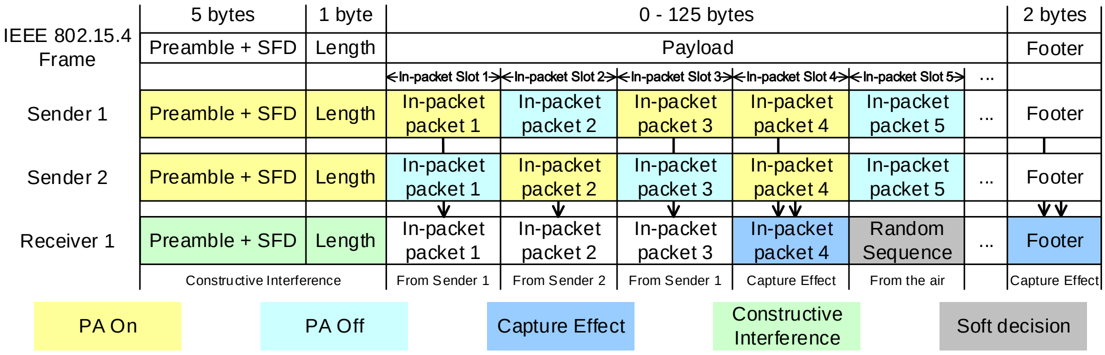


Figure 4.5.: Packet concatenation of two senders with one receiver. Both senders switch PA on and off to ensure that the receiver is able to receive various in-packet packets from all senders during one transmission slot. If both of two senders turn PA off, then the receiver makes a soft decision according to [Tex06], thereby, placing a random sequence in the corresponding part.

4.4.2.3. Time Synchronization

In this section, we explain how PiP exploits the designed frame structure to achieve time synchronization. Most of the CT-based protocols achieve accurate time synchronization by the receive-and-forward communication pattern proposed in Glossy [FZTS11]. The accurate time synchronization is based on identical packets in the whole network of a single CT round. However, in PiP, the packets cannot be identical throughout the network due to the “collection” nature. Therefore, we deliberately design the PiP frame structure and use one part of the frame to realize time synchronization. More precisely, this part can be regarded as the CT-based in-packet packet, which is identical through the network. Namely, it refers to the first 24 bytes (preamble, SFD, length, PiP Header, sequence number, relay counter, bitmap, and checksum) in a PiP packet (see Figure 4.4). Senders always turn on PA when transmitting this identical part of the packet. At the receiver, this identical part can be always received correctly due to CT. As a result, the back-to-back receive-and-forward pattern can continue to work and, thus, time synchronization can be achieved even though the remaining parts of the packets from different sources can be quite different.

4.4.2.4. Acknowledgment

The worst case of random in-packet packet injection is that all senders inject their packets into the same in-packet slot, so that collision or the capture effect happens and some of the packets never reach the receiver. In order to deal with this situation, we utilize a bitmap (as depicted in Figure 4.4) to enable the receiver to give an acknowledgment to all the senders (this is similar to Chaos [LFZ13a]). After the receiver receives a certain in-packet packet from a specific sender, an acknowledgment is broadcast to all the neighboring senders. Consequently, those nodes know that whose in-packet packet has been delivered to the receiver. If the in-packet packet from one sender has been delivered, then this sender would switch off its PA in the following TX rounds, so that more in-packet slots will be available for the other senders. In other words, the bitmap sent by the receiver notifies the senders that the in-packet packets from which node have been received correctly. Senders that get acknowledged, will switch PA off and inject nothing in the following transmission rounds. The acknowledgment scheme takes several rounds until all the in-packet packets have been acknowledged. As a result, various packets in in-packet slots can be received by the receiver.

4.4.3. Make it Work Network-wide

Previously, we explain how sensor nodes work with PiP. However, in order to make PiP effectively work in a network scale, here, we clarify a number of definitions of PiP with respect to the multihop network scale.

Figure 4.6 shows a small network consisting of four nodes — three source nodes and one sink within two hops. The whole process of data collection can be defined as a *PiP period*. Each *PiP period* consists of a number of *PiP rounds*. Each *PiP round* is made up of several *CT slots*. One CT slot is for the transmission or the reception of one single *PiP packet*.

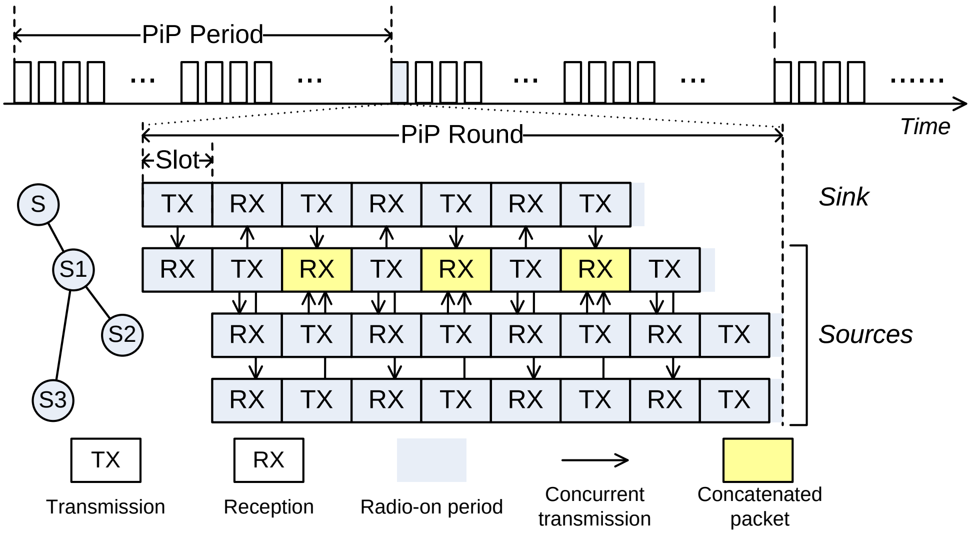


Figure 4.6.: Network-scale definitions and information exchange process in PiP. Basically, a data collection process is a PiP period. A PiP period consists of a number of PiP rounds. In each PiP round, there exists several CT slots. Note, the RX slot in yellow shows that multiple senders inject their own in-packet packets into the major packet, thus, generating a so-called concatenated packet.

In PiP, data collection is initiated by the sink and the source nodes broadcast their in-packet packets, relying on the attribute of concurrent transmissions. As a side effect, CTs could not control the direction of the data flow if without any central scheduling or processing, such as the one in LWB [FZMT12] or Chaos [LFZ13a]. In other words, this “flooding” attribute could cause that the packets are flooded away from the sink and never flow back.

For data collection applications, information generated by source nodes should not only be collected by a single-hop neighbor, but also forwarded in the direction of the sink hop-by-hop, and finally received by the sink. To achieve the oriented data propagation, PiP exploits the concept of a *rank* [MTH+17], i.e., the topology information representing the relative hop distance to the sink in each in-packet packet. For instance, the rank of the sink is set to 0. In PiP, nodes compare the rank in the received in-packet packet to their own rank. If the rank of the in-packet packet is greater than the rank of the receiver, then this means that the in-packet packet is propagated from the source nodes to the sink. On the contrary, an in-packet packet with an equal or smaller rank is neither stored nor forwarded by a node.

4.4.4. Implementation Aspects

Our implementation is based on Contiki OS [DGV04] and TelosB Tmote Sky. CC2420 [Tex06], the transceiver on the Tmote Sky, requires that the SFD pin must be active during the transmission of a data frame. Additionally, this pin (SFD) is the only one that is active during the transmission of a frame. Therefore, no normal register in the transceiver could be utilized to directly operate PA during a transmission. However, in our implementations, we operate with PA in the manner of busy-waiting for a certain fixed period of time during the transmission of a frame: PiP estimates the time period for a complete in-packet slot — in-packet slot. When the time is up, PiP forces to operate PA in the transceiver, namely switching PA either on or off.

Another big challenge in implementation is, that there is extremely limited time to parse, to restore, and to select in-packet packets that are to be forwarded. To overcome these problems, we utilize the Direct Memory Access (DMA) to speed up in writing the transmit buffer, reading from the buffer via the Serial Peripheral Interface (SPI) bus, copy and set memory in micro-controller. That saves a lot of time in read- and write-operations in memory.

4.5. Performance Evaluation

We evaluate PiP in terms of single-hop and multihop scenarios for data collections in WSNs, respectively. Additionally, we compare the performance of PiP to a state-of-the-art CT-based data collection protocol LWB [FZMT12] in each scenario.

4.5.1. Methodology

In this work, we use the FlockLab testbed [LFZ$^+$13b] for our experimental evaluation. We use the 26 indoor observers out of the 30, except outdoor nodes in the testbed, since they were not constantly available during our experiments.

Similarly to previous chapters, for the network performance of PiP, we focus on three key metrics to evaluate the performance of the selected protocols: (i) reliability (indicated by PDR), (ii) latency (indicated by collection rounds), and (iii) energy efficiency (indicated by radio duty cycle).

Specially, for CT-based protocols, it is however difficult to accurately measure the end-to-end latency by using the output messages from the serial ports, since the process of *print* is quite time-consuming and might cause clock drift. Besides, it is easy to compute the number of collection rounds in LWB [FZMT12] as well. For the sake of simplicity and fairness, in this chapter, we refer to the number of collection rounds as the notion of end-to-end latency in data collection applications.

We compare PiP to the LWB [FZMT12], a popular state-of-the-art communication protocol of low-power WSNs implemented in Contiki, following the trends of the 2018 EWSN Dependability Competition. LWB is realized by adding an application-level scheduler to Glossy in order to support many-to-one applications. The experiments of LWB are based on the TelosB-compliant code of LWB implemented by Sarkar [Sar16]. The basic configurations of LWB and PiP in multihop collection scenario (Section 4.5.3) are summarized in Table 4.2. The configurations apply in Contiki OS for a fair comparison.

4.5.2. Single-hop Capability of PiP

In this section, we review the single-hop performance of PiP. Basically, we show how PiP can actively “control” the capture effect and two packet concatenation schemes, i.e.,

Table 4.2.: Configurations of LWB and PiP in the multihop data collection scenario.

Parameter	**Configuration**		**Description**
	LWB	PiP	
IPI	2 s	1.5 s	Inter-packet interval (sensing frequency)
MINIMUM_LWB_ROUND	1 s	–	Duration of minimum LWB round
COOLOFF_PERIOD	30 s	–	Duration of the cool-off period
STABILIZATION_PERIOD	30 s	–	Duration of the stabilization period
MAX_PAYLOAD_LEN	3 bytes		Maximum payload of a LWB data packet
SINK_NODE_ID	1		ID of the sink node
NUM_TX	5		Number of transmissions

static packet injection and random packet injection.

4.5.2.1. Resisting the Capture Effect

In general, the capture effect is strongly used in CT-based WSNs. Namely, when senders transmit packets with different payloads to a receiver, then the receiver always receives the packet with the strongest signal. To some extent, the PA operation in PiP controls and eliminates the capture effect in order to ensure that the receiver successfully receives other packets instead of the strongest one only.

As shown in Figure 4.7, we measured the Received Signal Strength (RSS) of a set of neighboring nodes with node 3 as a receiver in FlockLab. The RSS values are measured by using 0 dBm TX power in all senders.

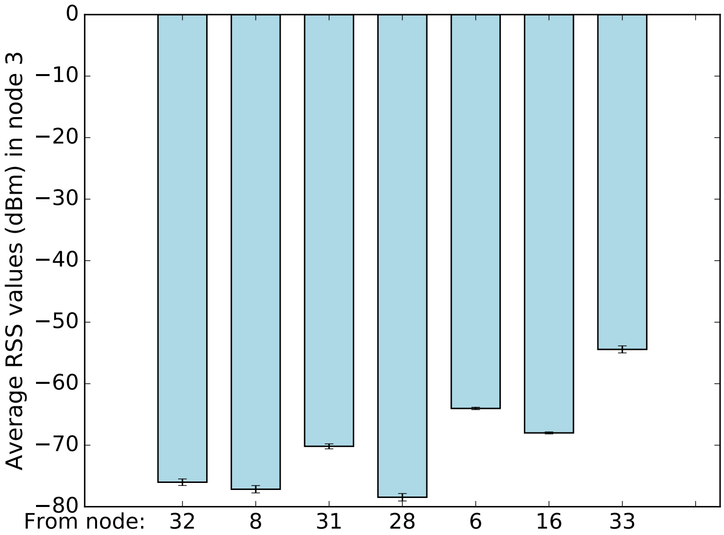
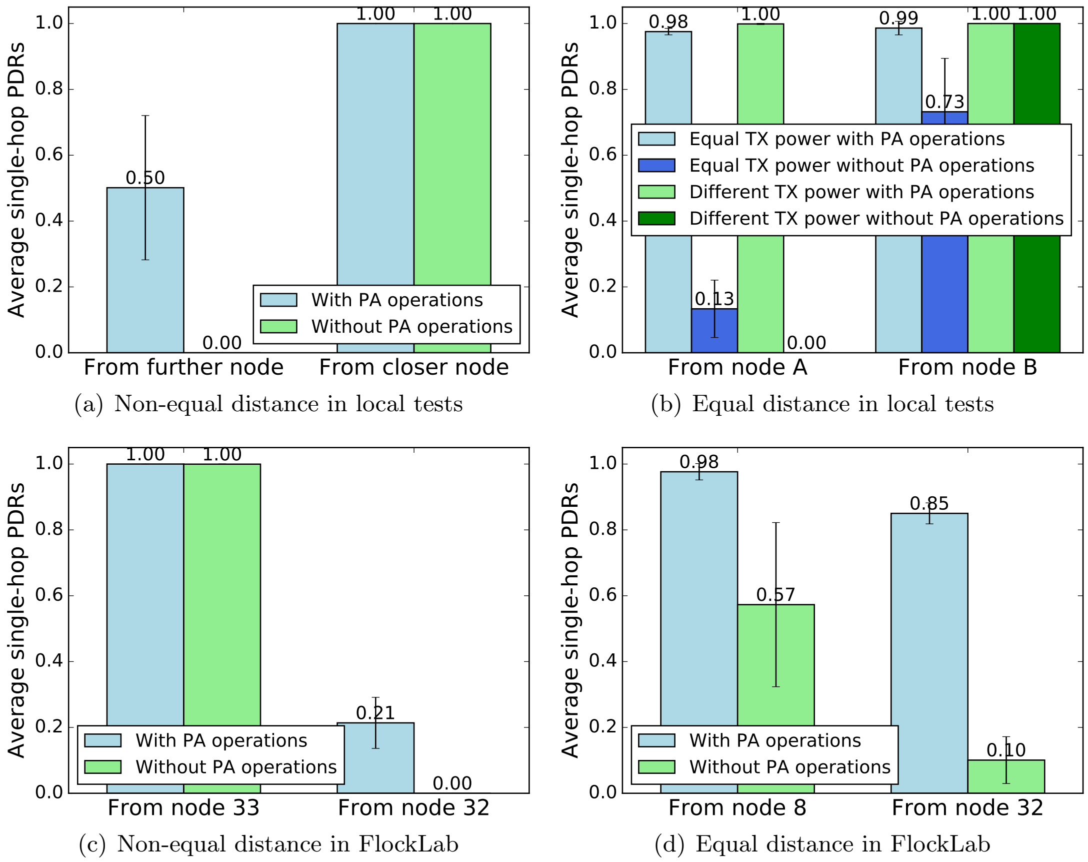


Figure 4.7.: Single-hop RSS values from various neighbors of node 3 in FlockLab. The higher the RSS value is, the closer to node 3 the node is.

Besides, we carry out a number of local tests and experiments in FlockLab in order to figure out the influencing factors on the performance of PiP, such as *RSS value*, *relative distance* between sender and receiver, and *TX power level*. In these local tests and

FlockLab experiments, two source nodes are pre-assigned to transmit their own packet in in-packet slot 1 and 2, respectively.

Figure 4.8(a) and Figure 4.8(b) show the results from the local tests with TelosB nodes. In the tests of *non-equal distance*, the further node is placed three meters away from the host (i.e., receiver) without obstacles, while the closer node is quite close (less than 0.5 meters) to the host. Thus, we ignore this short distance and assume that the closer node is at the same point as the host. The distance difference between the further node and the closer one to the host is roughly three meters. In the tests of *equal distance*, both nodes are placed quite close to the host with the same distance.



(a) Non-equal distance in local tests

(b) Equal distance in local tests

(c) Non-equal distance in FlockLab

(d) Equal distance in FlockLab

Figure 4.8.: Single-hop concatenations with non-equal/equal distance cases in local tests and in FlockLab.

In the tests of non-equal distance, the host receives the PA-on part from the further node when another sender turns PA off. As shown in Figure 4.8(a), the PDR of the closer node is 100% due to the capture effect. The packets from both nodes could be received by the host if they are transmitted using the PA operation. The PDR of the further node is only 50% if it transmits using the PA operation, which is still much better than its PDR without PA operation. This is, because, for the receiver, the concatenated packet is quite sensitive to the packet decoding error in CTs (also mentioned in [KW16]). However, the difference of distances are doubled in terms of the CT error due to the receive-and-forward scheme. The CT error for the two nodes is computed based on the distance differences and the speed of the radio signal: approximately 0.02 μs.

To further prove whether the CT with PA operation can function against the capture effect with respect to TX power, we used different levels of TX power in two source nodes (i.e., node A and B), i.e., 0 dBm and -25 dBm, in the equal-distance scenario. This is, because we attempt to exclude the influence of the relative distance difference to the host. As shown in Figure 4.8(b), with PA operation, high PDRs are achieved by PiP, which are independent of the TX power levels. Otherwise, PDRs would be worsened by the occurring capture effect, e.g., the case of equal TX power without a PA operation.

Similar experiments are carried out in FlockLab. Figure 4.8(c) shows the worst case, where node 33 is much closer to the host (node 3) than node 32, as indicated by the RSS values in Figure 4.7. Even though, node 32 with a PA operation can still obtain an average PDR of about 21%. Moreover, while in the case of equal distance to the host, nodes (node 8 and 32) with PA operation are able to maintain an average PDR of more than 80% as illustrated in Figure 4.8(d). In summary, experimental results reveal that PiP works effectively to concurrently concatenate various packets from neighboring source nodes in a single-hop manner.

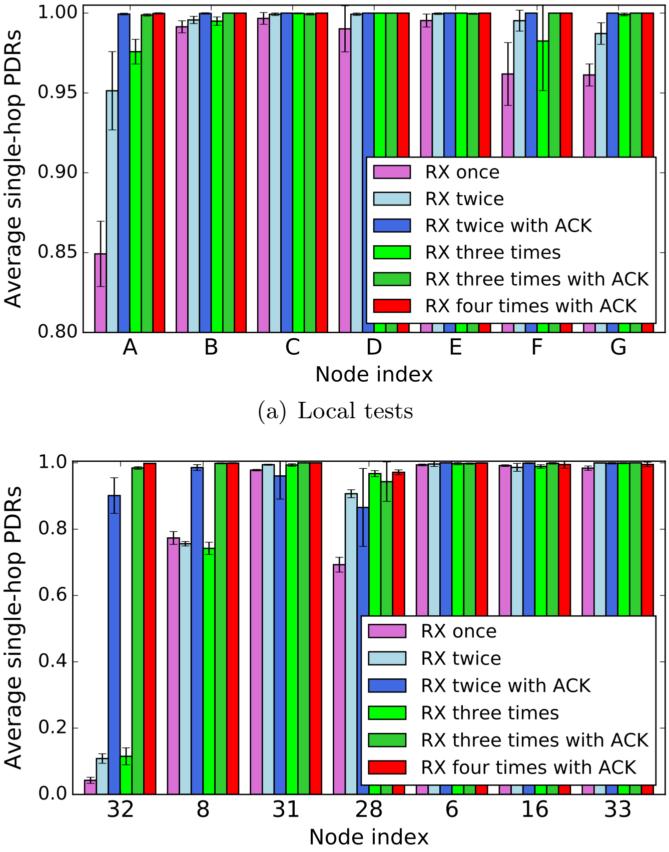
4.5.2.2. PiP with Reservation

Static pre-reservation means that the pre-defined corresponding in-packet slots are pre-reserved for the neighboring nodes to inject their own packet. In other words, each sender knows, which in-packet slot within the packet is preserved for its own transmission. Figure 4.9 illustrates the results for the single-hop cases with pre-reserved order in local tests and in FlockLab, respectively. Basically, the number of receptions equals to the number of transmissions (NUM_TX) (indicated in Table 4.2) plus one, e.g., receiving (RX) once means that the NUM_TX is set to two.

As Figure 4.9(a) and Figure 4.9(b) illustrate, the number of receptions has a positive effect on the PDRs in most cases. That is, higher PDRs can be achieved by increasing the number of transmissions. It is also clear that with our bitmap-based ACK mechanism, PiP is able to further improve the PDRs in both, local tests and experiments in FlockLab. The kind of PDR improvement is more noticeable, especially in the case of real-world testbed shown in Figure 4.9(b). Besides, in general, our results of the local tests in Figure 4.9(a) are better than the ones in FlockLab in Figure 4.9(b). This is, because we attempt to deploy the sender closer to the host in our local tests, which could alleviate the synchronization errors caused by the round time of transmission spent in the air.

4.5.2.3. Random Injection

Differ from the pre-reservation scheme, random packet injection refers that in a single in-packet slot, multiple senders randomly decide whether choosing the current in-pakcet slot to transmit their own in-packet packet or not: There is no pre-defined transmission order of multiple senders to avoid in-pakcet slot overlaps. Namely, while concurrently sending in-packet packets in one collection round, senders randomly inject their own in-packet packets into the current in-packet slot without any pre-defined order. Therefore, it combines the idea of an CSMA scheme with CT. The sender could back-off to the



(a) Local tests

(b) Experiments in FlockLab

Figure 4.9.: Single-hop concatenations with static pre-reservation in local tests and experiments in FlockLab.

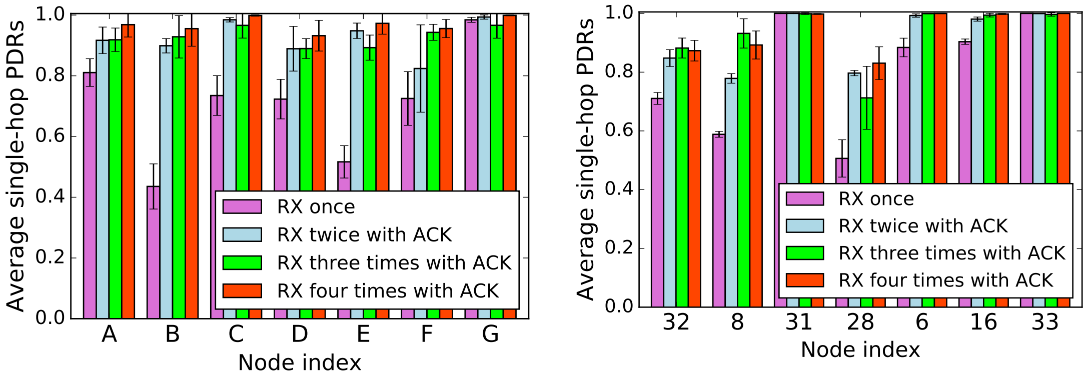
next PiP round if its in-packet packet overlapped with other in-packet packets from other nodes.

As Figure 4.10 shows, the overall performance is worse than the one with the pre-reservation in both, local and FlockLab experiments (see Figure 4.9), since the collision happens with a higher probability, and thus, deteriorates the quality of transmission.

However, the worst PDR in this scenario is still higher than 50%, i.e., node 28 in Figure 4.10(b), since node 28 is the furthest one to the sink node (also indicated by the RSS values in Figure 4.7).

4.5.3. Multihop Data Collection

In this section, we review the multihop collection performance of PiP. In general, we focus on four performance metrics: network initialization time, reliability, data collection round, and radio duty cycle. Besides, we evaluate the performance of PiP with random packet injection scheme and compare it to LWB in terms of those four metrics.



(a) Local tests

(b) Experiments in FlockLab

Figure 4.10.: Single-hop concatenations with random injections in local tests and experiments in FlockLab.

4.5.3.1. Initialization Time

In the TelosB-compliant version of LWB [Sar16], the cool-off period refers to the time spent on time synchronization of the nodes in the whole network. The sink sends a synchronization packet once every second. After the cool-off period, a network stabilization period is executed, which contains a request/reply slot and a contention slot of all the source nodes. In contrast, PiP does not require a network-wide initialization such as a cool-off period and a stabilization period as in LWB [Sar16], since PiP does not rely on the central scheduling in LWB.

Figure 4.11 illustrates two resulting initializations of PiP and LWB with TX powers of −5 and 0 dBm, respectively. As shown in Figure 4.11, PiP initializes the network within a much less period of time compared to LWB. The cool-off and stabilization time of LWB can be re-configured in the implementation [Sar16]: Even with least time for cool-off and stabilization, to settle down the network initialization and the time synchronization among all nodes in FlockLab, at least 25 rounds (or seconds) are required.

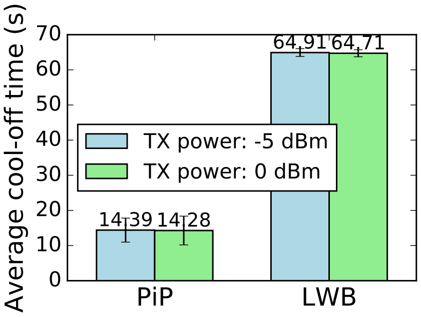


Figure 4.11.: Comparison of the network-wide cool-off time between PiP and LWB in FlockLab.

4.5.3.2. Reliability

We conduct a number of experiments of PiP in FlockLab with a number of different configurations: We tune the number of PiP rounds ranging from seven to thirteen and the level of TX power of −5 and 0 dBm, respectively. For one configuration, we run five times of experiments and compute the mean values and the corresponding standard deviations. In one single experiment, we run more than 1000 PiP periods of data collection.

Higher average PDRs are achieved by increasing the number of PiP rounds, as shown in Figure 4.12(a). The average PDR of LWB is 99% at 0 dBm and 98.9% at −5 dBm. Observing the experiments of PiP with nine rounds in one period, the worst-case PDRs amounts to 96.8% at 0 dBm and 97.6% at −5 dBm. By setting the round value to 11, the average PDR can reach 98.7% at 0 dBm. When the round value is set to 13, then the average PDR is 98.5% at 0 dBm and 97.7% at −5 dBm.

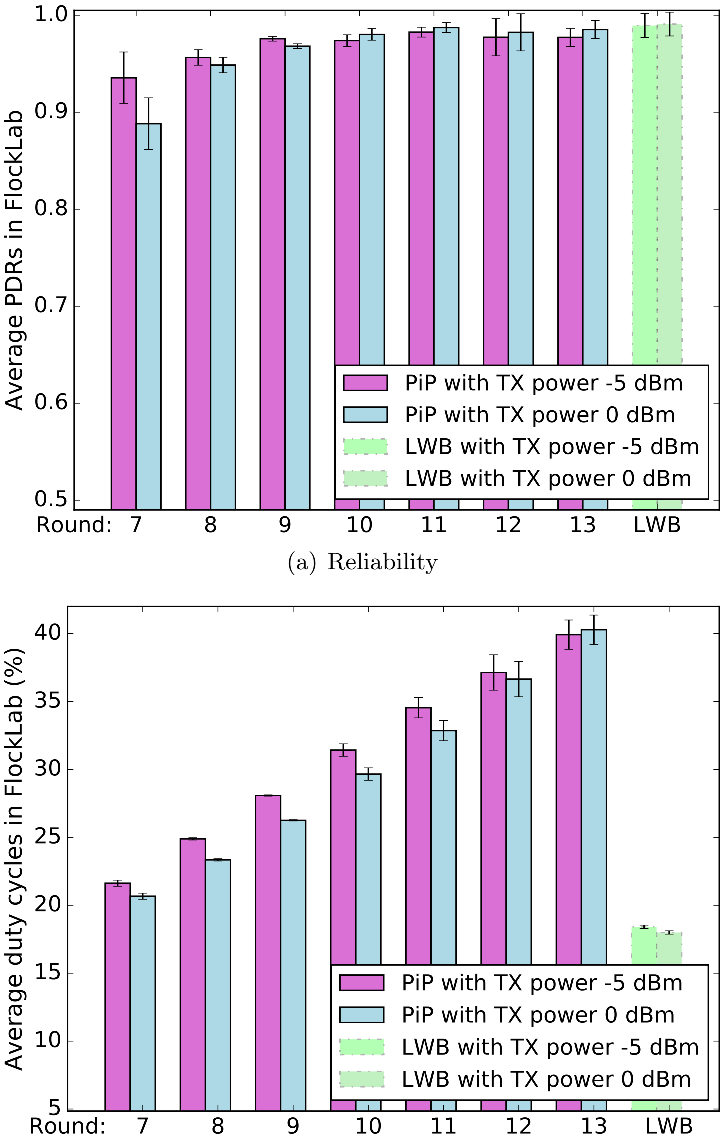
The dominant reason for packet loss of Glossy-like protocols is inaccurate time synchronizations. As a result, the sink does not receive anything in a few continuous LWB slots. This phenomenon becomes more apparent when the network diameter increases. However, in PiP, the packet loss is mainly led by the collisions among different in-packet packets. Moreover, the packet loss problem caused by different distances as mentioned in the single-hop scenario is thereby solved accidentally. The reason is that — once source nodes get an invalid packet — their "RX-TX" sequence is disrupted by accident. Thus, the source nodes at the same level have the opportunity to receive the in-packet packets from each other. As a side effect, the in-packet packets are relayed by those neighbors opportunistically. Therefore, the average PDRs in multihop scenarios are higher than the ones in the single-hop scenarios. The PDRs at lower TX power levels are lower than the ones at higher levels, when the PiP round number is relatively small (e.g., seven), as shown in Figure 4.12(a). A number of transmission rounds are insufficient, since severe collisions occur in the in-packet packets, resulting in degraded end-to-end PDRs.

4.5.3.3. Collection Round

Besides, we show the performance of different collection rounds of PiP. As shown in Figure 4.12(a) and Figure 4.12(b), PDR and the duty cycle of PiP are increased with the number of PiP rounds, e.g., PiP with eight rounds achieves more than 95% PDR and around 25% duty cycle. Note, that there is no so-called (collection) round in LWB. But according to our experiments, LWB requires at least 25 seconds to complete the data collection specifically in FlockLab.

4.5.3.4. Duty Cycle

Figure 4.12(b) presents the comparison of the duty cycle in PiP and LWB. The duty cycles of PiP are revised by Equation 4.1 with ω according to Table 4.1. All the duty cycles are converted based on the same total period for comparisons, i.e., the duty cycles of LWB are extended by a factor of 2/1.5. PiP exploits the duty cycles of 28.1% at −5 dBm and 26.3% at 0 dBm with nine rounds, while LWB maintains duty cycles of 18.4% at −5 dBm and 18% at 0 dBm, respectively. LWB performs better with respect to duty cycles (after network initialization). However, on the one hand, this result does not include the energy cost of the network initialization. On the other hand, the transmission order of senders (in a so-called contention slot) is scheduled in advance in LWB while nothing is pre-assigned in PiP. This also inevitably degrades the overall energy efficiency and the adaptation of LWB.



(a) Reliability

(b) Duty cycle

Figure 4.12.: Comparison of performance metrics between PiP (with random injection) and LWB with TX powers of −5 and 0 dBm, respectively, in FlockLab.

4.5.4. Discussion

The prototype of PiP has proved that the idea of packet concatenation in the air works in principle. A set of factors on practical data collection applications are discussed next.

4.5.4.1. Capability

Most of real-world WSNs are developed to collect data of sensors such as humidity and temperature. In this case, only a small-sized payload is required for this data. In PiP, the length of an in-packet packet is nine bytes. However, the payload of three bytes is reserved for users, excluding the node ID (two bytes), rank (one byte), header (one byte), and checksum (two bytes). The in-packet packet can also be extended if the payload size cannot satisfy the application requirement. Nevertheless, the maximal payload size of IEEE 802.15.4 (i.e., 125 bytes) limits the size and the number of in-packet packets. The network scale that PiP applies, depends on the size of the bitmap field, meaning that PiP currently supports a WSN with 80 nodes.

4.5.4.2. Behind MAC

CT also plays a critical role in PiP. The concatenation of packets cannot be implemented easily if there is no CT. This is because the packets in a packet are sensitive to the precision of the alignment of transmission (tight time synchronization). In CT-based network, receivers transmit the packets once the reception stops. They keep the alignment of transmission relying on the end of reception and the valid received packet. Therefore, the length of the forwarded packets must be identical and so do the contents. Moreover, the timeout scheme enables the process of receive-and-forward not to be stopped by invalid received packets in Chaos [LFZ13a]. The reception is triggered by the synchronization header and stopped according to the length of the frame. The alignment of transmission among nodes relies on their the accurate clocks. The part-by-part validations of the received packets rather than automatic validation of the whole packet by the hardware guarantee that the CTs with different contents do not be terminated.

4.6. Conclusion

In this chapter, we proposed an energy-efficient communication paradigm based on IEEE 804.15.4 for timely data collection in low-power WSNs. PiP exploits both, constructive interference and the capture effect to achieve high reliability and low latency. Moreover, PiP maintains a packet concatenation capability to gather single-hop information in a best-effort manner. We further compared PiP to LWB, a state-of-the-art CT-based collection protocol, by extensive experiments in FlockLab. Experimental results reveal that PiP highly reduces the collection duration and achieves a good performance in terms of high reliability and high energy efficiency in all experiments.

5 Application-oriented Adaptation

5.1. Introduction

WSNs have been recognized as promising tools to collect relevant, in-situ data for a wide range of application domains. Plenty of those real-world WSN-based applications have known predefined lifetimes. Several examples of such applications and their planned lifetimes are listed in Table 5.1 [RM04]. Basically, the applications have expected lifetimes within which the WSNs must function properly to fulfill the tasks. For instance, the College of Atlantic and Berkeley University conducted field research on the Great Duck Island (GDI) [MCP+02]. The research objective was to explore the usage pattern of the nesting burrows when one parent or both parents alternate between incubation and feeding. In the project, the monitoring application was designed to run for seven months.

Table 5.1.: Various WSN-based applications together with their corresponding expected lifetimes and affected performance metrics while maximizing the lifetime.

Project	Task Lifetime	Performance Metrics
Avalanche	days (duration of a hike)	reliability
Glacier	months	reliability
GDI	months (breeding period)	reliability, energy efficiency
Grape	months (growth period)	energy efficiency, timeliness
ZebraNet	one year	throughput, energy efficiency
Ocean	years	reliability, timeliness

Despite this valuable knowledge of lifetime expectancy, the WSN literature reports heavily on solving the lifetime maximization problem [HY08, YX10, ZWLL12] in order to enable the WSNs to run as long as possible. Such maximization sometimes comes at the expense of degrading other QoS parameters such as reliability, energy efficiency, latency, and throughput. On the contrary, exploiting design-time knowledge during run-time can be beneficial for the performance of QoS self-adaptation structures.

In this chapter, we introduce a strategy for QoS improvement, referred to as *Lifetime Planning (LP)*. The core idea behind our proposed strategy is to deliberately reduce the operational lifetime beyond a given maximal required lifetime to a predefined lifetime. Simultaneously, the nodes' lifetime has to meet the predefined time required to complete tasks. The amount of conserved energy is then utilized to provide a set of best-effort QoS metrics. According to the application scenario, the user can collect data on the basis of spatial mapping, target tracking, or both. Lifetime planning considers the case of a heterogeneous network in which data aggregation is achieved in an event-driven and time-driven manner. Namely, the proposed strategy re-configures the low-level controllable parameters similar to [ZFM$^+$12], such as transmission power, duty cycle, and sampling rate, so that event-miss probabilities are minimized and the sampling rates for continuous data flows are optimized.

Technically, LP is feasible through exploiting design-time knowledge. During design-time, a set of QoS maximal and minimal boundaries is estimated. These boundaries act as two thresholds where the instantaneous QoS metric should settle in between. At run-time, low-level parameters are tuned — within the allocated range — in the light of changes in the scenario and the environmental dynamics. Accordingly, a framework for self-adaptation is exploited to dynamically adapt the low-level parameters. The autonomic Monitor Analyze Plan Execute (MAPE) model [GSC09] is the basis of our self-adaptation technique. Afterwards, a reasoning engine is responsible for checking QoS boundary violations. Whenever a conflict occurs, then the low-level parameters are therefore triggered to be updated.

Throughout this chapter, we consider a cluster-tree topology where a hierarchical structure of the MAPE control loop is proposed. In this case, a system update — emerged from each sensor node — is only executed whenever permission is granted from its corresponding cluster head. This limitation is intuitive to prevent any selfish node strategy. To determine the system performance throughout the entire lifetime, we have designed an office monitoring scenario. The indoor environment dynamics are exploited to validate our proposed approach. A network of Tmote sky sensor nodes is deployed in the Contiki OS [DGV04] network simulator Cooja [ÖDE$^+$06].

The remainder of this chapter is organized as follows. Section 5.2 surveys previous endeavors to maximize lifetime and to design self-adaptation mechanisms in WSNs. Section 5.3 provides a brief overview of our protocol. Section 5.4 details the proposed strategy — LP. An analytical QoS model is introduced and validated in Section 5.4.3. The office monitoring scenario and its dynamics are elaborated in Section 5.4.4. Section 5.5 shows a comparative performance analysis in terms of reliability, latency, and network lifetime. Finally, the conclusion together with an outlook is given in Section 5.6.

5.2. Related Work

In this section, we motivate our proposed solution via exploring existent state-of-the-art for maximizing the lifetime and several self-adaptation scheme in WSNs.

5.2.1. Lifetime Maximization

Many efforts are exerted to maximize the operational lifetime of WSNs. Authors in [HY08] propose an optimal routing and data aggregation scheme to maximize network lifetime while reducing data traffic. Rao et al. [RLCL09] consider the trade-off between network performance and lifetime maximization in real-time WSNs as a joint non-linear optimization problem. Based on the solution to such a mathematical optimization problem, they developed an on-line distributed algorithm to achieve an appropriate trade-off. Alternatively, an adaptive fault-tolerant QoS control algorithm is designed in [CSE11] to meet the QoS requirements in query-based WSNs. They developed a mathematical model in which lifetime of the system is considered as a system parameter. Then, they determined the optimal redundancy level that satisfies QoS requirements while prolonging the lifetime. However, the network dynamics is not fully considered in their application. To some extent, the work is specified in the fault-tolerant and query-based WSNs, thus several QoS metrics are not mostly critical to the users. Similarly, authors [YX10] propose a framework for delay-tolerant applications that utilizes a mobile sink to maximize the lifetime of WSNs. Alternatively, the authors in [ZWLL12] present a sleep-scheduling technique, referred to as *Virtual Backbone Scheduling*. It provides multiple overlapped backbones working alternatively to maximize the network lifetime. However, none of these studies thoroughly considers or provides QoS support in WSNs, which are highly required based on the type of application.

5.2.2. Self-adaptation Scheme

A number of strategies of designing frameworks for self-adaptation are addressed in the literature [MGR09, ASB+14, SVNSO+11, JH13]. Munir et al. [MGR09] design a tuning algorithm based on a Markov Decision Process (MDP). This tuning scheme is scheduled to meet the user requirements and the dynamic environmental changes. Hence, messages in the network are flooded back and forth. They consider an energy consumption minimization scheme, which in turn increases the number of reconfigurations. The MDP algorithm is evaluated in MATLAB and therefore, the overhead of porting it to real sensor nodes is not investigated. Moreover, a proactive mechanism is proposed in [ASB+14] to optimize the system behavior through forecasting future conditions. Although this approach reduces the energy consumption, it increases the end-to-end delay due to the incurred computational overhead and the centralized nature of this algorithm.

Furthermore, an adaptation technique — similar to our approach — is introduced in [SVNSO+11], which also makes use of design-time knowledge. The system parameters are assigned in response to the expected and detectable scenario dynamics. Data flooding is used to distribute commands among sensor nodes. Despite the simplicity of this strategy, it does not react to unexpected environmental dynamics. Besides, the adaptation of network routing is manly based on flooding that completely contradicts with the goal of high energy efficiency.

The authors of [JH13] focus on the self-adaptation mechanism. They adapt the MAPE control loop over tree-based topologies. The cluster heads are completely responsible for planning the reconfigurations. In fact, this computational overhead burdens the cluster heads and leads to rapid battery depletion. Hence, their distribution of the MAPE four phases (M, A, P, and E) is arguable. In our self-adaptation mechanism, we balance the energy draw via planning the reconfigurations at each node. Then, the cluster heads utilize their knowledge of the cluster status to approve or disallow the reconfigurations of their children. However, relationships between the residual energy and other QoS metrics on the network-scale are still not carefully considered.

5.2.3. Summary

In this chapter, we present our cross-layer QoS optimization approach LP, in which the drawbacks of the aforementioned methods are sidestepped. LP makes use of design-time knowledge and relies on a light-weight self-adaptation procedure during run-time. Besides, based on the network model, LP carefully considers the WSN lifetime, QoS boundary, as well as the QoS metric conflicts.

5.3. Overview

In this chapter, we present LP, in which the drawbacks of the aforementioned methods are sidestepped. First, the algorithm makes use of design-time knowledge and relies on a light-weight self-adaptation procedure during run-time. Afterwards, we briefly describe the relevant definitions including WSN lifetime, QoS boundary, as well as the network model, based on which we need to concisely describe our approach. Moreover, we introduce assumptions made and define the problem of QoS degradation in response to lifetime maximization.

5.4. Design of Lifetime Planning

In this section, we explain the crux behind our proposed strategy LP for low-power WSNs. The main ideas are to (i) conserve energy by limiting the lifetime below a maximal possible lifetime, and (ii) exploit the design-time knowledge for controlling application-relevant QoS metrics. In fact, design-time knowledge of the application scenario is a significant resource. Such valuable knowledge can drastically decrease the computational burden on the self-adaptation mechanism. Furthermore, it can be utilized to engineer the lower and upper boundaries of QoS metrics. For instance, office monitoring applications do not have to provide extremely high service quality at night or during holidays. Based on the scenario dynamics, the instantaneous QoS metrics are confined between their boundaries.

The autonomic MAPE reference model [GSC09] is exploited to design a self-adaptive management element, as depicted in Figure 5.1. At run-time, the QoS metrics are continuously monitored. A set of secondary sensors — such as temperature and light sensors — forward their readings to the adaptation mechanism. Subsequently, the received data is analyzed to discover any interesting events. Accordingly, nodes find a course of action to adapt the low-level parameters once a problem has been detected.

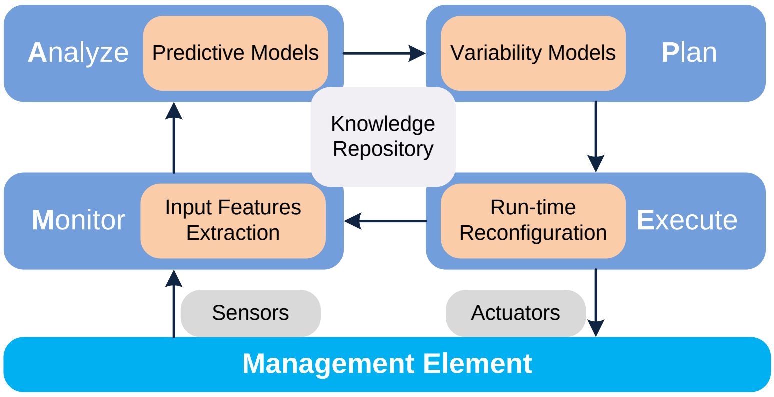


Figure 5.1.: Autonomic MAPE model [GSC09] in WSNs.

5.4.1. Comparative Analysis

To compare the proposed strategy LP to lifetime maximization, a network model has to be defined first. Consider a network consisting $\mathcal{N}$ wireless sensor nodes. A cluster-tree topology is used to avoid flooding unnecessary adaptation-oriented control packets in the network. In this case, the sensor nodes are grouped into $\mathcal{M}$ clusters controlled by a single sink node (base station).

Each cluster C_i has a (cluster) head node C_{hi} that manages its child nodes $S = \langle s_{i1}, s_{i2}, \ldots, s_{ij} \rangle$, where $j+1$ is the cluster capacity. Clusters are formed based on various criteria such as communication range, number and type of sensors, and geographical location. Generally, the head nodes are frequently elected for balancing the intra-cluster energy consumption. Please note, that the clustering method for such a homogeneous WSN is beyond the scope of this chapter.

We assume that all nodes s_{ij} are allocated an equal amount of energy E_0. Hence, we consider a worst case where the network has neither energy harvesters nor special nodes with a larger energy budget. It is assumed that m low-level controllable parameters $P = \langle p_1, p_2, \ldots, p_m \rangle$ have a range of adjustable values. The user can provide a set of QoS requirements $Q = \langle q_1, q_2, \ldots, q_n \rangle$ that are used for adapting the parameters of P. Moreover, the user has to define the task lifetime L_{task} in which the network must function with a best-effort performance. Accordingly, we define the WSN lifetime as the time span from deployment till the end of the intended task as depicted by L_{task} in Figure 5.2.

For maximization, on one hand, the lifetime is typically maximized by "squeezing" other quality metrics. The second row of Table 5.1 lists several relevant QoS metrics of real-world WSN-based applications. These metrics are severely affected by maximizing the operational lifetime. On the other hand, each sensor node spends an amount of energy to continue functioning beyond the expected task period. Such an amount of energy waste can be modeled as an area of a rectangle (in dark blue) shown in Figure 5.2.

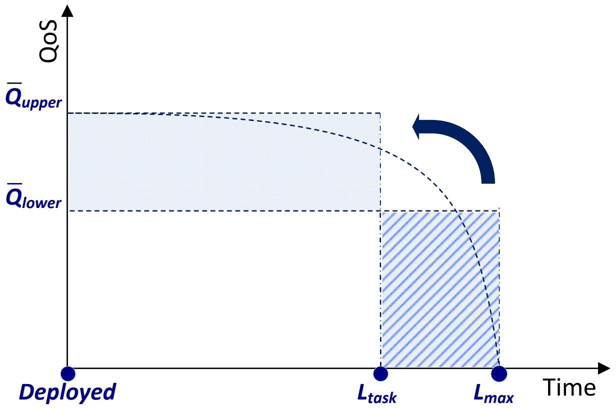


Figure 5.2.: QoS metrics in the light of heuristic lifetime maximization and planning.

The height of this rectangular represents the provided QoS level. Accordingly, the provided QoS after the completion of the task (hashed in the figure) $\bar{\Delta}$ can be characterized by the following equation:

$$\bar{\Delta} := \bar{Q}_{lower}(L_{max} - L_{task}) \tag{5.1}$$

where $\bar{Q}_{lower}$ is the QoS level that could be provided until the lifetime L_{max}. L_{task} denotes the task's actual lifetime. We consider level $\bar{Q}_{lower}$ as the worst case, where the service can be further improved. To sum up, sensor nodes mostly consume superfluous energy beyond the lifetime L_{task} at the expense of relaxing other service qualities. In the sequel, we discuss the architecture of our proposed framework in more detail.

The proposed LP exploits the residual energy thanks to the knowledge of expected lifetime — energy that still exists after the WSN task has been already accomplished — to improve the performance instead of wasting it beyond the lifetime L_{task}. Simultaneously, such a planning improves the provided QoS. Without planning, the average QoS resembles a lower boundary during the task lifetime. On the contrary, the planning strategy affords an upper level of QoS by investing the residual energy $\bar{\Delta}$ over the lifetime L_{task}. Equation 5.2 expresses the new level of the QoS $\bar{Q}_{upper}$ over the lifetime L_{task}.

$$\bar{Q}_{upper} := \left(\bar{Q}_{lower} + \frac{\bar{\Delta}}{L_{task}}\right) \tag{5.2}$$

In the following, we estimate a speedup ratio to examine the superiority of LP over lifetime maximization. A speedup ratio $\mathcal{S}$ between the naïve quality — obtained by solving the maximization problem — and the quality provided by LP is defined as given in Equation 5.3. The total power consumption is an approximation where a linear proportion exists with the average quality $\bar{Q}$. The term $P_2 = \left(\frac{E_0 - E_w}{L_{task}}\right)$ expresses the average consumed power during the lifetime L_{task}, where E_w is the residual energy in a sensor node. Similarly, the consumed power during the lifetime L_{max} is specified by $P_1 = \left(\frac{E_0 - E_w}{L_{max}}\right)$. The speedup ratio $\mathcal{S}$ is then given by Equation 5.3.

$$\mathcal{S} := \frac{\bar{Q}_{upper}}{\bar{Q}_{lower}} \cong \frac{\int_{t=0}^{L_{task}} P_2(t)dt}{\int_{t=0}^{L_{max}} P_1(t)dt} \tag{5.3}$$

where $P_2(t)$ and $P_1(t)$ are the instantaneous power consumption as functions of time. Knowing that the energy consumed in both cases is identical and that $L_{max} > L_{task}$ holds, we can infer that the speedup ratio is larger than 1 (i.e., $\mathcal{S} > 1$). After analyzing both strategies, i.e., maximizing and planning the lifetime, we then discuss the idea of self-adaptation in the perspective of network. In the next section, we explain how to implement the MAPE-based self-adaptation algorithm within the network.

5.4.2. Hierarchical Self-adaptation

In this work, we focus on the self-adaptation mechanism and performance of each sensor node under LP. We address the self-adaptation mechanism from the network perspective. Accordingly, we introduce a hierarchical WSN exploiting the self-adaptation scheme. Figure 5.3 depicts the MAPE loop assignment within a cluster-based WSN. The placeholders M, A, P, and E stand for the components of the MAPE loop, namely **M**onitor, **A**nalyze, **P**lan, and **E**xecute. As demonstrated in the figure, both cluster heads and their children implement the components of the MAPE loop, whereas the base station only analyzes and plans for the cluster heads.

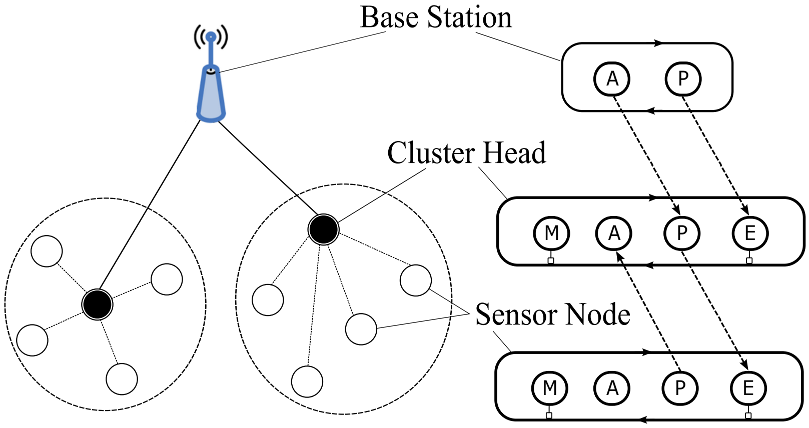


Figure 5.3.: An heuristic network architecture of a proactive WSN exploiting the MAPE scheme.

Under these settings, reconfigurations of the low-level parameters in each child node have to be reviewed by the corresponding cluster head, which is "data-rich" about its cluster. This constraint primarily sidesteps the execution of plans that harm the inter-sensors cooperation. The base station, in our setting, gives permission to execute proposed plans in the cluster heads. This hierarchy of data flow can highly improve the nodes' self-adaptation with emphasis on the global network parameters. In general, two research questions are raised when dealing with proactive adaptations: (i) How to select the best fitting target configuration? And (ii) how to deal with conflicting objectives?

Our proposed LP implicitly solves these two challenges. First, the best fitting is considered close to the upper QoS boundary as long as the lifetime is in its predefined range $[0, L_{task}]$. Second, the conflicts are sidestepped through the QoS model in lifetime planning. To confirm this hypothesis, we further investigate the QoS metrics to discover the possible conflicts. Next, we investigate QoS metrics of interest such as communication reliability, latency, and energy consumption.

5.4.3. QoS Modeling

In this section, we provide a QoS model that relates application-level QoS metrics to low-level parameters. The goal is to explore any possible conflicts between the various QoS metrics when adapting LP.

5.4.3.1. Analytical Model

For our analysis, we account for three major QoS metrics, i.e., reliability (PDR), latency (delay), and energy consumption. These mapping functions are devoted to control the node's behavior at run-time. Specifically, our QoS model is based on merging the Hoes's model [HBT+07] and the probabilistic Markov chain model of the IEEE 802.15.4 standard [ISA11] referred to as Park's model [PDMFJ13].

Reliability: In our model, reliability $\mathcal{R}(s_{ij})$ is defined as the probability of correctness and success of packet transmission between the nodes i and j. Specifically, our model differentiates itself from Park's model [PDMFJ13] by further considering the signal corruption in the accompanied noise (i.e., correctness), while Park's model accounts only for the contention-loss probability (i.e., successful transmission). The multiplicative reliability metric $\mathcal{R}(s_{ij})$ is expressed in terms of the Signal to Interference plus Noise Ratio (SINR) and the approximated probability of successful packet transmission $\tilde{R}(s_{ij})$ as given in Equation 5.4.

$$\mathcal{R}(s_{ij}) = \left(1 - Q\left(\sqrt{2 \times \mathit{SINR}}\right)\right)^{b} \times \tilde{R}(s_{ij}) \tag{5.4}$$

where b is the packet size (in bit) and $Q(\cdot)$ is the tail probability of the standard normal distribution. The second part of Equation 5.4 represents the probability of successful packet reception. It mainly depends on the packet generation rate, the operational duty cycle, and the (re-)transmission times. The term *SINR* has a direct relationship with the transmission power via Equation 5.5.

$$\mathit{SINR} = \left(\frac{P_{rx}}{N}\right) = \frac{K_r \times P_{tx} \times (d_0/d)^r}{K \times T \times B} \tag{5.5}$$

where P_{rx} and P_{tx} are the received and transmitted power respectively. The noise level N is expressed by the Boltzmann's constant K, the effective temperature in Kelvin T, and the receiver bandwidth B. The remaining terms are as follows: r is the path-loss coefficient ($r \geq 2$). K_r is the constant gain factor. d_0 is the reference distance, while d is the actual distance between a transmitting node Tx and a receiving one Rx.

Latency: In Park's model [PDMFJ13], the latency is defined as the time interval from the time point a packet is at the head of its MAC queue and ready to be transmitted until the transmission is successful and an acknowledgment has been received. In contrast, our model extends this notion via considering the time span from the arrival of a nearby objective until its detection (detection delay) as well. Equation 5.6 describes the additive delay metrics $\mathcal{D}(s_{ij})$ as a function of the sampling rate r_s, the detection duration D_s, and the transmission delay $\tilde{D}$.

$$\mathcal{D}(s_{ij}) = \left(\frac{1}{r_s} + D_s\right) + \tilde{D} \tag{5.6}$$

Energy Consumption: In this work, we assume that the radio transceiver is in sleep mode during the back-off mechanism specified by the IEEE 802.15.4 standard [ISA11]. Moreover, we presume that packet transmission and reception have identical energy consumption. Accordingly, the total energy consumption $\mathcal{P}(s_{ij})$ can be expressed by the Equation 5.7.

$$\mathcal{P}(s_{ij}) = f(s_{ij})\,(E_s \times r_s + P_{mcu}) + E_{radio}\,(r_o + \kappa \times r_i) \tag{5.7}$$

where E_s is the energy consumption of the sensing module, P_{mcu} is the consumed power for processing the sampled data, and $f(s_{ij})$ is the (active) radio duty cycle of the transceiver. The output traffic rate r_0 is the average rate of packet transmission while r_i represents the received traffic rate from κ neighboring nodes. Neglecting the consumed energy in sleep mode, the term E_{radio} reflects the average power consumption during channel sensing, MAC back-off state, and packet transmission including both, successful transmission and packet collision.

5.4.3.2. Model Validation

We discuss the validation of the proposed QoS analytical model. We implement our aforementioned analytical model in MATLAB and compare the results with the simulations by the Cooja simulator [ÖDE$^+$06] in Contiki OS [DGV04]. We estimate the contention loss and the transmission delay in the various simulation scenarios. Simultaneously, we measure the parameters in the MAC layer that would be applied later in our analytical model in MATLAB, such as the probability of the first CCA α and the probability of the second one β. During our simulations, we utilize the *powertrace* application [Dun11] to estimate the power consumption.

Table 5.2 presents the general configurations of the simulations. Considering the PER and the contention loss ratio, we carry out simulations by setting various values of parameters in Cooja, such as number of nodes, channel check rate, and idle time interval, based on the equations stated above.

For the number of nodes, we select the values of 4, 8, and 16 for a single cluster head. Besides, we set the channel check rate to 8, 16, 32, and 64 (Hz), respectively, in each simulation. For simplicity, we fix the idle time interval to 1000 milliseconds (ms) for all simulations. In this case, we run 12 ($3 \times 4 \times 1$) simulations in total. Taking the matter of time into account, we run each simulation in approximately 10 minutes for each scenario. In each simulation, multiple transmitters (i.e., child nodes) generate a data packet in every idle time interval. Then, each node performs two CCAs before transmission. If these two consecutive CCAs are both clear, meaning that the channel is idle at the moment, then the node transmits that data packet to the receiver (i.e.,

Table 5.2.: Simulation parameters.

Parameter	**Configuration**
Framer	802.15.4 framer
Radio Duty Cycling	ContikiMAC
MAC	CSMA/CA
Network	Rime
Radio Model	Unit disk graph medium
Simulated Node Type	Tmote sky
Packet Size	32 bytes
Transmission Power	-7 dBm
Transmission Range	10 meters
Idle Time Interval	1000 ms
Number of Nodes	4, 8, 16
Channel Check Rate	8, 16, 32, 64 Hz

cluster head).

Reliability. In our model, the reliability represents the probability that a sensor node correctly and successfully transmits a data packet. Thus, it describes the probability that neither a single bit error occurs in the packet during transmission, nor that the data packet is discarded as a result of a contention of the communication medium.

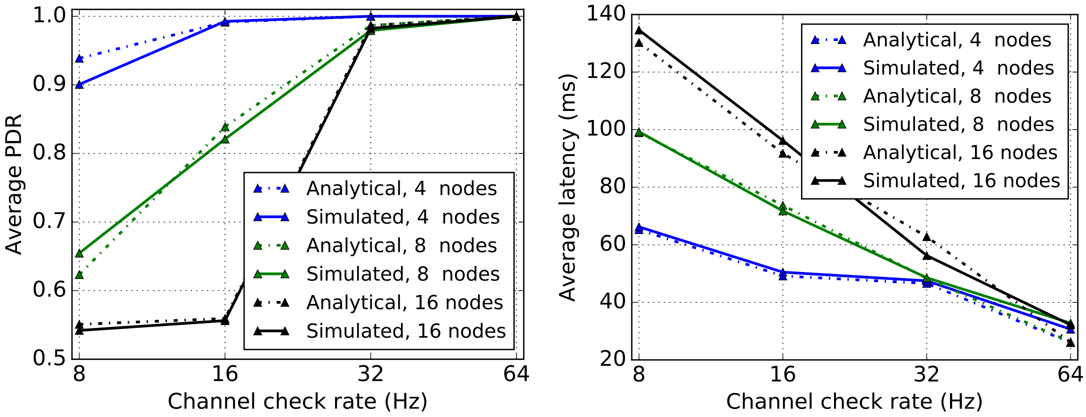
By utilizing the measurements from simulations, we obtain the average PDR of all the sensor nodes in different simulation scenarios. Then, using the reliability formula (Equation 5.4), we obtain the average PDR of our analytical model in the corresponding scenario via MATLAB.

Figure 5.4(a) shows the deviation of the average PDRs between the analytical model and the simulation. As shown in the figure, the deviation of the average PDR between the analytical model and the simulation decreases along with the increment of the channel check rate and the number of nodes. Even in the scenario with four nodes and 8 Hz channel check rate, the difference between the analytical and the simulated results is less than 4%.

Latency. As stated in Equation 5.6, the average transmission latency in our model contains two parts, D_s and $\tilde{D}$. The first part — the detection delay — is easy to obtain once the sensor's sample rate, the duration of sampling, and the duration of detection are fixed. The second part, the transmission delay, is the average delay for a successful packet transmission. It is defined as the time interval from the time instant the packet is at the head of its MAC queue and ready to be transmitted until the transmission was successful and the acknowledgment has been received. Therefore, we mainly discuss the validation of the average transmission delay $\tilde{D}$.

Similarly, we obtain the average transmission delay of all the sensor nodes from simulations in different scenarios. By using the settings of parameters in the MAC layer, we obtain the average transmission delay in the analytical model.

Figure 5.4(b) illustrates the deviation of average transmission delay between our analytical model and the simulations. As shown in the figure, the average transmission delay is affected by the number of nodes when the channel check rate is less than 32 Hz. Additionally, the delay is affected by the channel check rate whenever the rate is greater than or equal to 32 Hz. In the scenario of 16 nodes, the difference between the analytical model and the simulated one is still less than 10 ms. To conclude, our proposed analytical model considers several layers of the network stack. According to the validation work, the model is feasible enough to be invoked for detecting possible QoS conflicts due to LP.



(a) Average PDR versus channel check rate

(b) Average latency versus channel check rate

Figure 5.4.: Performance metrics with various channel check rates evaluated in Cooja simulations.

5.4.3.3. Discussion

In this section, we extracted the main model parameters to search for possible conflicts. Figure 5.5 summarizes the aforementioned relationships via delineating a mapping model between the low-level controllable parameters and the high-level QoS metrics. The rectangles represent constant values, while ovals are metrics that depend on the other parameters. The considered quality metrics are reliability, latency, and lifetime. Lines with a filled circle at the end embody direct proportions, i.e., if a low-level parameter is increased, then the corresponding metric increases as well. Similarly, lines with an open circle indicate inverse proportions.

The model depicts the possible trade-offs where adjusting a parameter has a positive influence on some quality metrics and a negative influence on others. These trade-offs are implicitly avoided in LP. This fact emerges from intentionally reducing the lifetime below the maximal operational lifetime. Moreover, other metrics — such as reliability and delay — have no inter-contradictions, as can be seen in Figure 5.5. Therefore, LP not only improves the provided QoS, but also avoids using any sophisticated algorithm to optimize contradicting metrics. However, to ensure the correctness of these remarks, we have to evaluate our novel QoS model. As a proof of concept, the next section discusses our implementation of an office monitoring scenario. This case study is investigated to examine the network performance with and without lifetime planning.

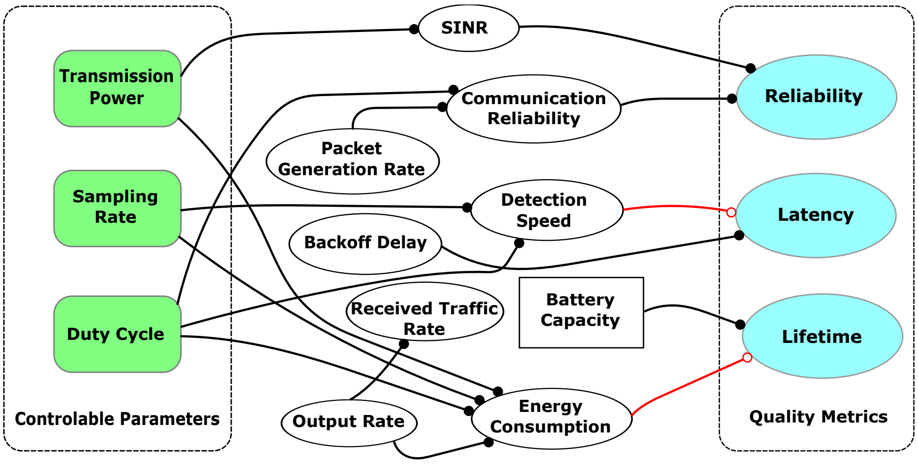


Figure 5.5.: Hierarchical relationships among QoS metrics.

5.4.4. Case Study: Office Monitoring Scenario

Energy efficiency in smart buildings is an important application objective of WSNs. To conserve energy, for example, online data has to be collected as a basis for dynamically controlling the lighting and heating systems. Such data also include the location of persons and their activities. In this section, we consider an active indoor localization scenario. A network of TelosB sensor nodes is simulated in the Cooja simulator. Specifically, the testbed consists of eleven Static Sensor Nodes (SSNs), which monitor the observed geographical areas, measure environmental changes, and forward event packets to the sink (base station). Another set of Mobile Sensor Nodes (MSNs) represents the indoor traffic. These MSNs broadcast identification packets to neighboring SSNs. To localize the MSNs, the sink node processes the RSS values for communication links between the MSNs and the nearby SSNs.

5.4.4.1. Scenario Dynamics

In real-world office monitoring applications, several interesting events emerge due to the environmental dynamics. The occurrence of such events is exploited to reconfigure the network in the light of LP. For instance, the mobility of a MSN is highly affected by a person's status: Detecting the person is "stationary" or "walking", thus, triggers a set of re-configurations. During the "walking" state, the RSS is more frequently detected based on the connectivity between the SSNs and the MSN. In this case, duty cycle f and the sampling rate r_s of the transceiver need to be set at a higher level. Moreover, the interference is waving when a person is walking. Thus, the transmission power T_{tx} should also be modified to overcome the imposed interference. The patterns of day and night are also of interest to our self-adaptation mechanism. The hallway in an office building, for example, has mostly low traffic at night. Less data has to be reported to the sink node during the night time. Accordingly, specific configurations could be customized to improve other performance metrics and save energy. In the next section, we discuss reasoning engines for modulating QoS metrics.

5.4.4.2. Reasoning Engine

According to the MAPE framework, the collected context information has to be processed to discover interesting events. In addition, reactions have to be generated in the light of these detected events. A prior knowledge of the application scenario leads to the definition of a set of possible events. Hence, reactions can be earmarked even before deployment such as the technique presented by [SVNSO$^+$11]. Nevertheless, environmental dynamics typically result in unexpected events that could also be exploited. Consequently, a "good" reasoning engine should make use of nearly any interesting event. Examples of such engines are the Constraint-Satisfaction Problem [GRF$^+$12], MDP [MGR09], Fuzzy inference [MPH07], and Event Condition Action (ECA) rules [ST09]. We use ECA rules to reconfigure the nodes, since such rules are simple and have low computational overheads. Formerly, ECA rules are used in active database systems [ST09]. Afterwards, they have been widely utilized as a flexible strategy to support management, reconfiguration and execution of reasoning rules.

Basically, ECA rules comprise three parts as follows. First, the *event* part specifies the signal that triggers the invocation of the rule. Second, the *condition* part is a logical test that, if evaluating to be true, then it causes the action to be executed. Finally, the *action* part is a function or a procedure that can be called by the condition evaluator. Handling various QoS metrics demands the modulation of manifold low-level parameters. Hence, an ECA rule has to execute multiple actions per single condition. In the next section, we introduce the performance evaluation through a comparative study among LP, unplanned adaptation, and lifetime maximization strategies.

5.5. Performance Evaluation

An experimental study on the office monitoring scenario has been performed to evaluate the proposed LP. The goal of LP is to improve the QoS of the network with adequate network lifetime. Therefore, a proactive adaptation mechanism based on the MAPE framework has been adopted. To fulfill this goal, we focus on answering the following three research questions: (i) Does LP improve the QoS metrics compared to static heuristics and unplanned adaptation? (ii) Does the actual network lifetime meet the application requirement? and (iii) Are the QoS metrics confined within boundaries throughout the entire lifetime?

As aforementioned, a scenario of office monitoring is engineered for evaluation purposes. The inherent dynamics in such a scenario are to be exploited to show the effect of planning the QoS levels throughout the entire lifetime. The simulator runs on a virtual machine with a 2.5 GHz processor and 8 GB RAM using an Ubuntu OS. Figure 5.6 shows the layout of the proposed office monitoring scenario. A mobile node broadcast its current coordinates to the neighboring nodes while moving from the right side to the left. As long as a cluster child node receives a packet, it forwards the packet to a so-call cluster head. The cluster head then delivers the packet to the sink via multiple hops. In this case, the simulation provides an application that tracks a moving object in an office monitoring scenario.

For a comparative analysis, we contrast LP to two different strategies, namely lifetime maximization and unplanned adaptation. The former strategy represents a fixed strat-

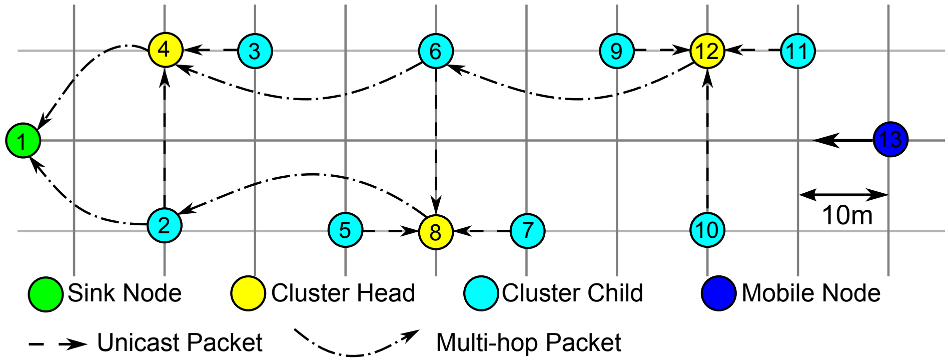


Figure 5.6.: An office monitoring scenario implemented in the Cooja simulator.

egy in which the controllable parameters P are assigned to the minimum values. The latter strategy — inspired from [SVNSO$^+$11] — exploits the environmental dynamics in a proactive manner in order to optimize system performance. Next, we discuss the implementation details of LP and of the unplanned adaptation strategy.

5.5.1. Unplanned Adaptation

Steine et al. [SVNSO$^+$11] introduce an adaptation method by exploiting design-time knowledge of the application scenario dynamics. At design-time, operation modes are defined, as well as the controllable parameters of the network stack. In this case, the parameters are adapted in response to the expected events. Such an approach is only dealing with a limited set of events. Thus, we refine the possible events and their corresponding conditions. The adaptation framework, designed in [SVNSO$^+$11], is not planned in terms of relevant QoS metrics. Hence, invoking the unplanned adaptation for the comparative study is able to clarify the advantages of our proposed approach.

5.5.2. Lifetime Planning

In this section, we detail LP in the simulated office monitoring scenario.

Algorithm 3 introduces the major details of applying LP in a sensor node. At design-time, lower and upper boundaries of QoS are estimated in the light of the expected task lifetime L_{task} and the initial energy budget E_0. Besides, the lower boundary is adjusted to the user requirements, if existing. Otherwise, it is assigned to the average QoS of the lifetime maximization. For the upper boundary, a set of mapping functions — extrapolated from the analytical model and simulations — determines the boundary, as stated in line 3. Considering only the analytical model is not practical due to the run-time data loss in the upper layers such as transmission collisions and failures to acknowledge packet reception. Moreover, the probabilities of a busy medium (busy CCAs lead to back-offs in the time domain) have to be used during the performance evaluation. We simulate the network in Cooja for each sensor node, depicted in lines 1-4 in Algorithm 3.

During run-time, ECA rules are continuously evaluated based on the environmental changes. Specifically, four rules have been designed in the light of the criteria listed in

Algorithm 3 Lifetime planning algorithm

```
Require: task lifetime $L_{task}$, energy budget $E_0$, user requirement $\{\mathcal{R}_{min}, \mathcal{D}_{max}, \mathcal{P}_{max}\}$
    /* Design-time estimation of the upper QoS boundary */
 1: for $0 \leq i < (\mathcal{M} - 1)$ do
 2:     for $0 \leq j < ((\mathcal{N}/\mathcal{M}) - 1)$ do
 3:         $P \leftarrow f(E_0, L_{task})$ where $P = \{r_s, f, P_{tx}\}$
 4:         determine $\mathcal{R}(s_{ij})$, $\mathcal{D}(s_{ij})$, $\mathcal{P}(s_{ij})$
 5:     end for
 6: end for
    /* Run-time processing */
 7: monitor $\text{QoS}_{instantenous} = \mathcal{R}, \mathcal{D}, \mathcal{P} \quad \forall \quad s_{ij} \in \mathcal{S}$
 8: if an ECA rule is fired then
 9:     update the parameters $P$                          ▷ mathematical model
10: end if
11: if (head $C_{hi} == 0$) then                           ▷ $C_{hi}$ rejects the plan
12:     go to line 8
13: else
14:     execute the update
15:     go to line 7
16: end if
```

Table 5.3. Two of them monitor the environmental events. The other rules confine the QoS metrics in their boundaries. Finally, the algorithm introduces a simple protocol between a child node and its cluster head for approving the system updates.

Besides, Table 5.3 summarizes the operational mode and all possible scenarios for lifetime maximization, unplanned adaptation, and LP, respectively. In fact, adapting general criteria — such as the traffic size and the speed of mobile nodes — mostly covers all possible events in office monitoring scenario. The settings are classified in the light of a mobile node's state i.e., mobile or stationary. The former has been classified in accordance with the speed and the number of mobile nodes. Thus, four cases emerge by considering only two linguistic variables *low* and *high*, as expressed in the table. Each strategy has different values of the transmission power P_{tx} and the channel check rate r_c, an indirect indicator of the duty cycle. For unplanned adaptation, the values indicated in the table are selected to reduce the power consumption, as proposed in [SVNSO^{+}11]. Alternatively, the values for LP are derived based on the required lifetime L_{task} via the mapping functions. Below, we discuss the obtained results in the context of the aforementioned research questions.

5.5.2.1. Evaluating the QoS metrics

In this section, we examine the impact of applying LP, unplanned adaptation, and lifetime maximization on the QoS metrics. Figure 5.7 and Figure 5.8 show a comparison between the three strategies in terms of the average PDR — representing a realistic measure of the reliability $\mathcal{R}$ — and the average delay $\mathcal{D}$ in several milliseconds. The horizontal axes gives the ID number of children according to Figure 5.6. In these exper-

Table 5.3.: Mode selection for office monitoring scenario.

Parameters		**Configurations**					
Mode of MSNs		Mobile				Stationary	
Mode of SSNs		Stationary				Stationary	
Scenario Settings	Traffic	Low	Low	High	High	–	–
	Speed	Low	High	Low	High	Low	High
Number of MSNs		1	1	4	4	1	4
Moving Speed (m/s)		0.5	1	0.5	1	–	–
Lifetime Maximization	P_{tx} (dBm)	-7	-7	-7	-7	-7	-7
	r_c (Hz)	8	8	8	8	8	8
Unplanned Adaptation	P_{tx} (dBm)	-7	-7	-7	-3	-7	-7
	r_c (Hz)	8	16	16	64	8	16
Lifetime Planning	P_{tx} (dBm)	-7	-3	-3	0	-7	-3
	r_c (Hz)	8	32	64	64	32	32

iments, we focus on the communication link between cluster heads and their children. Accordingly, QoS values of the sink and the cluster heads (node 1, 4, 8, and 12) have been eliminated from the figures.

As expected, LP achieves a higher reliability and a shorter latency than the other approaches, as can be seen in Figure 5.7 and Figure 5.8, respectively.

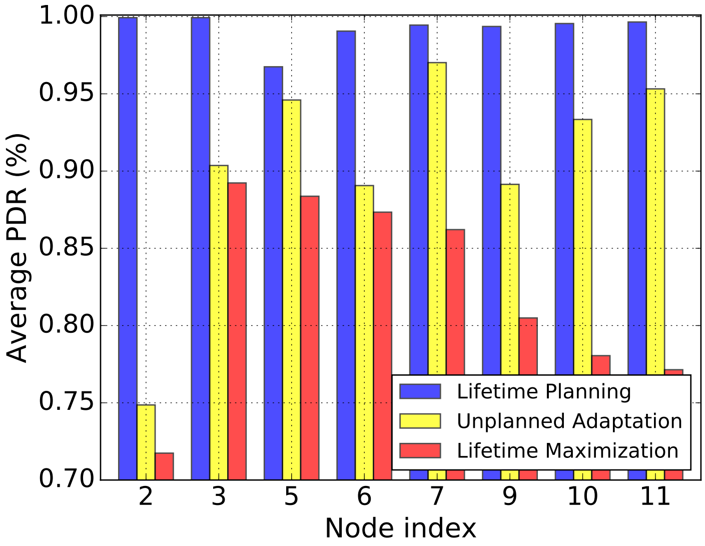


Figure 5.7.: Impacts of three strategies on the average PDRs for the office monitoring scenario.

Particularly, LP achieves an approximately 9.6% higher reliability than unplanned adaptation and a 20% higher reliability than lifetime maximization. Similarly, LP gets about 53% less delay than unplanned adaptation and 78% less delay than lifetime maximization. This excel is reasonable due to spending more energy in case of LP. However, we still need to double-check the impact of such improvements on the lifetime.

Figure 5.9 delineates the lifetime of cluster heads and children for all strategies. The

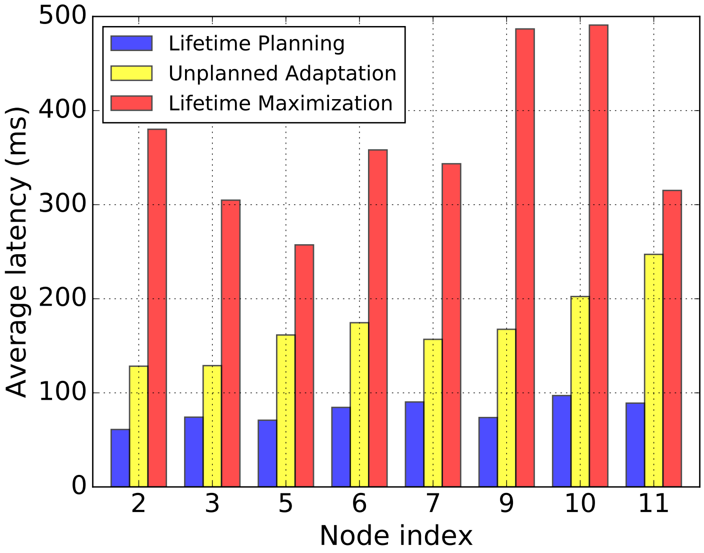


Figure 5.8.: Impacts of three strategies on the average delay for the office monitoring scenario.

average actual lifetime in case of LP is about 40% less than the one of unplanned adaptation, and 50% less than the one of lifetime maximization. Nevertheless, the obtained network lifetime (approximately 100 days) achieves the task lifetime used for estimating the QoS boundaries. Hence, we can conclude that LP (i) manages to improve the QoS metrics (i.e., reliability and latency), (ii) avoids any adaptation conflicts, and (iii) meets the expected task lifetime.

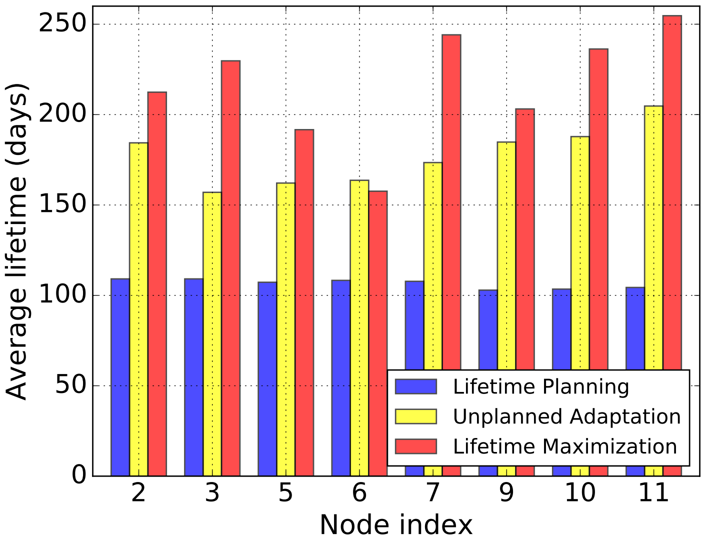
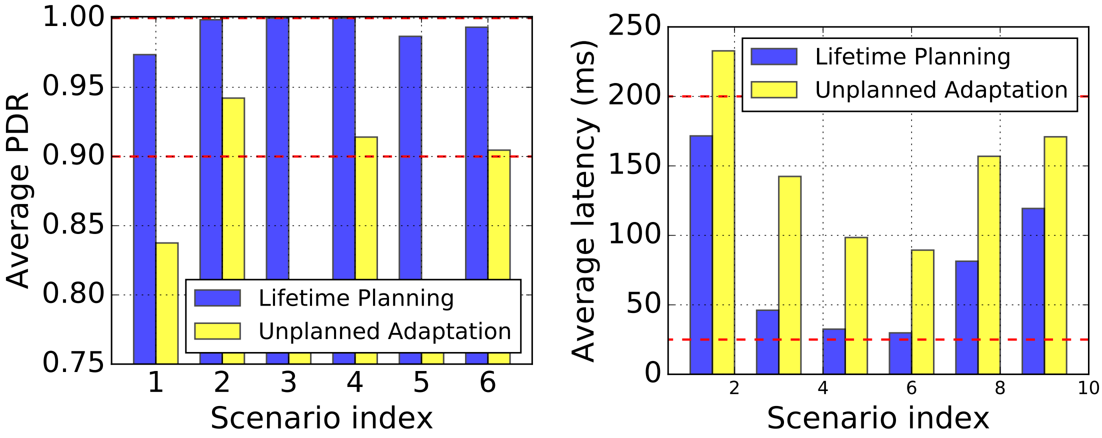


Figure 5.9.: Impacts on the lifetime for the office monitoring scenario.

5.5.2.2. Evaluating the QoS boundaries

Finally, we need to indicate how the expected lifetime is met. In this section, the average reliability and the average delay are examined for node 6 during several runs over the various scenarios. As it can be seen in Figures 5.10(a) and 5.10(b), the QoS boundaries are colored in gray and marked with triangles. Obviously, both strategies have the same behavior, but they reside at different levels. For the PDR, the LP (in blue) values are confined between the two gray thresholds, as shown in Figure 5.10(a). Alternatively, unplanned adaptation (in red) is reduced without any restrictions to reduce the energy consumption. Figure 5.10(b) shows a similar behavior for the delay metrics.



(a) PDR

(b) Latency

Figure 5.10.: Average end-to-end metrics in cases of LP and unplanned adaptation in terms of various scenarios.

5.6. Conclusion

Upcoming applications require WSNs to meet application-specific performance targets such as high reliability, low latency, and high throughput. In this chapter, we propose an application-oriented strategy to handle the self-adaptation of QoS metrics in WSNs, referred to as LP. Different from conventional QoS management designs, LP improves QoS metrics by exploiting an additional amount of energy. Such energy is gained from limiting the lifetime to the time required to fulfill the task. Based on a validated analytical model, we show that LP is able to avoid possible conflicts among the QoS metrics. Furthermore, an office monitoring scenario is engineered and used to examine the proposed strategy. The simulation results show that LP highly improves the reliability and lowers the latency. This profit comes at the expense of reducing the lifetime of a WSN. However, the shortened lifetime is still long enough to complete the assigned application tasks.

6 Conclusions and Outlook

This chapter concludes the thesis. It gives a summary of the presented work in Section 6.1 and a number of future prospects in Section 6.2.

6.1. Conclusions

The advancements in wireless technology and MEMS have enabled the boosting progress of WSNs, that in turn have fostered the emergence of a plethora of WSN-based applications in various fields such as agriculture, healthcare, industry, transportation systems, to name but a few. However, due to the constraint resource of battery-powered sensors, unreliable communication medium, and other limitations, these applications are still facing major QoS provision issues that prevent their widespread endorsements.

In this thesis, we concentrated to overcome the challenges through a number of research contributions provided in this thesis. Based on the WSN protocol stack, we began with the network and MAC layers: We proposed a multichannel opportunistic routing approach (*MOR*) so as to further improve the reliability again the adverse CTI, as well as to keep a high reliability, a low latency and a high energy efficiency. We evaluated MOR through extensive experiments to reveal the advantages (and also trade-offs) of the protocol especially in extreme adverse conditions.

Afterwards, covering MAC and physical layers, we proposed a machine learning-based strategy (*LiM*) to lessen the data redundancy in flooding. We modeled the data redundancy problem as a multi-armed bandit optimization problem and then, we exploited the Exp3 to overcome the optimization problem. To the best of our knowledge, we were the first to combine light-weight machine learning together with concurrent transmission in WSNs.

Subsequently, we introduced a physical- and MAC- layer communication scheme (*PiP*) in order to realize the packet concatenation in the air. We used so-called PA operations to control the packets in a packet to reduce the conflicts of the medium usage. We showed that PiP is able to significantly decrease data collection time while maintaining a high reliability and a high energy efficiency.

At last, we introduced an application-oriented adaptation strategy (*LP*) to relax the QoS contradiction and meanwhile to achieve a best-effort performance, which directly connects MAC layer to application layer. To realize this, we introduced an analytical model together with a so-called MAPE model for run-time adaptations in LP. Experiments showed that LP is able to satisfy the user requirement of sufficient lifetime while maintaining a high reliability and a high energy efficiency.

To sum up, all the protocols and algorithms discussed in this thesis provide approaches to improve, optimize, and analyze the QoS for WSNs. Compared with the existing state-of-the-art in the literature, the proposed protocols and algorithms substantially make efforts on the end-to-end QoS enhancement in WSNs, namely reliability, timeliness, and energy efficiency. We not only developed new models, communication protocols, and distributed algorithms to improve the end-to-end QoS of WSNs, but also implemented all our approaches based on the IEEE 802.15.4 standard [ISA11], TelosB sky mote (hardware), and Contiki OS (software). That is to say, we designed and developed the software for TeloB sensor node running in the Contiki OS. This enables us to study the WSN topics in a more realistic way — practically in real-world WSN-based applications. Moreover, for all proposed protocols, we evaluated their performances through extensive numerical real-world experiments in the FlockLab testbed [LFZ+13b] and through simulations by Cooja [ÖDE+06] in Contiki OS.

Generally, the guarantee and optimization of QoS are the key to the adoption of WSNs in real-world applications so as to accomplish IoT and CPS in Industry 4.0. Specifically, the main goals of the thesis are to (i) propose and implement novel communication algorithms and protocols to optimize multi-objective QoS in multihop low-power WSNs; (ii) validate and evaluate the effectiveness of the proposed algorithms and protocols in terms of several key QoS metrics, i.e., reliability, timeliness, and energy efficiency, through simulations and real-world experiments; and (iii) compare the performance of solutions to the existing state-of-the-art solutions and then analyze performance based on the QoS trade-offs.

In conclusion, this thesis explains that opportunistic routing, machine learning, concurrent transmission, and application adaptation provide enabling technologies for the purpose of accomplishing the main goals: MOR (Chapter 2) provides a dependable solution for WSNs against interference in real world; LiM (Chapter 3) exploits a reinforcement learning algorithm to improve the energy efficiency of WSNs, especially for data dissemination in WSN applications; By actively manipulating of concurrent transmissions, PiP (Chapter 4) aims to enhance the timeliness for mission-critical data collection in WSN applications; LP (Chapter 5) designs a dynamical tuning strategy between MAC layer and application layer so as to achieve the target lifetime of a WSN. Additionally, the evaluations based on real-world testbeds as well as simulations validate the excellent performances and anticipate a bright future for the full endorsement of WSNs. Moreover, the performance comparisons between the proposed solutions and existing state-of-the-art, as well as the QoS trade-off analysis, provide a number of valuable research insights of WSNs towards IoT and CPS in realization of Industry 4.0.

6.2. Outlook

The end-to-end communication performance is a key point that matters a lot for many real-world low-power wireless applications, ranging from data collection to other CPS scenarios. Existing WSN research tends to focus on protocol design for specific applications. However, many WSNs simultaneously need to share a common communication infrastructure for various applications, e.g., data collection and data dissemination. Thereby, WSN protocols are required to support heterogeneous applications.

Regarding the proposed protocols and algorithms of this thesis, as future work, we envisage to consider wrapping up the proposed strategies as a generic cross-layer solution aiming at various QoS trade-offs. Correspondingly, the generic protocol can be adjusted to the specific application by the users to fast-boost the design and development. For instance, the protocol is able to dynamically handle different application scenarios in WSNs, such as one-to-many, many-to-one, one-to-one and many-to-many scenarios, similar to LWB [FZMT12] and ORPL [DLV13].

Additionally, in the future, we plan to extend LiM to the frequency domain, i.e., adding a channel hopping strategy provided by MOR, so as to enhance the dependability against the uncertain interference in real-world scenarios. Besides, we are also interested in applying PiP in the next EWSN Dependability Competition in order to evaluate the performance in a much larger WSN testbed — D-Cube [SBWR17].

Moreover, as one of the most critical QoS metrics, security in WSNs is a must to be contemplated for the fulfillment of the real-world IoT within Industry 4.0. This is a great challenge that requires numerous efforts for the communities that are working and contributing in wireless technologies, standards, and network security.

Last but not least, in general, we also believe a number of novel approaches — such as artificial intelligence [RN16], bio-mimic optimization strategies [ARAI13], machine learning algorithms [ALNT14], and battery-less wireless communication (e.g., Radio Frequency Identification (RFID) and ambient backscatter [LPT^{+}13]) — can be further employed together with WSNs, Thereby, contributing to pushing forward smart and efficient IoT and CPS in Industry 4.0.

Appendices

A

Competition: Using Enhanced OF∂COIN to Monitor Multiple Concurrent Events under Adverse Conditions

Monitoring various states of multiple nodes reliably and timely in narrow-band and power-constrained wireless sensor network is extremely challenging, not to mention in existence of interference. An enhanced Oriented Flooding protocol with Partial COnstructive INterference (*eOF∂COIN*) [MZT+18] is proposed to collect the states of sources via many-to-many communication. Basically, eOF∂COIN is based on two phenomena, i.e., constructive interference and the capture effect. To boost the dependability against interference, eOF∂COIN further exploits a channel-hopping scheme. Furthermore, eOF∂COIN maintains a light-weight topology model to achieve oriented many-to-many communication.

A.1. Introduction

Data packets can be received correctly by a commercial IEEE 802.15.4 device, if the wireless signal (i) is stronger than noise, or (ii) is diverse from noise in the domain of time or frequency. In terms of the time domain, transmitting messages when the channel becomes clear is intuitive. It is easy to be implemented via low-power medium access control (MAC) mechanisms with clear channel assessment (CCA) such as ContikiMAC [Dun11]. However, it might not be the best choice since distinguishing a wireless signal from noise is quite challenging for CCA. Moreover, differentiating a signal from noise in the time domain, e.g, wait-and-transmit, introduces latency as most of unknown interference is stochastic. Therefore, our enhanced Oriented Flooding protocol with Partial COnstructive INterference (eOF∂COIN) wakes up all devices globally without CCA and makes the signal as strong as possible (using constructive interference) in different channels (using a frequency hopping mechanism) to avoid the noise. In the given scenario, which is more complicated than the scenario of last year, multiple sources need to be monitored and the data should be reported to several destinations in real-time. The observed 8th pin of each source is required to be *OR*ed together in every destination, i.e. many-to-many communication. The states of other pins are sent to the designated destination hop-by-hop, i.e. in one-to-one communication. However, it is necessary that the one-to-one message is replicated on other routes in the network to achieve reliability in spite of consuming more energy such as opportunistic routing [LGDJ12], MOR [ZLT17]. Therefore, eOF∂COIN, a many-to-many scheme, is applicable in this scenario to achieve high reliability and low latency.

A.2. Enhanced OF∂COIN

A.2.1. Oriented Flooding

An off-the-shelf IEEE 802.15.4 transceiver can be triggered to start recording the wireless signal when the preamble is strong enough and the rate of successful reception can be increased [FZTS11,KW16]. OF∂COIN [MTH$^+$17] is an approach which affords feedback. That is, different information which is located at the end of the packet can be transmitted via the capture effect after the constructively interfered identical information. The feedback including the rank of the receiver, which represents how far it is from the given node, allows the flooding to be oriented. If the given node is the destination node, then it can make messages propagate in the right direction as shown in Figure A.1.

Nodes decide whether it is in the yellow area in Figure A.1 by calculating the ranks to the source and the destination. Gray nodes outside the yellow area will not participate in the communication and sleep. Green nodes on the routes, i.e., inside the yellow area, turn off the radio immediately to save energy once having successfully forwarded the received messages. In this scenario, nodes in colorful blocks relay messages and others sleep as shown in Figure A.2. eOF∂COIN reserves the oriented flooding from the original protocol. Besides, it is improved with respect to many-to-many mechanism, message synchronization, frequency hopping and network scalability.

A.2.2. Many-to-many Mechanism

As mentioned above, we need to add the many-to-many characteristic to the original OF∂COIN protocol. Each source fills its states of pins in the private information field (shown in Figure A.3) to inject its own information to the network. The lengths of the public information and private information are to be determined by the overall number of sources. Relay nodes mix their private information field to the common information field. All nodes, i.e., sources, relays, and destinations, fill their own rank in the topology information field that are calculated during the set-up phase. The identical part and the different part of one packet could be checked separately.

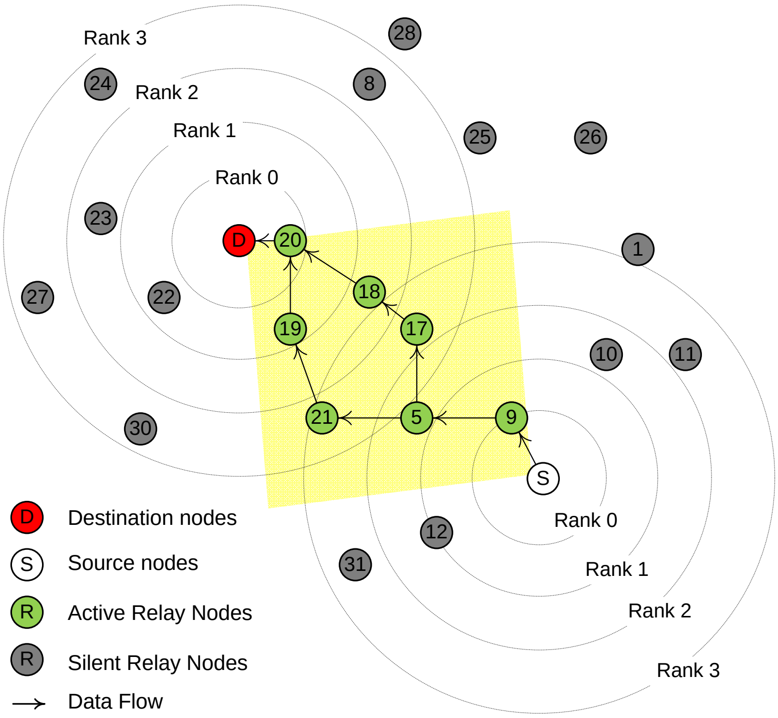


Figure A.1.: Oriented flooding.

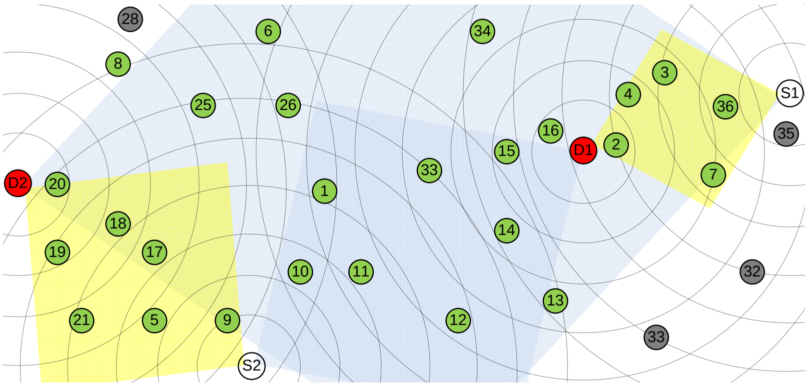


Figure A.2.: Oriented flooding using multiple sources and destinations.

Octets:								
4	1	1	TBD	1	TBD	1	1	2
Preamble Sequence	Start of Frame Delimiter	Frame Length	Public Information	Identical Part Checksum	Private Information	Topology Information	Different Part Checksum	Footer /CRC
Identical Part					Different Part			

Figure A.3.: Frame structure of eOF∂COIN.

A.2.3. Message Synchronization

Since the states of 8th pin on the sources are required to be *OR*ed in all destinations, all the messages in the network should be highly synchronized, as depicted in Figure A.4. Packet loss, latency, or jitters in the network could lead to asynchronous messages, which can induce errors. Every sample of sources is marked and related with each other. The *OR* operation is not executed if the 8th pins of other sources are out of date.

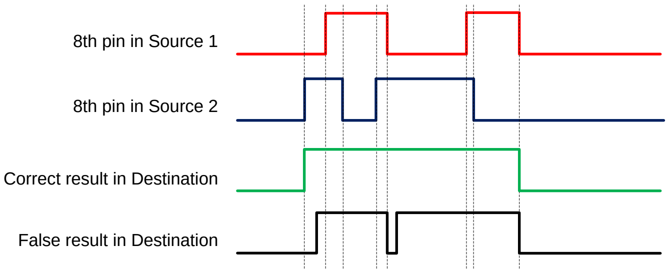


Figure A.4.: *OR*ed messages are required to be synchronized in the network.

A.2.4. Hopping More in Frequency

The hopping frequency sequence in the original OF∂COIN protocol is only of length three. Although three channels are well selected so that they can hardly be interfered at one same moment, the performance is still affected when the noise reaches the strongest level, according to our experience from the last year. We intend to extend the sequence appropriately to ensure the reliability during strong interference. The timing is similar to the one used in the original protocol [MTH+17].

A.2.5. Scalability

As mentioned in [NPPS+15] and [WHM+13], concurrent transmission in large-scale and dense networks induces failed reception. In order to avoid false-positive cases as they have occurred last year, we adopt additional methods, such as transmitting power control to limit the density of concurrent transmission in the network.

A.3. Result

In the EWSN 2018 Dependability Competition[1], at the end, we achieved the third place[2] among all teams from academia and industry.

[1] `https://ewsn2018.networks.imdea.org/competition-program.html`

[2] `https://iti-testbed.tugraz.at/blog/page/11/ewsn-18-dependability-competition-final-results/`

B Competition: Using DeCoT+ to Collect Data under Interference

Toward the scenario of data collection in EWSN 2019 Dependability Competition, we base our design on the efficient concurrent transmission. Moreover, there are two critical mechanisms to guarantee the dependability of the protocol, i.e., the channel-hopping and the network initiation. The Dependable Concurrent Transmission-based protocol (DeCoT) [MZL^{+}18] performs effectively in the past competitions where an event was represented as a short payload. Variable lengths of payloads and dynamic traffic loads are new challenges appearing in the EWSN 2019 Dependability Competition. A Consistency Strategy and a Network Coding (ConNec) are functional to overcome the challenges. Therefore, we propose *DeCoT+* [MZL^{+}19], DeCoT with ConNeC, that ought to work effectively in a dynamic (heavy or light) traffic loads under interference. Besides, the network coding strategy could improve the reliability of the network, where nodes communicate with long packets, e.g., 64 bytes.

B.1. Introduction

In 2017, we proposed OF∂COIN [MTH^{+}17] based on the concurrent transmission (CT) to propagate simple events from one source to one destination dependably under interference. In 2018, an enhanced OF∂COIN (eOF∂COIN) [MZT^{+}18], supporting many-to-all communications, was proposed to monitor multiple concurrent events under adverse conditions. Both protocols achieved high reliability under interference. Specifically, Scan-and-Lock mechanism, a continuous transmission with channel hopping mechanism proposed in OF∂COIN [MTH^{+}17] and eOF∂COIN [MZT^{+}18], maintains usable links under interference. By using Force-Initiated mechanism, not only the host but also the synchronization agents are able to initiate the network, which is quite different from the mechanisms in most current CT-based protocols. It was applied in eOF∂COIN [MZT+18] to decentralize the network, thereby improving reliability. We name the CT-based protocol with the Scan-and-Lock mechanism and the Force-Initiated mechanism as DeCoT [MZL+18]. Each event in these scenarios of previous competitions can be described with ONE bit. That is to say, the payload is so tiny that we can repeat the payload continuously to guarantee a high reliability. As presented in [MZT+18], putting all the events from different sources into one packet does not drastically lengthen the payload. However, the scenario of this year is more challenging. An event could be several bytes rather than one bit. That means simple repetitions of an event in the network is not feasible and energy-efficient since each event can not be represented as ONE bit any more. The traffic would be more dynamic, i.e., the period between two events would be either 1 s or 30 s. To this end, we propose DeCoT+, which combines DeCoT with a Consistency Strategy and a Network Coding (ConNeC).

B.2. DeCoT+

DeCoT is based on CT and supports many-to-all communications. It exploits Scan-and-Lock mechanism and Force-Initiated mechanism. On the fundamental of DeCoT, DeCoT+ combines the consistency strategy with the network coding.

B.2.1. Consistency Strategy

Considering such a scenario with eight sources, each source has 4-byte packets to transmit. That is to say, a direct concatenation of these packets requires at least a payload of 32 bytes, if there is no data compression used. Even worse, some source nodes probably have 64-byte packets to transmit at the same time. It is impossible to put all the packets into one payload due to the limitation of the maximum length of the payload in IEEE 802.15.4 standard. Therefore, we need a consistency strategy that all nodes in the network agree in advance and abide by. The goal of the consistency strategy is to let the information, which is unknown to the sink, to be flooded to the sink as soon as possible. In a CT period, both node A and node B have packets to send at the same moment. According to the consistency strategy, packets from node A always have a higher priority. Therefore, the payload would be filled with the packet from node A. Eventually, in this period the packet from node A is forwarded by relays and received by the sink. The packet from node B would be scheduled to be forwarded in the following period. However, the specific consistency strategy can be optimized according to a concrete application scenario and benchmarks of the protocol.

B.2.2. Network Coding

Assuming that we put a 64-byte payload in one packet of IEEE 802.15.4, i.e., to send a long packet directly rather than to split it into multiple small packets, all the nodes need to forward this long packet in one single period. That means, the period would have to be long. The sink needs to wait for a relative longer period of time, e.g., another period, if the long payload is not received successfully in the current period. Then, the latency would increase in this case. Consequently, we decide to divide a long packet into several small blocks to deliver. Some approaches are based on a handshaking mechanism, e.g., Crystal [IMPR16, ITMP18], work well when the handshaking packet can be received, such as an acknowledgment packet. However, this mechanism is not dependable enough under harsh interference. Therefore, relying on an ACK might not be reliable under intensive interference. Compared to the handshaking mechanism, the intra-session network coding is more reliable, because no handshaking packet is required at all. In summary, we divide a long packet into several small blocks to deliver. Then we apply an intra-session network coding, e.g., LT Codes [Lub02], to those blocks. The sink can recover the long packet after a certain amount of coded blocks have been received.

B.2.3. Many-to-all Communication

Data collection is a many-to-one communication scenario. This does not mean that the source in the network does not need to communicate with other sources. On the contrary, one source is required to give away the opportunities to others if it has nothing to send. One source can repeat a message for a number of times to guarantee the reliability when others have nothing new to send. Thus, in this competition scenario (a many-to-one scenario), we can not avoid many-to-all and many-to-many communications. As mentioned above, DeCoT+ does not rely on ACK to ensure the high reliability. However, the ACK from the sink to the others helps to optimize the allocation of the network resources. That is, to make the allocation reasonably, DeCoT+ disseminates the ACKs from the sink representing from which source the message has been received.

B.2.4. Node Failure

A dependable network should be able to recover from any failure states. If a network is partitioned unexpectedly by interference, the traditional centralized CT protocols such as Glossy [FZTS11], LWB [FZMT12], Crystal [IMPR16, ITMP18] and Chaos [LFZ13a] can not even complete the initialization phase [MZL^{+}18]. Generally, the host in these protocols is the sink in the scenario of data collection. Packets from the sink in these protocols can not reach all the nodes at all. That means these nodes consume energy without any contribution until they are initialized, i.e., synchronized with the host. Relay nodes, in the assumed scenario, may suffer a power failure at any time and reboot after a random period of time. To let the rebooted nodes be initialized as fast as possible, the Force-Initiated mechanism is used.

B.3. Result

We conducted extensive experiments in the first topology (layout 1). In all the scenarios — even when the interference level is the most intensive — we can achieve a reliability of more than 95%, with the payload of 64 bytes and the message generation period of five seconds. A main lesson was learned in the preparation phase. A high reliability can be achieved if dividing a long packet into several short blocks in the harsh interference (level 3). However, the latency increases especially when the number of source nodes increases. Therefore, the reliability and the latency needs to be balanced when the inter-network coding is applied in our protocol. Finally, in the EWSN 2019 Dependability

Competition[1], we won the first place[2] among all teams from both academia and industry.

[1] http://ewsn2019.thss.tsinghua.edu.cn/competition-scenario.html

[2] https://iti-testbed.tugraz.at/blog/page/21/ewsn-19-dependability-competition-final-results/

List of Figures

List of Tables

List of Algorithms

List of Acronyms

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
ACK	Acknowledgment
CCA	Clear Channel Assessment
CoAP	Constrained Application Protocol
CPS	Cyber-Physical Systems
CRC	Cyclic Redundancy Check
CS	Carrier Sense
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CT	Concurrent Transmission
CTI	Cross-Technology Interference
CTP	Collection Tree Protocol
DAC	Digital-to-Analog Converter
DCO	Digitally Controlled Oscillator
DMA	Direct Memory Access
ECA	Event Condition Action
ED	Energy Detection
EDC	Expected Duty Cycle
ETX	Expected Transmission Count
EWSN	international conference on Embedded Wireless Systems and Networks
Exp3	Exponential-weight algorithm for Exploration and Exploitation
FIFO	First In First Out
FIFOP	First In First Out (Positive)
GDI	Great Duck Island
I/O	Input/Output
IC	Integrated Circuit
IoT	Internet of Things
IP	Internet Protocol
IPv6	Internet Protocol version 6
ISM	Industrial, Scientific and Medical

ITU	International Telecommunication Union
LiM	Less is More
LP	Lifetime Planning
LPL	Low-Power Listening
LPP	Low-Power Probing
LWB	Low-power Wireless Bus
MAC	Medium Access Control
MAPE	Monitor Analyze Plan Execute
MDP	Markov Decision Process
MEMS	Micro-Electro-Mechanical System
MIM	Message In Message
MOR	Multichannel Opportunistic Routing
MSN	Mobile Sensor Node
ORPL	Opportunistic RPL
ORW	Opportunistic Routing for Wireless sensor networks
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PA	Power Amplifier
PDR	Packet Delivery Ratio
PER	Packet Error Rate
PiP	Packet-in-Packet
QoS	Quality of Service
RAM	Random-Access Memory
RF	Radio Frequency
RFID	Radio Frequency Identification
ROM	Read-Only Memory
RPL	IPv6 Routing Protocol for Low-Power and Lossy Networks
RSS	Received Signal Strength
RX	Receive
SFD	Start of Frame Delimiter
SINR	Signal to Interference plus Noise Ratio
SOSUS	Sound Surveillance System
SPI	Serial Peripheral Interface
SSN	Static Sensor Node
TDMA	Time Division Multiple Access
TSCH	Time Slotted Channel Hopping
TTR	Time-To-Rendezvous

TX	Transmit
UDP	User Datagram Protocol
WAP	Wireless Access Point
WSN	Wireless Sensor Network

List of Publications

In Journals

[1] **Peilin Zhang**, Mohamed Abdelaal, and Oliver Theel. Quality of service control in proactive wireless sensor networks via lifetime planning. *International Journal of Sensor Networks (IJSNET)*, 26(4), 2018.

[2] **Peilin Zhang**, Alex Yuan Gao, and Oliver Theel. Bandit learning with concurrent transmissions for energy-efficient flooding in sensor networks. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, 18(13), 2018.

[3] Xiaoyuan Ma, **Peilin Zhang**, Oliver Theel, and Jianming Wei. Gathering data with Packet-in-Packet in wireless sensor networks. *Computer Networks*, 170, 2020.

[4] Xiaoyuan Ma, **Peilin Zhang**, Xin Li, Weisheng Tang, Jianming Wei, and Oliver Theel. DeCoT: A dependable concurrent transmission-based protocol for wireless sensor networks. *IEEE Access*, 6, 2018.

[5] Jenny Röbesaat, **Peilin Zhang**, Mohamed Abdelaal, and Oliver Theel. An improved BLE indoor localization with kalman-based fusion: An experimental study. *Sensors*, 17(5), 2017.

[6] Dennis Lisiecki, **Peilin Zhang**, and Oliver Theel. CONE: A Connected Dominating Set-Based Flooding Protocol for Wireless Sensor Networks. *Sensors*, 19(10), 2019.

[7] Mohamed Abdelaal, Oliver Theel, Christian Kuka, **Peilin Zhang**, Yang Gao, Vasilisa Bashlovkina, Daniela Nicklas, and Martin Fränzle. Improving energy efficiency in QoS-constrained wireless sensor networks. *International Journal of Distributed Sensor Networks*, 12(5), 2016.

In Conference Proceedings

[8] **Peilin Zhang**, Xiaoyuan Ma, Oliver Theel, and Jianming Wei. Packet-in-Packet: Concatenation with concurrent transmission for data collection in low-power wireless sensor networks. In *Proceedings of the 24th IEEE International Conference on Parallel and Distributed Systems (ICPADS '18)*, Sentosa, Singapore, Dec. 2018.

[9] **Peilin Zhang**, Xiaoyuan Ma, Oliver Theel, and Jianming Wei. Concurrent transmission-based packet concatenation in wireless sensor networks. In *Proceedings of the 43nd IEEE Conference on Local Computer Networks (LCN '18)*, Chicago, USA, Oct. 2018.

[10] **Peilin Zhang**, Alex Yuan Gao, and Oliver Theel. Less is More: Learning more with concurrent transmissions for energy-efficient flooding. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services (MobiQuitous '17)*, Nov. 2017.

[11] **Peilin Zhang**, Olaf Landsiedel, and Oliver Theel. MOR: Multichannel opportunistic routing for wireless sensor networks. In *Proceedings of the 14th International Conference on Embedded Wireless Systems and Networks (EWSN '17)*, pages 36–47, Uppsala, Sweden, Feb. 2017.

[12] Xiaoyuan Ma, **Peilin Zhang**, Ye Liu, Carlo Alberto Boano, Hyung-sin Kim, Jianming Wei, and Jun Huang. Harmony: Saving concurrent transmissions from harsh RF interference. In *Proceedings of the 38th IEEE International Conference on Computer Communications (INFOCOM '20)*, Apr. 2020.

[13] Xiaoyuan Ma, **Peilin Zhang**, Ye Liu, Xin Li, Weisheng Tang, Pei Tian, Jianming Wei, Lei Shu, and Oliver Theel. Competition: Using DeCoT+ to collect data under interference. In *Proceedings of the 16th International Conference on Embedded Wireless Systems and Networks (EWSN '19)*, pages 290–291, 2019. *1st place.*

[14] Xiaoyuan Ma, **Peilin Zhang**, Weisheng Tang, Xin Li, Wangji He, Fuping Zhang, Jianming Wei, and Oliver Theel. Competition: Using enhanced OF∂COIN to monitor multiple concurrent events under adverse conditions. In *Proceedings of the 15th International Conference on Embedded Wireless Systems and Networks (EWSN '18)*, pages 211–212, 2018. *3rd place.*

[15] Mohamed Abdelaal, **Peilin Zhang**, and Oliver Theel. QoS improvement with lifetime planning in wireless sensor networks. In *Proceedings of the 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN '15)*, pages 8–17, Shenzhen, China, Dec. 2015.

[16] Xin Li, Xiaoyuan Ma, **Peilin Zhang**, Pei Tian, and Jianming Wei. Escape or exploit?: A noise-modulation-based communication under harsh interference. In *Proceedings of the 7th International Workshop on Real-World Embedded Wireless Systems and Networks (RealWSN@SenSys '18)*, pages 31–36, Shenzhen, China, Nov. 2018.

[17] Awais Usman, **Peilin Zhang**, and Oliver Theel. A highly available replicated service registry for service discovery in a highly dynamic deployment infrastructure. In *Proceedings of the 15th IEEE International Conference on Services Computing (SCC '18)*, pages 265-268, San Francisco, CA, USA, Jul. 2018.

[18] Awais Usman, **Peilin Zhang**, and Oliver Theel. A component-based highly available data replication strategy exploiting operation types and hybrid communication mechanisms. In *Proceedings of the 14th IEEE International Conference on Services Computing (SCC '17)*, pages 495–498, Hawaii, USA, Jun. 2017.

[19] Awais Usman, **Peilin Zhang**, and Oliver Theel. An efficient and updatable item-to-item frequency matrix for frequent itemset generation. In *Proceedings of the 2nd*

International Conference on Internet of Things and Cloud Computing (ICC '17), pages 85:1–85:6, New York, NY, USA, 2017.

Bibliography

[AB15] Ericsson AB. Ericsson mobility report: On the pulse of the networked society. *Ericsson, Sweden, Technical Report EAB-14*, 61078, 2015.

[Abd16] Mohamed Abdelaal. *Enabling Energy-Efficient Wireless Sensing with Improved Service Quality.* PhD thesis, University of Oldenburg, Germany, 2016.

[ABK$^+$09] Mário Alves, Nouha Baccour, Anis Koubaa, Ricardo Severino, Gianluca Dini, Nuno Pereira, Rui Sá, Ida Savino, and Paulo Gandra de Sousa. Quality-of-service in wireless sensor networks: State-of-the-art and future directions. Technical report, CISTER-Research Centre in Realtime and Embedded Computing Systems, 2009.

[ACBFS95] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th IEEE Annual Foundations of Computer Science*, pages 322–331, Oct 1995.

[ACBFS02] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[ALNT14] Mohammad Abu Alsheikh, Shaowei Lin, Dusit Niyato, and Hwee-Pink Tan. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, 16(4):1996–2018, 2014.

[ANDIV14] Beshr Al Nahas, Simon Duquennoy, Venkatraman Iyer, and Thiemo Voigt. Low-power listening goes multi-channel. In *Proceedings of the 2014 IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS '14, pages 2–9, Washington, DC, USA, 2014. IEEE Computer Society.

[ARAI13] Md Akhtaruzzaman Adnan, Mohammd Abdur Razzaque, Ishtiaque Ahmed, and Ismail Fauzi Isnin. Bio-mimic optimization strategies in wireless sensor networks: A survey. *Sensors*, 14(1):299–345, 2013.

[ASB$^+$14] Ivan Dario Paez Anaya, Viliam Simko, Johann Bourcier, Noël Plouzeau, and Jean-Marc Jézéquel. A prediction-driven adaptation approach for self-adaptive sensor networks. In *Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '14, pages 145–154, 2014.

[Ash09] Kevin Ashton. That 'Internet of Things' thing. *RFID journal*, 22(7):97–114, 2009.

[AZT15] Mohamed Abdelaal, Peilin Zhang, and Oliver Theel. QoS improvement with lifetime planning in wireless sensor networks. In *Proceedings of 11th International Conference on Mobile Ad-hoc and Sensor Networks*, MSN '15, pages 8–17, Dec 2015.

[BLS16] Martina Brachmann, Olaf Landsiedel, and Silvia Santini. Concurrent transmissions for communication protocols in the internet of things. In *Proceedings of the 41st IEEE Conference on Local Computer Networks*, LCN '16, pages 406–414, Nov 2016.

[BM04] Sanjit Biswas and Robert Morris. Opportunistic routing in multi-hop wireless networks. *SIGCOMM Computer Communication Review*, 34(1):69–74, January 2004.

[BP11] Kaigui Bian and Jung-Min Park. Asynchronous channel hopping for establishing rendezvous in cognitive radio networks. In *Proceedings of the 2011 IEEE International Conference on Computer Communications*, INFOCOM '11, pages 236–240, April 2011.

[BSL10] Joris Borms, Kris Steenhaut, and Bart Lemmens. Low-overhead dynamic multi-channel MAC for wireless sensor networks. In *Proceedings of the 7th European Conference on Wireless Sensor Networks*, EWSN '10, pages 81–96, Berlin, Heidelberg, 2010. Springer-Verlag.

[BVN$^+$11] Carlo Alberto Boano, Thiemo Voigt, Claro Noda, Kay Römer, and Marco Zúñiga. JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks*, IPSN '11, pages 175–186, April 2011.

[BYAH06] Michael Buettner, Gary V Yee, Eric Anderson, and Richard Han. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 307–320, New York, NY, USA, 2006. ACM.

[CBF98] Nicolo Cesa-Bianchi and Paul Fischer. Finite-time regret bounds for the multiarmed bandit problem. In *Proceedings of the 5th International Conference on Machine Learning*, ICML '98, pages 100–108, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[CCT$^+$13] Doug Carlson, Marcus Chang, Andreas Terzis, Yin Chen, and Omprakash Gnawali. Forwarder selection in multi-transmitter networks. In *Proceedings of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS '13, pages 1–10, May 2013.

[CSE11] Ray Chen, Anh Phan Speer, and Mohamed Eltoweissy. Adaptive fault-tolerant QoS control algorithms for maximizing system lifetime of query-based wireless sensor networks. *IEEE Transactions on Dependable and Secure Computing*, 8(2):161–176, March 2011.

[DANLW15] Simon Duquennoy, Beshr Al Nahas, Olaf Landsiedel, and Thomas Watteyne. Orchestra: Robust mesh networks through autonomously scheduled TSCH. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, pages 337–350, New York, NY, USA, 2015. ACM.

[DCL13] Manjunath Doddavenkatappa, Mun Choon Chan, and Ben Leong. Splash: Fast data dissemination with constructive interference in wireless sensor networks. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, NSDI '13, pages 269–282, Berkeley, CA, USA, 2013. USENIX Association.

[DDHC$^+$10] Prabal Dutta, Stephen Dawson-Haggerty, Yin Chen, Chieh-Jan Mike Liang, and Andreas Terzis. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 1–14, New York, NY, USA, 2010. ACM.

[DEFT11] Adam Dunkels, Joakim Eriksson, Niclas Finne, and Nicolas Tsiftes. Powertrace: network-level power profiling for low-power wireless networks. Technical report, Swedish Institute of Computer Science, 2011.

[DGV04] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki – a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th IEEE Annual International Conference on Local Computer Networks*, LCN '04, pages 455–462, Washington, DC, USA, 2004. IEEE Computer Society.

[DLV13] Simon Duquennoy, Olaf Landsiedel, and Thiemo Voigt. Let the tree bloom: Scalable opportunistic routing with ORPL. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 2:1–2:14, New York, NY, USA, 2013. ACM.

[DLZL15] Wan Du, Jansen Christian Liando, Huanle Zhang, and Mo Li. When pipelines meet fountain: Fast data dissemination in wireless sensor networks. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, SenSys '15, pages 365–378, New York, NY, USA, 2015. ACM.

[DOTH07] Adam Dunkels, Fredrik Österlind, Nicolas Tsiftes, and Zhitao He. Software-based on-line energy estimation for sensor nodes. In *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets)*, EmNets '07, pages 28–32, New York, NY, USA, 2007. ACM.

[dPAP12] Rodolfo de Paz Alberola and Dirk Pesch. Duty cycle learning algorithm (DCLA) for IEEE 802.15.4 beacon-enabled wireless sensor networks. *Ad Hoc Networks*, 10(4):664 – 679, 2012.

[Dun11] Adam Dunkels. The ContikiMAC radio duty cycling protocol. Technical Report 2011:13, Swedish Information and Communications Technology (ICT), 2011.

[För16] Anna Förster. *Introduction to wireless sensor networks.* John Wiley & Sons, 2016.

[Fun06] HART Comunication Fundation. HART field communication protocol specification. *Revision HFC_SPEC12*, 6, 2006.

[FZMT12] Federico Ferrari, Marco Zimmerling, Luca Mottola, and Lothar Thiele. Low-power wireless bus. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 1–14, New York, NY, USA, 2012. ACM.

[FZTS11] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. Efficient network flooding and time synchronization with glossy. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '11, pages 73–84, April 2011.

[GAKS11] Shyamnath Gollakota, Fadel Adib, Dina Katabi, and Srinivasan Seshan. Clearing the RF smog: Making 802.11n robust to cross-technology interference. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, pages 170–181, New York, NY, USA, 2011. ACM.

[GBM$^+$11] Travis Goodspeed, Sergey Bratus, Ricky Melgares, Rebecca Shapiro, and Ryan Speers. Packets in packets: Orson welles' in-band signaling attacks for modern radios. In *Proceedings of the 5th USENIX Workshop on Offensive Technologies*, WOOT '11, pages 54–61, 2011.

[GFJ$^+$09] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 1–14, New York, NY, USA, 2009. ACM.

[GHG$^+$14] Shuo Guo, Liang He, Yu Gu, Bo Jiang, and Tian He. Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. *IEEE Transactions on Computers*, 63(11):2787–2802, Nov 2014.

[Gil16] Alasdair Gilchrist. *Industry 4.0: The Industrial Internet of Things.* Apress, Berkely, CA, USA, 1st edition, 2016.

[GRF$^+$12] Nadia Gamez, Daniel Romero, Lidia Fuentes, Romain Rouvoy, and Laurence Duchien. Constraint-based self-adaptation of wireless sensor networks. In *Proceedings of the 2Nd International Workshop on Adaptive Services for the Future Internet and 6th International Workshop on Web APIs and Service Mashups*, pages 20–27, 2012.

[Gro08] ITU-T Study Group. Definitions of terms related to quality of service, 2008.

[GSC09] David Garlan, Bradley R Schmerl, and Shang-Wen Cheng. Software architecture-based self-adaptation. In *Autonomic Computing and Networking*, pages 31–55. Springer US, 2009.

[HBT$^+$07] Rob Hoes, Twan Basten, Chen-Khong Tham, Marc Geilen, and Henk Corporaal. Analysing QoS trade-offs in wireless sensor networks. In *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, pages 60–69. ACM, 2007.

[HBT$^+$09] Rob Hoes, Twan Basten, Chen-Khong Tham, Marc Geilen, and Henk Corporaal. Quality-of-service trade-off analysis for wireless sensor networks. *Performance Evaluation*, 66(3-5):191–208, 2009.

[HXS$^+$13] Pei Huang, Li Xiao, Soroor Soltani, Matt W Mutka, and Ning Xi. The evolution of MAC protocols in wireless sensor networks: A survey. *IEEE Communications Surveys Tutorials*, 15(1):101–120, First 2013.

[HY08] Cunqing Hua and Tak-Shing Peter Yum. Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks. *IEEE/ACM Transactions on Networking*, 16(4):892–903, Aug 2008.

[IMPR16] Timofei Istomin, Amy L Murphy, Gian Pietro Picco, and Usman Raza. Data prediction + synchronous transmissions = ultra-low power wireless sensor networks. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems*, SenSys '16, pages 83–95. ACM, 2016.

[Ins07] Texas Instruments. CC2420 datasheet. *Reference SWRS041B*, 2007.

[ISA11] IEEE-Standards-Association. IEEE standard for local and metropolitan area networks – Part 15.4: Low-rate wireless personal area networks (LR-WPANs). *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pages 1–314, Sept 2011.

[ITMP18] Timofei Istomin, Matteo Trobinger, Amy Lynn Murphy, and Gian Pietro Picco. Interference-resilient ultra-low power aperiodic data collection. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 84–95. IEEE, April 2018.

[IVHJH11] Ozlem Durmaz Incel, Lodewijk Van Hoesel, Pierre Jansen, and Paul Havinga. MC-LMAC: A multi-channel MAC protocol for wireless sensor networks. *Ad Hoc Networks*, 9(1):73 – 94, 2011.

[IWL11] Venkatraman Iyer, Matthias Woehrle, and Koen Langendoen. Chrysso – a multi-channel approach to mitigate external interference. In *Proceedings of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, SECON '11, pages 449–457. IEEE, 2011.

[JH13] Ahmed Jemal and Riadh Ben Halima. A QoS-driven self-adaptive architecture for wireless sensor networks. In *Proceedings of the IEEE 22nd International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 125–130, June 2013.

[JSS05] David Jea, Arun Somasundara, and Mani Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *Proceedings of the International Conference on Distributed Computing in Sensor Systems*, pages 244–257. Springer, 2005.

[KF14] Yoshiaki Kadono and Naoki Fukuta. LAKUBE: An improved multi-armed bandit algorithm for strongly budget-constrained conditions on collecting large-scale sensor network data. In *Proceedings of the 13th Pacific Rim International Conference on Artificial Intelligence*, pages 1089–1095. Springer International Publishing, 2014.

[KHHW13] Henning Kagermann, Johannes Helbig, Ariane Hellinger, and Wolfgang Wahlster. Recommendations for implementing the strategic initiative industrie 4.0: Securing the future of german manufacturing industry. final report of the industrie 4.0 working group, 2013.

[KSC08] Youngmin Kim, Hyojeong Shin, and Hojung Cha. Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, IPSN '08, pages 53–63, Washington, DC, USA, 2008. IEEE Computer Society.

[KW07] Holger Karl and Andreas Willig. *Protocols and architectures for wireless sensor networks*. John Wiley & Sons, 2007.

[KW16] Michael König and Roger Wattenhofer. Effectively capturing attention using the capture effect. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems*, SenSys '16, pages 70–82, New York, NY, USA, 2016. ACM.

[LBD$^+$05] Benyuan Liu, Peter Brass, Olivier Dousse, Philippe Nain, and Don Towsley. Mobility improves coverage of sensor networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 300–308. ACM, 2005.

[Lee06] Edward A. Lee. Cyber-physical systems - are computing foundations adequate? In *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, volume 2, pages 1–9. Citeseer, 2006.

[Lee08] Edward A. Lee. Cyber physical systems: Design challenges. In *Proceedings of the 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369. IEEE, 2008.

[LF76] Krijn Leentvaar and Jan Flint. The capture effect in FM receivers. *IEEE Transactions on Communications*, 24(5):531–539, May 1976.

[LFZ13a] Olaf Landsiedel, Federico Ferrari, and Marco Zimmerling. Chaos: Versatile and efficient all-to-all data sharing and in-network processing at scale. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 1:1–1:14, New York, NY, USA, 2013. ACM.

[LFZ$^+$13b] Roman Lim, Federico Ferrari, Marco Zimmerling, Christoph Walser, Philipp Sommer, and Jan Beutel. Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, IPSN '13, pages 153–166, New York, NY, USA, 2013.

[LGDJ12] Olaf Landsiedel, Euhanna Ghadimi, Simon Duquennoy, and Mikael Johansson. Low power, low delay: Opportunistic routing meets duty cycling. In *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, IPSN '12, pages 185–196, New York, NY, USA, 2012. ACM.

[LLCL11] Zhiyong Lin, Hai Liu, Xiaowen Chu, and Yiu-Wing Leung. Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks. In *Proceedings of the 2011 IEEE International Conference on Computer Communications*, pages 2444–2452. IEEE, 2011.

[LPT$^+$13] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R Smith. Ambient backscatter: wireless communication out of thin air. *ACM SIGCOMM Computer Communication Review*, 43(4):39–50, 2013.

[Lub02] Michael Luby. LT codes. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 271–280, Nov 2002.

[LW09] Jiakang Lu and Kamin Whitehouse. Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks. In *Proceedings of the 28th Conference on Computer Communications*, INFOCOM '09, pages 2491–2499, April 2009.

[Mac05] David JC MacKay. Fountain codes. *IEE Proceedings-Communications*, 152(6):1062–1068, 2005.

[MB07] Ali Motamedi and Ahmad Bahai. MAC protocol design for spectrum-agile wireless networks: Stochastic control approach. In *Proceedings of the 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 448–451, April 2007.

[MCP$^+$02] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97, 2002.

[MGC16] Mobashir Mohammad, XiangFa Guo, and Mun Choon Chan. Oppcast: Exploiting spatial and channel diversity for robust data collection in urban environments. In *Proceedings of the 15th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '16, pages 1–12, April 2016.

[MGR09] Arslan Munir and Ann Gordon-Ross. An MDP-based application oriented optimal policy for wireless sensor networks. In *Proceedings of the 7th IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis*, pages 183–192, 2009.

[Mit97] Tom Mitchell. *Machine Learning*. McGraw Hill, 1997. ISBN: 0070428077.

[Moy97] John Moy. Ospf version 2. RFC 2178, Internet Engineering Task Force (IETF), 1997.

[MPH07] Mihai Marin-Perianu and Paul Havinga. D-FLER – A distributed fuzzy logic engine for rule-based wireless sensor networks. In *Ubiquitous Computing Systems*, volume 4836, pages 86–101. 2007.

[MTH$^+$17] Xiaoyuan Ma, Weisheng Tang, Wangji He, Fuping Zhang, and Jianming Wei. Competition: Using OF∂COIN under interference. In *Proceedings of the 14th International Conference on Embedded Wireless Systems and Networks*, EWSN '17, pages 266–267, USA, 2017.

[MTX$^+$11] Xufei Mao, Shaojie Tang, Xiahua Xu, Xiang-Yang Li, and Huadong Ma. Energy-efficient opportunistic routing in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(11):1934–1942, Nov 2011.

[MZL$^+$18] Xiaoyuan Ma, Peilin Zhang, Xin Li, Weisheng Tang, Jianming Wei, and Oliver Theel. DeCoT: A dependable concurrent transmission-based protocol for wireless sensor networks. *IEEE Access*, 6:73130–73146, 2018.

[MZL$^+$19] Xiaoyuan Ma, Peilin Zhang, Ye Liu, Xin Li, Weisheng Tang, Pei Tian, Jianming Wei, Lei Shu, and Oliver Theel. Competition: Using DeCoT+ to collect data under interference. In *Proceedings of the 2019 International Conference on Embedded Wireless Systems and Networks*, pages 290–291, USA, 2019. Junction Publishing.

[MZT$^+$18] Xiaoyuan Ma, Peilin Zhang, Weisheng Tang, Xin Li, Wangji He, Fuping Zhang, Jianming Wei, and Oliver Theel. Competition: Using enhanced OF∂COIN to monitor multiple concurrent events under adverse conditions. In *Proceedings of the 15th International Conference on Embedded Wireless Systems and Networks*, pages 211–212, USA, 2018.

[MZTW20] Xiaoyuan Ma, Peilin Zhang, Oliver Theel, and Jianming Wei. Gathering data with Packet-in-Packet in wireless sensor networks. *Computer Networks*, 170:107–124, 2020.

[NPPS$^+$15] Claro Noda, Carlos M Pérez-Penichet, Balint Seeber, Marco Zennaro, Mário Alves, and Adriano Moreira. On the scalability of constructive interference in low-power wireless networks. In *Proceedings of the 12th European Conference on Wireless Sensor Networks*, pages 250–257. Springer, 2015.

[NTCS99] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162. ACM, 1999.

[ÖDE$^+$06] Fredrik Österlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. Cross-level sensor network simulation with Cooja. In *Proceedings of the 31st IEEE Conference on Local Computer Networks*, LCN '06, pages 641–648, Nov 2006.

[PDMFJ13] Pangun Park, Piergiuseppe Di Marco, Carlo Fischione, and Karl Henrik Johansson. Modeling and optimization of the IEEE 802.15.4 protocol for reliable and timely communications. *IEEE Transactions on Parallel and Distributed Systems*, 24(3):550–564, March 2013.

[RCBG10] Bhaskaran Raman, Kameswari Chebrolu, Sagar Bijwe, and Vijay Gabale. PIP: A connection-oriented, multi-hop, multi-channel TDMA-based MAC for high throughput bulk transfer. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 15–28, New York, NY, USA, 2010. ACM.

[RKP$^+$16] Vijay S Rao, M Koppal, R Venkatesha Prasad, Tamma V Prabhakar, Chayan Sarkar, and Ignas Niemegeers. Murphy loves ci: Unfolding and improving constructive interference in WSNs. In *Proceedings of The 35th Annual IEEE International Conference on Computer Communications*, INFOCOM '16, pages 1–9, April 2016.

[RLCL09] Lei Rao, Xue Liu, Jian-Jia Chen, and Wenyu Liu. Joint optimization of system lifetime and network performance for real-time wireless sensor networks. In *Quality of Service in Heterogeneous Networks*, pages 317–333. Springer, 2009.

[RM04] Kay Römer and Friedemann Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61, Dec 2004.

[RN16] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach.* Malaysia; Pearson Education Limited,, 2016.

[Rob85] Herbert Robbins. *Some aspects of the sequential design of experiments*, pages 169–177. Springer New York, New York, NY, 1985.

[RSMQ09] Eric Rozner, Jayesh Seshadri, Yogita Mehta, and Lili Qiu. SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks. *IEEE Transactions on Mobile Computing*, 8(12):1622–1635, Dec 2009.

[SAL$^+$03] John A. Stankovic, Tarek Abdelzaher, Chenyang Lu, Lui Sha, and Jennifer C. Hou. Real-time communication and coordination in embedded sensor networks. *Proceedings of the IEEE*, 91(7):1002–1022, July 2003.

[Sar16] Chayan Sarkar. LWB and FS-LWB implementation for Sky platform using Contiki. *CoRR*, abs/1607.06622, 2016.

[SB98] Richard Stuart Sutton and Andrew G. Barto. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[SBWR17] Markus Schuß, Carlo Alberto Boano, Manuel Weber, and Kay Römer. A competition to push the dependability of low-power wireless protocols to the edge. In *Proceedings of the 14th International Conference on Embedded Wireless Systems and Networks*, pages 54–65, USA, 2017.

[Sch97] Robert R. Schaller. Moore's law: past, present and future. *IEEE Spectrum*, 34(6):52–59, June 1997.

[SCM$^+$08] N. Santhapuri, R. R. Choudhury, J. Manweiler, S. Nelakuduti, S. Sen, and K. Munagala. Message in Message MIM: A case for reordering transmissions in wireless networks. In *Proceedings of the 7th ACM Workshop on Hot Topics in Networks*, HotNets '08, 2008.

[SG09] Ryo Sugihara and Rajesh K. Gupta. Optimizing energy-latency trade-off in sensor networks with controlled mobility. In *Proceedings of the 28th Conference on Computer Communications*, INFOCOM '09, pages 2566–2570, April 2009.

[SGJ08] Yanjun Sun, Omer Gurewitz, and David B. Johnson. RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 1–14, New York, NY, USA, 2008. ACM.

[She14] Z. Shelby. The constrained application protocol (coap). RFC 7252, Internet Engineering Task Force (IETF), 2014.

[ST09] Mazeiar Salehie and Ladan Tahvildari. Self-adaptive software: Landscape and research challenges. *ACM transactions on autonomous and adaptive systems (TAAS)*, 4(2):14:1–14:42, 2009.

[Sut84] Richard Stuart Sutton. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, 1984. AAI8410337.

[SVNSO$^+$11] Marcel Steine, Cuong Viet Ngo, Ramon Serna Oliver, Marc Geilen, Twan Basten, Gerhard Fohler, and Jean-Dominique Decotignie. Proactive reconfiguration of wireless sensor networks. In *Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '11, pages 31–40, 2011.

[SWM07] Hoi-sheung Wilson So, Jean Walrand, and Jeonghoon Mo. McMAC: A parallel rendezvous multi-channel MAC protocol. In *Proceedings of the 2007 IEEE Wireless Communications and Networking Conference*, pages 334–339, Washington, DC, USA, 2007. IEEE Computer Society.

[SYM13] Makoto Suzuki, Yasuta Yamashita, and Hiroyuki Morikawa. Low-power, end-to-end reliable collection using glossy for wireless sensor networks. In *Proceedings of the IEEE 77th Vehicular Technology Conference*, VTC Spring '13, pages 1–5, June 2013.

[Tex06] Texas Instruments. CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF transceiver, 2006. www.ti.com/lit/gpn/cc2420.

[TNCS02] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8(2/3):153–167, March 2002.

[TSGJ11] Lei Tang, Yanjun Sun, Omer Gurewitz, and David B. Johnson. EM-MAC: A dynamic multichannel energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '11, pages 23:1–23:11, New York, NY, USA, 2011. ACM.

[TTRJ12] Long Tran-Thanh, Alex Rogers, and Nicholas R. Jennings. Long-term information collection with energy harvesting wireless sensors: A multi-armed bandit based approach. *Autonomous Agents and Multi-Agent Systems*, 25(2):352–394, 2012.

[VM05] Joannes Vermorel and Mehryar Mohri. *Multi-armed bandit algorithms and empirical evaluation*, pages 437–448. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[VRP12] Berta Carballido Villaverde, Susan Rea, and Dirk Pesch. Inrout – a QoS aware route selection algorithm for industrial wireless sensor networks. *Ad Hoc Networks*, 10(3):458 – 478, 2012.

[Wat89] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards.* PhD thesis, King's College, Cambridge, 1989.

[Wei93] Mark Weiser. Ubiquitous computing. *Computer*, (10):71–72, 1993.

[Whi05] Edward C. Whitman. Sosus: The "secret weapon" of undersea surveillance. *Undersea Warfare*, 7(2):256, 2005.

[WHM$^+$13] Yin Wang, Yuan He, Xufei Mao, Yunhao Liu, and Xiangyang Li. Exploiting constructive interference for scalable flooding in wireless networks. *IEEE/ACM Transactions on Networking*, 21(6):1880–1889, 2013.

[Win12] Tim Winter. RPL: IPv6 routing protocol for low-power and lossy networks. 2012.

[WMP09] Thomas Watteyne, Ankur Mehta, and Kris Pister. Reliability through frequency diversity: Why channel hopping makes sense. In *Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, PE-WASUN '09, pages 116–123, New York, NY, USA, 2009. ACM.

[WPG15] Thomas Watteyne, Maria Palattella, and Luigi Grieco. Using IEEE 802.15.4e time-slotted channel hopping (TSCH) in the Internet of Things (IoT): Problem statement. RFC 7554, IETF, 2015.

[WWJ+05] Kamin Whitehouse, Alec Woo, Fred Jiang, Joseph Polastre, and David Culler. Exploiting the capture effect for collision detection and recovery. In *Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors*, pages 45–52. IEEE, 2005.

[YH15] Dingwen Yuan and Matthias Hollick. Ripple: High-throughput, reliable and energy-efficient network flooding in wireless sensor networks. In *Proceedings of the 16th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks*, WoWMoM '15, pages 1–9, June 2015.

[YRH14] Dingwen Yuan, Michael Riecker, and Matthias Hollick. Making 'Glossy' networks sparkle: Exploiting concurrent transmissions for energy efficient, reliable, ultra-low latency communication in wireless control networks. In *Proceedings of the 11th European Conference on Wireless Sensor Networks*, EWSN '14, pages 133–149, New York, NY, USA, 2014. Springer-Verlag New York, Inc.

[YX10] YoungSang Yun and Ye Xia. Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications. *IEEE Transactions on mobile computing*, 9(9):1308–1318, 2010.

[ZAT18] Peilin Zhang, Mohamed Abdelaal, and Oliver Theel. Quality of service control in proactive wireless sensor networks via lifetime planning. *International Journal of Sensor Networks (IJSNET)*, 26(4), 2018.

[ZFM+12] Marco Zimmerling, Federico Ferrari, Luca Mottola, Thiemo Voigt, and Lothar Thiele. ptunes: Runtime parameter adaptation for low-power MAC protocols. In *Proceedings of the 11th International Conference on Information Processing in Sensor Networks*, IPSN '12, pages 173–184, New York, NY, USA, 2012. ACM.

[ZGT17] Peilin Zhang, Alex Yuan Gao, and Oliver Theel. Less is More: Learning more with concurrent transmissions for energy-efficient flooding. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services (MobiQuitous)*, pages 323–332, New York, NY, USA, 2017. ACM.

[ZGT18] Peilin Zhang, Alex Yuan Gao, and Oliver Theel. Bandit learning with concurrent transmissions for energy-efficient flooding in sensor networks. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, 18(13), 3 2018.

[ZLT17] Peilin Zhang, Olaf Landsiedel, and Oliver Theel. MOR: Multichannel opportunistic routing for wireless sensor networks. In *Proceedings of the 14th International Conference on Embedded Wireless Systems and Networks*, EWSN '17, pages 36–47, USA, 2017. Junction Publishing.

[ZMTW18a] Peilin Zhang, Xiaoyuan Ma, Oliver Theel, and Jianming Wei. Concurrent transmission-based packet concatenation in wireless sensor networks. In *Proceedings of the 43nd IEEE Conference on Local Computer Networks (LCN)*, pages 421–424, Oct 2018.

[ZMTW18b] Peilin Zhang, Xiaoyuan Ma, Oliver Theel, and Jianming Wei. Packet-in-packet: Concatenation with concurrent transmission for data collection in low-power wireless sensor networks. In *Proceedings of the IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 868–877, Dec 2018.

[ZRHK15] Jin Zhang, Andreas Reinhardt, Wen Hu, and Salil S. Kanhere. RFT: Identifying suitable neighbors for concurrent transmissions in point-to-point communications. In *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '15, pages 73–82, New York, NY, USA, 2015. ACM.

[ZSJS16] Jiang Zhu, Yonghui Song, Dingde Jiang, and Houbing Song. Multi-armed bandit channel access scheme with cognitive radio technology in wireless sensor networks for the internet of things. *IEEE Access*, 4:4609–4617, 2016.

[ZWLL12] Yaxiong Zhao, Jie Wu, Feng Li, and Sanglu Lu. On maximizing the lifetime of wireless sensor networks using virtual backbone scheduling. *IEEE Transactions on Parallel and Distributed Systems*, 23(8):1528–1535, Aug 2012.