CARL VON OSSIETZKY universität OLDENBURG

Faculty II – Computing Science, Business Administration, Economics and Law
Department of Computing Science

Enabling Energy-Efficient Wireless Sensing with Improved Service Quality

Thesis to obtain the degree of
Doctor of Engineering
- Dr. Ing. -

presented by

M.Sc. Mohamed Abdelaal

Reviewers:
Prof. Dr.-Ing. Oliver Theel
Prof. Dr. Daniela Nicklas

Public defense on: 04 November 2016

Acknowledgments

First and foremost, I would like to express my deepest gratitude to Prof. Dr.-Ing. Oliver Theel, for his excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research. During my study at the University of Oldenburg, he spent endless amount of time proofreading my research papers and giving me excellent suggestions. His professional support and advises guided me to overcome several difficulties and lead to this dissertation. Many thanks also go to Prof. Dr. Daniela Nicklas, who kindly agreed to act as my co-advisor. In addition, I would like to thank the rest of my thesis committee: Prof. Dr. Oliver Kramer and Dr. Jörg Bremer, for their insightful comments and encouragement, but also for the hard question which motivated me to widen my research from various perspectives.

The members of the *System Software and Distributed Systems* group have contributed immensely to my personal and professional time at Oldenburg University. I am especially grateful to Eike Möhlmann, Oday Jubran, Robert Schadek, Dilshod Rahmatov, Peilin Zhang, and Awais Usman. Other past and present group members that I have had the pleasure to work with are Vasilisa Bashlovkina, Khudoyor Shogurezov, Mack Goodstein, Jenny Röbesaat, Nils Müllner, and Kinga Lipskoch. A special thank also goes to Ms. Petra Oetken and Ms. Ira Wempe for their continuous support. I would like to thank my colleagues in the SCARE research training group – including Björn Engelmann, Nils Erik Flick, Yang Gao, Christoph Peuser, Saifullah Khan, Christian Kuka, Heinrich Ody, Evgeny Erofeev, Peter Mossad, Habibur Rahman, and Hendrik Radke – for many fruitful discussions and colloquiums. Prof. Dr. Ernst-Rüdiger Olderog deserves my thanks for his continuous support since the time I joined the SCARE group.

A special thank my parents and my siblings for their continuous support and encouragement over all these years. Last but not least, I would like to thank my wife, Hasnaa Kandil and my son, Majd Abdelaal. They are always there cheering me up and stood by me through the good and bad times. Without their support and encouragement, I could not have finished this work.

Abstract

Wireless Sensor Networks (WSNs) are spatially distributed autonomous sensor nodes (SNs) over the sensing field to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. Thereupon, measurements are conveyed in a cooperative manner to the application manager via a sink node. In general, WSNs have some features that differentiate them from other wireless networks including 1) deployment in harsh environments and 2) strong restrictions on hardware and software capabilities in terms of processing speed, memory storage, and energy supply. Such sensors usually carry limited, irreplaceable energy resources. Therefore, lifetime adequacy is a significant feature of all WSNs.

The crux behind this work is to break the "downward spiral" between reducing certain quality-of-service (QoS) measures and extending the application's lifetime. In fact, the WSN literature is rich of endeavors in the field of energy efficiency and QoS control. For the sake of identifying research challenges and possible gaps, we conduct a comprehensive survey of the state-of-the-art. The reported methods have been classified into *node-oriented methods*, *data-oriented methods*, and *network-oriented methods*. The impact of each method on the provided service quality is briefly discussed. Furthermore, we report on QoS control methods in WSNs. Based on this survey, it was obvious that several single- and multi-objective optimization methods have been proposed to handle this relationship. In fact, single-objective methods are not a practical solution in most WSN applications in which many QoS parameters are deliberately engaged. The *multi-objective optimization* (MOO) methods provide reasonable solutions but they suffer – in most cases – from poor scalability and lack of flexibility against run-time dynamics.

Alternatively, the thesis introduces an *energy-centric* strategy to manipulate the energy-QoS relationship in the light of the relevant context information. As its name implies, it is a design procedure in which the amount of saved energy – during run-time – is dynamically adjusted according to changes in the application and environmental conditions. These adjustments provide the required energy to enhance the provided service qualities. We propose exploiting design-time knowledge to leverage self-adaptive mechanisms. For example, knowing the application scenario helps in estimating the minimum amount of energy, required to achieve the expected lifetime. Hence, the sensor nodes can freely adjust their energy-saving method provided that they possess the predefined minimum energy level. Due to diversity of WSN applications, we introduce three different methods, based on the energy-centric strategy, namely Fuzzy transform-based precise data compression, reliable virtual sensing, and lifetime planning.

In the first method, the radio energy consumption, as a dominant energy-consumer, is reduced via multi-objective data encoding. The literature has some lossy compressors such as discrete wavelet transform (DWT), discrete cosine transform (DCT), and lightweight temporal compression (LTC) methods. However, those methods suffer from drawbacks, such as being ill-suited for resources-taxed devices and having weak immunity against possible outliers. These limitations motivate toward exploring a recently-developed Fuzzy transform to be used for sensor data compression. Actually, the Fuzzy compressor shows a comparable precision performance with the aforementioned methods. However, further improvement of the recovery precision via adapting the transform is sought. Then, a modified version, referred to as FuzzyCAT, has been proposed. FuzzyCAT

has high compression ratios and precision via detecting the input signal curvature and dynamically modifying the transform's approximating function.

In the second method, virtual sensors (VSs) are proposed as a novel technique for reducing the excessive energy consumed by some sensors (such as GPS and Gas sensors); and simultaneously slashing the event-miss probability. Generally, VSs are orchestrations of HW/SW components whose output describes a certain phenomenon. Such a phenomenon can be directly acquired through adopting an "energy-hungry" sensor. The method has been evaluated through two case studies including object tracking and gas leaks detection. In both cases, the lifetime of the main sensors has been significantly extended. Moreover, reliability of the VSs is improved via adopting ontology-based generated rules for sensor selection (sensing quality and environmental conditions are the selection criteria).

In the third method, a novel concept is developed for the interplay between energy efficiency and QoS requirements. Instead of maximizing the lifetime, we can only meet the application-expected lifetime, and simultaneously improve the provided QoS. A self-adaptive framework has been proposed to respond to the environmental dynamics. Moreover, a hierarchical monitor-analyze-plan-execute (MAPE) architecture has been introduced to form a global control loop. The results show that lifetime planning highly improves the QoS characteristics. This profit came at the expense of reducing the WSN lifetime. But the resultant lifetime, shortened lifetime is still long enough to complete the assigned WSN task.

The following publications were a result of research work conducted during the doctoral study:

[ZAT16] P. Zhang, M. Abdelaal, O. Theel, "Quality of Service Control in Proactive Wireless Sensor Networks via Lifetime Planning," In: *the International Journal of Sensor Networks (IJSNet).* to appear, 2016.

[ATK$^+$16] M. Abdelaal, O. Theel, C. Kuka, P. Zhang, G. Yang, V. Bashlovkina, D. Nicklas, M. Fränzle, "Improving Service Quality of Energy-Efficient Wireless Sensor Networks," In: *the International Journal of Distributed Sensor Networks, special issue on Energy and Spectrum Efficient Wireless Sensor Networks.* Vol. 12, No. 5, May 2016.

[AZT15] M. Abdelaal, P. Zhang, O. Theel, "QoS Improvement with Lifetime Planning in Wireless Sensor Networks," In: *Proceedings of the 11th International Conference on Mobile and Ad-hoc and Sensor Networks (IEEE MSN2015).* pages 8-17, Dec. 2015.

[AMT15] M. Abdelaal, P. Mosaad and O. Theel, "liteDTW: A Lightweight Dynamic Time Warping for Tiny Sensing Devices," In: *Proceedings of the 4th International conference on Wireless Sensors (IEEE ICWiSE).* pages 29-35, August 2015.

[BAT15a] V. Bashlovkina, M. Abdelaal, O. Theel, "FuzzyCAT: A Lightweight Fuzzy Compression Adaptive Transform for Wireless Sensor Networks," In: *Proceedings of the Workshop on Next Generation Green ICT in conjunction with IEEE Conference on Communications (ICC 2015)*. London, UK, pages 10809-10815, June 2015.

[BAT15b] V. Bashlovkina, M. Abdelaal, O. Theel, "FuzzyCAT: A Novel Procedure for Refining the F-transform Based Sensor Data Compression," In: *Proceedings of the 14th International Conference on Information Processing in Sensor Networks (IPSN '15)*. Seattle, WA, USA, pages 340-341, April 2015.

[AKT+15] M. Abdelaal, C. Kuka, O. Theel, and D. Nicklas, "Reliable Virtual Sensing for Wireless Sensor Networks," In: *Proceedings of the 2015 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. Singapore, pages 1-7, April 2015.

[A15a] M. Abdelaal, "Fuzzy Compression Refinement via Curvature Tracking," In: *Proceedings of the Frontiers of Formal Methods (FFM 2015)*. Aachen, Germany, pages 31-35, March 2015.

[A15b] M. Abdelaal, "Distributed Techniques for Energy Conservation in Wireless Sensor Networks," In: *Proceedings of the Doctoral Consortium in conjunction with the 4th International Conference on Sensor Networks (SENSORNETS)*. Angers, France, pages 9-20, February, 2015.

[AT14] M. Abdelaal and O. Theel, "Recent Energy-preservation Endeavors for Long-life Wireless Sensor Networks: Concise Survey," In: *Proceedings of the 11th International Conference on Wireless and Optical Communications Networks (WOCN 2014)*. India, pages 1-7, September 2014. Note: an extension of the article: Power Management in Wireless Sensor Networks: Challenges and Solutions.

[AYF+14] M. Abdelaal, G. Yang, M. Fränzle, O. Theel, "EAVS: Energy Aware Virtual Sensing for Wireless Sensor Networks," In: *Proceedings of the 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. pages 1-6, Singapore, April 2014.

[AT13a] M. Abdelaal, O. Theel, "An Efficient and Adaptive Data Compression Technique for Energy Conservation in Wireless Sensor Networks," In: *Proceedings of the 2013 IEEE Conference on Wireless Sensors (ICWiSe)*. pages 124-129, Malaysia, December 2013.

[AT13b] M. Abdelaal, and O. Theel, "Power Management in Wireless Sensor Networks: Challenges and Solutions," In: *Proceedings of the 2013 International Conference in Central Asia on Internet (ICI 2013)*. pages 139-144, October 2013.

Zusammenfassung

Wireless Sensornetzwerke (WSNs) sind über ein Messfeld räumlich verteilte, autonome Sensorknoten (SNs), die physikalische Parameter wie Temperatur, Schall, Druck, usw. beobachten. Die verteilt ermittelten Messdaten werden kooperativ an einen zentralen Knoten, eine sogenannte Senke, weitergeleitet. In der Regel haben WSNs einige Eigenschaften, die sie von anderen drahtlos-Netzwerken unterscheiden, einschließlich 1) Einsatz in rauen Umgebungen, 2) starke Einschränkungen von Hardware- und Software in Bezug auf die Verarbeitungsgeschwindigkeit, Speicher und Energieversorgung. Solche Sensoren haben in der Regel eine begrenzte, nicht austauschbare Energieversorgung. Deshalb ist Berücksichtigung einer angemessenen Lebenszeit eine wesentliches Merkmal aller WSNs. Der springende Punkt hinter/ Das Kernanliegen dieser Arbeit ist es, die Lebenserwartung von WSNs zu verlängern, insbesondere ohne dabei die anwendungsspezifische Quality-of-Service (QoS) zu verringern. Da in der WSN-Literatur viele Ansätze im Bereich Energieeffizienz und QoS-Steuerung existieren, geben wir einen umfasssenden Überblick über den Stand der Forschung, auch um offene Fragen und ungelöste Probleme aufzuzeigen. Die in der Literatur beschriebenen Verfahren sind in *Knoten-orientierte Methoden*, *datenorientierte Methoden* und *netzwerkorientierte Methoden* eingeteilt. Die Auswirkungen der einzelnen Methoden auf derdie QoS wird kurz diskutiert.

In dieser Dissertation wird alternativ eine Optimierung der *energiebasierten* Strategie vorgestellt, um die Beziehung von Energieverbrauch und QoS unter Berücksichtigung der Kontextinformationen der Anwendung zu beeinflussen. Wie der Name andeutet, handelt es sich um ein Entwurfsverfahren, bei dem Energieeinsparungen zur Laufzeit an die Veränderungen der Umgebung angepasst werden. Die durch diese Anpassungen erzielten Einsparungen werden bewusst verwendet, um die Qualitäten der erbrachten Dienstleistung zu verbessern. Konflikte können jedoch weiterhin zwischen einige QoS-Metriken auftreten. Zum Beispiel führt die Übertragung genauer Messwerte an die Basisstation zu großen Übertragungsverzögerungen aufgrund der zusätzlichen Verarbeitungsschritte in jedem Sensorknoten. Ebenso kann eine erwartete Lebensdauer bei ungezielter Veränderung des Energieverbrauches unterschritten werden. Um solche Konflikte zu lösen, schlagen wir vor, beim Entwurf vorliegendes Wissen einzubeziehen, und in selbstanpassenden Mechanismen zu nutzen. Zum Beispiel hilft es bei der Schätzung der Energie, die notwendig ist, um die zu erwartende Lebensdauer zu erreichen, das Anwendungsszenario zu kennen. Sensorknoten können dann ihren Energieverbrauch frei einstellen, vorausgesetzt, dass sie ihre jeweiligen vordefinierten minimalen Energieniveaus besitzen. Aufgrund der Vielfalt der WSN-Anwendungen stellen wir drei verschiedene, auf auf der energiebasierten Strategie, beruhende Methoden vor, nämlich 1) Fuzzy-basierte präzise Datenkomprimierung, 2) zuverlässige virtuelle Sensorik, und 3) Lebensdauerplanung.

Zunächst konzentrieren wir uns auf die Verringerung des Funkenergieverbrauches durch Datenkodierung. In der Literatur beschriebene Verfahren sind verlustbehaftete Kompressoren wie Wavelet, DCT und LTC-Methoden. Solche Methoden haben viele Nachteilen, daher nutzen wir eine kürzlich entwickelte Fuzzy-Transformation für die Sensordatenkompression. Eigentlich hat der Fuzzy-Kompressor eine vergleichbare Genauigkeit und Leistung wie die vorher genannten Verfahren. Deshalb suchen wir jedoch eine weitere Verbesserung der Recovery-Präzision über die Anpassung der Transformation zu erreichen. Es wird eine modifizierte Version, die wir FuzzyCAT nennen, vorgeschlagen. FuzzyCAT

hat eine hohe Kompressionsrate und Präzision durch die Erfassen der Krümmung des Eingangssignals und das dynamische Modifizieren der Näherungsfunktion.

In der zweiten Kategorie werden virtuelle Sensoren (VSS) als neuartige Technik zur Verringerung des Energieverbrauchs durch energiehungrige Sensoren (wie Z.B. GPS, Gas-Sensoren) und gleichzeitig zur Verringerung der Event-miss-Wahrscheinlichkeit vorgestellt. Im Allgemeinen sind virtuelle Sensoren Kombinationen von HW / SW-Komponenten. Diese beschreiben ein Phänomen, das durch einen „energiehungrigen" Sensor direkt gemessen werden kann. Das Verfahren wird in zwei Fallstudien – Objektverfolgung und Gasleck-Erkennung – ausgewertet. In beiden Fällen hat sich die Lebensdauer der Hauptsensoren erheblich verlängert. Darüber hinaus wurde die Zuverlässigkeit des VSS über die Annahme ontologiebasiert erzeugter Regeln für die Sensorauswahl (Sensor Qualität und Umweltbedingungen sind die Auswahlkriterien) verbessert.

In der dritten Kategorie wurde ein neues Konzept für das Wechselspiel zwischen Energieeffizienz und QoS-Anforderungen entwickelt. Anstelle der Maximierung der Lebensdauer können wir die QoS an die erwartete Einsatzzeit des Sensors anpassen. Es wird selbstanpassender Rahmen vorgeschlagen, um auf die Umweltdynamik zu reagieren. Darüber hinaus bildet eine hierarchische MAPE-Architektur einen globalen Regelkreis. Die Ergebnisse zeigen, dass die Lebenszeitplanung die QoS-Eigenschaften stark verbessert. Dieser Gewinn wurde auf Kosten der Verringerung der WSN-Lebensdauer erzielt. Aber die resultierende, verkürzte Lebensdauer ist noch lang genug, um die zugewiesene WSN Aufgabe auszuführen und abzuschließen.

Contents

List of Figures

List of Tables

List of Acronyms

μOS Micro-Operating System

ARMA Auto-Regressive Moving Average

CCA Clear Channel Assessment
CE Cognitive Engine
CH Cluster Head
COTS Commercial Off-the-Shelf
CR Compression Ratio
CRs Cognitive Radios
CRSN Cognitive Radio Sensor Network
CS Compressive Sensing
CSMA Carrier Sense Multiple Access
CT Cooperative Transmission

DBP Derivative-based Prediction Modelling
DCT Discrete Cosine Transform
DMS Dynamic Modulation Scaling
DPM Dynamic Power Management
DTMC Discrete-Time Markov Chain
DTW Dynamic Time Warping
DVS Dynamic Voltage Scaling
DWT Discrete Wavelet Transform

ECA Event-Condition-Action Rules
EEDC Energy Efficient Data Collection

FFT Fast Fourier Transform
FLC Fuzzy Logic Controller
FTC Fuzzy Transform Compression
FuzzyCAT Fuzzy Compression Adaptive Transform

GA Genetic Algorithm

LEACH Low-Energy Adaptive Clustering Hierarchy
liteDTW Lightweight Dynamic Time Warping
LTC Lightweight Temporal Compression

MAC Medium Access Protocol
MAPE Monitor, Analysis, Plan, Execute
MDP Markov Decision Process

MIMO Multiple Input and Multiple Output
MOO Multi-Objective Optimization

NC Network Coding

OTSN Object Tracking Sensor Networks

PDF Probability Density Function
PDR Packet Delivery Ratio
PEGASIS Power-Efficient Gathering in Sensor Information Systems
PER Packet Error Ratio
PWM Pulse Width Modulation

QF Quality Factor
QoS Quality-of-Service

RMSE Root Mean Square Error
RSS Received Signal Strength

SDR Software-Defined Radio
SINR Signal-to-Interference Noise-Ratio
SISO Single Input Single Output
SN Sensor Node
SNR Signal-to-Noise Ratio
SSN Semantic Sensor Network Ontology

TDMA Time Division Multiple Access

UWB Ultra Wideband

WPT Wireless Power Transfer
WSN Wireless Sensor Network

1. Introduction

1.1. Motivation

On 1st August 2007, a highway bridge in downtown Minneapolis loaded with rush-hour traffic dropped more than 60 feet into the Mississippi River, sending at least 50 vehicles and passengers into the water. In this crisis, 13 people were killed and 145 were injured. The potential causes behind the accident were the heat stress, the weight of construction materials, and the heavy traffic load, as reported by the engineering consultant firm Thornton Tomasetti Inc [Pre2009]. By the same token, forest fires provide ample field to put human life in danger. Table 1.1 summarizes the fire losses in some European countries during the past 25 years [VSXea2009]. These incidents can be sidestepped if right precautions have been taken on time. These precautions normally rely on collecting ample information such as weather forecasting, temperature, and high density smoke zone.

Table 1.1.: Number of victims in forest fires-related incidents in some European Countries [VSXea2009]

Country	**Period**	**Number of victims**
Portugal	1982-2007	110
Spain	1982-2005	186
France	1982-2007	20
Croatia	1980-2007	28
Greece	1980-2007	177

The technological needs of wide-area environmental monitoring, as used in incidents avoidance, precision farming [VGP$^+$2007] or pollution control [GHIGGHPD2007], have recently driven a rapidly growing interest in *Wireless Sensor Networks* (WSNs). A WSN is a set of autonomously operating *Sensor Nodes* (SNs) which are spatially distributed over the area of interest to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. [ASSC2002]. WSNs provide a bridge between the real physical and virtual worlds. They are able to observe the previously unobservable at a fine resolution over large spatio-temporal scales.

Designing a WSN is mostly challenging due to the limited allocated resources, such as memory footprint, energy supply, and node size. However, the most challenging task during WSN design is how to handle the natural trade-off between energy consumption and other quality of service (QoS) metrics, like latency, accuracy, throughput, and reliability. In practice, significance of each QoS metric is application-dependent. Specifically, each WSN application is only concerned in a subset of QoS metrics, referred to as the application-relevant QoS metrics. For instance, object tracking WSN is typically designed to monitor possible intruders in a certain area. In such a scenario, detection reliability and latency is certainly crucial.

The WSN literature is rich of energy-efficiency methods [SMSC2011, ACDFP2009, RBC2014, SLGT2011, YY2015]. Most of these methods save energy in an aggressive manner via deliberately "squeezing" the application-relevant QoS metrics. For instance, Raza et al. [RCM+2012] propose a model-based compression method, referred to as the derivative-based prediction (DBP), to reduce the reporting rate of luminosity readings inside a tunnel. Although their method triples the lifetime, the recovery accuracy is highly affected due to being sensitive to packet loss and channel interference. Similarly, Chu et al. [CS2015] introduce a game-theoretic method to control the amount of transmitted power. However, limiting the transmitted power negatively affects communication latency, interference and connectivity.

To overcome the aforementioned limitation, multi-objective optimization (MOO) methods, like ant colony optimization and genetic algorithms, have been proposed. In MOO methods, there exist multiple optimal solutions. Then, a decision maker selects the most convenient solution, depending on the priorities of the objectives to be achieved [And2000]. Depending upon the preference of the multiple objectives, the optimization problem can be tackled using various techniques. For instance, Jameii et al. [JFD2015] propose a genetic algorithm-based method for maximizing the coverage rate, minimizing the percentage of active sensor nodes, and minimizing the unbalanced energy consumption. Similarly, Liu [Liu2016] introduces an ant colony-based MOO optimization method. His objectives comprise improving energy efficiency and balancing energy consumption locally (i.e. node-level) and globally (i.e. network-level). Despite their ability to handle many application relevant QoS metrics, MOO methods lack flexibility where they mostly do not consider the application context information. In addition, they incorporate complex optimization algorithms. Consequently, the computational overhead is drastically increased along with a cumbersome convergence.

This thesis advances the current state of the art in WSN research through proposing a novel strategy of how to deal with conflicting objectives, referred to as an *energy-centric* strategy. The proposed strategy is a design procedure which can be applied to various WSN application scenarios. The core idea lies in exploiting design-time knowledge and enabling self-adaptation mechanisms which react to environmental dynamics. At the outset, typical energy-efficiency methods are to be designed to conserve energy. Subsequently, more energy is spent at run-time to enhance the provided QoS metrics. To this end, self adaptation mechanisms open the door for reacting to the environmental dynamics. As a proof of concept, three different methods have been proposed, namely FuzzyCAT data compression, reliable virtual sensing, and lifetime planning.

1.2. WSN Background

This section gives a detailed description of the system architecture. Moreover, it discusses the main WSN characteristics and the accompanied challenges. In general, there are four basic components in a sensor network: (1) an assembly of distributed or localized sensors; (2) an interconnecting network (usually, but not always, wireless-based); (3) a central point of information clustering, aka base station; and (4) a set of computing resources at the central point (or beyond) to handle data correlation, event trending, status querying, and data mining. As it can be seen in Figure 1.1, the SNs cooperatively pass their data through the network to a main location called base station. Some deployed nodes may have additional resources (super SNs), or may have additional tasks (routing nodes).

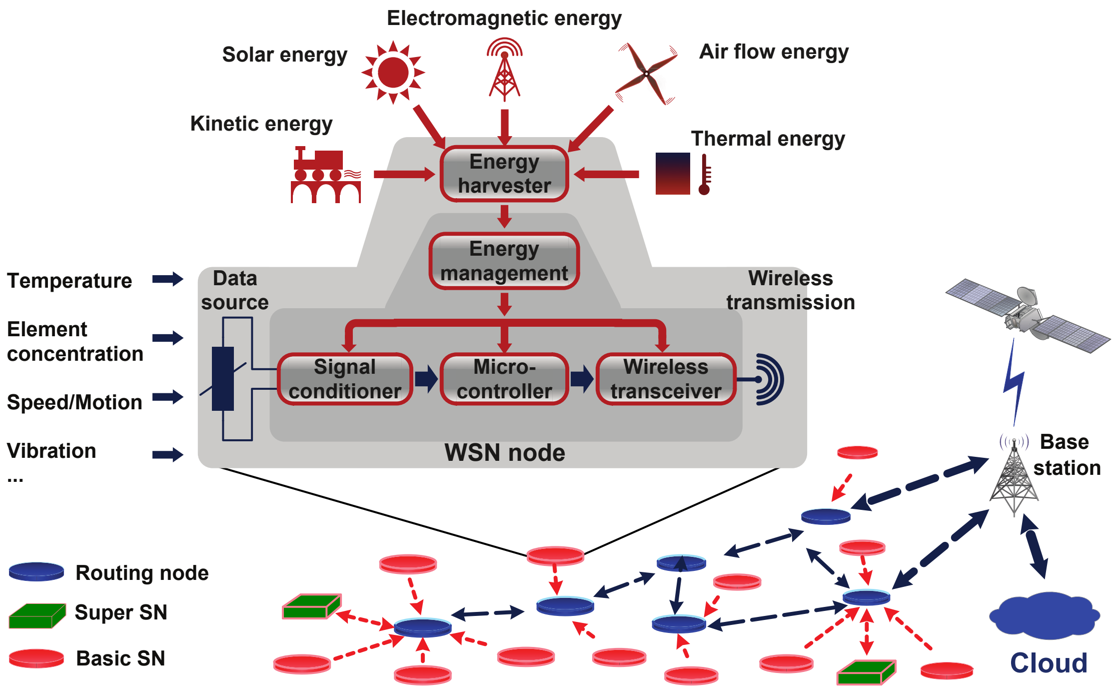

Figure 1.1.: A general WSNs architecture

The deployed SNs typically observe phenomena, perform circumscribed types of signal processing, and broadcast their event detections. In each of those nodes, dedicated software manages the allocation of node resources in a controlled manner. A μ-operating system has to subsume the special necessities of WSN applications and also the resource constraints in WSN hardware platforms. As delineated in Figure 1.1, the SNs are generally composed of four main subsystems: (1) energy supply, (2) sensing module, (3) processing unit, and (4) communication module. Memory types consist of in-chip flash memory, the processor RAM and the external flash memory. In general, these SNs are highly constrained in terms of

- power consumption where batteries might have to last for years,
- operation in harsh environments (e.g. heat, dust, moisture, interference),

- bandwidth (Maximum of 250 KB/s, lower rates are the norm),
- memory (few tens of kilobytes) [PH2007].

Several design challenges present themselves to designers of WSN applications. However, achieving an adequate lifetime subject to extremely limited energy supply thereby is one of the hardest design constraints. Generally, the energy module comprises energy harvesters and an energy control component. The former, on the one hand, energizes a sensor node via exploiting the renewable energy sources like solar energy, thermal energy, and kinetic energy. On the other hand, energy control exploits the available energy in an efficient manner. In practice, SNs usually carry tiny batteries of correspondingly low capacity and replacement in the field tends to be economically infeasible. For instance, Table 1.2 comprises the energy profile of TelosB sensor nodes [Adv2015]. The listed energy profile defines the exact energy consumption per module in the various operational states. Assume adjusting a SN on the idle listening mode where no transmission or reception is carried out. As a result, the SN continuously works for approximately four days, with 2000 mAh AA battery. However, most WSN applications typically last for months or years. Then, unusual methods have to be devised to wisely consume the allocated energy while meeting the expected lifetime. Moreover, efficient power management is a significant feature of all WSNs, which may be complicated in practice by applications requiring integration of different sensor nodes with different capabilities (heterogeneity) or applications utilizing mobile SNs or relay nodes for data aggregation [ASSC2002].

Table 1.2.: Power model for TeloSB sensor nodes

Mode	**Current (μA)**	**Mode**	**Current (mA)**
CPU		**Radio**	
Active (1 MHz, 2.2 V)	300	Rx	18.8
Standby Mode	1.1	Idle listening	
Off Mode (RAM retention)	0.2	Tx (0 dBm)	17.4
LPM0	50	Tx (-1dBm)	16.5
LPM1	50	Tx (-3 dBm)	15.2
LPM2	11	Tx (-5 dBm)	13.9
LPM3	2.5	Tx (-7 dBm)	12.5
LPM4	1.1	Tx (-10 dBm)	11.2
LEDs	0.2	Tx (-15 dBm)	9.9
Sensor board		Tx (-25 dBm)	8.5
Temperature and Humidity Sensors	550	Idle	0.426
Light Sensor	1.3	Sleep	0.02

WSN has many features that differentiate it from other wireless networks. Significance of each feature is application-dependent. Therefore, WSN design lacks the required standardization. For instance, dynamicity covers many situations such as phenomena as node failure, link fluctuations, node attacks, and mobile nodes. The WSN literature has many studies in routing, coverage, scheduling or topology control. Many of them seek to determine solutions where these events occur. However, considering such events in optimization problem models remains a challenge.

Scalability results in changing the network dimensions. In the context of optimizations, scalability increases the computation time. Real-time design is crucial in many applications where it is mandatory to produce an instant action. For example, fire detection is an application that requires real-time sensing and acting to prevent the fire from being uncontrollable. Similarly, health monitoring applications requires sensing and reporting the detected events in a timely-manner. Heterogeneity is another feature which emerges from deploying nodes with different capabilities, such as sensor/actuator-level, higher-level nodes (e.g. gateways, data processing sinks). Additionally, heterogeneity sometimes is necessary to enable the same network infrastructure to support several applications/services. Specifically, heterogeneity brings more challenges in terms of interoperability between different protocols.

1.3. Contributions

The underlying idea behind this work – presented here – is to extend the lifetime expectancy of WSNs. We would like to break the "downward spiral" between reducing a certain quality-of-service (QoS) measure and extending the network's lifetime. Actually, the application-relevant QoS metrics cannot be ignored during energy saving. Moreover, considering single QoS metric for optimization is not practical in most WSN applications. Therefore, MOO methods showed promising results in the context of WSNs. However, most of the existent MOO methods suffer either from lack of flexibility or poor scalability.

The thesis contributes to this arena via introducing a novel "energy-centric" strategy which relaxes the problem from finding optimal values of the QoS metrics to determining context-aware values. The primary idea behind the proposed strategy is to improve both, lifetime and QoS metrics leveraging the context information such as the design-time knowledge, the WSN application dynamics, and the environmental conditions. Considering these information leads to providing additional energy for enhancing the provided service qualities along with meeting the expected WSN lifetime. As a proof of concept, three different methods have been proposed and evaluated. In each method, a set of QoS metrics are compromised with energy consumption. Throughout the thesis, various evaluation methods have been used ranging from practical to empirical methods, including real experiments with the TelosB sensor nodes and Arduino boards, probabilistic model checking, and the Cooja simulator of the Contiki μ-operating system.

In the following, all contributions of this thesis are described in more detail.

a) **Energy-Efficiency Taxonomy.** The WSN technical literature has been surveyed to identify dominating energy consumers in different WSN application scenarios, existent energy-efficiency methods, and QoS control methods. A new taxonomy of energy consumers in WSNs has been constructed to identify interesting research questions (cf. Figure 2.1). Identifying energy consumers in each WSN application, enables us to understand how QoS is manipulated during saving energy. Energy consumers have been divided according into *component level* and *functional level*. The first category, on the one hand, comprises the energy consumed per single sensor node for sensing, memory, communication, processing, and state transition. On the other hand, the second category lists local and global consumers related to networking and QoS provisioning.

Similarly, we construct a comprehensive taxonomy of energy-efficiency methods in WSNs (cf. Figure 3.1). Each reported method has been explained and assigned to a corresponding category. The taxonomy is mainly constructed to study how energy-efficiency methods affect the various QoS metrics. Energy management in WSNs has been divided into energy supply and energy conservation. The former denotes providing each node with additional energy during run time to fully (or partially) energize sensor nodes. The latter is further divided into three subcategories, namely *data-oriented methods*, *node-oriented methods*, and *network-oriented methods*.

In addition, we construct a novel taxonomy for QoS control in WSNs (cf. Figure 2.4). The taxonomy distinguishes itself from other work by listing single- and multi-objective QoS control methods. Specifically, QoS control methods have been classified into, *service-oriented architecture*, *node-oriented methods*, *QoS-aware communication*, and *multi-objective optimization* methods. Throughout this taxonomy, we realize the importance of considering multi-objective problems in the context of WSNs. Moreover, it motives toward developing a simplified procedure to achieve the multi-objective processing. Main parts of this contribution have been published in [AT14], [ZAT16], and [ATK+16].

b) **Energy-Centric Strategy.** The main motivation is to simultaneously improve more than one QoS metric along with reaching the operational lifetime. The energy-centric strategy is a design procedure which targets modifying the energy-efficiency methods to achieve multi-objective optimality. The underlying idea behind the proposed energy-centric strategy is to dynamically adjust the amount of saved energy – during run-time – in the light of the environmental dynamics. These adjustments are deliberately executed to improve the QoS metrics. To avoid possible conflicts, we utilize all available knowledge of the application scenario to leverage self-adaptations. In other words, energy-efficiency methods are normally developed, and then they are refined at run-time to improve the QoS parameters. To validate such a strategy, three different methods have been proposed, including *FuzzyCAT compression*, *reliable virtual sensing*, and *lifetime planning*. Main parts of this contribution have been published in [ATK+16].

c) **FuzzyCAT Data Compression.** We utilize Fuzzy transform in the context of one dimensional sensor data compression for the first time. Moreover, the transform is successfully modified to decrease the recovery errors. Initially, reducing radio energy consumption via data encoding is investigated. Despite the existence of much research work in this field, the proposed method differentiates itself via adapting the algorithm parameters according to characteristics of the signal of interest. In this context, exploiting a recently-developed Fuzzy transform for sensor data compression is proposed. Actually, the Fuzzy compressor shows a comparable precision performance with other lossy compressors such as the wavelet transform, and the model-based methods. However, further improving the recovery precision through adapting the transform is intended. Accordingly, a modified transform, referred to as FuzzyCAT, is proposed. The FuzzyCAT achieves high compression ratios while offering high recovery precision. Specifically, it detects the input signal curvature and dynamically modifies the transform. Subsequently, we propose to shrink the reporting delay via exploiting the interplay between compression and prediction. Base stations are enabled to predict the missing readings, tanking into consideration the spatio/temporal data correlation. To sum up, FuzzyCAT shows an example of improving both of accuracy and latency while offering an adequate operational lifetime. Main parts of this contribution have been published in [BAT15a], [A15a], and [AT13a].

d) **Reliable Virtual Sensing.** The method represents an alternative way to save sensing energy consumption. Although adaptive sampling methods are successful in reducing sensing energy, they suffer from high event-miss probabilities. Virtual sensing is exploited being a novel method for reducing energy consumed by "energy-hungry" sensors (e.g. GPS, Gas sensors) and simultaneously reducing the event-miss probability. Generally, virtual sensors are orchestration of HW/SW components that are able to sense a phenomenon which can be directly sensed by an "energy-hungry" sensor. In addition, sensing reliability is tackled via an Ontology-based decision-making algorithm for selection between main and virtual sensors at run-time. The method is evaluated through two case studies, including object tracking and gas leak detection. In the object tracking WSN, virtual sensors have been implemented using seismic sensors along with a pattern matching algorithm, called the *dynamic-time warping* algorithm. Whereas, gas leak is virtually detected via a light sensor together with a special chemical film, whose color changes with the presence of gases. In both studies, lifetime of the main sensors is significantly extended. Moreover, virtual sensing reliability is improved through adopting the ontology-based generated rules for sensor selection where sensing quality and environmental conditions are the selection criteria. To conclude, reliable virtual sensing illustrates how to improve accuracy and detection reliability while offering an adequate lifetime. Main parts of this contribution have been published in [AKT+15], [AMT15], and [AYF+14].

e) **Lifetime Planning.** In lieu of maximizing the lifetime, we restrict ourselves to only meet the application expected lifetime, but improving the QoS provided. A novel idea based on exploiting the design-time knowledge for planning the entire lifetime of each SN is proposed. A self-adaptive framework – based on the the *event-condition-action* (ECA) rules – is proposed to respond to the environmental dynamics. Moreover, a hierarchical *MAPE* (**M**onitor, **A**nalysis, **P**lan, and **E**xecute) architecture forms the global control loop. A case study of office monitoring WSN is employed. Evaluating the performance of such a method reveals meeting the expected task lifetime while offering significant improvement in communication reliability and latency. Main parts of this contribution have been published in [ZAT16], [AZT15].

Note that for all above listed publications, the idea, the concept and the implementation were developed by the author of this thesis. Moreover, large parts of the papers were written by the author, while Vasilisa Bashlovkina wrote the C codes of FuzzyCAT and LTC. Gao Yang modeled virtual and real sensors in the PRISM model checker. Christian Kuka designed the ontology-based automatic rule generation mechanism. Finally, Peilin Zhang validated the QoS analytical model and participated in the comparative study between lifetime planning, blind adaptation, and lifetime maximization methods. Important to note that this research work is funded by the German Research Foundation through the Research Training Group DFG-GRK 1765: "System Correctness under Adverse Conditions" (SCARE, scare.uni-oldenburg.de).

1.4. Organization of the Dissertation

The dissertation is organized as follows. After this introduction in Chapter 1, Chapter 2 defines the underlying research problem and explains the proposed energy-centric strategy. Chapter 3 reviews the main recent efforts on energy-efficiency and possible impacts on the provided QoS. A Fuzzy transform-based data compression technique is elaborated in Chapter 4. Moreover, the idea behind FuzzyCAT data compression is discussed in the light of improving the recovery precision and the detection latency. Chapter 5 introduces virtual sensing for energy efficiency and demonstrates how the system reliability can be significantly improved. To this end, the chapter explains an ontology-based technique for automatically generating rules to select – during run-time – between main and virtual sensors. Additionally, an object tracking case study is presented. Lifetime planning with a self-adaptive mechanism is demonstrated in more detail in Chapter 6. A conclusion together with an outlook is addressed in Chapter 7. Finally, Appendix A introduces a novel method for speeding up the dynamic-time warping algorithm.

2. System Overview

In this chapter, an overview of the underline problem is presented. As aforementioned, the thesis's main target is to develop a straightforward method for manipulating the energy-QoS contradictions. Energy-efficiency methods are typically designed to tackle excessive energy consumption. Accordingly, we begin by identifying the main energy consumers in WSNs along with an energy model for quantifying the dominating energy consumers in many WSN applications such as data logging, processing, and transmission. To understand impacts of tackling those energy consumers on the service quality, the energy-QoS trade-off is analyzed through mapping the relationship between a set of low-level parameters, such as transmission power and sampling rate, a set of QoS metrics, such as detection latency and communication reliability. Such mapping is described in the following section before formally defining the energy-QoS relationship.

In addition, the chapter investigates the main recent research endeavors in the context of QoS control in WSNs. A special focus is given to how such methods deal with energy along with manifold QoS metrics. Based on this survey, it is obvious that multi-objective methods are well-suited to solve the underlying research problem. However, the existent methods are relatively complex and incur additional computational overhead. As an alternative to the existent endeavors, we explain a novel strategy, referred to as the *energy-centric* strategy. The proposed method is not a mathematical method per se, instead it is a procedure for handling the contradicting parameters. To this end, exploiting both of the design-time knowledge and the scenario dynamics is crucial.

2.1. Problem Definition

In this section, the main energy consumers in WSNs are discussed and the research problem is formulated. As said, efficient power management is a key enabler for practical applicability and economical feasibility of WSNs. It is obviously mandatory to ensure the functionality of the WSN till the completion of the assigned tasks, which may amount to year-long or, in the future, even decade-long data sampling in the field under harsh environmental conditions detrimental to battery capacity. Traditional approaches to extending lifetime include shutting off high-power components, like radio transmitters, for longer phases of inactivity, deep sleep of the whole sensor node between sampling instances, and "smart" forms of sampling to permit shut-offs of reasonable duration amortizing the transient costs of changing power modes. Nevertheless, such energy-efficiency methods negatively affect the application-relevant QoS metrics. To fully recognize the problem significance, a formal definition of lifetime and an energy model are given, in this section, to systematically proceed with the underline research problem.

Symbolically, the underline research problem can be denoted as shown in Equation 2.1. Under the assumption Asm of allocating an amount of energy for each SN, a sensing system Sys (operating in the environment Env) has to satisfy the user's specifications $Spec$. In fact, users typically expect optimal performance, including achieving the maximum lifetime together with providing the highest level of service quality. At first glance, these objectives look contradicting. Therefore, they are relaxed as formally given in Equations 2.2a–2.2c. Equation 2.2a expresses the objective function, denoted by $\tilde{E}_{tot}$, which is the total consumed energy by a WSN consisting of n nodes $s_1, \cdots, s_n$. The term V constitutes a set of energy consumers including the *component level* and the *functional level* (cf. Figure 2.1).

$$Asm \vdash (Env \parallel Sys) \; sat \; Spec \tag{2.1}$$

$$\underset{V}{min} \quad \tilde{E}_{tot}(V) \tag{2.2a}$$

$$L_{task} \leq \tilde{L}(s) < L_{max} \qquad \forall\, s \in [s_1, s_2, \cdots, s_n] \tag{2.2b}$$

$$100\% \geq \tilde{Q}(s) = [q_1, q_2, \cdots, q_m] \geq \psi \tag{2.2c}$$

Specifically, such an objective function is governed by two paramount criteria:

- As denoted in Equation 2.2b, the actual lifetime $\tilde{L}$ of each node s may not be maximized (i.e. L_{max}), but at least has to reach the expected time L_{task}, which is required to complete the assigned task;

- A quality set $\tilde{Q}$, defined in terms of m quality measures, should satisfy the minimum application requirements ψ. Hence, a small space could tolerate the trade-offs as defined in Equation 2.2c.

Through relaxing the optimal objectives, the hard optimization problem can be easily tackled. Since energy-saving is a main objective, we start with analyzing the main energy consumers in WSNs. In fact, sources of energy consumption are application-dependent. Hence, finding general methods – that handle both of energy-saving and QoS control in WSNs – is far-fetched. Hence, we provide below a novel taxonomy of energy consumers in WSNs. Furthermore, we introduce an energy model which formulates the main sources of energy consumption, namely sensing, processing and radio communication.

2.1.1. Energy Model

A WSN's lifetime can be defined as the time span from its deployment to the instant when the network is considered nonfunctional due to the death of a crucial SN or of a critical percentage of SNs. Mathematically, the expected lifetime $\mathbb{E}[\mathcal{L}]$ can be modeled as shown in Equation 2.3.

$$\mathbb{E}[\mathcal{L}] = \frac{E_0 - \mathbb{E}[E_w]}{\sum_{n=1}^{N} (P_s + P_c + P_p) + \sum_{i=1}^{k} (P_{t,i}) + P} \tag{2.3}$$

The numerator comprises the expected energy-waste $\mathbb{E}[E_w]$, which is unused energy remaining in the network when it dies, subtracted from the initial energy reservoir E_0. The denominator is a summation of the sources of energy consumption in a network comprising N sensor nodes. Among them are the energy consumed in sensing (P_s), communication (P_c), processing (P_p), the k different transitions ($P_{t,i}$) between the components operational states, and the continuous energy P needed to sustain the network during its lifetime when not collecting data. Figure 2.1 depicts a taxonomy of all possible energy consumption sources in WSNs. The consumers are classified according to their scope into *component level* and *functional level* sources.

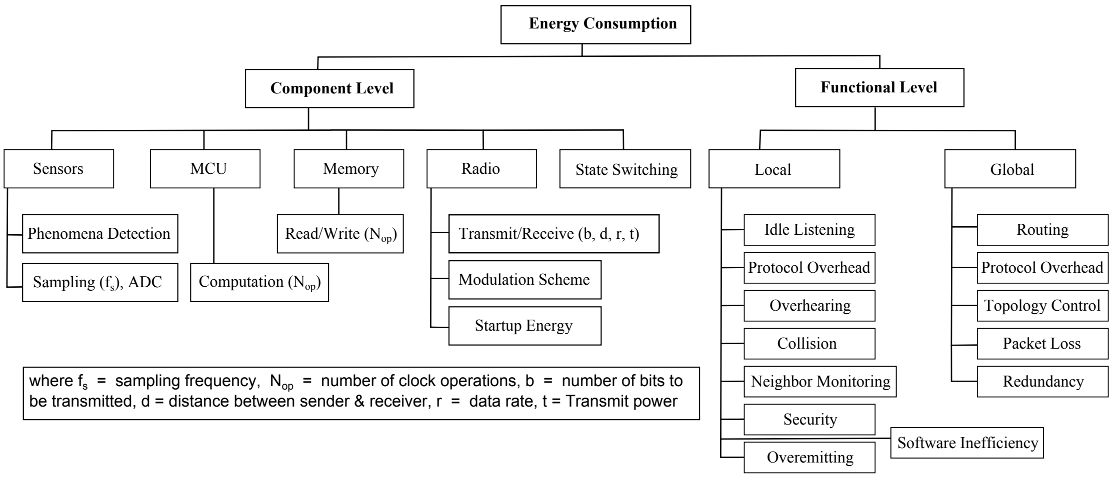

Figure 2.1.: Taxonomy of energy consumption sources in WSNs

The first category comprises the energy consumed per single node s for sensing, memory, communication, processing, and state transition. For example, sensor modules convert a phenomenon of interest into an electrical signal. Equation 2.4 expresses the energy consumed through the activities for detecting b bits per round including sensing activity E_{sens} and data logging E_{logg}. Each term in this equation represents the power in terms of the drawn current, supply voltage, and the activity period $T_i, i \in \{\text{sens, read, write}\}$. While, I_{write} and I_{read} are the current consumption of writing and reading one byte of data from the memory, respectively.

$$E_{sensor}(b) = E_{sens}(b) + E_{logg}(b) = \frac{bV_{sup}}{8}\left(8I_{sup}T_{sens} + I_{write}T_{write} + I_{read}T_{read}\right) \quad (2.4)$$

Recent processors are designed with multiple low power modes, like the MSP430 processor embedded in TeloSB sensor nodes. As depicted in Table 1.2, the MSP430 processor has an active mode and five low power modes through disabling the CPU electronics and various clock signals [Tex2006]. Equation 2.5 describes the energy dissipation of processing b bits per round as the sum of leakage and switching power. Generally, the latter consumes approximately 80% of the allocated energy. The terms V_{sup}, f, C, I_0 and v_t represent the supply voltage, the operational frequency, the total capacitance, the leakage current, and the thermal voltage, respectively [OR2007].

$$E_{processor}(f,b) = b \cdot V_{sup} \cdot f \cdot \left(C \cdot V_{sup} + \frac{I_0}{s} e^{\frac{V_{sup}}{n_0 v_t}}\right) \quad (2.5)$$

Radio communication is typically an “energy-hungry” task. In particular, the energy cost of transmitting one bit is approximately equal to the energy of serving one thousand queries [OR2007]. As it can be seen in Table 1.2, reception consumes more energy than packets transmission due to increasing the receiver’s complexity to detect weak signals. Equation 2.6 models the energy of sending b bits within a distance of d, where E_{elec} is 50 nJ/bit and ϵ_{amp} is 100 $pJ/bit/m^2$. Equation 2.7 computes the energy consumed while listening for data during T_{on} seconds [OR2007]. The overall energy dissipated in radio communication is computed through summing up the power consumed in each state. Another example is state switching between modes of operation. This operations highly increases the power consumption due to leakage currents.

$$E_{Tx}(k,d) = b \cdot \left(E_{elec} + d^2 \cdot \epsilon_{amp}\right) \tag{2.6}$$

$$E_{Rx} = P_{rec} \cdot T_{on} \tag{2.7}$$

The second category of energy consumers in WSNs, as depicted in Figure 2.1, lists local and global consumers related to networking and QoS provisioning. Examples of such a category are as follows.

- *control packet overhead*: Network exchanging of these packets is extremely crucial to guarantee the expected QoS parameters such as reliability and throughput. Unfortunately, WSNs incur excessive overhead due to these packets. For instance, the authors in [RCM$^+$2012] managed to reduce the transmitted readings up to 99%. However, the lifetime is only tripled due to the control packets overhead.
- *idle listening*: Due to the unknown traffic behavior, the SNs have to activate their receivers, waiting for possible incoming packets. Transceivers mostly consume the same energy for reception and for idle listening [ZS2012].
- *overhearing*: Throughout idle listening, a SN could detect false packets which are intentionally sent to other nodes.
- *over-emitting*: Due to synchronization problems, a SN could transmit packets to a neighbor which is not ready for receiving packets at this moment.
- *collisions*: They occur when multiple transmissions are initiated simultaneously. The reasons might be the bad link quality or the inefficient medium access protocol (MAC). However, plenty of MAC protocols endorse collisions in favor of avoiding the energy-costly synchronization.

After identifying the main energy consumers in most WSNs, below we define the energy-QoS relationship. Through a novel QoS model, we illustrate how the energy-QoS relationship is contradicting. Moreover, we explain other conflict-free situations in which the QoS metrics are independent.

2.1.2. Energy-QoS Relationship

Aside from energy conservation, QoS metrics are highly sensitive to aggressive energy saving plans. To illustrate, an analytical model (described in Chapter 6) has been developed to search for possible energy-QoS conflicts. Energy consumption is typically controlled via a set of low-level parameters, namely transmission power P_{tx}, sampling rate r_s, and duty cycle f. Whereas, QoS metrics comprise transmission reliability R, delay D, and lifetime L. Figure 2.2 summarizes the 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 parameters. Lines with a filled circle at the end embody direct proportions. For instance, if the low-level parameter is increased, then the corresponding metric increases as well. Similarly, lines with an open circle indicate inverse proportions.

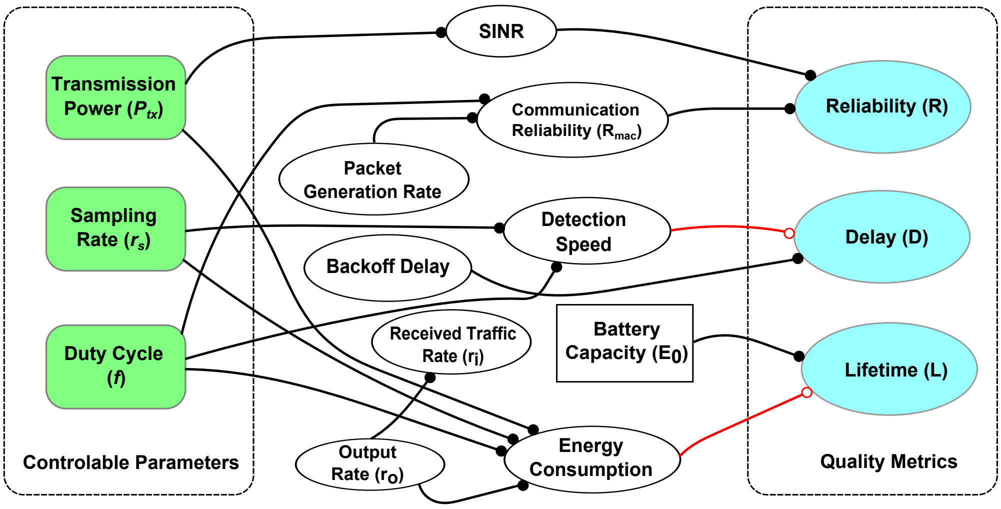

Figure 2.2.: Hierarchical trade-off QoS model

Obviously, 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. For instance, reducing transmission power P_{tx}, leads to reducing both of energy consumption and signal to interference noise ratio $SINR$. As a result, communication reliability is improved at the expense of shrinking the operational lifetime. Accordingly, the main challenge in WSN design lies in considering energy efficiency while offering an acceptable level of other application-relevant QoS metrics. Note that QoS metrics may conflict with each other. For example, maximization of coverage conflicts with the packet error rate, delay, network/battery lifetime and the overall cost of the system. Whereas in some cases, there exist multiple objectives having no direct relationship with each other, rather they are design dependent; e.g., maximization of coverage has no direct relationship with the throughput, energy efficiency and the QoS. Similarly, reducing the detection delay has certainly no impact on accuracy of the reported readings. Hence, considering the nature of application, the sensing scenario, input and output of the problem, is highly beneficial for compromising the energy-QoS relationship.

In the next section, we discuss the main efforts to improve the provided service quality in more detail. The QoS control methods have been classified into single- and multi-objective optimization methods, forming a novel QoS control taxonomy.

2.2. QoS Control in WSNs

The widespread of WSNs technology pushes towards further improvement of the accompanying service quality. We define the terms QoS as a measure of, both the network functionality and the user satisfaction. In other words, the network has to work properly to deliver the required task while offering the user minimal requirements. The relationship among QoS metrics, WSNs, and users are illustrated in Figure 2.3.

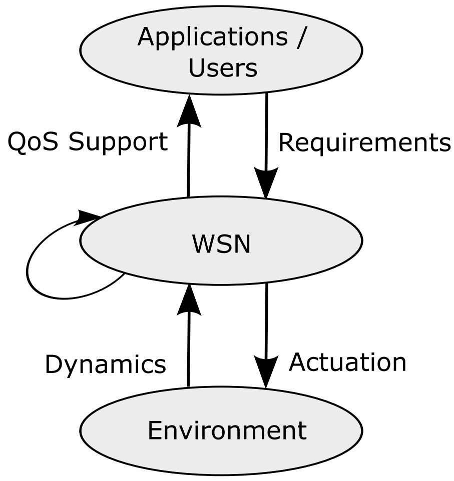

Figure 2.3.: A simple QoS model

Initially, the users convey their minimum requirements to the network. Based on the requirements and the environmental dynamics, the network provides the user with an adequate level of the service quality – which is beyond the user requirements. The rotating arrow (see Figure 2.3) denotes the internal context information such as the residual energy and the transmission range. This additional information is exploited by the network to optimize the natural trade-offs among the conflicted QoS parameters. In some cases, the network also activates actuators to modify the surrounding conditions. This proposed model differentiates itself from the one presented in [CV2004] via considering the expected and the unexpected dynamics imposed by the environment.

The WSN literature reports highly on QoS control as well as trading-off the various QoS metrics. Figure 2.4 shows a novel taxonomy of QoS control methods in WSNs. Specifically, the taxonomy distinguishes itself from the one presented in [ABK^{+}2009] by listing single- and multi-objective QoS control methods. As depicted in Figure 2.4, the methods are classified into, *service-oriented architecture*, *node-oriented methods*, *QoS-aware communication*, and *multi-objective optimization* methods. The former category discusses a recently-developed design framework for WSNs which supports the QoS enhancement. The second category comprises different ideas for proactively modifying the sensor nodes' parameters. The proactive behavior stems from changes in the internal and external context information, such as the environmental conditions and the residual energy. The QoS-aware communication category discusses networking protocols which consider the QoS parameters as an optimization criterion, before the multi-objective methods are investigated. Below, each category is discussed in more detail.

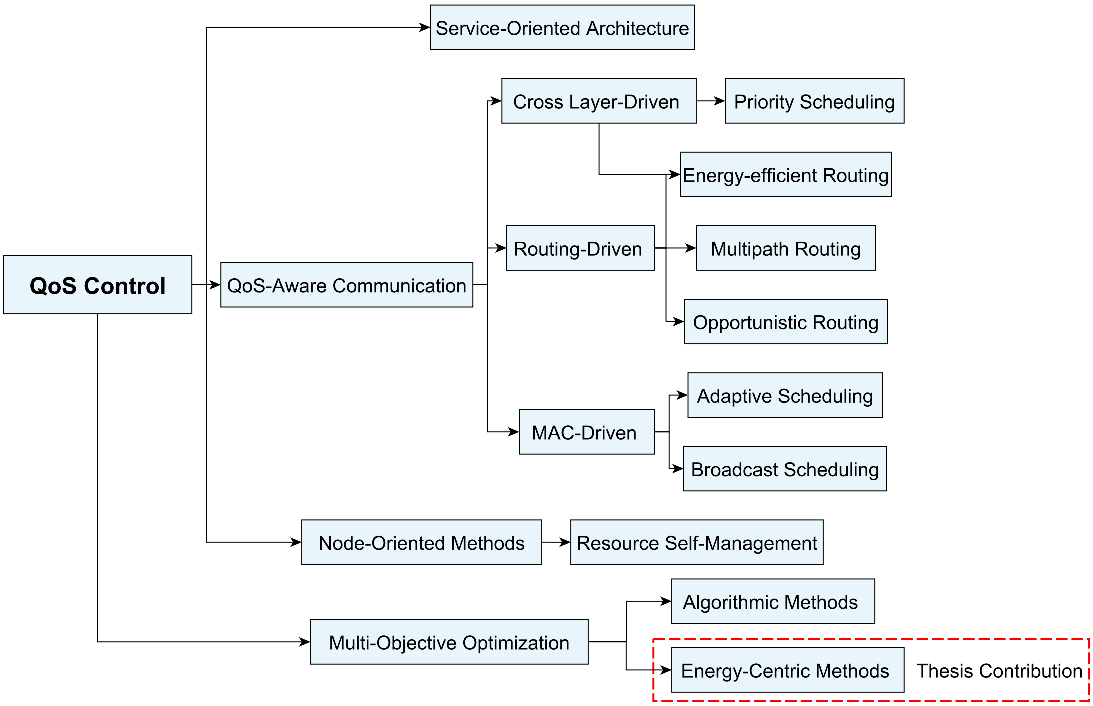

Figure 2.4.: Taxonomy of QoS control in WSNs

2.2.1. Service-Oriented Architecture

The idea behind service-oriented architecture (SoA) is to decompose the system into smaller, distinct modules [Ort2013]. Each of these modules contributes to the overall task. The SoA models enable a rapid and a cost-effective composition of interoperable scalable systems. Accordingly, QoS parameters can broadly be controlled even with WSNs' integration into large-scale cyber-physical systems in which multiple applications run on diverse technologies and platforms.

Ortega [Ort2013] proposes a SoA middleware layer in WSNs to achieve higher level of QoS for WSNs. To this end, Ortega uses sensors in a cluster and dynamic service selection. Specifically, a service or function is achieved by a group of sensors called a sensor cluster. The quality metrics are then captured at the sensor cluster level each time it is executed within the WSN. The QoS improvement results from optimally selecting the sensor cluster. Similarly, Delicato et al. [DPR+2005] propose a middleware layer which monitors both network and application execution states, and accordingly it performs a network adaptation whenever needed.

According to our QoS definition, the QoS metrics can reflect both the WSN performance, such as delay, jitter, throughput; and the non-functional criteria, such as scalability, programmability, and adaptability. Recently, research efforts have considered the non-functional criteria like TinSoA [ALGM2009], OASiS [XS2007] or Service-Oriented middleware [ABRL2010]. Lopez et al. [ALGM2009] introduce TinySoA, a service-oriented architecture to improve the way developers access and control WSNs and integrate them into their applications. The TinySoA consists of a simple service-oriented API which enables programmers to access WSNs via a language of their choices. A more detailed overview of the SoA-based WSNs is given by [MAJ2011] and by [WCLD2008].

2.2.2. Node-Oriented Methods

In this section, we present the recent endeavors for internally – inside each sensor node – controlling the QoS provided. Resources management significantly contributes to QoS provision in the "resources-taxed" WSNs. Our proposed lifetime planning method belongs to this category. Generally, WSNs consist of embedded devices with limited computing, communication and power resources. Increasing the provided QoS level is feasible via extensively consuming the allocated resources. To overcome such a limitation, autonomous *resource managers* have to be exploited to improve the provided QoS through adapting the allocated resources. However, the incurred overhead due to adopting these resource managers has to be minimized.

Several research work has been devoted in this arena [XZST2007, JH2013, MGR2009, SVS+2011, JHH+2013]. For instance, the authors in [MGR2009] design a tuning algorithm based on a *Markov Decision Process* (MDP). This tuning is scheduled to meet the dynamic user requirements and the environmental changes. Hence, the network is flooded with messages going back and forth. Moreover, they consider an energy consumption minimization, which in turn increases the number of reconfigurations. The MDP algorithm is evaluated in MATLAB, thus the overhead of porting it to real sensor nodes is not investigated. Alternatively, a proactive mechanism is proposed in [ASB+2014] 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.

An adaptation technique – similar to our approach – is introduced in [SVS+2011]. It 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 communicate commands among sensor nodes. Despite the simplicity of this strategy, it does not react to unexpected environmental dynamics. Moreover, the adoption of network routing based on flooding completely contradicts with the goal of energy savings. In [JH2013], the authors focus on the self-adaptation mechanism. They adopt 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 their children's reconfigurations.

Alternatively, Xia et al. [XZST2007] propose a Fuzzy logic-based QoS management method, referred to as FLC-QM. The main idea behind FLC-QM is to exploit the feedback control technology to deal with the impact of unpredictable changes in traffic load on the QoS. Specifically, the sampling period of each source sensor node is dynamically adapted in the light of packet miss ratio between the source sensor node and the actuator node.

In the following section, we report on methods dedicated to improve the QoS metrics through optimizing the networking tasks, including routing and medium access. Moreover, we discuss the case of cross-layer design.

2.2.3. QoS-Aware Communication

The cooperative manner of WSNs imposes several requirements to efficiently achieve the sensing task. When to access the communication medium and what is the next transmission hop to deliver the sensed data, are two important questions to be individually investigated for each WSN application. Accordingly, many routing and MAC protocols are developed to optimize networking and data forwarding [SS2012, HN2015, CNC+2014, AA2013b, ZLZ+2014, NKVD2013, YIE2011, AMS2013, SBK2014, AA2013a, TSKJ2013, SWLC2014, CQXB2014]. Several of these protocols consider the QoS parameters as optimization criteria. In this section, we discuss the network-oriented methods in detail. Furthermore, we discuss these methods and also provide a summary of gains and impacts of applying such methods in various WSN applications.

Routing-Driven Methods

Designing an efficient QoS-based routing protocol to meet the requirements of WSN-based applications while preserving high energy efficiency is a challenging task, especially for mission-critical applications. [SS2012] provides a survey which summarizes the state of the existing work on QoS-based routing protocols until the year of *2012*. QoS-based routing protocols such as Sequential Assignment Routing (SAR), Multi-Path and Multi-SPEED Routing (MMSPEED), Multi-Constrained QoS Multi-Path routing (MCMP), Message-Initiated Constrained-Based Routing (MCBR), and Energy efficient and QoS aware multi-path routing protocol (EQSR) are presented along with their solutions to meet QoS requirements. Departing from their survey [SS2012], in this part, we present the state-of-the-art of routing-driven methods for QoS provision and enhancement in WSNs.

Hammoudeh et al. [HN2015] propose a cluster-based route optimization and load-balancing protocol (ROL), which utilizes QoS metrics to meet the applications requirements. The protocol combines the optimizable routing metrics to build energy efficient clusters with a cluster balancing method. By simulations, it shows ROL can prolong the network lifetime, reduce the end-to-end delay, and improve the data delivery ratio.

In [CNC+2014], based on the geographic opportunistic routing (GOR), Cheng et al. propose an efficient QoS-aware GOR protocol for QoS provisioning in WSNs. The protocol builds the forwarding candidate set and prioritizes the forwarding candidates in an efficient manner, which is well-suited for WSNs in terms of energy efficiency, latency, and time complexity. Their simulation results demonstrate that the protocol significantly enhances the energy efficiency, reduces the end-to-end delay, and is characterized by low time complexity.

Alwan et al. [AA2013b] present a heuristic neighbor selection mechanism in WSNs. They combine the geographic routing mechanism with QoS requirements to provide multi-objective routing in WSNs, dealing with different application requirements under different network constraints. The proposed scheme partitions the QoS metrics into two sub-network cost metrics: 1) the node cost metric, where the link condition and the available resources at each node have an effect in selecting the next hop, and 2) the path cost metric, where the QoS metrics are calculated to achieve the requirements while minimizing the overall consumption. Their evaluation results show that the protocol improves the performance in terms of energy consumption, data delivery ratio, and end-to-end delay.

An energy-balanced routing method based on forward-aware factor is proposed by Zhang et al. [ZLZ+2014]. In their work, the next-hop node is selected based on the awareness of the link weight and the forward energy density. Additionally, an autonomous reconstruction mechanism is designed for local topology. Their experimental results show that the method balances the energy consumption, prolongs the functional lifetime and guarantees high QoS of WSNs, which outperforms Low Energy Adaptive Clustering Hierarchy (LEACH) and Energy-Efficient Uneven Clustering (EEUC) protocol.

In [NKVD2013], Nikolidakis et al. propose a Equalized Cluster Head Election Routing Protocol (ECHERP) to maintain energy conservation through balanced clustering. The protocol uses Gaussian elimination algorithm to calculate the combinations of nodes which can be chosen as cluster heads, thus to extend the network lifetime. Simulation results show that their protocol improves the energy efficiency when compared to other well-known protocols in WSNs.

MAC-Driven Methods

Due to the highly resource-constrained nature of sensor nodes, unreliable wireless links, harsh operation environments, and other factors, designing QoS-aware MAC protocols to provide QoS support for WSNs is another challenging issue. In [YIE2011], Yigitel et al. provide a good survey of QoS-aware MAC protocols for WSNs, which focuses on the QoS support at the MAC layer. The MAC layer forms the basis of communication stack and has the ability to tune key QoS-specific parameters, such as duty cycle of the sensor node. The authors survey the QoS mechanisms and classify the state-of-the-art QoS-aware MAC protocols in WSNs, along with their advantages and disadvantages. In this part, to extend their survey after the year of *2013*, we present a few MAC-driven methods for QoS provision in WSNs.

The Intelligent Hybrid MAC (IH-MAC), a low-power and QoS guaranteed medium access control protocol for WSNs, is proposed in [AMS2013]. The main idea behind the IH-MAC is that it utilizes both broadcast and link scheduling depending on the dynamic network loads. The IH-MAC suitably adjusts the transmit power to reduce energy consumption and uses parallel transmission to reduce delay. The analytical and simulation results show the efficiency of the protocol.

Another Adaptive MAC Protocol for Heterogeneous WSNs with fair QoS support (AMPH) is presented in [SBK2014]. To provide QoS support for WSNs, the protocol maintains high channel utilization with a hybrid adaptive behavior and an efficient prioritization scheme. The simulation and analytical results demonstrate that AMPH's hybrid behavior outperforms contention-based protocols such as CSMA/CA and Diff-MAC, in terms of channel utilization, latency, and reliability.

Cross-Layer-Driven Methods

In this part, we mainly discuss the cross-layer-driven methods based on the interactions between MAC layer and network layer. In [AA2013a], a cross-layer scheme is designed to deliver data according to the required end-to-end QoS in WSNs. The scheme utilizes a QoS-aware priority scheduling to ensure the real-time and non-real-time traffic can achieve their desired QoS. Their results of simulations demonstrate the effectiveness of the proposed scheme. [TSKJ2013] proposes an energy-efficient multi-layer MAC (ML-MAC) protocol, which is designed to achieve low duty cycle, prolonged network lifetime and low number of collisions. Simulation results show that the protocol outperforms IEEE 802.15.4 in terms of residual battery capacity, network lifetime, number of packets dropped, total data units received, throughput, average end-to-end delay and average jitter.

A Cross-Layer Adaptive Duty Cycle (CLA-DC) control is proposed in [SWLC2014]. By dynamically adjusting the sleep interval, the desired end-to-end delay guarantees can be achieved. The availability of the single-hop delay mode for end-to-end delay guarantees in CLA-DC is verified by experiments. The experimental results show that CLA-DC outperforms Simple-DC on meeting the end-to-end delay requirements. In [CQXB2014], Chen et al. utilize a cross-layer optimization scheme to jointly consider routing, relay selection, and power allocation strategies for the reliability constraint wireless sensor networks. The results of simulations demonstrate that the proposed cross-layer cooperative strategies considerably prolong the network lifetime. To sum up, Table 2.1 provides a summary of these methods in terms of relevant QoS metrics, applied strategies and their evaluation approaches.

2.2.4. Multi-Objective Optimization

Most WSN applications depend on more than one objective function. To this end, various objective functions are optimized simultaneously and in a systematic manner. This process is referred to as *multi-objective optimization* (MOO). At the outset, an optimizer receives a set of low-level parameters $\{x_1, \ldots, x_n\}$, such as transmit power, reporting frequency, and sensing range, as inputs (cf. Figure 2.5a). The MOO methods mostly utilize complex frameworks to produce multiple optimal plans $\{P_1, \ldots, P_m\}$, and a decision maker typically chooses the best among them, depending on the priorities of the objectives to be achieved. Formally, a MOO problem with k objectives can be expressed as follows.

$$\underbrace{Min/Max}_{x}\ F(x) = [F_1(x), \cdots, F_k(x)]^T \tag{2.8}$$

Subject to

$$g_i(x) \leq 0, \quad i = \{1, 2, \cdots, n\} \tag{2.9}$$

$$h_j(x) = 0, \quad j = \{1, 2, \cdots, m\} \tag{2.10}$$

Table 2.1.: Reviewed network-oriented QoS improvement approaches

Protocol	Layer	Influenced QoS metrics	Strategies	Evaluation methods
ROL [HN2015]	Network	Reliability, latency, lifetime	Route optimization, load balancing	Simulation: Dingo simulator
EQGOR [CNC+2014]	Network	Energy efficiency, latency, time complexity	Opportunistic routing	Simulation: NS-2 simulator
MQoSR [AA2013b]	Network	Reliability, latency, lifetime	Geographic routing	Simulation: C++-based simulator
FAF-EBRM [ZLZ+2014]	Network	Reliability, lifetime	Load balancing	Simulation: TOSSIM, OMNeT++, ATEMU simulator (stated by the authors)
ECHERP [NKVD2013]	Network	Lifetime	Hierarchical routing, Gaussian elimination	Simulation: Java-based simulator
IH-MAC [AMS2013]	MAC	Energy efficiency, latency	Broadcast scheduling, carrier sense multiple access, link scheduling	Simulation: OMNET++
AMPH [SBK2014]	MAC	Reliability, latency, throughput	Adaptive prioritization scheme	Simulation: OMNET++
[AA2013a]	Cross-layer	Energy efficiency, reliability, latency	Multipath routing	Simulation: C++-based simulator
ML-MAC [TSKJ2013]	Cross-layer	Energy efficiency, reliability, latency, throughput	Duty cycle scheduling	Simulation: QualNet 5.2 simulator
CLA-DC [SWLC2014]	Cross-layer	Latency	Sleep scheduling	Experiment: TelosB testbed
[CQXB2014]	Cross-layer	Reliability, lifetime	Cooperative diversity	Simulation: Matlab

As depicted in Equation 2.8, several objective functions $F_k(x)$ are to be maximized or minimized, according to nature of the application. For instance, detection latency has to be minimized in delay-intolerant networks, while data delivery ratio has to be maximized in critical applications, such as smart grids and healthcare monitoring. Equations 2.9–2.9 express n inequality constraints, denoted by $g_i(x)$, and m equality constraints, denoted by $h_i(x)$. An expected solution represents a set of reconfigurations applied to independent variables $X = \{x_1, \cdots, x_s\}$, such as transmission power and sampling rate. Note that an optimal plan can only be determined after solving the above problem, i.e. Equations 2.8–2.10, for s independent variables.

The WSN literature reports on MOO methods [And2000, JFD2015, KH2014, FT2007]. Lei et al. [RLCL2009] 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 the appropriate trade-off. Alternatively, an adaptive fault-tolerant QoS control algorithm is designed in [CSE2011] to meet the QoS requirements in query-based WSNs. They develop a mathematical model in which lifetime of the system is considered as a system parameter. Then, they determine the optimal redundancy level that could satisfy QoS requirements while prolonging the WSN lifetime. Unfortunately, in their application the network dynamics were not fully considered.

Hoes [HBT+2007] proposes a Pareto-algebraic method to determine sets of parameters for sensor nodes that trade-off several application-level QoS metrics. The core idea behind Pareto optimality is to precisely capture all the trade-offs in a multi-dimensional optimization space. Then, a central criterion is employed to compare configurations and discard the ones that cannot be optimized. The main shortcoming of metrics optimization methods is their complexity. Some of them have a random nature while using a flat search space, hence finding all solutions may require an exhaustive search of the whole space. Other methods [HBT+2007, ARAI2013] bypass such huge burden via a selective strategy from the search space. However, following this strategy may lead to forgo the optimal parameters.

Kellner et al. [KH2014] propose multi-objective ant colony optimization (MOACO) algorithms that are capable of considering multiple objectives at the same time. The MOACO algorithms provide a compromise between security and efficient routing. Konstantinos et al. [FT2007] utilize *Genetic Algorithms* (GA) to optimize the self-organizing network and energy management, taking into consideration communication constraints and energy-conservation characteristics. Similarly, Martins et al. [MCW+2011] introduce a hybrid MOO optimization algorithm to solve the coverage and connectivity problem. Moreover, their algorithm enhances the network performance in terms of network lifetime, by joining a multi-objective on-demand algorithm employing GA and a local online algorithm.

Aside from the profits cultivated from employing these MOO algorithms, their computational overhead is questionable. The impact of such additional burden is highly notable, especially with resources-taxed sensor nodes. A difference still exists between theoretical studies and implementation in real WSNs. Many of the available MOO methods are theoretical studies which remain centralized and require heavy computation. In some cases, the optimization methods assume continuous variables sampling, despite the discontinuous nature of the corresponding events such as power transmission and data flow. Alternatively, the thesis proposes an energy-centric strategy to avoid these complex algorithmic MOO optimizations. In the next section, the core idea behind the energy-centric strategy is discussed in more detail.

2.3. Energy-Centric Strategy

As earlier explained, energy-efficiency methods – when are solely considered during WSN design – lead to “squeezing” other QoS metrics, for the sake of conserving energy. Hence, many QoS control methods have been devised to handle this conflict. However, most of these methods have single-objective, such as increasing the operational lifetime while offering precise measurements. Nevertheless, most WSN applications require high levels of many QoS metrics. For instance, Object tracing WSNs should provide the user with accurate detections in a timely-manner along with satisfying the planned lifetime. Therefore, MOO methods have been introduced. However, they drastically suffer from:

- adding a computational overhead due to adopting relatively complex algorithms;
- lacking the real-time flexibility to react to possible dynamics in the application scenario and the environmental dynamics;
- the slow convergence speed – to determine the optimal solution among many other solutions – due to the incurred complexity.

Figure 2.5 shows structure of the proposed energy-centric strategy. At the design time, we collect all information relevant to the WSN application. Subsequently, we engineer an energy saving method with several energy saving levels, i.e. the amount of energy saved can be dynamically adjusted. During the run time, we analyze changes in the environmental and application conditions. Afterward, this knowledge leverages adapting the energy saving level to provide additional energy to enhance the relevant QoS metrics.

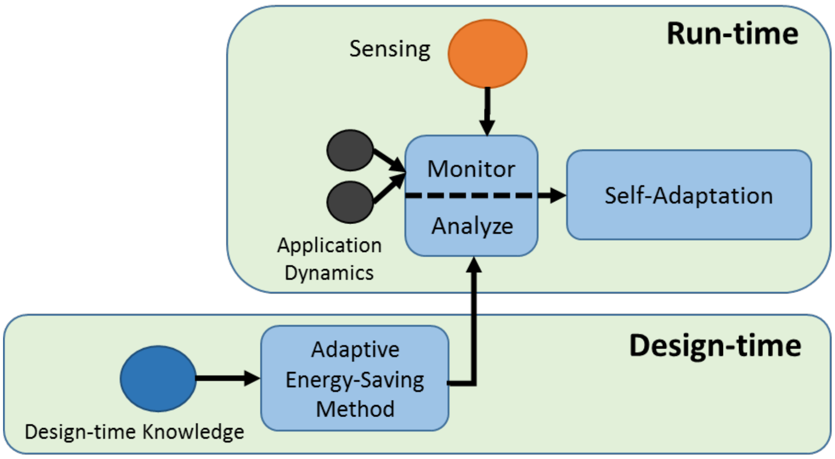

Figure 2.5.: Structure of the proposed energy-centric strategy

Fortunately, most energy-efficiency methods have a high degree of freedom, i.e. they can be adjusted to several levels of energy-saving. For instance, the data reduction strategy seeks to reduce energy-overhead of excessive radio communication. Usually, data reduction utilizes prediction or compression algorithms. Important to realize that data compressors, as an example, can have different degrees of granularity while adopting different compression ratios. Hence, it is straightforward to adjust the energy-saving level, whenever the environment has interesting events.

These adjustments are deliberately executed to improve the provided service qualities. Nevertheless, conflicts may still occur between some QoS metrics. For instance, reporting accurate readings after compression – to the base station – comes at the expense of a long reporting delay, due to additional processing steps at each sensor node. In other WSN applications, an expected operational lifetime may be violated (i.e. $\tilde{L}(s) \leq L_{task}$), due to blindly adapting the energy-saving level. Blindness normally emerges due to lack of knowledge about the application scenario. Hence, sensor nodes have to be designed to be more "intelligent".

To resolve such conflicts, we propose exploiting design-time knowledge to leverage self-adaptation behavior. For example, knowing the application scenario helps in estimating the minimum amount of energy, required to achieve the expected lifetime. Hence, sensor nodes can freely adjust their energy-saving method provided that they possess the predefined minimum energy level. Due to diversity of WSN applications, we introduce three different methods, based on the energy-centric strategy, namely Fuzzy-based precise data compression, reliable virtual sensing, and lifetime planning. Initially, each method is designed to minimize the energy consumption. During run-time, sensor nodes react to the surrounding dynamics via spending more energy for the sake of improving the application-relevant QoS metrics. Note that self-adaptation are designed in a simple way to minimize their computational overhead at run-time.

WSN applications are unfortunately diverse. Each application has its own QoS requirements. Additionally, dominating energy consumers are application-dependent. For instance, gas leak detection utilizes "energy-hungry" gas sensors. Consequently, sensing modules dominate energy consumption in the gas leak application. Whereas, environmental monitoring typically uses energy-efficient sensors, such as temperature and pressure sensors. Below, we provide a taxonomy of WSN applications. We classify the WSN applications according to how sensor readings are reported to the base station. Through this classification, each category comprises WSN applications with similar QoS/energy requirements.

2.3.1. WSN Applications

As aforementioned, it is necessary to categorize the WSN's applications which have common features. As it can be seen in Figure 2.6, WSN applications can be distinguished, according to the data aggregation method, into *time-driven*, *event-driven*, or *hybrid* scenarios.

Time-Driven Applications. In this category, the measured data has to be sent periodically to the gateway for further processing. Thus, data traffic will be extremely dense, especially with short acquisition periods. Moreover, probabilities of collisions, idle listening, and inefficient routing are therefor raised. Obviously, applications – belong to this category – suffer from excessive data reporting, which burdens the nodes' batteries.

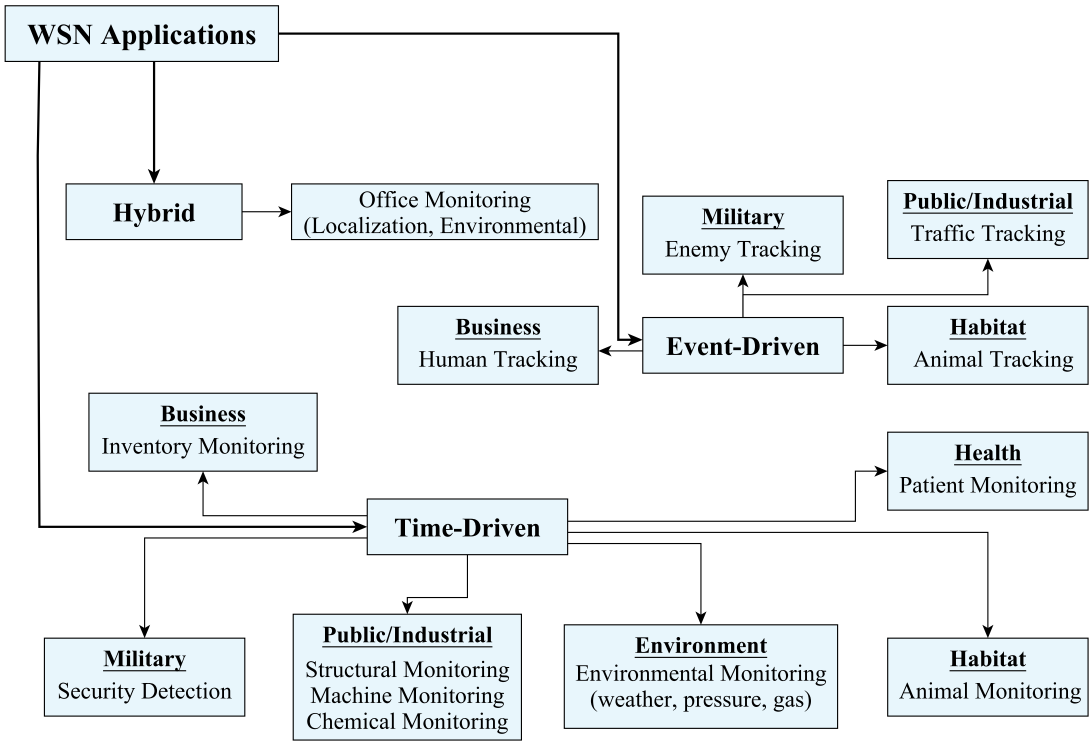

Figure 2.6.: Classification of WSN application scenarios

Event-Driven Application. SNs are typically deployed to detect a prior-known event such as intrusion, predators, or forest fires. Whenever the required event is monitored by a single node or a set of SNs, the event is then reported to the base station or commands are conveyed to the surrounding actuators. As it can be extrapolated, radio communication is not frequent. Therefore, other modules like sensing or processing may dominate the energy consumption due to the long data acquisition.

Hybrid Applications. This category includes heterogeneous application scenarios which collect data in time-driven and event-driven manners. The office monitoring scenario stands as a clear example of this category. In this application, individuals have to be monitored on an event-based manner, whereas indoor environmental parameters have to be periodically reported.

2.3.2. Strategy Validation

Based on the aforementioned categorization of WSN's applications, the thesis revolves around proving feasibility of the proposed energy-centric strategy to manipulate the natural tradeoffs between energy consumption and other QoS parameters. We provide three examples of energy conservation methods that consider QoS improvement. Each of these examples covers one of the aforementioned categories of WSN application scenarios. Specifically, the thesis answers the following three research questions.

a) Can data transfer in a time-driven application set be "energy-cheap" in an accurate and timely manner? Reducing the energy consumption of data transfer requires "shrinking" the measurements' size. However, processing the readings at run-time adds delay which makes it ill-suited for delay-intolerant applications. Moreover, "shrinking" the measured values removes important signal details such as the curvatures or abrupt signal changes. Hence, developing an approach for reducing the radio traffic in an accurate and timely manner is substantial for time-driven WSN's applications.

b) How can "energy-expensive" sensors in the event-driven application set be deactivated without missing the prescribed events? In the set of event-driven applications, data transmission is highly less frequent than that of the previous set. However, other sources may dominate the energy consumption. More attention should be given to decrease such an overly consumption in data acquisition.

c) What does lifetime planning achieve for the tradeoff between energy consumption and other QoS parameters? Answering this question can be achieved by exploiting design-time knowledge and adopting self-adaptation mechanisms. In particular, the design-time knowledge, such as expected lifetime and environmental dynamics, permits energy dissipation in a deterministic manner. For improving QoS parameters, a self-adaptive mechanism is employed to make use of the environmental dynamics.

To sum up, Table 2.2 summarizes the proposed methods which are based on the energy-centric strategy. Three different methods – corresponding to the three WSN applications categories– have been proposed to answer the above research questions. Each method has been designed to consider a trade-off between energy consumption and two application-relevant QoS metrics. For example, lifetime planning targets achieving an expected lifetime together with improving throughput – in terms of packet delivery ratio – and reporting latency. Lifetime planning has been evaluated through a case study of object tracking WSN. Although we consider only three QoS metrics for optimizations, other metrics can be optimized in a similar manner. We only select those QoS metrics, since they exhibit conflicts among each other. Accordingly, advantages of the proposed methods can be clearly demonstrated.

Table 2.2.: Some proposed methods based on the energy-centric strategy

WSN Applications	**Proposed Methods**	**Energy**	**Latency**	**Reliability**	**Accuracy**	**Throughput**
Time-Driven	FuzzyCAT	✓	✓		✓	
Event-driven	Virtual Sensing	✓		✓	✓	
Hybrid	Lifetime Planning	✓	✓			✓

As the first step in implementing the energy-centric strategy is to utilize typical energy-efficiency methods. In the next chapter, we introduce a comprehensive survey of energy-efficiency methods in WSNs. Through this extensive survey, we identify the main research trend in the context of energy-efficiency in WSNs. Furthermore, we assign the contributions of this thesis to corresponding categories of the energy-efficiency taxonomy.

3. Literature Review

Generally speaking, energy conservation is a key issue in the design of systems based on WSNs. As earlier explained in Chapter 1, many energy consumers exist which may quickly deplete the allocated energy budget. However, a WSN has to properly functions, at least, till the completion of the assigned task. Therefore, an outsized and growing body of the WSN technical literature has investigated the energy efficiency problem. Specifically, energy is afforded either through preserving the already allocated energy budget such as batteries or super-capacitors; or through external energy provision; or by utilizing both approaches. Below, we provide a comprehensive energy-efficiency taxonomy. We aim at identifying how each method affects the application-relevant QoS metrics.

3.1. Introduction

In fact, the aforementioned energy consumers are application-dependent. For instance, a WSN is being used in the *ZebraNet* project to observe the behavior of wild animals within a spacious habitat (e.g., wild horses, zebras, and lions) at the Mpala Research Center in Kenya [RM2004]. The observation period is scheduled to last a year or more. The observation area may be as large as hundreds or even thousands of square kilometers. Animals are equipped with sensor nodes. Each node logs readings from its sensors every three minutes. Obviously, radio communication dominates the energy consumption here due to the large-scale network and the short acquisition time. Tracking military vehicles is another kind of WSN applications. A WSN is being used to track the path of military vehicles (e.g., tanks). Tracking results should be reported within given deadlines. Magnetometer sensors are attached to the nodes in order to detect the proximity of tanks. Tracking results are transmitted to an unmanned aerial vehicle. As it can be deduced, sensing and radio communication consumes much more energy than the radio module. This fact emerges from the nearly continuous data logging and processing, whereas reporting is solely done after detecting military vehicles.

Several taxonomies of the WSNs energy conservation exist, however considering the WSN applications behavior and their impact on the various energy consumers has not been clearly discussed [SMSC2011, ACDFP2009, RBC2014, SLGT2011, YY2015]. For example, the authors in [SMSC2011] divided energy conservation methods into *duty cycle*, *data-driven*, and *mobility-based methods*. The former comprises methods for putting the radio transceiver in the low-power sleep mode whenever communication is not required. Unneeded samples and power consumption of the sensing module is targeted in the data-driven category. The latter category deals with mobility utilization for energy-efficient data aggregation. This taxonomy does not classify which methods are general and which are dedicated. Moreover, the aforementioned taxonomies are outdated or do not consider the recent methods such as network coding, compression sensing, etc. It is important to mention that most of these taxonomies did not discuss the application-relevant QoS parameters. Therefore, the impact of applying these energy-efficient methods on the service quality provided is vague.

In this chapter, recent research efforts for saving the allocated energy are highlighted. A unique taxonomy of energy saving techniques dedicated to WSNs is provided. The taxonomy comprises the main recent ideas for solving the energy consumption problem. Moreover, the question of how these methods deal with the various QoS is briefly answered. Therefore, this chapter represents a valuable source for new researchers in this field. As it can be seen in Figure 3.1, energy management in WSNs is divided into *energy supply* and *energy conservation*. The former denotes providing each node with additional energy during run-time to fully (or partially) energize the sensor nodes. The energy provision is typically achieved either through scavenging surrounding energy sources; such as kinetic energy, solar energy, electromagnetic energy, air flow energy, and thermal energy; or through wireless power transfer. Below, the energy saving methods and how they deal with other QoS parameters are elaborated. The taxonomy is divided into three main roots as follows.

- **Data-oriented methods** - this category is divided according the WSN applications into time-driven methods and event-driven methods. On the one hand, the former category comprises methods for reducing radio energy consumption which dominates total consumption in the time-driven scenarios. On the other hand, methods utilized to alleviate the burden of frequent data acquisition or energy-hungry sensors belong to the latter category.

- **Node-oriented methods** - this category encompasses local methods for modulating the hardware, the software components, or both of each node. Indeed, these reported methods are data-agnostic. Hence, they can be utilized in cases of event- and time-driven scenarios.

- **Network-oriented methods** - methods belong to this category mainly targets the energy consumed for maintaining the network functionality. This includes energy-aware routing, sleep/wake-up protocols, and mobility-based methods.

In the sequel, the large taxonomy is divided into smaller blocks. Then, the various roots are explained in more details where examples of recent work are given. Moreover, the impacts on application-relevant QoS are analyzed.

3.2. Energy Supply

Figure 3.2 depicts the higher level of the taxonomy of energy management methods. As it can be seen, energy management is tackled through either energy saving or energy provision during run-time.

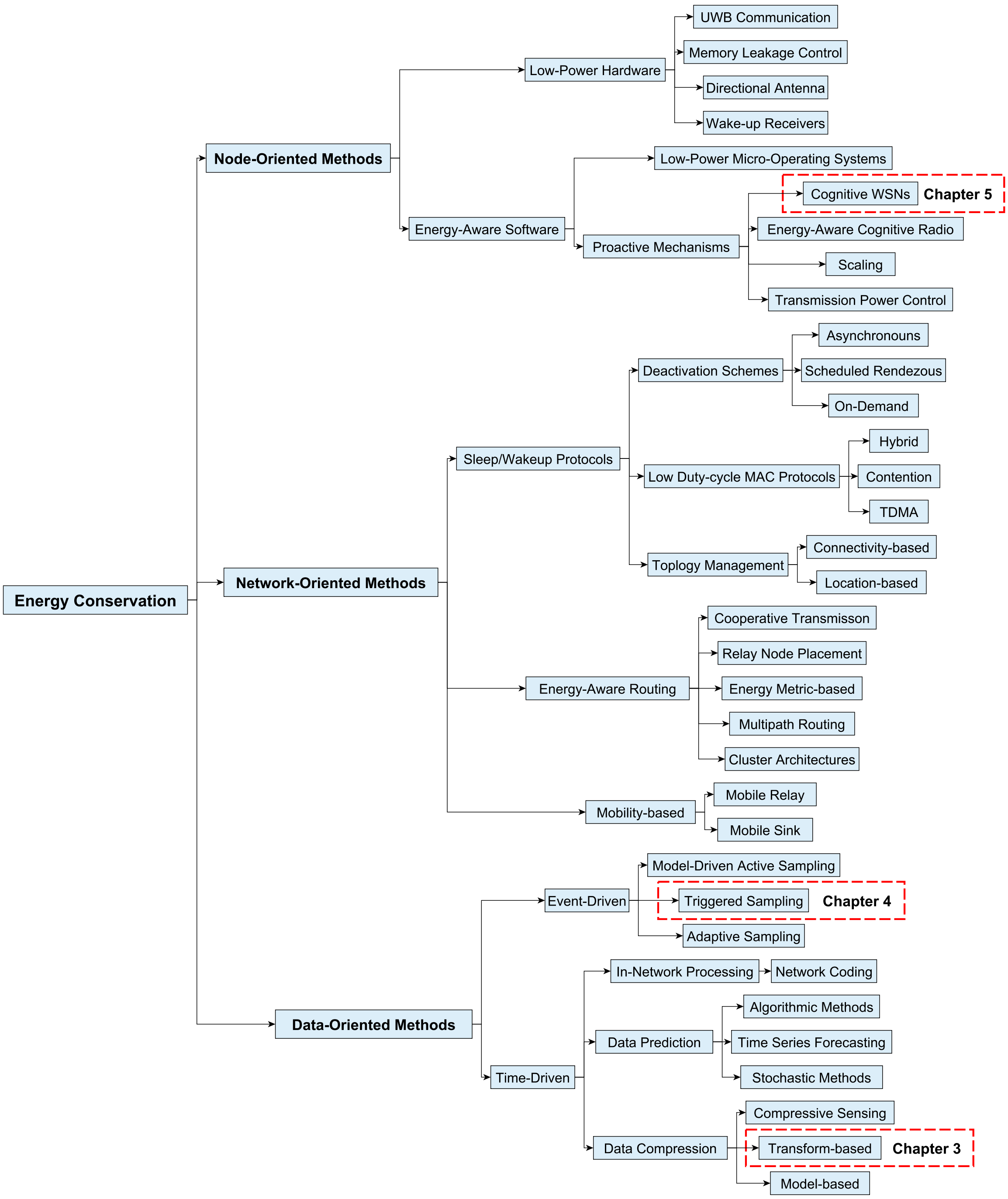

Figure 3.1.: Taxonomy of energy management techniques in WSNs

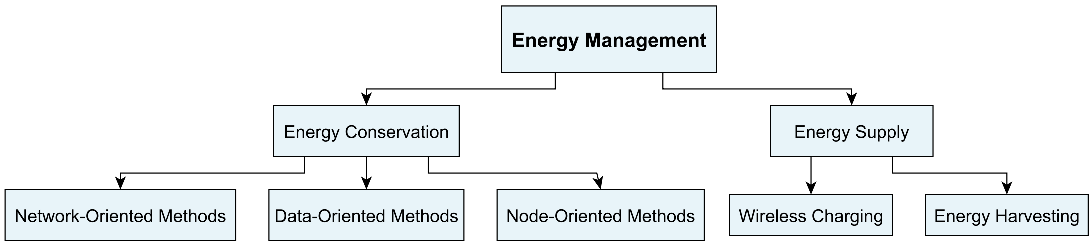

Figure 3.2.: Main energy management methods in WSNs

3.2.1. Energy Harvesting

Energy-harvesting devices efficiently and effectively capture, accumulate, store, condition, and manage this energy to energize the deployed nodes. They are increasingly attractive alternatives to costly batteries which have a relatively limited lifetime. Energy is usually everywhere in the various WSN environments. This energy is available in the form of thermal energy, light (solar) energy, wind energy, and mechanical energy [SK2011]. Currently, several sources of energy harvesting exist as follows.

- Mechanical Energy – from sources such as vibration, mechanical stress and strain
- Thermal Energy – waste energy from furnaces, heaters, and friction sources
- Light Energy – captured from sunlight or room light via photo sensors, photo diodes, or solar panels
- Electromagnetic Energy – from inductors, coils, and transformers
- Natural Energy – from the environment such as wind, water flow, ocean currents, and solar
- Human Body – a combination of mechanical and thermal energy naturally generated from bio-organisms or through actions such as walking and sitting
- Other Energy – e.g. from chemical and biological sources

For example, Innowattech Ltd. advertised recently that piezoelectric devices, if planted along a one-kilometer stretch of road, could provide an average of 400 kW of electricity, enough to power 600–800 homes annually [Dia2009]. In general, an energy harvesting system requires an energy source and three other key electronic components. First, an energy conversion device, such as a piezoelectric element, translates the energy into electrical form. Second, an energy harvesting module captures, stores, and manages the power for the device. Finally, a ZigBee-enabled WSN is to be fully or partially powered by the collected energy. These harvesting systems are increasingly attractive alternatives to inconvenient wall plugs and costly batteries. Theoretically, rechargeable motes can operate continuously for an unlimited length of time. Tables 3.1 lists, for each energy source, the amount of energy available, the harvesting technology, the conversion efficiency and the amount of converted energy. The source of energy may be ambient, active human power, or passive human power.

Table 3.1.: Listing and characteristics of various energy sources [GB2008]

energy source	amount of available energy	harvesting technology	conversion efficiency	amount of harvested energy
solar	100 mW/cm^2	solar cells	15 %	15 mW/cm^2
wind	–	anemometer	–	1200 mWh/day
finger motion	19 mW	piezoelectric	11 %	2.1 mW
footfalls	67 W		7.5 %	5 W
vibrations (indoor)	–	electromagnetic induction	–	0.2 mW/cm^2
exhalation	1 W	breath masks	40 %	0.4 W
breathing	0.83 W	ratchet-flywheel	50 %	0.42 W
blood pressure	0.93 W	micro-generator	40 %	0.37 W

Selection between these energy sources highly depends on the WSN application requirements and constraints. Obviously, the available energy harvesting technologies provide new possibilities for extending the WSN operational lifetime. However, WSN applications running out of harvested energies may have difficulties due to the following peculiarities:

- the available harvested power is limited although not the surrounding energy,
- the power availability varies over time,
- the power availability may be extremely anisotropic in different nodes,
- there is no single node with complete knowledge of the entire WSN's energy budget.

The impact of these peculiarities affects several aspects related to task scheduling and networking. Energy harvesting architectures often require energy prediction schemes in order to efficiently manage the available power via dynamically adjusting their behavior. In other words, the low-level parameters, such as transmit power and sampling frequency, are frequently modified according to the periodicity and magnitude of the harvested energy and the scenario dynamics. For instance, rechargeable motes using solar panels may intensively function during daytime and conservatively at night. Moreover, residual energy distribution over the scattered nodes is mostly uneven due to the difference in the energy collected. Accordingly, most research work investigates the question of where to spend energy in lieu of how to conserve the energy budget at each SN. This implies that WSN protocols should be designed to intensively operate at energy-rich areas, whereas they have to conserve as much energy as they can in other areas [BTS2012, JT2014].

On the one hand, the development of protocols that consider the degradation of batteries over time is crucial. Causes of such degradation comprises current leakage and storage loss. Hence, the network performance may negatively affected [TYU2015]. On the other hand, area and cost of such harvesting devices have to be manipulated. Given the typical maximum power density available of around 1 $\mu W \cdot mm^{-3}$ and the typical average power demands of a SN of circa 3 mW for a node volume of around 10,000 mm^3 (excluding the power source) means that the volume of the energy harvesting device at around 3,000 mm^3 is a significant proportion of the overall node volume [GB2008]. Accordingly, a further decrease in size and cost of such harvesting devices is a pre-requisite for WSN applications to become widely accepted.

3.2.2. Wireless Charging

Wireless power transfer (WPT) is a recently developed method for increasing the sustainability of WSNs and making them perpetually operational. In such a method, electricity is transmitted between SNs without the need of any contact between the transmitter and the receiver. In general, wireless charging in WSNs is feasible through two techniques, including *omnidirectional electromagnetic* (EM) radiation and *magnetic resonant coupling*. In [XSHL2013], the authors show that omni-directional EM radiation technology is well-suited for ultra-low power WSN applications with low sensing activities. This harsh limitation emerges due to the rapid drop of power efficiency over distance. Moreover, active radiation is highly undesirable for health-related reasons. Whenever WSN applications require a higher efficiency within a several-meter range, then magnetic resonant coupling has to be employed. It has high efficiency over several meters under omni-direction. Moreover, it is not requiring line-of-sight and insensitive to weather conditions.

The technical literature has many examples of using wireless power transfer to replenish the SNs' batteries. Table 3.2 summarizes some efforts in this arena. As it can be seen in the table, the exerted efforts rang from cross-layer optimization frameworks, utilization of unmanned aerial vehicles (UAVs) to power the terrestrial SNs, and mobile chargers (i.e. mainly robots), to the application of WPT technology in the various WSN application scenarios.

Table 3.2.: Examples of research work on WPT-powered WSNs

reference	**how it works**
[**ZHL+2009**]	WPT is applied to power medical sensors
[**JG2011**]	WPT replenishes sensors embedded in concrete
[**GD2012**]	WPT powers ground sensors from a UAV
[**XSH+2015**]	mobile chargers directly deliver power to deployed SNs
[**BA2012**]	controlling number and power level of RF sources, significant energy savings can be achieved
[**RL2014**]	jointly considering routing, scheduling, and power control in the WPT-powered WSN

In this arena, many research topics are still open and require extensive and widespread efforts. The network scalability is a main technical challenge. This barrier emerges due to the limited charging range and the slow battery charging time. For instance, recharging Nickel Metal Hydride batteries typically lasts for several hours. Additionally, the existing recharge scheduling policies have to be extended to support multihop wireless charging. Specifically, the WTP-powered SNs are envisioned to be capable of harvesting energy from the environment and transferring energy to other nodes. To realize such an idea, multi-hop energy transfer has to be investigated, which open new perspectives for the design of wireless charging protocols and energy cooperative systems.

3.2.3. Discussion

In this section, different methods for providing SNs with additional energy, during run-time, have been discussed. For energy harvesting, external power supply sources, in many cases, exhibit a non-continuous behavior which can cause system malfunctioning [YWL2015]. However, extensive research is being conducted to improve their efficiency through upgrading the power management circuits as well as the transducers and the storage devices. It is expected that energy harvesting will make continuous advancements in all the three areas, transducers, power management circuits, and storage. Furthermore, current research trends will continue for the next five to ten years owing the key applications of energy harvesting.

Environmental energy harvesting techniques provide safe, eco-friendly, renewable sources and much higher power density than wireless charging. However, as mentioned earlier, main challenges are uncertainty and variation from the power source. Therefore, possible future work should answer a question of how to combine advantages from energy harvesting and wireless charging.

As explained above, energy-supply methods – either through energy harvesting or through WPT – are not adequate to guarantee sustainable operation of the scattered SNs. Therefore, there should be interplay between energy harvesting and conservation in a unified framework that avoids wasting the collected energy. From the QoS perspective, energy provision during run-time enhances the application-relevant service quality. This hypothesis is subjected to the implementation of efficient self-adaptive mechanisms moving the load to energy-rich sectors of the sensing area. Below, the three energy-conservation methods including node-oriented methods, data-oriented methods, and network-oriented methods are explained.

3.3. Node-Oriented Methods

In this category, energy-saving methods – whose scope is within the individual SNs – are briefly discussed. In fact, the methods listed here are general enough to cover event-driven and time-driven WSN application scenarios. These energy-saving methods are designed to optimize the SN's performance without prior knowledge of the assigned task or application scenario. These methods are divided into *low-power hardware* and *energy-aware software*. The former branch encompasses some methods for manipulating the SN's hardware components. This manipulation is achieved either by adding additional modules such as the wake-up receivers, or by employing alternative technologies such as directed antenna and an Ultra Wideband (UWB) communication system. The latter branch comprises examples of methods for saving energy through modifying the embedded software. The mentioned ideas range from developing energy-aware μ-operating systems to designing self-managing (self-configuration, self-healing, self-optimization, and self-protection) software components.

3.3.1. Low-Power Hardware

Figure 3.3 depicts examples of methods which are developed to force the hardware components to consume as less energy as possible. In this field, there are several efforts, however this section discusses some of these approaches. Below, four methods, including directed antenna methods, wake-up receivers, UWB communication technology, and leakage current control are discussed.

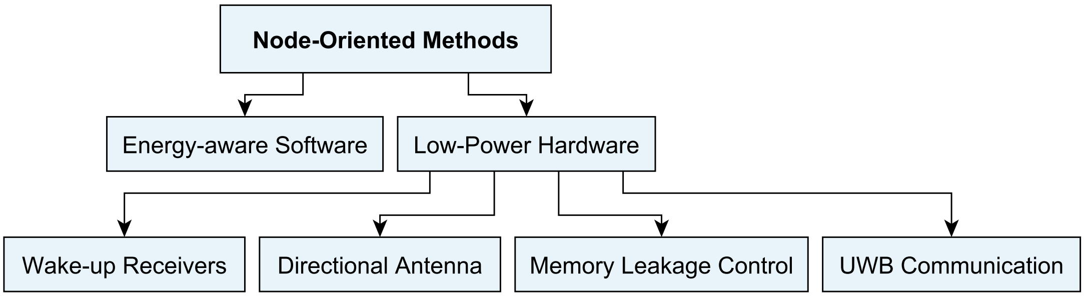

Figure 3.3.: Classification of low power hardware methods

Directed Antenna

The core idea here is to improve the transmission range and throughput via signal transmission and reception in one direction in lieu of the omnidirectional manner. Directed antenna schemes are advantageous due to allowing multiple communications in close proximity within the same bandwidth. In other words, directed antenna schemes permit the spatial reuse of bandwidth. Directed transmission also sidesteps possible overhearing, for a given range. However, their main advantage is consuming less energy than that drawn by an omnidirectional antenna. The main disadvantage of directed antenna schemes is their need for localization methods to determine the direction of transmission and reception [KKW2004].

Staniec et al. [SD2012] introduce a simplified model for the antenna radiation characteristic. Based on this model, they define a threshold for the antenna beam-width such that the signal-to-interference noise-ratio (SINR) is optimized even in scenarios with simultaneous emissions from all SNs. Alternatively, Rout et al. [RGG2012] propose a combination of directional antenna and network coding which results in a significant reduction in the number of transmissions and receptions. They indicate that lower beam angles can be chosen to transmit data packets for longer distances with lower energy consumptions. Choudhury et al. [CYRV2006] considers the problem of designing medium access control protocols for ad hoc networks using directional antennas. However, some problems that are specific to directional antennae have to be considered: signal interference, antenna adjustment and deafness problems.

Wake-up Receivers

In the following paragraphs, ideas for engaging wake-up receivers are briefly described. In general, wake-up receivers are used to trigger the main receivers whenever an incoming signal is detected by the wake-up receivers. Accordingly, unnecessary activation of the main power-hungry receivers for idle listening is avoided, leading to significant energy savings. Figure 3.4 demonstrates a classification of the recent efforts in this filed. Based on their energy sources, wake-up receivers can be active in which the internal battery is utilized to power the wake-up receiver. However, the power source of a wake-up radio can highly affect this energy gain. Recent hardware developments have provided realistic approaches to power the wake-up radio passively. Such that the wake-up radio is entirely powered by the wake-up signal and does not need any additional battery supply [CCB+2013].

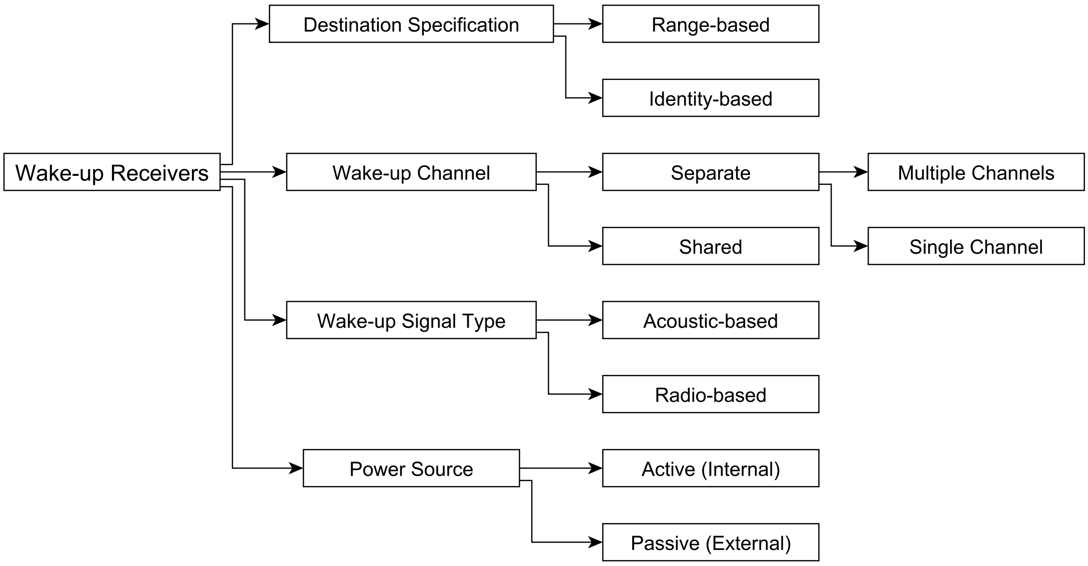

Figure 3.4.: Classification of wake-up receiver schemes

The channel on which the wake-up signal is sent can be the same as the main radio communication channel (i.e., shared channel), or a separate channel can be used for the wake-up signaling. This separate wake-up channel may consist of multiple channels to be able to wake up specific nodes utilizing frequency division [DEO2009]. Although a separate channel increases the cost and complexity of the SN, the energy gain of deactivating the main receiver outweighs this additional overhead.

The wake-up signal can be a single wake-up tone or a bit sequence. In range-based wake-up receivers, all the SNs that receive the tone activate their main transceiver. In identity-based wake-up receivers, the wake-up signal may consist of a bit sequence to address the destination. After the reception of a wake-up signal, nodes check if the bit sequence refers to them; if so, the destination wakes up. Mostly, radio signals are employed as wake-up signals in radio-based wake-up receivers [DEO2009]. Alternatively, acoustic wake-up receivers are triggered by acoustic signals. When the observed level of the external sound reaches a threshold, the wake-up circuitry is turned on.

Ba et al. [BDH2013] consider the programmable RFID tags to implement a passive wake-up radio for WSNs. The wake-up radio is realized using a passive RFID tag as the wake-up signal receiver, whereas an RFID reader acts as the wake-up radio transmitter. However, the results reveal that the wake-up range is relatively limited compared to the ZigBee-compliant sensor mote communication range. Moreover, the radio wake-up transmitter requires high energy consumption that cannot be applied to all SNs. Later, the authors propose in [CCB+2013] a RFID range extension method through energy harvesting.

UWB Communication

Ultra Wideband (UWB) is a short-range wireless communication technology based on transmission of very short impulses emitted in periodic sequences. The advantages of UWB communication technology over WiFi and ZigBee are as follows [ZWYX2006].

- It has good localization capabilities.
- It is able to share previously allocated radio-frequency bands by hiding signals under the noise floor.
- It can transmit high data rates with low power.
- It has good security characteristics due to the unique mode of operation.
- It is able to cope with multipath environments.

Compared to narrow band Zigbee and WiFi, UWB offers significant advantages with respect to robustness, energy consumption and location accuracy. UWB spreads the transmit signal over a very large bandwidth (typically 500 MHz or more). However, UWB is not a viable approach for communication over longer distances. Moreover, several challenges emerge in the hardware development, dealing with MAC and multipath interference, and understanding propagation characteristics [ZOS+2009].

Memory Leakage Control

The leakage current can be controlled to save the energy waste [VIK+2005]. Different approaches are proposed in this arena such as 1) ones which make their leakage management decisions based on performance feedback, 2) techniques that manage cache leakage in an application-intensive manner (e.g. by periodically turning off cache lines), and 3) techniques that utilize feedback from the program behavior. Swaran et al. [SM2009] examine several leakage reduction techniques applied to binary and ternary content addressable memories. Accordingly, they show that leakage can be reduced by a factor of 168 over non-optimized designs.

3.3.2. Energy-Aware Software

Figure 3.5 depicts a classification of the energy-aware methods which rely on optimizing the software components. As shown in Figure 3.5, the methods are divided into energy-aware μ-operating systems and self-managing software components. First, current operating systems are ill-suited for WSNs. This limitation stems from the taxed WSNs resources. Moreover, WSN applications mostly undergo several changes at run-time due to the environmental dynamics, the topology change, or the application requirements. Accordingly, μ-operating systems have to be designed in such way that conserve energy and dynamically allocate the SN's resources.

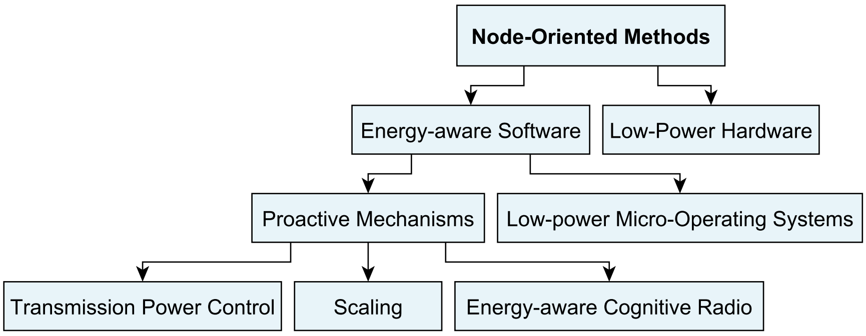

Figure 3.5.: Classification of energy-aware software methods

Second, self-managing methods are recently developed to increase the WSN "intelligence". The term "managing" refers to various tasks including parameters configuration, self-diagonals, self-healing, self-optimization, and self-protection. In the sequel, examples of these methods are only given, such as scaling, transmission power control, and cognitive radio sensor networks.

Low-Power Micro-Operating Systems

Micro-operating systems (μOSs) in WSNs are classified into *event-driven* μOS and *multi-threaded* μOS. The former are efficient in terms of resources utilization. The latter μOSs have superior event processing capabilities [FK2011]. Recent μOSs, such as Contiki and SOS, comprise generic abstractions to manage the power consumed by peripherals of the sensor devices. On one hand, μOSs can accomplish significant energy reduction by performing energy-aware task scheduling and resource management. On the other hand, compilers have been studied to generate efficient code in terms of power consumption [VIK$^+$2005].

Proactive Mechanisms

In this section, self-adaptive (or referred to as "proactive") mechanisms are briefly introduced. Three methods, including energy-aware cognitive radio sensor networks, scaling, and transmission power control, are explained. The common denominator between these methods is the proactive behavior in changing the low-level parameters according to the internal and external context information.

Energy-Aware Cognitive Radio In general, cognitive radio (CR) is a new technology by which the wireless devices can "intelligently" sense and exploit portions of the unused spectrum of the licensed users/networks. Recently, the CR concept has been exploited for resource allocation in existent network infrastructures such as WSNs and smartphones. This leads to founding a new paradigm, referred to as the cognitive radio sensor networks (CRSNs) [AARH2015]. CRSNs are WSNs which encompass "intelligent" wireless communication systems able to determine the most favorable operating parameters (cognition) based on the radio environment and its own capability (awareness). Accordingly, they reconfigure the radio parameters (reconfigurability). Such dynamic adaptations lead to more efficient utilization of the allocated radio resources. A CRSN permits not only modulation adjustment, coding, and radiated power control, but also learning and adjusting component characteristics to reduce the energy dissipation [MHDN+2012]. Another track in this field is the software-defined radio (SDR) technology. This method enables fully programmable wireless transceivers which automatically adapt their communication parameters to network demands, which improves context-awareness.

Ayaz et al. [AARH2015] survey radio resource allocation schemes for CRSNs. Such schemes have been divided according to the source of reconfiguration decisions into centralized, distributed, and cluster-based methods. He et al. [HSR+2008] propose a cognitive radio framework to minimize radio energy consumption and to improve the throughput. Figure 3.6 demonstrates the cognitive radio framework for optimizing the radio energy consumption. The solid lines, as they can be seen in Figure 3.6, represent the existing components in conventional wireless communication devices. Whereas, the cognitive engine (CE) block is to be used in learning the characteristics and capabilities of other building blocks and frequently configures these blocks.

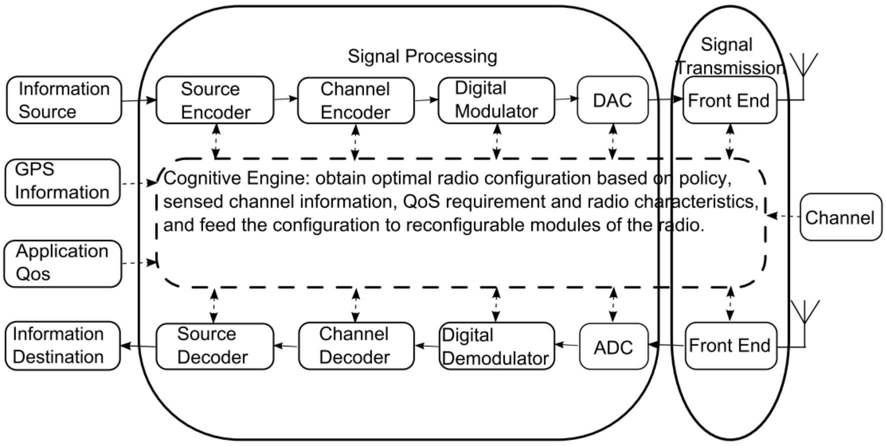

Figure 3.6.: CR self-adaptive mechanism for saving energy [HSR+2008]

Vijay et al. [VBI2010] provide a comparative study of the different cognitive techniques applied to recent WSN applications. They discuss applying the concept of cognition based on knowledge and learning to all SNs' modules so that end-to-end goals of the WSN are realized. Recent cognitive radio studies are interested in the power control of transmitters, residual energy-based channel assignment, and combining network coding and CR. Open research issues include the development of cross-layer approaches for MAC, routing or clustering protocols that take advantage of cognitive radio opportunities.

Scaling The core idea of scaling methods is to dynamically adapt the processor's operating voltage and frequency based on instantaneous computational load requirements. As a result, significant processing power can be saved. This method is referred to as *dynamic voltage scaling* (DVS) [TSH2010]. Additionally, *dynamic modulation scaling* (DMS) is a well-known method for reducing broadcasting energy with respect to the number of packets that need to be transmitted at that particular time interval [JDiJR2007].

However, DMS methods may increase the overall system latency due to the inherent complexity.

Kulau et al. [KBW2013] present a theoretical approach to estimate the gain and the ideal usage of DVS on wireless SNs. For the sake of further decreasing the consumed energy, they engineer a hybrid framework of DVS and dynamic power management (DPM). In the DPM method, the SN continuously moves from one operational state to another according to the residual energy and the instantaneous network load. However, modern SNs do not support DVS due to the overhead of DC-DC converters. Furthermore, DVS mostly does not achieve enough gain because of the power loss of DC-DC converters [CKC2007]. To resolve such limitations, Youngjin et al. [CKC2007] introduce passive voltage scaling without DC-DC converters.

Transmission Power Control In this category, the radio transmission power is dynamically adjusted in accordance with the residual energy and the network load. Chu et al. [CS2015] propose a game-theoretic approach to control transmission power of every SN taking into consideration the uneven energy consumption distribution. However, transmission power decrease may have negative impacts on communication latency, interference, and connectivity. Specifically, reducing the transmission power leads to increasing the number of hops to convey data. These additional hops result in a superfluous delay. Moreover, connectivity can change; hence, transmission power control has to be accompanied with a suitable topology control method. Although transmission power control are relatively related with topology control, they cannot be tagged as a topology control technique.

Zheng et al. [ZWM+2010] propose an adaptive transmission power control algorithm for WSNs. This transmitter detects the minimum transmission power required to connect with the neighboring SNs. Their algorithm considers the power levels as “distances” to solve a shortest path problem. The solution gives the minimum transmission power levels for all nodes and the optimized routing information for the whole network. Transmission power control techniques can achieve higher energy gains when properly integrated with communication protocols. For instance, transmission power can be used as a routing metric; thus energy-aware routes are estimated. Moreover, transmission power control techniques have to be extended to support broadcast and multicast packets [CMdS+2007].

3.3.3. Discussion

The aforementioned methods deal with saving energy at the node-level. Several ideas have been examined either on real WSN testbeds or using simulations. Indeed, all these methods provide uneven contributions to the energy efficiency problem. However, negative impacts on other service qualities often stem from only focusing on the energy problem. Table 3.3 summarizes the discussed methods and their influences on other service qualities. For instance, the additional complexity of directed antennae emerges from the need for an accurate localization system; hence, the energy consumption and the processing time are increased. Moreover, the coverage of such directed antenna-equipped SNs is arguable where many hops have to be achieved in order to reliably communicate between two nodes with no line of sight.

Table 3.3.: Summarizing the various node-oriented energy saving methods

Method	Explanation	Influenced QoS	Reference
Directed Antenna	signal transmission and reception in one direction in lieu of the omnidirectional manner	coverage, more complexity	[KKW2004, SD2012, RGG2012, CYRV2006]
Wake-up Receivers	triggering the deactivated receiver whenever an incoming packet is detected	communication delay	[CCB+2013, DEO2009, BDH2013]
UWB Communication	transmission of very short impulses emitted in periodic sequences	connectivity, communication reliability	[ZWYX2006, ZOS+2009]
Memory Leakage Control	reducing the memory leakage current	–	[VIK+2005, SM2009]
Low-Power μ-OSs	energy-aware task scheduling and resource management	run-time operation	[FK2011, VIK+2005]
Energy-Aware CRSNs	a self-adaptive mechanism for allocating radio resources	latency, complexity	[AARH2015, MHDN+2012, HSR+2008, VBI2010]
Scaling	adapting the operating voltage and frequency based on instantaneous computational load	processing latency	[TSH2010, JDiJR2007, KBW2013, CKC2007]
Transmission Power Control	Adapting the transmission power according to the residual energy and network load	connectivity, interference, communication reliability	[CS2015, ZWM+2010, CMdS+2007]

To sum up, node-oriented methods are extremely beneficial for reducing the energy consumption. Moreover, they do not depend on the application scenario. However, the significance of each method relatively relies on the application features. For instance, wake-up receivers are ideal for high-frequency data reporting, whereas their gain is reduced with less frequent data transmission. In this thesis, a contribution to the category of energy-efficient software is introduced through proposing a self-adaptive framework to plan the entire lifetime of each SN. The crux of such planning is to exploit the residual energy for improving the provided service quality. More details about this novel method are given in Chapter 6.

3.4. Data-Oriented Methods

In this section, several ideas of reducing the sampled and transmitted data are discussed. The reported methods are broadly divided, according to the data aggregation scheme, into *event-driven* methods and *time-driven* methods. The former method mainly focuses on reducing the burden of data sampling and processing to prolong the event-based WSN application scenarios. The latter method circles around minimizing the radio energy burden of time-driven WSN scenarios. Several research work has been devoted to handle these two objectives. Below, examples of such efforts are briefly described.

3.4.1. Event-Driven Methods

We introduce methods for reducing the sensing overhead. Mainly, two methods are considered to reduce the energy consumed by a sensor: *duty-cycling* and *adaptive sensing* [AADFR2009]. In the former, sensing modules are only woken up for the time needed to acquire a new set of samples. This strategy allows to optimally manage energy, provided that the dynamics of the monitored phenomenon are time-invariant and known in advance. Alternatively, adaptive sampling is proposed to avoid wasting energy with oversampling or to reduce the event-miss probabilities with under-sampling. The core idea behind adaptive sampling is to reduce the number of samples, consequently reducing the amount of data to be processed and (possibly) transmitted.

As depicted in Figure 3.7, duty cycling and adaptive sensing are complementary approaches that form a unified framework. In this figure, the μ-OS provides a set of primitives for activating and deactivating the sensing module to support duty cycle mechanisms. Based on these primitives, the application implements an adaptive sampling strategy to dynamically acquire data. Currently, most available μ-OSs do not have automatic duty cycle management. Instead, the application programmer has to manually decide when to activate and deactivate the sensing module. Future μ-OSs have to adopt the automated and sensor-specific approach for both, relieving the application programmer from manual handling and improving the effectiveness of the duty-cycling mechanism.

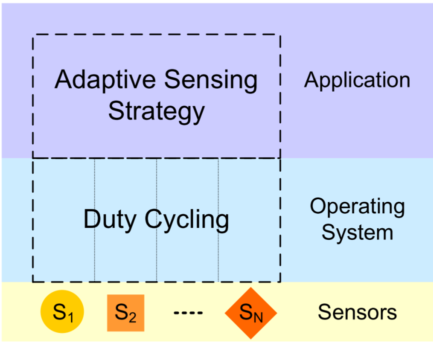

Figure 3.7.: A general framework for sensing module's energy management [AADFR2009]

Specifically, adaptive sensing can be implemented by exploiting three different approaches, i.e., *model-based active sensing*, *adaptive sampling*, and *triggered sensing*. Below, the philosophy behind each technique is briefly introduced.

Model-Based Active Sampling

The core idea behind the model-based active sampling is to reduce the number of data samples by using a computed model. Figure 3.8 demonstrates the core idea behind the model-based active sampling. Initially, an abstraction of the sensed phenomenon is determined though a forecasting model. This model predicts the next readings, hence avoiding the burden of sensing each data sample. In such a method, both, source and sink nodes have to run the same predication model. Both of the forecasting model's accuracy and nature of the monitored phenomenon have a significant impact on the effectiveness of such a model-based active sampling.

Gedik et al. [GLY2007] propose ASAP – an adaptive sampling approach to energy-efficient periodic data collection in WSNs. ASAP splits the network into clusters such that SNs with close readings are assigned to the same clusters. Within each cluster, a dynamically changing subset of the SNs is selected as samplers such that the sampler nodes' readings are directly collected, whereas the values of the non-sampler nodes are predicted by probabilistic models. Mietz et al. [MR2011] introduce a new method for learning the correlation structure from past sensor data and model it as a Bayesian Network (BN). The BN allows estimating the probability that a sensor currently outputs the expected state. Alternatively, Jiang et al. [JJW2011] describe an adaptive framework

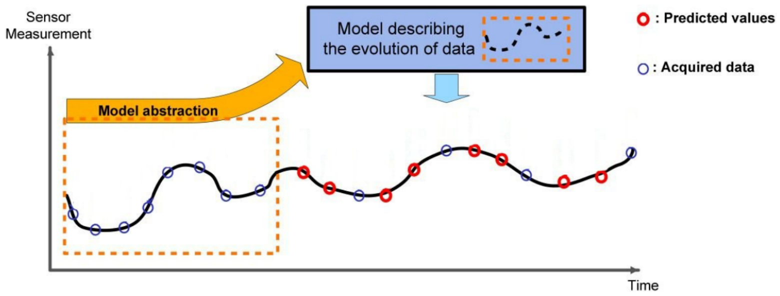

Figure 3.8.: The core idea behind the model-based active sampling method [AADFR2009]

for enabling/disabling the prediction scheme. This framework analyzes the performance tradeoff between reducing communication cost and limiting prediction cost. In general, more efficient algorithms to reduce the computational overhead of prediction schemes have to be developed to maximize the energy gain of such model-based active sampling.

Adaptive Sampling

According to the Nyquist theorem, the minimum sampling frequency (F_s) needed for correct reconstruction of the original signal should be $F_s = 2 \times F_{max}$ where F_{max} is the maximum frequency in the power spectrum of the considered signal. Unfortunately, choosing F_{max} is not trivial, because

- it cannot be known a priori, thus leading to choose an unnecessary high sampling frequency,
- the maximum frequency may vary over time.

To overcome this problem, several research work propose an adaptive algorithm that dynamically estimates the current maximum frequency in accordance with the tempo/spatial correlation among acquired data and/or the residual energy. For instance, Alippi et al. [AAG+2007] propose an adaptive sampling algorithms for snow monitoring applications. The current maximum frequency of a signal is determined by using a first set of acquired samples. The change is detected when the current maximum frequency happens to be above or below a threshold for some consecutive samples. Alternatively, Jain et al. [JC2004] propose an adaptive sampling method based on the outcome of a Kalman filter. They implement the algorithm in a decentralized fashion, i.e., the Kalman filter is executed on each SN.

Such solutions might not be feasible in WSNs consisting of tiny devices with limited computational capabilities. It is conclusive that with adaptive sampling a compromise between sampling rate which will influence the resources usage and accuracy needs to be achieved. The degree of complexity of the model will also influence both, the accuracy of the estimated model and resource usage.

Triggered Sampling

The core idea behind triggered sensing (sometimes referred to as hierarchical sampling) is to equip the SN with different sensors to measure the same phenomenon. Each of such different sensors is characterized by its own accuracy and power consumption. On one hand, simple sensors are energy-efficient while offering a relatively limited precision. On the other hand, complex sensors are more accurate at the expense of higher energy consumption. In trigger sampling, low-power sensors enable coarse-grained characterization of the sensing field. The energy-hungry sensors are activated whenever needed to improve the coarser description. In Chapter 5, a contribution in this arena is proposed. The concept of virtual sensing is exploited to enhance the triggered sampling method. A full description of the related work is given in Chapter 5.

3.4.2. Time-Driven Methods

In time-driven WSN applications, such as environmental, machinery and patient monitoring, radio communication for reporting the data samples represents a heavy burden. To provide a solution, several methods have been proposed, as depicted in Figure 3.9. For example, *compressive sensing* (CS) is a distributed data reduction technique in which the reading is processed to remove redundancy by exploiting spatial correlation. CS depends on reconstructing the sparse WSN's data from a small number of random-linear readings [BRK2011]. *Coding by ordering* and *pipelined in-network* are samples of the distributed techniques which can be applied in the context of WSNs for reducing transmissions [KL2005]. *Data prediction* is another technique to minimize the number of transmitted packets via implementing identical predictors in the source and sink nodes [BD2008]. Many prediction algorithms have been utilized in WSNs including time series forecasting, stochastic and algorithmic approaches.

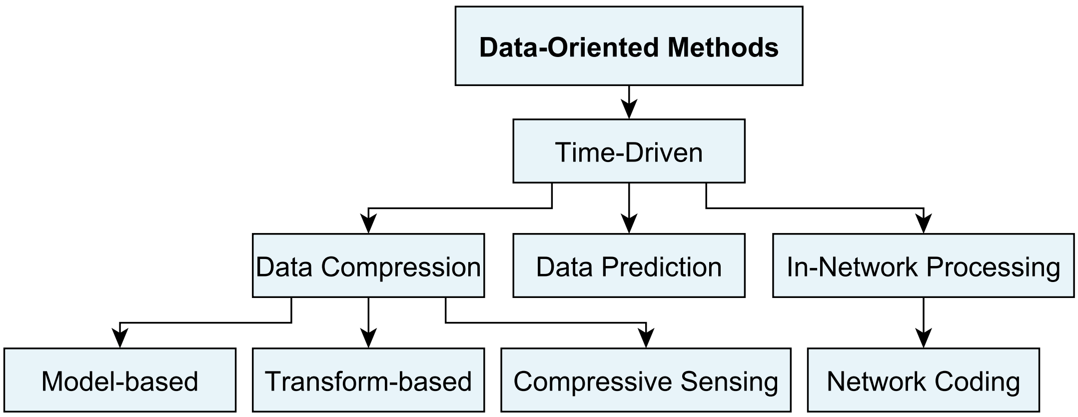

Figure 3.9.: Classification of data reduction methods for time-driven scenarios

Data Compression

As expressed in Eq. 2.6, on page 30, radio power consumption strongly depends on the packet size. Therefore, removing data redundancy is essential to find a more compact representation. Compression (sometimes called *encoding*) may be *lossless* or *lossy*. The former grants the exact original data to be reconstructed from the compressed data. *Huffman coding* and *Arithmetic coding* are samples of the lossless compression. The latter method is one where compressing data and then decompressing it, retrieves data that may well be different from the original, however it is "close enough" to be useful in manifold applications. Examples of this method comprise the *wavelet transform* and the *Fourier transform* [KL2005]. The point of interest of lossy strategies over lossless ones is that, in some cases, a lossy method can produce much smaller compressed signals than any known lossless method, whereas still meeting the requirements of the application [RBD2013]. A new contribution to this category is proposed in Chapter 4. The proposed data compression method circles around examining a recently-developed transform, referred to as Fuzzy transform, for compressing/decompressing sensor readings. Moreover, the algorithm is modified twice to enhance the accuracy and latency of the Fuzzy-based compressor. More information about existent compression methods is summarized in Section 4.2.

In-Network Processing

The basic idea behind in-network processing methods is to reduce the traffic at intermediate nodes between the sources and the sink through performing data aggregation (e.g., computing average of sensor readings within a predefined time window). Several ideas exist for the aggregating function. However, the most appropriate in-network processing method is application-dependent. Fasolo et al. [FRWZ2007] provide a comprehensive survey about in-network processing techniques.

Network coding (NC) is a new in-network processing method which exploits the characteristics of the wireless medium (in particular, the broadcast communication channel). It is mainly developed to reduce the traffic in broadcast scenarios by sending a linear combination of several packets instead of a copy of each packet. Figure 3.10 depicts an example of the network coding strategy [RBC2014]. In this example, a five-node topology is constructed such that node 1 has to broadcast two packets, a and b. Without the NC approach, if the nodes 1, 2, and 3 store and omni-directionally forward the data packets, this will generate six packet transmissions (2 per each node). Alternatively, nodes 2 and 3 can transmit a linear combination of data items a and b with the NC approach. Accordingly, the nodes 2 and 3 have to send only a single packet. Nodes 4 and 5 can decode the packet by solving linear equations. As a result, two packets are saved in total in this example. In general, NC approach exploits the trade-off between computation and radio communication.

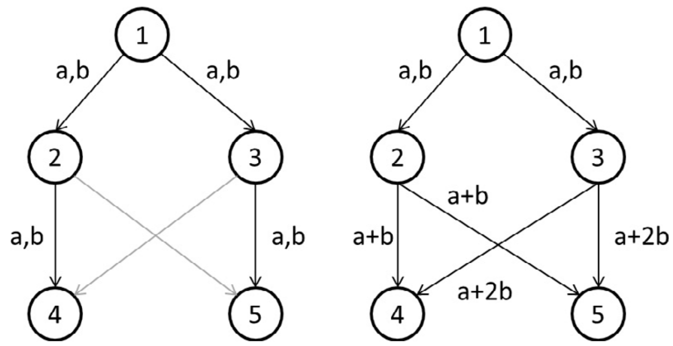

Figure 3.10.: An example of network coding strategy [RBC2014]

Ostovari et al. [OWK2014] classify the methods based on their objective, application, and network topology assumption. Despite the advantages of the NC strategy for saving energy and improving the communication reliability, Voigt et al. [VRL+2012] report on several drawbacks, including: (1) strongly increased delay, and (2) high overhead due to lack of adaptability. Accordingly, research efforts have to be exerted in this arena to overcome such limitations.

Data Prediction

This strategy is similar to the model-based active sampling in which an abstraction of the sensed phenomenon is created. The model frequently predicts the sensor readings within certain error bounds. Identical predictors have to be implemented at the sensors and the sink nodes. If the prediction model is accurate enough, queries issued by users can be evaluated at the sink without sampling real values. Otherwise, explicit data transmission between SNs and the sink is mandatory. Such approaches need to periodically validate and update their models in order to avoid a rapid deterioration in the predicted values. Indeed, data prediction reduces the number of data packets sent by the various SNs, and the energy needed for communication as well. The main forecasting schemes are classified into *algorithmic methods*, *time series forecasting*, and *stochastic methods*. Below, the philosophy of each branch is briefly described with mentioning examples of the existent work.

Stochastic Methods The core idea behind the algorithmic methods is to estimate a stochastic characterization of the phenomenon to be measured [ACDFP2009]. This estimation can be achieved in two different ways. In the first way, data is to be mapped into a random process described in terms of a *probability density function* (PDF). Through combining the computed PDFs with the observed samples, the data prediction can be easily obtained. In the second way, a state space representation of the phenomenon can be derived, so that forthcoming samples can be predicted by filtering out a non-predictable component modeled as noise. Chu et al. [CDHH2006] propose a robust approximate technique that utilizes replicated dynamic probabilistic models to minimize communication from SNs to the sink node. Although this approach is general, its computational overhead makes it ill-suited for tiny SNs with limited computational capacities.

Time Series Forecasting The main idea here is to use a set of historical values to predict a future value in the same series. The time series method explicitly considers the internal structure of data [ACDFP2009]. In general, a time series can be decomposed into three components, a trend, a season, and a remainder. The trend component can be described by a monotonically increasing or decreasing function that can be approximated using common regression techniques. Once the trend is fully characterized, the resulting model can be used to predict future values in the time series.

The moving average (MA), the auto-regressive (AR), or the auto-regressive moving average (ARMA) models are simple examples of time series predictors which are easy for implementation and provide an acceptable accuracy. Santini and Römer [SR2006] choose the Least Mean Square (LMS) method over a Kalman filter since it does not require a priori knowledge of the desired measurements, which implies that the sink and the sensors do not need to agree on a pre-defined model. Miranda et al. [MRRR2013] show that a well-tuned AR estimator may be used to estimate data series in cluster-based one-hop WSNs.

Algorithmic Methods This category encompasses methods which rely on a heuristic or a state-transition model describing the sensed phenomenon [SR2006]. Such algorithmic approaches derive methods to construct and update the model on the basis of the chosen characterization. For instance, Han et al. [HMV2004] propose an energy-efficient data collection (EEDC) method which is well-suited in inquiry-based applications. In such scenarios, each SN relates with an upper and lower bound and difference between bounds denotes the accuracy of sensed values. In general, algorithmic approaches incur complex computations and sometimes generate additional communication overheads.

3.4.3. Discussion

The aforementioned methods deal with saving energy through data manipulation. The methods have been split into two broad categories: event-driven and time-drive methods. The event-driven methods focus on the WSN scenarios in which processing or sensing is dominating the energy consumption. Most ideas in this category revolve around reducing the sensed samples and decreasing the sensors' duty cycle. The time-driven methods are clearly developed for the WSN scenarios in which data transmission is frequent. In these applications, radio communication consumes the majority of the allocated energy budget. Accordingly, the proposed ideas in this category premise on reducing number and size of the transmitted data packets.

Table 3.4 summarizes the discussed methods and their influences on other service qualities. Although these methods prove to be successful in reducing the energy consumption, they often have negative impacts on other service qualities. As an example, data compression highly reduces the burden of radio communication by removing data redundancy and shrinking the number of packets. In most cases, lossy data compressors are utilized due to their ability to achieve a higher compression ratio than what can be obtained via lossless compressors. Therefore, users have to accept a certain level of accuracy degradation in the recovered data after decompression. Moreover, data compression methods, in many cases, spend a considerable amount of time for storing the uncompressed pattern and for evaluating the compression algorithm. The resultant delay may be harmful in many critical WSN applications such as smart grid monitoring. In Chapter 4, the aforementioned limitations of data compression methods are addressed through proposing a novel compression algorithm, referred to as *FuzzyCAT*. In this compressor, accuracy is highly improved via dynamically adapting the algorithm characteristics in accordance with the original signal's curvatures. Moreover, the latency is investigated in the light of a new cooperative multi-source prediction scheme.

As listed in Table 3.4, trigger sampling is another example in which the duty cycle of the energy-hungry sensors are highly reduced. However, most ideas in this arena suffer from the complexity of the new heterogeneous sensing system. In Chapter 5, this problem is targeted via proposing the invocation of virtual sensors as secondary sensors. Such sensors are extremely energy-efficient and can be easily designed by commercial

Table 3.4.: Summarizing the various data-oriented energy saving methods

Method	Explanation	Influenced QoS	Reference
Model-Based Active Sampling	reducing the number of data samples by using a computed model	accuracy	[AADFR2009, GLY2007, MR2011, JJW2011]
Adaptive Sampling	dynamically estimating the sensors' sampling rate	event-miss, complexity	[JC2004, AAG+2007]
Triggered Sampling	equipping the SN with different sensors to measure the same phenomenon	complexity, sensing reliability	Chapter 5
Data Compression	reducing the packet size by removing data redundancy	latency, recovery accuracy	Chapter 4
In-Network Processing	reduce the traffic at intermediate nodes through performing data aggregation	complexity, latency	[FRWZ2007, RBC2014, VRL+2012]
Data Prediction	constructing an abstraction of the sensed phenomenon	accuracy	[ACDFP2009, CDHH2006, SR2006, HMV2004]

off-the-shelf (COTS) components. In addition, reliability of such a new heterogeneous sensing system is improved. An ontology-based automatic rule generation method is developed to dynamically select between the main sensors and the virtual ones in the light of the virtual sensors' accuracy and the environmental conditions.

3.5. Network-Oriented Methods

In this category, energy-saving methods – whose scope is within the entire network – are briefly discussed. Indeed, the methods listed are general enough to cover event-driven and time-driven WSN application scenarios. These energy saving methods are designed to optimize the WSN's performance without prior knowledge of the assigned task or application scenario. As it can be seen in Figure 3.11, the taxonomy has three main roots, including *mobility-based methods*, *energy-aware routing*, and *sleep/wake-up protocols*. The former employs mobile sinks or mobile relay nodes in order to reduce the number of multi-hops. Energy-aware routing methods deal with data propagation within the network in a way that reduces the total energy consumption. Finally, sleep/wake-up protocols focus on reducing the number of active SNs to eliminate possible redundancy. Below, the philosophy behind each category is briefly explained.

3.5.1. Mobility-Based Methods

As it can be shown in Figure 3.11, mobility-based methods rely on employing either mobile sinks or mobile relay nodes in order to reduce the number of multi-hops and thereby minimizing the transmission cost [DD2012]. These mobile nodes are often attached to mobile entities in the environment such as vehicles, animals, or dedicated robots.

Specifically, mobility-based methods increase the network lifetime through reducing the burden on bottleneck nodes. In general, SNs closer to the sink have to relay more packets so that they are subject to premature energy depletion, even when applying other energy-efficiency techniques mentioned above. Through adding mobility, the traffic flow can be altered with mobile data collectors. Ordinary nodes wait for the passage of the mobile device and route messages towards it. Accordingly, the number of multi-hops of radio communication is highly reduced. As a consequence, ordinary nodes can save energy thanks to reduced link errors, contention overhead, and forwarding.

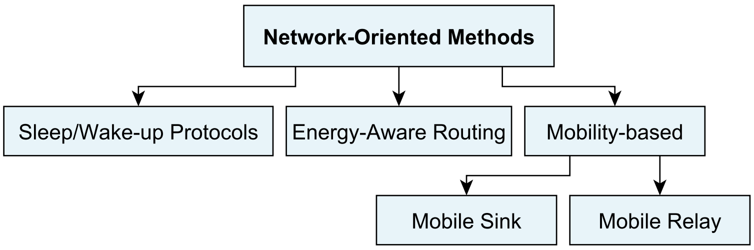

Figure 3.11.: Classification of mobility methods

Silva et al. [SSB2014] introduce a comprehensive survey of mobility models in WSNs. This survey considers the mobility feature from different perspectives, including the MAC layer and the network layer. The authors also propose the network of proxies (NoP) concept to relieve SNs from performing complex mobility tasks by moving them to the network side. Jain et al. [JSB$^+$2006] present the MULE architecture as an alternative to ad-hoc networks. The MULE architecture is a three-tiered design, including sensors, mobile ubiquitous entities, and sink nodes. The key idea of MULE is to exploit the presence of mobile nodes in the environment by using them as forwarding agents. Sugihara and Gupta [SG2009] investigate the trade-off between saving energy by employing mobile collectors and the increased data delivery latency. Generally, with controllable mobile nodes, the mobile displacement can be studied to prevent high latency, buffer overflow, and energy depletion.

3.5.2. Energy-Aware Routing Methods

Routing is the process of delivering information to the destination through a – hopefully – short path. Many optimization techniques have been proposed to improve the routing performance in terms of energy consumption [Cir2011]. Some existent work utilize special nodes with unlimited energy sources for assisting the battery-powered sensors in aggregating information. Examples of protocols include *information-driven sensor query* (IDSQ) and *cluster-head relay routing* (CHR). Alternatively, several other work has been developed with the assumption of even energy allocation [SSS2010]. As depicted in Figure 3.12, a classification of such energy-aware WSN routing protocols includes relay node placement, energy metric-based methods, cooperative communication, multipath routing, and clustering architectures.

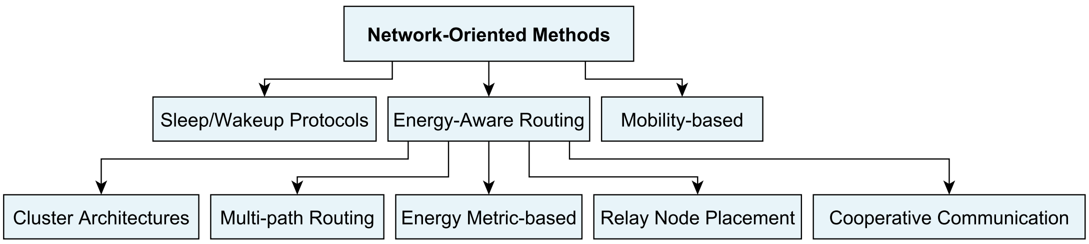

Figure 3.12.: Classification of energy-efficient routing methods

Relay Node Placement

Network partitioning and energy holes can be sidestepped by the optimal placement of SNs through even distribution or by adding a few relay nodes with enhanced capabilities. Ergen and Varaiya [EV2006] focus on optimally locating the relay nodes to prolong the network lifetime. Alternatively, Misra et al. [MMH2011] define the location of each relay node at the design-time. These locations are governed by the energy harvesting potential. The authors focus more on estimating the minimum number of relay nodes, to achieve connectivity or survivability, while ensuring that the relay nodes harvest large amounts of ambient energy.

Energy Metric-Based Methods

The core idea of this category is to consider the consumed energy as a criterion in the path setup phase. In other words, selecting routes can be accomplished in the light of the residual energy in lieu of a shortest path strategy. For example, Liu et al. [LRL+2012] propose an adaptive double cost function based routing (DCFR) algorithm. In DCFR algorithm, the cost function is composed of end-to-end energy consumption and residual energy. Akkaya and Younis [AY2003] present an energy-aware QoS routing mechanism which tries to balance energy consumption and throughput. The trick of this protocol is to look for a delay-constrained path with the least possible cost based on a cost function defined for each link. Alternative paths with higher costs are tried until one is found, which meets the end-to-end delay requirement and maximizes the throughput.

Cooperative Communication

Several single-antenna devices are used to improve the received signal's quality. Such devices cooperate to form a virtual multiple-antenna transmitter. Overhearing is typically a superfluous phenomenon in which data directed to a certain node, is also received by this node's neighbors. The core idea of cooperative transmission (CT) or communication schemes is to engage the overhearing neighboring nodes in the data retransmission phase. Many versions of the same data are then combined at the receiving node to improve the signal-to-noise ratio (SNR) relative to the conventional non-CT communication [LTW2004]. The CT schemes deliberately combat multi-path fading and shadowing. Accordingly, transmission power can be reduced, the data rate can be increased, and the transmission range can be extended.

In most cases, sensor nodes in the proximity of the sink node rapidly die due to frequently relaying packets from the other nodes in the network. In battery-powered WSNs, this unbalanced energy consumption eventually leads to network partitioning. In other WSNs powered by energy harvesting devices, access to sink node may be constrained therefor. The authors in [JWI2011] apply CT range extensions to extend network lifetime by exploiting the energy of less-burdened nodes. Then, data packets are "hoped over" the more burdened or "bottleneck" nodes. Accordingly, duty cycling of nodes over the entire network is balanced, as normal relay sensors can be replaced by other cooperative nodes.

Jayaweera et al. [Jay2006] compared the energy consumption of both SISO (Single Input Single Output) and virtual MIMO (Multiple Input Multiple Output) systems and indicate that MIMO systems can provide better energy savings and smaller end-to-end delays over certain transmission range distances, even with the extra overhead energy required for MIMO training. In [AE2008], the authors investigate the optimal number of cooperating nodes per hop to minimize the end-to-end total energy consumption while satisfying an outage probability requirement at each hop. Finally, Sadek et al. [SYL2010] study the energy gain of utilizing CT schemes in WSNs. The results show that below a certain threshold of the distance between the source and the destination, the use of CT methods is not recommended. Otherwise, CT methods can achieve a better energy gain.

Multi-Path Routing

In fact, multi-path routing enables energy to be balanced among nodes by alternating forwarding nodes. The underlying idea is to determine the k-shortest forwarding paths to the sink node. Afterward, a single route is chosen that minimizes the energy consumption. Radi et al. [RDBL2012] provide a comprehensive analysis of the most recently proposed multi-path routing protocols for WSNs. Ming et al. [MLWSW2007] exploits the path diversity provided by the multi-path routing approach to prolong network lifetime. This gain is achieved by distributing network traffic over multiple node-disjoint paths using a cost function depending on the energy levels and hop distances of the nodes. Accordingly, it allocates the traffic rate to each selected path. Alternatively, Yahya and Ben-Othman [YBo2009] presented the REER protocol; a robust and energy efficient multi-path routing protocol. The REER protocol uses the residual energy, the node available buffer size, and the signal-to-noise ratio to predict the next hop through the paths construction phase. To sum up: multi-path routing protocols are energy-efficient relative to the single-path protocols. However, the incurred burden of handling and examining multi-paths expand the computational overhead. Therefore, simplifying those methods may be considered as an open research issue.

Clustering Architectures

These protocols are suitable in case of continuous transmission due to the presence of redundant data. Specifically, this approach is based on splitting the network into groups called *clusters*. In each cluster, one node is elected as a *cluster head* (CH) which aggregates the packets from its cluster members. The CH node is responsible for coordinating the members' activities and communicating with other CHs or the base station. This architecture limits energy consumption as follows [LLC+2013]:

- it reduces the communication range inside the cluster which requires less transmission power,
- it limits the number of transmissions thanks to fusion performed by the CH,
- it reduces energy-intensive operations such as coordination and aggregation to be done by the cluster head,
- it enables to power-off some nodes inside the cluster while the CH takes forwarding responsibilities, and

- it balances energy consumption among nodes via CH rotation.

Additionally, cluster architectures also improve network scalability by maintaining a hierarchy in the network. Optimizing the collected information might be accomplished by the cluster head in order to decrease energy and traffic. Examples of hierarchical clustering include *low-energy adaptive clustering hierarchy* (LEACH), *power-efficient gathering in sensor information systems* (PEGASIS), and *adaptive periodic threshold sensitive energy efficient sensor network protocol* (APTEEN) [RBC2014].

3.5.3. Sleep/Wake-up Protocols

Sleep/wake-up protocols exploit network redundancy to extend the network longevity by switching a number of redundant SNs into sleep mode. Radio transceivers, in most cases, consume the majority of the energy available. Hence, switching the transceiver into sleep mode helps to greatly prolong the network lifetime. Specifically, active SNs can be switched off according to the workload.

Deactivation Schemes

This section discusses the main sleep/wake-up schemes implemented as independent protocols on top of the MAC protocol (i.e. at the network or the application layer). As depicted in Figure 3.13, independent sleep/wake-up protocols can be further subdivided into three main categories, including *on-demand*, *scheduled rendezvous*, and *asynchronous schemes*. Explanations of these categories and examples are given below.

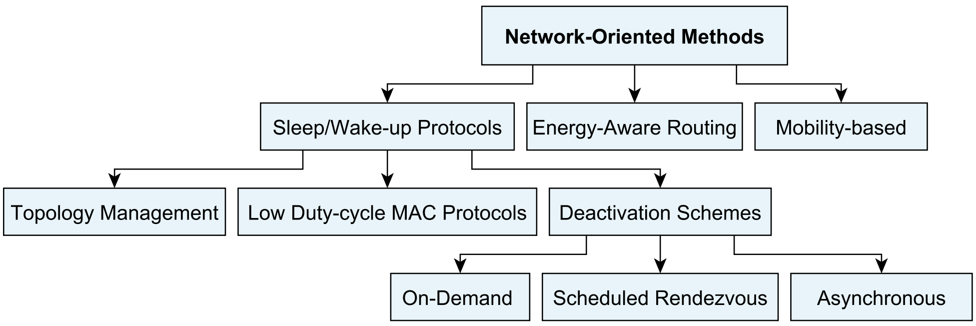

Figure 3.13.: Classification of deactivation schemes for low duty-cycle operation

Scheduled Rendezvous The key idea is to achieve a synchronous activation for a SN and its neighbors. Initially, SNs wake up according to a wake-up schedule, and they remain active until communication with their neighbors is finished. Afterward, they are deactivated until the next rendezvous time. Time synchronization among the neighboring SNs is generally assumed. Keshavarzian et al. [KLV2006] present a multi-parent scheme which assigns forwarding nodes with different wake-up schedules, as depicted in Figure 3.14. The active neighboring SNs have to be partially overlapping to allow SNs to communicate with their neighbors. This method takes a cross-layer approach and exploits the existence of multiple paths between the SNs to improve the energy-efficiency of the wake-up process while meeting the message delay constraints.

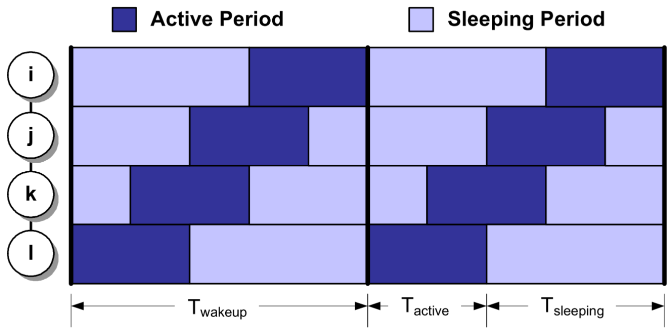

Figure 3.14.: Staggered sleep/wake-up pattern [KLV2006]

Asynchronous Methods Asynchronous protocols enables each SN to wake up whenever it wants and still be able to communicate with its neighbors. In such a scheme, no explicit information exchange is required among the neighboring SNs. Although asynchronous methods are simpler to implement, they are not as efficient as synchronous schemes, and in the worst case their guaranteed delay can be very long. Paruchuri et al. [PBD+2004] present a randomized approach, referred to as RAW, to address the protocol design issues of asynchronous wake-up mechanisms. The RAW protocol enables each SN to make local decisions on whether to sleep or to be active. It allows the existence of several paths between a source and a destination and, thus, a packet can be forwarded through any of such available paths.

On-Demand The main idea behind on-demand protocols is to deactivate a SN and then switch it on only when another SN wants to communicate with it. The main challenge is how to timely trigger the sleeping SN whenever other SNs are willing to communicate. In particular, multiple radios with different energy/performance tradeoffs are utilized. In other words, a low-rate and low-power radio can be dedicated for signaling, and a high-rate but more power hungry radio can be used for data communication. Schurgers et al. [STS2002] introduce STEM, a topology management technique that trades off power savings versus path setup latency. The proposed technique consists of a separate radio operating at a lower duty cycle. Upon receiving a wake-up message, it turns on the primary radio, which takes care of the regular data transmissions.

Topology Management

In many cases, SNs are deployed with a high level of redundancy to ensure space coverage and to cope with possible node failures occurring during or after the deployment. The idea behind topology management protocols is to deactivate some nodes while maintaining network coverage and connectivity. The decision of either activating or deactivating nodes typically depends on the application's needs. Accordingly, topology management protocols dynamically modulate the WSN topology for the sake of minimizing the number of active nodes, hence, prolonging the network lifetime.

Choosing the active SNs can be accomplished in two ways, using a *location-based approach* or a *connectivity-based approach*. In the former procedure, the sensing field is divided into cells as shown in Figure 3.15. In each cell, a single SN is activated while others are switched to sleep mode [BPC+2007]. Consequently, power consumption and collisions are reduced. The latter procedure determines the minimum number of nodes that still guarantee network connectivity. Redundant SNs are deactivated [LY2006].

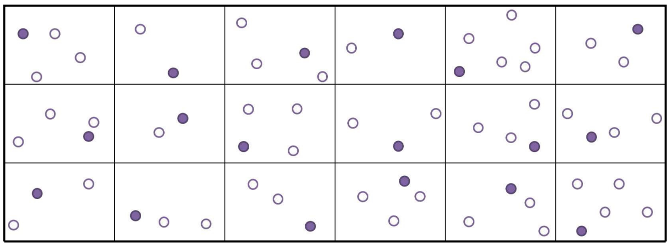

Figure 3.15.: Location-based duty cycling [BPC+2007]

Low Duty-Cycle MAC Protocols

MAC protocols are responsible for the coordination between neighbors. Optimizing MAC protocols leads to significant reduction in power consumption. For instance, *time division multiple access* (TDMA) is a well-known MAC protocol. Data collision is avoided by dividing the time frame into slots where each node is assigned two fixed time slots for transmitting and receiving packets. As a result, SNs are active during their assigned slots and inactive during other slots. Advantages of the TDMA protocol comprise eliminating data collision and conserving significant amount of energy. However, this technique requires a precise synchronization among the various nodes which may be difficult in many situations [Cza2006].

Alternatively, contention-based MAC protocols allow nodes to independently access the shared wireless medium [Anw2006]. These protocols propose minimizing collisions rather than avoiding them completely. Contention-based protocols depend on a carrier sensing mechanism called *carrier sense multiple access* (CSMA). In this mechanism, transceivers are switched on only for listening to the traffic before broadcasting in order to check the availability of free channels [ACDFP2009]. Finally, hybrid protocols dynamically modify the protocol behavior to the contention's level. They behave as a contention-based protocol when the level of contention is low, and switch to a Time Division Multiple Access (TDMA) scheme when the level of contention is high.

3.5.4. Discussion

The aforementioned methods deal with saving energy at the network-level. Although all these methods provide solutions to the networking overhead, negative impacts on other service qualities may emerge. Table 3.5 summarizes the discussed methods and their influences on other service qualities. As an example, deactivation schemes strive to minimize the network duty cycle to overcome idle listening and redundancies. Nevertheless, this action increases the communication delay where each node has to wait for its neighbors to wake up and only then it can start broadcasting. Even the synchronized version of these algorithms may suffer from latency due to time deviations and additional computational overhead of the synchronization algorithm.

Table 3.5.: Summarizing the various network-oriented energy saving methods

Method	Explanation	Influenced QoS	Reference
Mobility	employing mobile sinks or mobile relay nodes to reduce the number of multi-hops	latency	[DD2012, SSB2014, JSB+2006, SG2009]
Relay Node Placement	adding few relay nodes with enhanced capabilities	complexity	[Cir2011, EV2006, MMH2011]
Energy Metric-Based	considering the consumed energy as a criterion in the path setup phase	throughput, latency	[LRL+2012, AY2003]
Cooperative Communication	using several single-antenna devices to improve the received signal's quality	complexity	[LTW2004, JWI2011, Jay2006, SYL2010, AE2008]
Multi-Path Routing	alternating the forwarding paths for balanced energy distribution	computational overhead	[RDBL2012, MLWSW2007, YBo2009]
Clustering Architectures	splitting the network into groups for limiting multi-hops	latency	[LLC+2013]
Deactivation Schemes	sleep/wake-up schemes implemented as independent protocols on top of the MAC protocol	connectivity, coverage, latency	[KLV2006, ACDFP2009, PBD+2004, STS2002]
Topology Management	deactivating some nodes while maintaining network coverage and connectivity	latency	[BPC+2007, LY2006]
Low Duty-Cycle MAC Protocols	reducing the sensors' duty cycle to eliminate idle listening	throughput, latency	[Cza2006, Anw2006, ACDFP2009]

3.6. Discussion

Through this chapter, we discuss the main energy-efficiency methods in WSNs. The underlying idea behind each method has been clarified. Additionally, we provide a new classification for these reported methods. We conclude that many efforts have been exerted in this field of energy-efficiency in WSNs. Nevertheless, many of the reported methods did not consider the energy-QoS trade-offs. Therefore, the proposed energy-centric MOO optimization significantly contribute to the WSN literature.

In the remainder of the thesis, novel methods are proposed with a new concept which calls for saving energy and simultaneously improving the provided service quality. Chapter 4 discusses the implementation details of local adaptive data compression algorithm. In this field, the energy consumption as well as the recovery accuracy and the reporting delay are all considered. Similarly, Chapter 5 investigates the energy burden of energy-hungry sensors. In this arena, event-miss probabilities, detection delay, and energy consumption are considered. Finally, Chapter 6 discusses a novel strategy for QoS control and performance improvement, referred to as lifetime planning. With lifetime planning, a self-adaptive mechanism controls the relation between energy consumption and other service qualities.

4. Fuzzy-based Sensor Data Compression

4.1. Introduction

In this chapter, the first category of time-based WSNs scenarios is targeted. As earlier declared for this category, transmissions are scheduled either beyond sampling immediately or periodically. A question may arise here about why acquiring a data history could be wholesome. Generally, WSNs capabilities open the door for diverse applications in which sensors are to continuously monitor phenomena. In many situations, sensor data is useful not only for its present use in some application, but for its potential future utility as well. For instance, wireless medical sensors track patients' status and location. Accordingly, databasing this data is crucial [MFjWM2004]. Interesting events such as earthquakes, ecological disasters, and volcano eruptions are scarce, and hence the value of data collected about them is significant. Scientific applications such as bird observation, ocean water monitoring, and glacier monitoring [RM2004], aim at tracking the behavioral changes over long term. Thus, preserving these estimations is of profitable concern.

Accordingly, the energy tank will be rapidly depleted thanks to the additional processing, storage and communication overhead. In such a category, radio communication is considered as the dominant factor of energy consumption. To prove this claim, an example was given in [BA2006]. The authors investigated the power consumption of a simple 32-bit addition instruction and transmitting one bit. Despite, they utilized a Compaq Personal Server for data collection rather than a SN; their findings can provide conspicuous insights of processing and transmission power dissipation. Specifically, an amount of 0.86 nJ was consumed for processing one ADD instruction. Therefore, transmitting a single bit is roughly equivalent to performing 485-1267 ADD operations depending on the quality of the network link. From the aforementioned discussion, it is concluded that performing more processing could significantly reduce the communication overhead and hence extend the node's lifetime.

One intelligent solution is to move the burden from the limited-resources sensors to base stations through data compression. Data Compression is a genuine approach for encoding the readings into smaller vectors. Initially, it reduces transmit-rates in terms of bits per seconds. Additionally, it alleviates the burden of network flooding with superfluous radio traffic. As a result, probabilities of packet collision and retransmission requests are tremendously reduced. In this work, a novel localized data compression technique based on the so-called Fuzzy transform is proposed. Through evaluations, the proposed technique is examined using real environmental data. Moreover, the novel *fuzzy transform compression* (FTC) is contrasted to a *lightweight temporal compression* (LTC) method [SGOea2004].

Despite the acceptable recovery error of the Fuzzy transform compressor, further improvement of the application-relevant QoS such as the recovery accuracy and the reporting delay is sought. For the accuracy, two methods have been examined. The first attempt is – a data-centric approach – extrapolated from the fact that the Fuzzy-transform resembles a low pass filter. Accordingly, removing fluctuation by quick sorting could highly reduce the losses. Nonetheless, the approach is found to be limited due to the "back-sorting" overhead. Afterward, the proposed strategy to improve the FTC precision is altered from avoiding fluctuations to tracking them. Thus, a novel version of the FTC, referred to as *fuzzy compression adaptive transform* (FuzzyCAT) is presented. Its core idea is to adapt the transform parameters to the signal's curvature inferred from the time derivatives. The chapter comprises a variety of simulations and real experiments with Telosb sensor nodes. These evaluations aim at comparing the performance of FTC, FuzzyCAT and LTC in terms of time/space complexity and energy consumption. For data recovery precision, the results show the superiority of FTC and FuzzyCAT over the LTC method for compression ratios (CRs) above 50. Moreover, FuzzyCAT saves 96.07% of the radio energy consumption. The main FuzzyCAT's disadvantage lies in the incurred delay due to additional processing time. However, a novel cooperative-prediction mechanism is proposed to resolve the latency problem.

The contributions of this chapter are summarized as follows.

- refining the FTC via tracking the signal curvatures (FuzzyCAT),
- presenting comparative analysis of the FTC and the FuzzyCAT techniques with the well-known LTC approach,
- validating the excel of FuzzyCAT via simulations and real experiments,
- proposing an inter-sensor prediction scheme for mitigating the impact of FuzzyCAT delay.

The remainder of this chapter is organized as follows. Section 4.2 summarizes the recent efforts in data compression for WSNs. Section 4.3 briefly explains Fuzzy transform (F-transform) preliminaries and its modality for sensor data compression. Section 4.4 explains refining the recovery precision via data manipulation, and the FuzzyCAT technique. The cooperative prediction scheme is also elaborated. Moreover, plenty of performance evaluations, outlining the merits and flaws, are presented. Finally, Section 4.5 concludes the chapter, and discusses the main ideas and possible extensions.

4.2. Related Work

In general, data compression methods have been classified into *lossless* and *lossy* approaches. The former has zero recovery error with relatively small compression ratios, which makes it suitable for applications requiring high precision like patient monitoring in health care. Lossy compression methods, on the other hand, incur recovery errors but achieve higher compression ratios. Numerous applications tolerate limited precision reduction, like environmental monitoring and other types of data logging. In such scenarios, lossy compressors provide an excellent solution, delivering high compression ratios with tolerated information loss. Furthermore, lossy algorithms tend to be less complex than their lossless counterparts, hence are easier to implement on the computationally constrained sensor nodes. In this work, the lossy compression algorithms suited for WSNs is under focus. The lossy compression methods can be generally further classified into:

- *transform-based techniques* such as the discrete cosine transform (DCT), the discrete Wavelet transform (DWT), etc.,
- *model-based techniques* like derivative-based prediction (DBP), lightweight temporal compression (LTC), etc.,
- *compressive sampling* (CS) [RWC2014].

The first category typically consists of transformations that map data on a space where computation is simpler. Given the temporal correlation of sensed data, most resulting coefficients approach zero and are discarded, so the mapped space can be easily entropy-coded [DBF2007]. Many techniques exploit the cooperative nature of the network to run distributed compression algorithms [GEH2003], [WHBD2005]. Such an approach takes advantage of the signal's spatial correlation since neighboring nodes cooperate in compressing their presumably correlated data. However, as the authors of [DBF2007] note, the data gathered by nearby nodes may not necessarily be correlated. Moreover, by employing both, a linear transform and a lossless entropy-based compression of resulting coefficients, such algorithms end up being quite computationally complex which is not suitable for WSNs. Additionally, they may increase the inter-node radio communication [ARAI2013].

In the second category, the readings are approximated by a simple mathematical model. The most well-known example of such a technique is *lightweight temporal compression* [SGOea2004]. Targeting environmental applications such as temperature, humidity, and light sensing, the algorithm exploits the signal's high temporal correlation to approximate it by a sequence of line segments (cf. Figure 4.1). Information loss is controlled by a user-set error margin: whenever the approximating line deviates from the next data point by more than the error margin (gray area), the current line parameters are sent and a new approximation is started. We note that the authors of [RCM$^+$2012] utilize the same idea in designing derivative-based prediction (DBP) modelling - a greedy algorithm that linearly approximates the signal - although their work is not data compression per se. The key difference between LTC and DBP is that LTC transmits the parameters of a line once it has exhausted its approximating potential, whereas DBP sends out the line parameters (the model) immediately after the learning phase and waits until the model adheres to the "incorrectness" definition to compute a new one. Admittedly, both algorithms proved to be successful by finding application in real WSN.

Compressive sensing is a novel method which displaces the traditional mantra of "sample then compress" with "compress while sampling". Its core idea is to sample below the Nyquist rate and then use numerical optimization methods to recover full-length signals from a small number of randomly collected samples [RWC2014]. For the CS

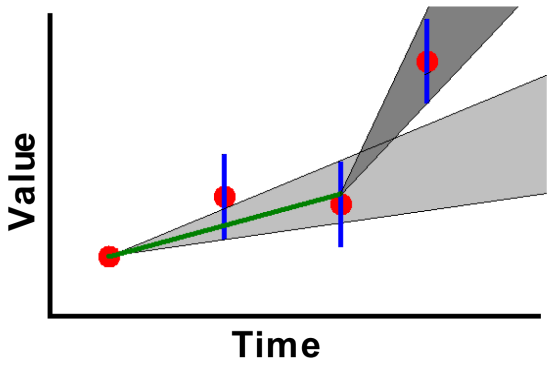

Figure 4.1.: The idea behind the LTC compression method [SGOea2004]

method, the computational burden is customarily transferred to the sink during signal recovery, and not on-mote during signal compression. Despite this advantage of the CS method, its conditions – such as the data sparseness and the unlimited power supply of gateways are not always met. Figure 4.2 depicts a realistic comparison between various compression algorithms including the K-run-length encoding (KRLE), the LTC method, the DWT method, the run-length encoding (WQTR), and the low-pass filtered fast Fourier transform (FFT). As it can be seen, the different methods result in approximately the same battery lifetime. However, the CS and the FFT methods have an advantage of avoiding negative values.

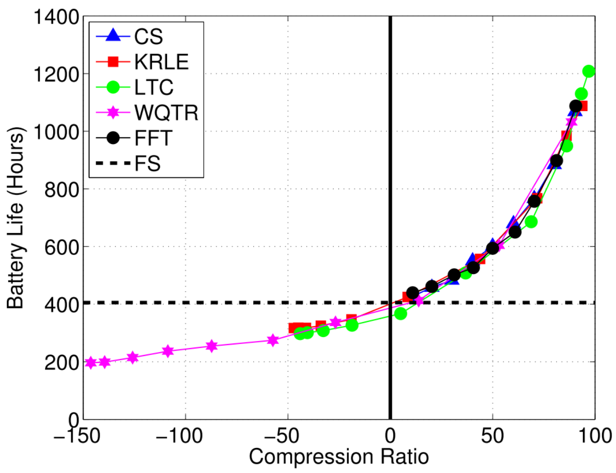

Figure 4.2.: The SN lifetime running various compression algorithms [RWC2014]

In the context described above, the proposed work lies in the realm of lossy compression at individual source nodes. Therefore, to give a fair point of reference, the proposed algorithm is compared with that of the LTC method, as the original and the more well-known version of data linear approximation. Below, the theoretical aspects of the F-transform and how to use it for compression during run-time operation are discussed in details.

4.3. Fuzzy Transform

The foregoing discussion highlights the significance of data compression for extending WSN lifetime. In this section, the concept behind the F-transform and its application in data abstraction are introduced.

4.3.1. Preliminaries

Primarily, data transforms convert an original signal into a special space promoting naïve manipulation. Accordingly, the *fuzzy transform* can be characterized as a fuzzy set mapper of a continuous or a discrete function into an n-dimensional vector [Per2004]. Presume a time series is confined into an interval $\phi = [a, b]$ as universe. This domain is fuzzy-partitioned by Fuzzy sets given by their basic function. This function can be defined as follows.

Definition 1. *Suppose uniformly distributed points $x_1 \leq \cdots \leq x_n$ within ϕ such that $n \geq 2$. The fuzzy sets $A_1, \cdots, A_k, \cdots, A_n$ are referred to as a uniform basic function whenever they conform to the following conditions for $k = 1, \cdots, n$:*

a) $A_k : [a, b] \to [0, 1], A_k(x_k) = 1$

b) $A_k(x) = 0$ *if* $x \notin [x_{k-1}, x_{k+1}]$

c) A_k *is continuous over* ϕ

d) A_k *rigorously increases on* $[x_{k-1}, x_k]$ *and rigorously decreases on* $[x_k, x_{k+1}]$

e) $\sum_{k=1}^{n} A_k(x) = 1 \quad \forall\, x \in [a, b]$

Conditions (a)–(c) state that a basic function has to cover the entire universe in a continuous manner. An example of a uniform triangular basic function is given in Figure 4.3. As can be shown in the figure, each basic function is composed of a number of Fuzzy sets. Each Fuzzy set has to be increasing through a half period, i.e. $[x_{k-1}, x_k]$, and has to be decreasing during the following half period, $[x_k, x_{k+1}]$, as stated in Conditions (b)–(d). Moreover, Fuzzy sets are zeroed at equidistant nodes given by Equation 4.1. The red line delineates condition (e) where summation of any two vertical points should equal one.

Generally, the shape of the basic function forges the approximating function. Hence, the F-transform is well-suited for dealing with linear and non-linear sensor readings. Moreover, the basic function characteristics, such as their shape and length, devote a fine-grained control over the recovery process. The *direct F-transform* de facto converts the original signal into an n-dimensional vector, where n corresponds to the number of Fuzzy sets applied. The *inverse F-transform*, on the other hand, approximates the original signal utilizing the Fuzzy vector. The F-transform is explicitly defined for discrete as well as continuous functions. Definition 2 formulates the direct F-transform in terms of the uniform basic function.

$$x_k = \begin{cases} a & \text{if } k = 0 \\ b & \text{if } k = n \\ \frac{a \times (n-k)}{n-1} + \frac{b \times (k-1)}{n-1} & 1 < k < n \end{cases} \tag{4.1}$$

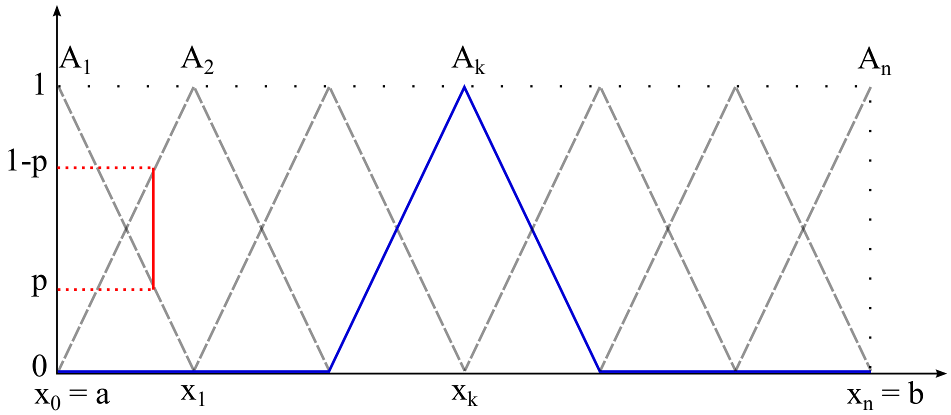

Figure 4.3.: Uniform triangular basic function

Definition 2. *Assume a fuzzy partition of ϕ be given by Fuzzy sets $A_1, \cdots, A_n \subset \phi$ and $n > 2$. If an F-transformer receives a discrete function $f : \phi \rightarrow R$ known at points $x_1, \cdots, x_l$ as an input such that for each $k = 1, \cdots, n$, there exists $j = 1, \cdots, l$: $A_k(x_j) > 0$, then, the n-tuple of real numbers $[F_1, \cdots, F_n]$ is given by*

$$F_k = \frac{\sum_{j=1}^{l} f(x_j)A_k(x_j)}{\sum_{j=1}^{l} A_k(x_j)} \tag{4.2}$$

To understand the F-transform essence, it is necessary to invoke Fuzzy control theory. Specifically, the direct F-transform resembles the defuzzification process (Center of gravity) through which linguistic variables ("low", "medium", "high", etc.) are mapped onto real numbers. This implies that each vector element F_k constitutes the weighted average of the data points $f(x_j) \in [x_{k-1}, x_{k+1}]$. The transformed data can be retrieved by means of the inverse operation, given in Definition 3. The recovery error is defined as the measure of convergence between the original function and the recovered signal. Particularly, the F-transform behaves as a low pass filter where it removes the signal's high frequency components [Ste2007]. The interested readers can find more properties and proofs in [Per2004].

Definition 3. *Suppose a fuzzy vector $F_n[f] = [F_1, \cdots, F_n]$ w.r.t. $A_1, \cdots, A_n$ has been applied to an inverse F-transformer. The recovered signal, defined at points points $x_1, \cdots, x_l$, is given by*

$$f_{F,n}(x) = \sum_{k=1}^{n} F_k A_k(x). \tag{4.3}$$

In the next section, the question of how the F-transform can be applied for lossy data compression in WSNs is answered.

4.3.2. Fuzzy Transform Compression

In the context of WSNs where data communication is extremely costly, compressing the payload data is an efficient technique to reduce energy consumption, radio traffic, and data collision. Algorithm 1 depicts pseudocode of the F-transform compression (FTC) technique. At the outset, the fuzzy sets A_k are delineated as a rising and a falling edge. Their length (*half_period*) is expressed in terms of the number of data points in the universe (ω, assuming a fixed sampling rate) and the number of fuzzy sets (n). The equidistant points x_k are positioned in the light of Equation 4.1. However, the basic function is solely determined once and then stored in a non-volatile memory. The sensor readings $f(x_j)$, $x_j \in [a, b]$, are windowed forming the universe ϕ. A nested loop iterates over a half period to evaluate two successive vector elements. The process repeats until filling the F_k array. For reducing the source nodes' overhead, the fuzzy vector normalization is performed at the sink node.

Algorithm 1 The FTC compression

Require: a, b, ω, and n

Determine *half_period* $= \frac{\omega}{(n-1)}$;
Compute x_k and $A_k(x_j)$;
Record $f(x_j)$ where $x_j \in [a, b]$;
for $k : 1 \rightarrow n$ **do**
    for $j : 1 \rightarrow$ *half_period* **do**
        $F_k := F_k + (f(x_j) \times A_k(x_j))$;
        $F_{k+1} := F_{k+1} + (f(x_j) \times A_{k+1}(x_j))$;
    end for
end for
Transmit the fuzzy vector $F_k = [F_1, ..., F_n]$;

Data recovery at the sink node is straightforward. The length n is inherent in the F_k array size. Algorithm 2 shows the data recovery mechanism which commences with normalizing the received F_k array. Within each half period, two Fuzzy sets are utilized to generate approximated data points.

Algorithm 2 The FTC decompression

Require: a, b, and ω

Calculate x_k and $A_k(x_j)$;
Determine *half_period*$= \frac{\omega}{(n-1)}$;
Receive $F_k = [F_1, ..., F_n]$
Normalize F_k by area$(A_k(x_j))$
for $k : 1 \rightarrow n$ **do**
    for $j : 1 \rightarrow$ *half_period* **do**
        $f_{F,k}(x_j) := F_k \times A_k(x_j) + F_{k+1} \times A_{k+1}(x_j)$
    end for
end for

A series of experiments on authentic sensor data acquired by the Berkeley Lab in 2005 [IBRL] is utilized to contrast the FTC performance against that of the LTC method. The compression ratio (CR) and the normalized root mean square error (RMSE) are used as metrics for the performance evaluation, as given in Equations 4.5, 4.6, and 4.4.

In particular, the signal, used for the assessment, was measured by the light sensor of node #50. Figure 4.4 represents a segment of time series light intensity data in its original form, as well as two recovered time series utilizing the LTC and the FTC decompressors. As it can be seen in the Figure, the LTC and the FTC methods exhibit similar performance. The compression ratio (CR) delivered by the FTC method is 10 whereas the LTC method compresses the data by a ratio of 9.17. The normalized RMSE is 3.67% for the FTC method, and 3.89% for the LTC method. However, by taking a closer look at the graph of the absolute error (see Figure 4.5), one can notice an interesting trend at the end of the time series segment.

$$normalized\, RMSE = \frac{RMSE}{y_{max} - y_{min}} \times 100\% \tag{4.4}$$

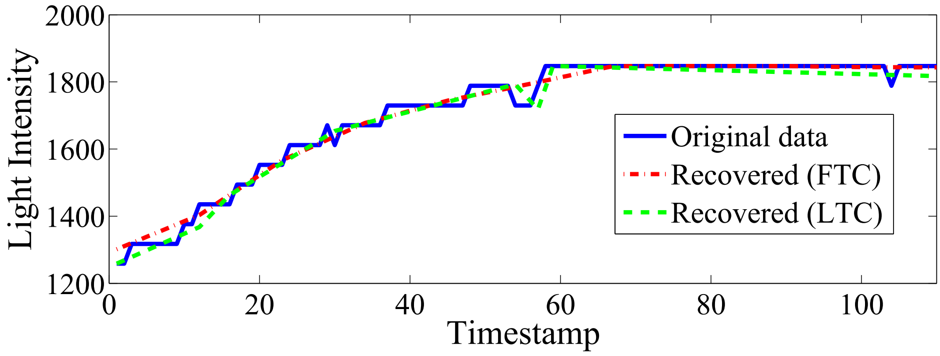

Figure 4.4.: Data recovery with FTC and LTC methods

$$CR = \frac{Uncompressed\ size}{Compressed\ size} \tag{4.5}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - y_{i\,reconstructed})^2} \tag{4.6}$$

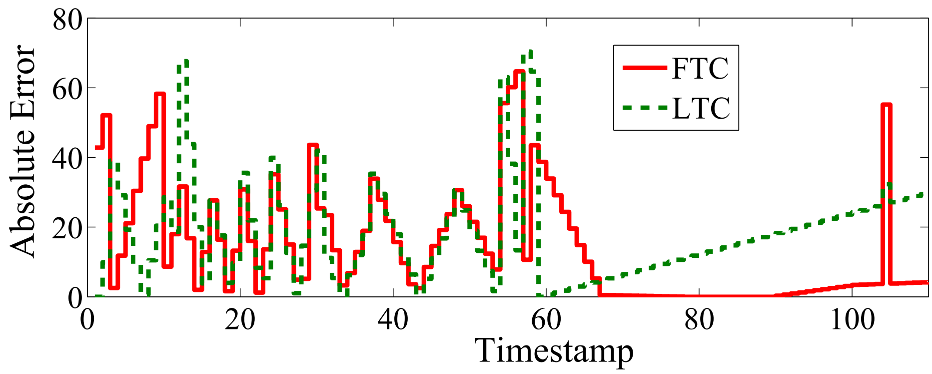

Figure 4.5.: Absolute recovery error of the FTC and the LTC methods

Despite being almost linear data which the FTC method reconstructs with a relatively high precision, the LTC method continuously accumulates the errors. This happens because it is a greedy algorithm and the linear fit it produces is not optimal. The approximating line has a slope that differs from the slope of the actual time series but stays within the error margin set by the user. A similar issue is described in [RCM+2012]. In the meantime, the FTC method does not follow small fluctuations in the signal as closely as the LTC method, but also does not deviate from it, thus avoiding error accumulation. On the other hand, FTC periodicity in data reporting is advantageous since

- neighboring SNs might schedule their reawakening, consequently, idle listening is lessened,
- the impact of missing packets is not critical. In case of LTC, loosing one packet might corrupt the readings for lasting until a new packet reaches the sink node.

However, both methods suffer from the increased reconstruction error around the points with high curvature. The next section introduces the proposed approach for further improving the accuracy of data compression in WSNs.

4.4. Accuracy Refinement

In this section, the proposed data compression algorithm is refined with the goal of enhancing the most significant QoS metric, namely the recovery accuracy. During this work, two methods have been examined, including the data sorting method, and the curvature tracking algorithm. Although the former is found to be not successful due to the non-trivial back sorting, it is convenient to be mentioned as an open research topic. The latter approach achieves smaller recovery error for high compression ratios. Therefore, a fine-grained description of the algorithm, referred to as the FuzzyCAT compressor is provided.

4.4.1. Data Sorting Method

One way to ensure smoothness, thus minimizing the reconstruction error of the FTC on interesting intervals of the signal, is to implement data preconditioning for instance by using the quick sort algorithm. It is known that many lossy compressors act as low-pass filters, only preserving signals of low frequency, or low curvature. Therefore, the idea of sorting the discrete signal is not specific to FTC. It can rather be adapted to other lossy algorithms. In this case, applying a quick sort to the data before compression indeed reduces the RMSE dramatically, since all high fluctuations are removed. However, this comes at a high cost to the overall performance; since the sorted and compressed signal requires a “back-sorting” at the sink node on top of performing the decompression. The array of resorted indices of each data point has to be sent along with the Fuzzy vector. As a consequence of this naïve approach, the compression ratio plummets.

Figure 4.6 depicts the normalized RMSE as a function of the compression ratio where FTC has been applied to different window sizes. Note that the purple data points corresponding to the performance of the transform with sorting, exhibit a strong upward trend already at very low compression ratios. It seems like the resulting curve has an asymptote at $CR = 10$, which makes it highly undesirable compared to regular FTC. The uncompressed array of indices transmitted together with the FT vector completely undermines the compression. That being said, it is maintained that sorting combined with compressing can be a viable technique, especially for less smooth signals. An idea for future work is to implement an injective function between permutations and long integers in which sending the uncompressed array of indices will be replaced by transmitting a single long integer denoting a unique permutation.

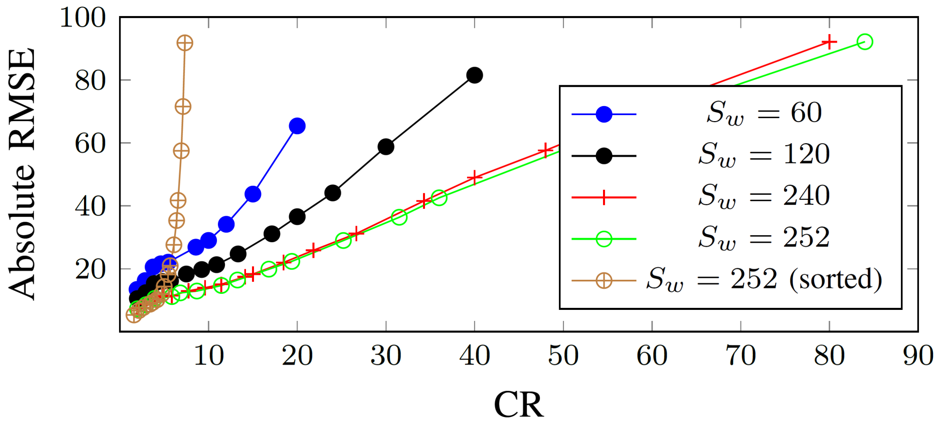

Figure 4.6.: The reconstruction error of regular and sorted data for different window sizes

4.4.2. FuzzyCAT Method

Sorting the data points causes recovery problems. How else can we address the issue of increased reconstruction error around the points with high curvature? A natural way to proceed is to simply increase the resolution of the F-transform by applying narrower Fuzzy sets whenever the signal exhibits high curvature. This is the idea behind the modified FTC, called "Fuzzy Compression: Adaptive Transform" (FuzzyCAT). As aforementioned discussed, Fuzzy vectors comprises weighted average of the original readings over the processing window, denoted by $[a, b]$. Utilizing narrower Fuzzy sets within curvature regions is beneficial, since the resultant averages – by the narrow Fuzzy sets – represent small areas within the curvature. This adjustment leads to a better approximation. *Lemma 3* given in [Per2004] proves this fact as follows.

$$\int_{x_1}^{x_k} f(x)dx = h(\frac{1}{2}F_1 + F_2 + ... + F_{k-1} + \frac{1}{2}F_n) + \mathcal{O}(h^2) \tag{4.7}$$

where $\mathcal{O}(h^2)$ is the local error as a function of the length (h) of the support of A_1 or A_n. Obviously, adding such narrower Fuzzy sets reduces the local error between the original signal and the fuzzy components. Accordingly, the modified F-transform, implemented by the FuzzyCAT, is as follows.

$$A_{Hybrid} = A_k(x) + B_k(x) + C_k(x) \tag{4.8}$$

$$F_k = \sum_{j=1}^{l} f(x_j) \cdot A_{Hybrid} \tag{4.9}$$

$$f_{F,n}(x) = \frac{\sum_{k=1}^{n} f(x_j) \cdot A_{Hybrid}}{\sum_{j=1}^{l} A_{Hybrid}} \tag{4.10}$$

Particularly, the FuzzyCAT technique consists of two stages. The first stage detects the signal's curvature, while the second stage adapts the F-transform and evaluates the Fuzzy vector. Below, these stages are explained in more details.

Curvature Detection

In the series of experiments on Intel Lab Data [IBRL], it became evident that the algorithm yields a higher reconstruction error when the signal has high fluctuations (see Figure 4.4). An easy, low-computation way to detect fluctuations is to monitor the second derivative of the signal, which is the indicator of curvature. To confirm the hypothesis, the absolute error of the reconstructed signal and the first as well as the second derivative of the original signal as functions of time are graphed.

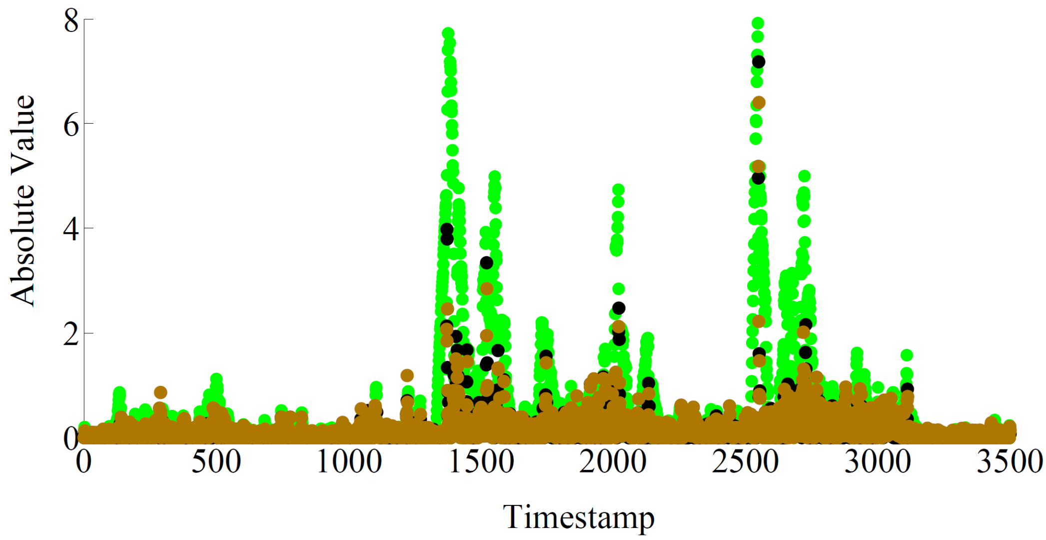

Figure 4.7.: The first and second derivatives as measures of smoothness

Figure 4.7 shows that there exists a certain correlation between the absolute error and the first and the second derivatives. Although it is not true that for all points with high reconstruction error the derivative is high too, but it holds that for all points p with high derivative, the *interval* of the time series around p exhibit an increased error. Thus, a problem arises: environmental data, such as temperature or humidity, is intrinsically smooth, with few abrupt changes. But, when such sudden fluctuations do occur, then they are often of particular interest to the scientists studying the phenomena, and therefore require minimal reconstruction error.

Algorithm 3 depicts the mechanism by which signal curvature is detected. The number ω of data points to be compressed, the base number n of Fuzzy sets to be applied to the

Fuzzy universe, as well as the number e of extra Fuzzy sets to be applied per half period, are set by the user such that (1) $(n-1)|\omega$ and (2) $(e+1)|\frac{\omega}{n-1}$. The data is acquired through iterating over half periods of a basic function. Throughout the process, the program maintains a *meta* array where each cell is set if the corresponding half period requires higher resolution. For each data point, with an obvious exception of the first two, the second derivative $\frac{d^2f(t)}{dt^2}$ is computed. Knowing the high noise level seen in sensed environmental data, it is important to only increase the resolution of the transform if the fluctuations detected are significant. To ensure this, two stages of filtering are applied. In the first stage, the current $\frac{d^2f(t)}{dt^2}$ is compared to the derivative threshold T_{deriv} set by the user. If the threshold is exceeded, then a counter of data points with excessive derivative is incremented. In the second stage, that counter is compared to another user-set threshold $T_{percent}$ representing the maximum percentage of the data points in one half period with excessive derivatives. If that threshold is exceeded too, then the current half period likely contains significant fluctuations. Hence, it is marked as needing higher resolution in the *meta* array. This double threshold approach is influenced by the notion of measurement correctness in [RCM+2012].

Algorithm 3 FuzzyCAT curvature detection

Require: ω and n

Determine $half\_period = \frac{\omega}{(n-1)}$;
Construct $meta[\frac{\omega}{half\_period}]$;
for $i : 0 \rightarrow \frac{n}{half\_period}$ **do**
  for $j : 0 \rightarrow half\_period$ **do**
    Acquire x_k;
    if $k > 2 \wedge (x_k - 2 * x_{k-1} - x_{k-2}) > T_{deriv}$ **then**
      Increment $high\_derivative\_counter$;
      if $high\_derivative\_counter > T_{percent}$ **then**
        Set $meta[i] = 1$;
      end if
    end if
  end for
end for

Transform Adaptation

Once the signal has been assessed on the matter of fluctuations, the modified F-transform is applied. Algorithm 4 commences with constructing two kinds of Fuzzy sets. The first function (A_k) is delineated based on the number of coefficients, the data window needs to be compressed into assuming there are no significant fluctuations. The other function, a narrower one (E_k), is based on the number of extra fuzzy sets to be added in a half period marked as requiring higher resolution. Note that to minimize computation, the Fuzzy sets are only computed once in a node's lifetime and stored away. The program iterates over half periods of the window size and applies the transform choosing the Fuzzy sets based on the information about the current half period recorded in the *meta* array. To ensure that the decompressor distinguishes between coefficients resulting from regular Fuzzy sets and the ones added for higher resolution, the extra coefficients have their sign bit flipped. This way, no further information needs to be transmitted, unlike in the case of data sorting. The obvious limitation of this approach is that it does not work if the signal's range can span both, positive and negative values. But in that case, it is possible to offset the signal with a known constant so that it always remains "on the same side of zero."

Algorithm 4 FuzzyCAT at the source node

Require: ω, n, e, and *meta* array
Compute A_k and E_k
Determine *half_period*= $\frac{\omega}{(n-1)}$;
for $i : 1 \rightarrow \frac{\omega}{half\_period}$ **do**
  if $meta[i] = 0$ **then**
    Compute F_k and F_{k+1} using A_k and A_{k+1}
  else
    Compute $-F_k$ and $-F_{k+1}$ using E_k and E_{k+1}
  end if
end for
Transmit the fuzzy vector $F_k = [F_1, ...]$;

If one were to graph the resulting Fuzzy sets over the whole time window, then one would see something like the graph given in Figure 4.8. The sample signal is shown on top, and the fuzzy sets constructed by FuzzyCAT for that signal are displayed on the bottom. On the half periods where the signal is smooth, the regular Fuzzy sets are applied. In the half period where fluctuations were detected, narrower Fuzzy sets are applied (in blue). Note that to fulfill the requirement (5) of *Definition* 1, the Fuzzy sets adjacent to the high-resolution half period (in red) are asymmetric. They represent so-called "hybrids" because they are constructed using part of a regular and an additional basic function. As a result, the area around the points with high $\frac{d^2 f(t)}{dt^2}$ value is transformed using narrower Fuzzy sets, thus ensuring a higher precision.

Performance Evaluations

Figure 4.9 presents an example of performance comparison between FTC and FuzzyCAT on a segment of the temperature signal from the Berkely lab dataset. Both algorithms aimed to compress the 1000 data points into 26 coefficients, while FuzzyCAT was set to add three additional Fuzzy sets per half period when needed. The scaled pink line, representing the difference between the signal reconstructed by the regular FTC and FuzzyCAT, reveals that the algorithms yielded identical results on most of the segment, only deviating on the intervals with high fluctuations. The FTC yields compression ratio of 38.46, with normalized RMSE of 8.72%. The adaptive transform added 9 extra Fuzzy sets, decreasing the compression ratio to 28.57 and bringing the normalized RMSE down to 4.22%. Adding extra Fuzzy sets cut the RMSE by more than half – a 52% decrease, while the resulting compression ratio was only 25% percent smaller than the original. Thus, FuzzyCAT exhibits a compelling advantage over the naïve F-transform.

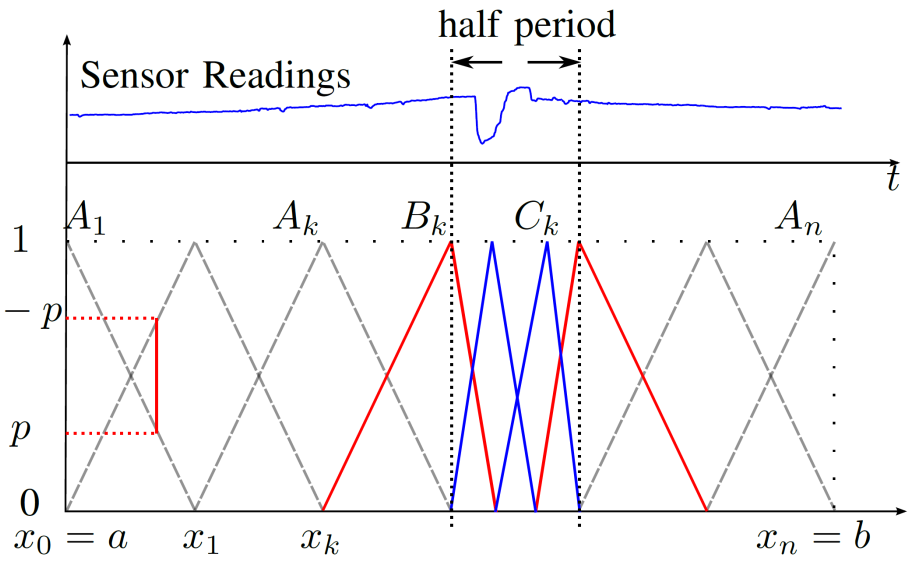

Figure 4.8.: Structure of the adaptive basic function

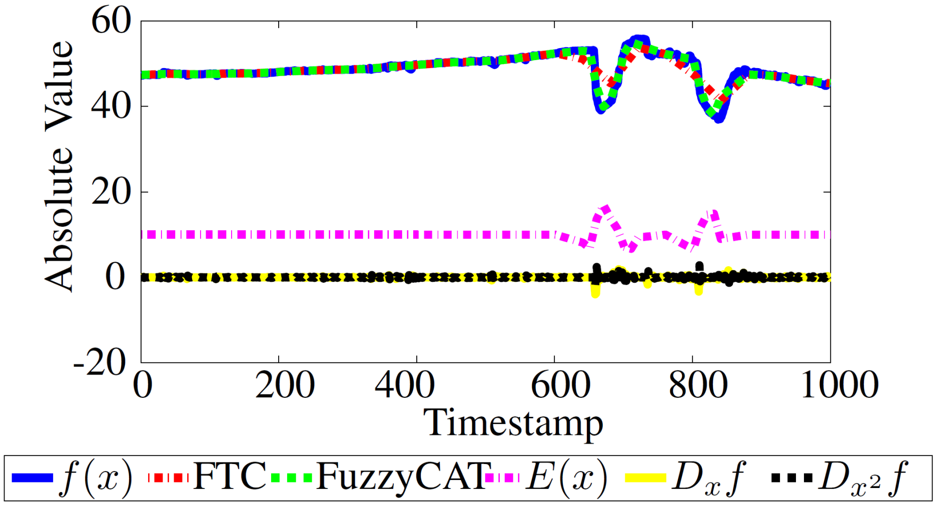

Figure 4.9.: FuzzyCAT outperforms the naïve FTC

Comparing the performance of LTC, FTC, and FuzzyCAT involved compressing and recovering a 1000-point segment of the temperature dataset while varying the parameters of each algorithms: error margin for LTC, number of coefficients for FTC and the number of additional Fuzzy sets per half period for FuzzyCAT. The two remaining parameters of FuzzyCAT – T_{deriv} and $T_{percent}$ were optimized such that the maximum quality factor (QF), defined in Equation 4.11, is achieved (Figures 4.10–4.12) [KPea2013].

$$QF := \frac{CR^2}{RMSE} \tag{4.11}$$

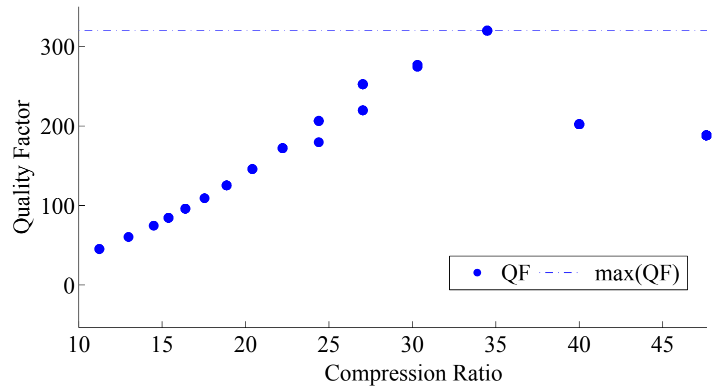

Figure 4.10.: CR optimization

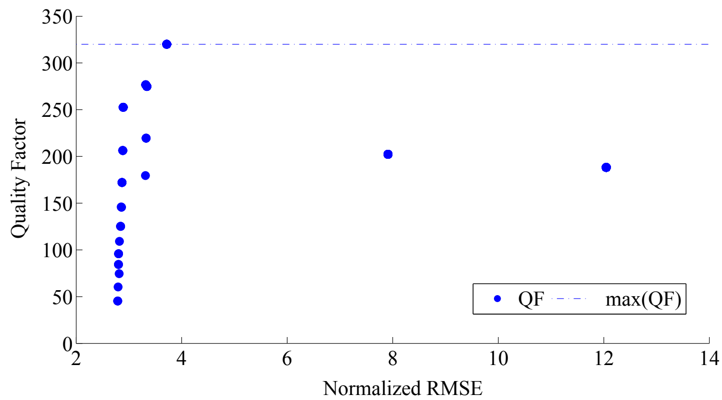

Figure 4.11.: RMSE optimization

Figure 4.10 depicts the process of finding the optimal compression ratio CR_{opt} that maximizes the quality factor. Afterward, this CR_{opt} is used while optimizing the RMSE. Figure 4.11 depicts the quality factor versus RMSE values at a fixed CR $= CR_{opt}$. The RMSE value which satisfy maximum QS is an optimal value E_{opt}. Finally, Figure 4.12 shows how we select values of T_{deriv} and $T_{percent}$ so that FuzzyCAT is correctly adapted. Both of CR_{opt} and E_{opt} have been used to estimate the points shown in Figure 4.12. Accordingly, pairs of thresholds that yield the maximum QF were picked from the curve on Figure 4.12. Figure 4.13 shows the results of the comparison. Note that depending on

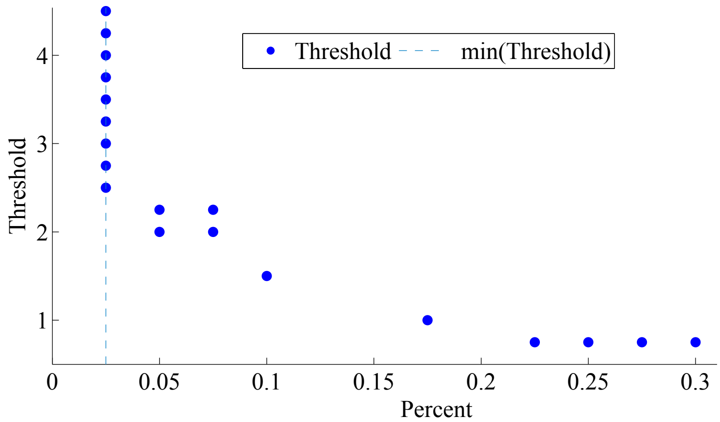

Figure 4.12.: Determining the optimum thresholds for Berkeley lab data

the error margin, LTC can yield different reconstruction errors with the same compression ratio. Obviously, LTC has slightly less error than FuzzyCAT for compression ratios below 50. Beyond this threshold, the LTC error curve sharply increases, giving an advantage to FuzzyCAT.

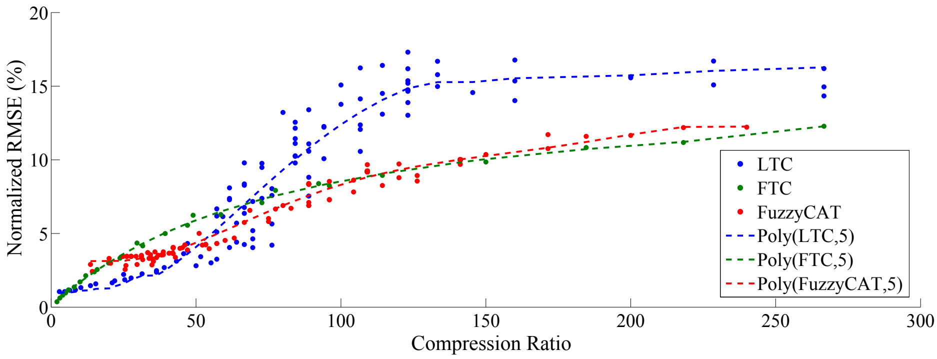

Figure 4.13.: The normalized error versus compression ratio of LTC, FTC, and FuzzyCAT

To evaluate how FuzzyCAT and LTC affect the energy consumption, a series of experiments using TelosB sensor nodes with Contiki OS is run. A network of five SNs was deployed to collect environmental data (temperature, humidity, and light intensity). Half of the sensor nodes were utilizing FuzzyCAT for data compression, while the other half were using LTC. The indoor light level has been frequently changed during the experiment to create data fluctuations. The setup involved a network under ConikiMAC radio duty cycling protocol with the *unicast* communication primitive in the Rime stack. Power consumption was estimated utilizing the *Energest* module in the Contiki OS. For the fairness of the experiment, the parameters of each algorithm were set such that both resulted in the same normalized RMSE.

The LTC nodes sent, on average, 53 packets whereas the FuzzyCAT transmitted solely 11 packets for the same data received at the sink. Figure 4.14a delineates the power consumed via broadcasting the LTC and the FuzzyCAT vectors. In fact, the FuzzyCAT consumes 96.07% less energy than the LTC for a fixed throughput. This significant gain comes together with a similar reduction in processing power consumption as can be shown in Figure 4.14b.

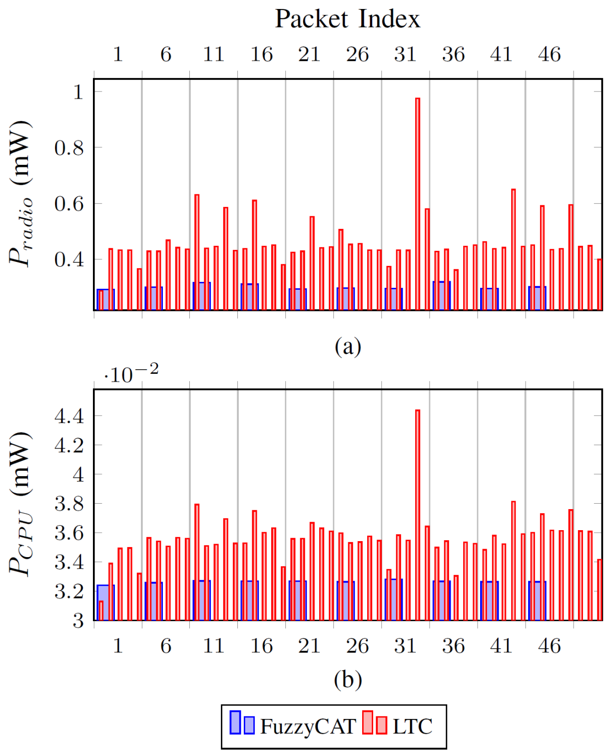

Figure 4.14.: Transmission and processing costs of LTC and FuzzyCAT

Part of the reason why the FuzzyCAT nodes reduced the transceiver activity, was that the algorithm requires conveying a single array of compressed measurements per data acquisition window, whereas the LTC transmits a separate packet for each approximated linear segment. Thus, FuzzyCAT efficiently spreads the overhead involved in sending each packet. This property of FuzzyCAT also results in periodicity of transmissions, unlike the unpredictable nature of LTC's sending patterns. Periodicity of transmissions is valuable because it allows (1) to implement scheduling algorithms thus minimizing idle listening and packet collisions and (2) to easily detect lost packets: the sink expects a packet and sends a NACK message if the packet did not arrive in time. Neither feature can be used with LTC since the packets are sent irregularly [RCM+2012].

Limitations of FuzzyCAT

Intuitively, the proposed FuzzyCAT algorithm has some drawbacks that should be here identified. However, solutions which could highly mitigate the incurred burden of such drawbacks are provided.

- The most obvious disadvantage of FuzzyCAT hides in the fact that the computation of the transform requires iterating through the whole window of data points before sending the compressed vector, this increases the *latency* of measurements. Contrary to DBP where a new model is sent as soon as the signal deviates from the previously established model, FuzzyCAT cannot be used in systems where immediate feedback is required. At each round, the latency is limited to a certain value, and then reset to zero on the next scheduled transmission.

- Since FuzzyCAT records information about the curvature of each half period of the time series window before applying the transform, it needs to store the whole array of uncompressed measurements in the mote's memory. This is problematic, given that most motes have very *limited storage* left after an operating system is installed. During the experiments with CM5000 MSP motes, it is found that compressing over 1000 data points at a time is not trivial without running out of space. However, this problem would not occur if motes with extended memory are used or via dividing the data into smaller blocks.

- Spreading the overhead by sending arrays of values comes at a price of having to respect the *maximum packet size*. Rime, the employed networking protocol, set that limit to 127 bytes, which accounting for the header allowed us to send no more than 28 floats at a time. This issue can be easily addressed by splitting the array of compressed values into smaller sub-arrays in case it exceeds the maximum packet size. This feature is automatically implemented in some other network protocols.

- Another potential difficulty is that the *parameters* of FuzzyCAT, of which there are five, must be carefully chosen to suit the application. For instance, the window size ω, the number n of Fuzzy sets and the number e of extra Fuzzy sets added per half period must satisfy the preconditions; otherwise, integer division results in incorrect compression and higher reconstruction errors. The thresholds T_{deriv} and $T_{percent}$ must also be adapted to the type of signal expected by the application. Fortunately, this fine-tuning needs to occur only before the deployment of the WSN and no further adjustment is required.

- Finally, as mentioned previously, the measurements that include both positive and negative values must be *offset* by a constant for the decompressor to correctly compute the inverse transform.

As one can see, the disadvantages of FuzzyCAT are certainly workable and by no means undermine the overall value of the algorithm, especially in consideration with its beneficial characteristics such as stellar energy efficiency, low complexity, and periodicity. Next, the proposed method for shortening the reporting delay is explained.

4.4.3. Cooperative Prediction Scheme

In this section, an efficient prediction mechanism is set up at the aggregating node for reducing the delay incurred through compression. As it can be seen in Figure 4.15, a general structure of a prediction-based technique is engineered. Assume, a sensor node x_1 sends its compressed readings F_{x,t_1} at time base t_1. The sink node typically has to wait Δt time unit for the next round data at t_2, where $\Delta t = t_2 - t_1$. Instead of waiting, the sink can forecast the missing readings $f(x_1)$ at $t \in [t_1, t_2]$ via exploiting the current readings at t_1. For rounds at $t = 2, \cdots, n$, the predictor parameters are trained via feeding back the prediction error.

Unfortunately, most predictors do not exhibit high stability over time, i.e. long-term forecast between t_1 and t_2 is not trivial. To resolve this limitation, we propose to exploit the spatial data correlation between a sensor x_1 and its n adjacent nodes $\{x_2, \cdots, x_n\}$. Accordingly, we can significantly improve the prediction accuracy. Knowing that data is typically sent to the sink node in an asynchronous manner. Hence, accurately predicting the readings $f(x)$ at $t \in [t_1, t_2]$ – at the sink node – is feasible whenever reported readings from neighbors of x_1 are highly correlated to readings from the node x_1. If this condition is satisfied, then we can use these readings $f(x_2), \cdots, f(x_n)$ to provide a prediction of $f(x_1)$ at $t \in [t_1, t_2]$. Formally, the proposed cooperative prediction scheme is defined as follows.

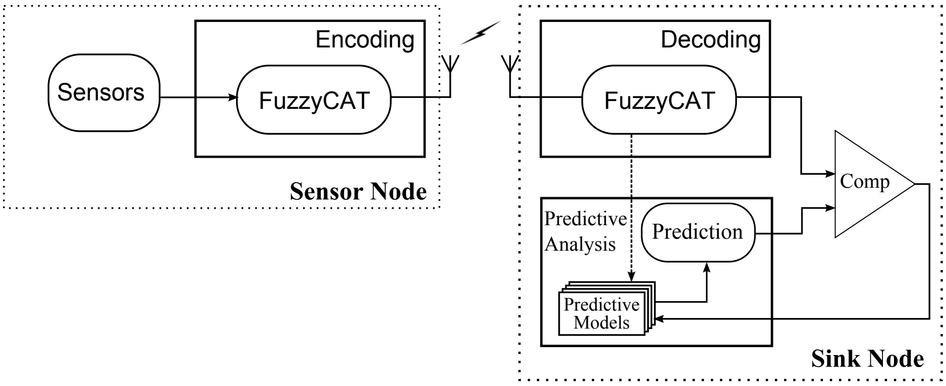

Figure 4.15.: A predictive analysis for an acceptable delay

$$\hat{f}(x_1) = \Psi\left(f(x_2), \cdots, f(x_n)\right) \quad t_1 < t < t_2, \tag{4.12}$$

subject to

$$\rho_{x_1,x_k} < H \quad \forall\, k \in [2, n] \tag{4.13}$$

where $\hat{f}(x_1)$ is the prediction of $f(x_1)$ at $t \in [t_1, t_2]$. The symbol Ψ represents the prediction process which depends on the employed predictor. Finally, Equation 4.13 denotes that cross-correlation ρ_{x_1,x_k} between readings from x_1 and any adjacent node x_k should be within a certain boundary H. Advantages of this method is that

- it does not require additional communications,
- it is independent of the prediction model,
- the computation overhead is done at the sink nodes, where energy is assumed to be affordable.

The applicability of such a method is tested with the real readings extracted from Intel lab (nodes #3, #4, and #6). These nodes are close enough to each other in order to generate highly correlated data. A combination from the three nodes stimulates four predictors, as denoted in Equation 4.12 where $t_1 = 0, \Psi = 40,$ and $\beta = 70$.

For evaluating this study, the predictive analysis of the collected time series is exploited. Initially, prediction can be classified into: 1) *classification*, i.e. forecasting the outcome from a set of finite possible values, 2) *regression*, i.e. forecasting a numerical value, 3) *clustering*, i.e. summarizing data and identifying groups of similar data points, 4) *association analysis*, i.e. extrapolating relationships between attributes, and 5) *deviation analysis*, i.e. estimation of exceptions in major trends or structures [ASB+2014]. In fact, the adopted criteria to select the prediction model were (1) the simplicity to reduce any computational overhead, and (2) the ability to precisely predict times series such as temperature, humidity, and light intensity. Particularly, three different predictors are evaluated, including:

- *Single exponential smoothing* [Kal2004], i.e. it assigns exponentially decreasing weights as the observations get older.
- *Linear regression model* [GLQC2011], i.e. it estimates the forecast by finding the best-fitting straight line through the time series points.
- *Holt-winter smoothing* [Kal2004], i.e. it takes into account seasonal changes as well as trends of the time series. Seasonality can be characterized as the tendency of time series data to exhibit behavior that repeats itself every fixed periods.

As depicted in Figure 4.16, the original temperature values, from node #3 (in blue), and various time series predictions are delineated. Obviously, both of the linear regression and the Holt-winter smoothing predictors approximately follow the envelope of the original data. Although the single exponential smoothing method is examined with two different weights ($\alpha = 0.3$ and $\alpha = 0.7$), it did not exhibit high precision. This could be explained in the light of the fact that such predictors typically do not consider the trend and the seasonality components of the time series. In fact, these predictors were given here to only examine the proposed cooperative prediction method. Nevertheless, other predictors have to be examined in terms of their accuracy and their ability to shorten the compression delay. Finally, it is concluded that the proposed cooperative prediction method is feasible for mitigating the burden of any incurred delay. Actually, evaluating the performance of such an approach in terms of the computational overhead and energy costs are to be considered in possible extensions of this work.

4.5. Discussion

In this chapter, a novel lossy compression algorithm for WSN, called FuzzyCAT is presented. It is designed and implemented in the C programming language for Contiki OS. Three QoS metrics were on focus including, energy consumption, recovery accuracy, and reporting delay. Testing the FuzzyCAT algorithm against the well-known LTC using a network of TelosB sensor nodes revealed that – in this particular setting – FuzzyCAT consumes 96.07% less transmission energy than LTC for a fixed throughput, which is the dominant source of power consumption in WSN. Thus, FuzzyCAT is a very competitive WSN compressor. Besides being more energy-efficient and less computationally complex than LTC, it is characterized by periodicity, a property that increases the resilience to lost packets and makes the algorithm compatible with scheduling protocols. On

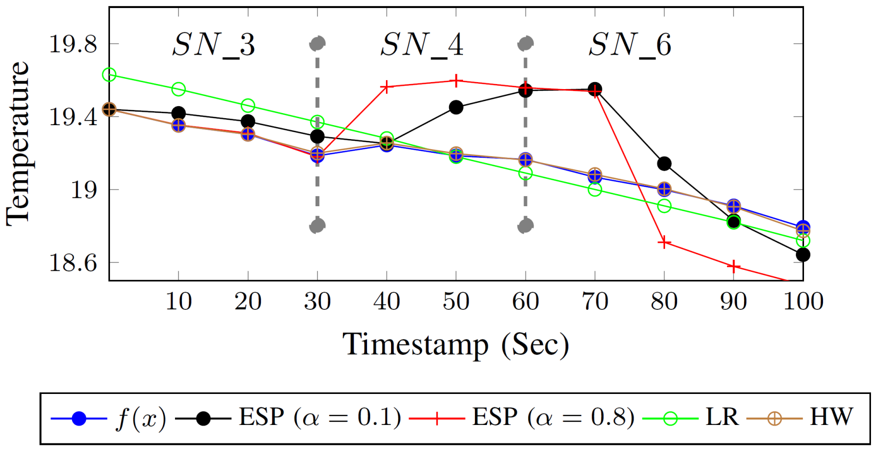

Figure 4.16.: A cooperative prediction of the temperature readings

the other hand, FuzzyCAT possesses such negative qualities as latency, and potentially overusing the limited storage capacity of the mote. However, it is believed that FuzzyCAT is a promising method for data compression in WSN targeting such applications as environmental monitoring and general data logging.

As opportunities for future work, setting up predictors for mitigating the effect of a long delay inherent in the compression process is considered. The viability of data sorting with permutation encoding is another interesting topic worth exploring. It might result in a general enhancement tool for lossy compression algorithms in WSN.

5. Reliable Virtual Sensing

5.1. Introduction

Efficient power management is a key enabler for practical applicability and economical feasibility of WSNs. It obviously is mandatory to ensure the functionality of the WSN till the completion of the assigned tasks. Those tasks may amount to year-long or, in the future, even decade-long data sampling in the field under harsh environmental conditions detrimental to battery capacity. Traditional approaches to extending lifetime include shutting off high-power components, like radio transmitters, for longer phases of inactivity, deep sleep of the whole sensor node between sampling instances, and "smart" forms of sampling to permit shut-offs of reasonable duration amortizing the transient costs of changing power modes. The aim of this chapter is to contribute to this set of techniques for prolonging the WSN lifetime. Simultaneously, the improvement of other application-relevant QoS metrics influenced by such lifetime extension is considered.

In general, sensors are devices stimulated by one property or more from the physical environment. Their expected outputs are normally electrical signals representing the phenomena of interest. From the operational perspective, sensors may either have a *passive* nature, like temperature sensors, humidity sensors, light intensity sensors, and cameras; or an *active* nature, like radars, and light detection and ranging (LiDARs) systems. Table 5.1 lists the energy drawn by some common sensors. As it can be seen in the table, many sensors – such as light intensity and temperature – predominantly consume a negligible amount of energy, compared to radio modules' energy consumption. Other sensing modules, such as GPS and gas sensors, are under no circumstance energy-efficient.

Table 5.1.: Energy consumption of some hardware detectors [AR2008]

Category	**Sensor Type**	**Power Consumption**
energy-efficient	temperature	0.5mW - 5mW
	acceleration	3mW
	pressure sensor	10mW - 15mW
energy-expensive	hall effect sensor	80mW
	soil moisture sensor	36mW - 225mW
	image sensor	150mW
	gas sensor	500mW - 800mW

In a typical sensor node, the amount of energy drawn by each module is typically application-dependent. For instance, environmental monitoring applications usually employ energy-efficient sensors and may require periodic transmission of the collected data. In this setting, radio communication consumes the majority of the residual energy [OR2011]. In other settings, sensing modules may dominantly contribute to the battery depletion, as it may

- utilize active sensors, such as radar and laser rangers, or energy-hungry passive sensors, such as chemical and biological sensors [LzHbGT2011];
- demand high-rate and highly accurate A/D converters, e.g. for acoustic or seismic transducers [APM2005]; or
- prohibit energy-saving sleep modes due to long data acquisition.

Two different examples are given here to support this fact. First, the authors in [KWIea2012] utilized a μ-radar sensor of 175 mA consumption for surveillance. Suppose that the radar is the dominant energy-dissipation source, neglecting other consumers and with a 100% duty cycle. Thus, such nodes will drain their batteries after circa 11.2 hours with an AA alkaline battery of 2000 mAh capacity. Second, Figure 5.1 depicts sources of energy consumption of the H-mote with a Hybus sensor board which contains five sets of air quality monitoring sensors [KCC2008]. Power consumption of both, the NO_2 and VOC sensors, is over 30mW, about 66% of the total energy consumption of the node. Accordingly, more efforts should be exerted to decrease such an overly energy consumption for data acquisition.

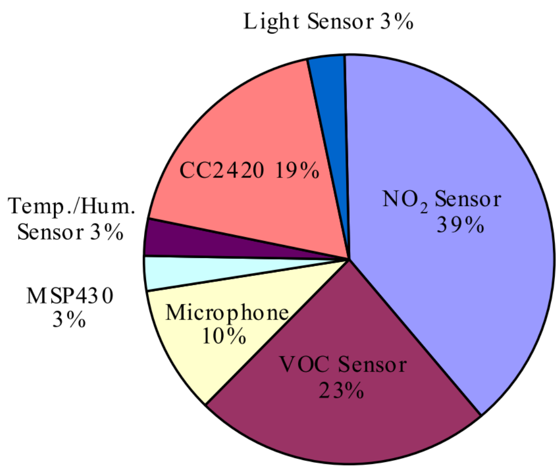

Figure 5.1.: Breakdown of H-mote's energy consumption due to various sensors [KCC2008]

This chapter deals with the latter setting, where the sensing module dominates all other contributions to energy dissipation, such that techniques like sleep modes for radio transmitters or sensor nodes between sensing phases have limited impact. We therefore exploit the concept of *virtual sensing* to mitigate the energy burden of the sensing module due to the aforementioned situations. In order to extend the operational span of WSNs, trading-off energy consumption and average response time, by extending the deactivation periods of sensors with a particularly high energy consumption, is proposed. To compensate for the temporal unavailability of these sensors, alternative, low-power hybrid sensors generate estimates on the probabilities of the occurrence of interesting events, waking up the corresponding main sensor when detection probability justifies it.

For examining the efficiency of the proposed approach, a case study of virtually detecting gas leaks is presented. Moreover, a series of evaluations is performed using *Probabilistic Model Checking*, which is a well-established technique for exhaustively analyzing systems with stochastic and probabilistic behavior in a formal way. To apply probabilistic model checking techniques, one has to model the system as a *Markov Chain* or *Markov Process*, thereafter formally verifying the properties of interest using a probabilistic model checking tool. In this work, the model checker PRISM [KNP2011] is used to evaluate the expected lifetime, energy consumption, and response time obtained by a WSN employing the proposed approach. PRISM is a mature probabilistic model checker. Due to its exhaustive and thus mathematically exact analysis, it produces more accurate results than could be obtained by simulation, which provides statistical figures only. In this work, it is shown how PRISM can be utilized to provide a detailed performance analysis of a gas sensing application, and the results obtained confirm that the lifetime of the sensing system is extended significantly by the novel scheme while retaining an acceptable response time.

Although the WSN lifetime has been extended via utilization of virtual energy-efficient sensors, a question arises about the reliability of such virtual sensors. At a first glance, the replacement of a real sensor $\mathcal{S}_r$ by virtual sensors $\mathcal{S}_v$ appears to be reasonable and simple. However, using n virtual sensors could result in precision shortcomings where a sensing quality set $Q = \{q_1, ..., q_n\}$ may have a negative impact on the detection probability of important events. Specifically, the impacts may be relatively harmful when these replacements consist of an orchestration of heterogeneous sensors like magnetic, radar, thermal, acoustic, electric, seismic, or optical sensors.

Hence, an ontology-based technique is proposed for improving the sensing reliability. Based on design-time knowledge, switching between real and virtual sensors during run-time is feasible via ontology-generated rules. These rules represent the relationship between sensors, their observations and environmental conditions. We demonstrate the applicability of the proposed technique by an example of an object tracking network. The authors in [KWIea2012] designed a network of μ-radars to monitor intruders. Such networks dissipate excessive energy due to the persistent data acquisition. The proposed virtual sensor is composed of seismic sensors, a look-up table of targeted vibration signals (expected intrusions) called, *codebook*, and a pattern-matching algorithm. *Dynamic time warping* (DTW) is selected to contrast the measured vibrations and the codebook.

First, the energy gain of adopting the virtual sensors using a set of TelosB sensor nodes is validated. Then, the ontology-based reliability method is evaluated and the demands of a network of μ-radar and virtual sensors are implemented. Due to the lack of such μ-radars, the proposed method is examined via an event-driven simulator developed for large-scale wireless networks, called *the WSNet* simulator [CFH]: To sum up, this chapter has the following contributions

- the concept of virtual sensing is exploited to highly reduce the energy consumption of energy-expensive sensors,

- the probabilistic model checking is customized to evaluate the profit of engaging virtual sensors into the WSNs,
- the virtual sensing reliability is improved via the sensor network ontology,
- virtual object tracking sensors are designed to substitute the “energy-expensive” sensors
- the virtual sensors’ excel is verified compared to the naïve solutions in terms of energy consumption and accuracy, and
- a benchmark for reliability parameters versus lifetime and event-miss probability is generated via large-scale simulations.

The remainder of this chapter is structured as follows. Section 5.2 briefly discusses the recent endeavors for reducing the sensing module energy consumption. Moreover, previous work utilizing some tools, such as probabilistic model checking and ontologies, in the context of WSNs are discussed. Section 5.3 introduces the concept of virtual sensing where an example of gas leak detection is provided. Next, the problem of virtual sensing reliability is discussed in Section 5.4. Furthermore, the role of ontologies in rule generation to switch between real and virtual sensors is elaborated. Section 5.5 presents a case study of object tracking via a combination of real and virtual sensors. The overall system evaluation in terms of energy consumption and event-miss probability is presented in Section 5.6. Finally, the chapter is concluded and future work is addressed in Section 5.7.

5.2. Related Work

In this section, the state-of-art of the proposed technique and their related tools are discussed. First, the endeavors for reducing the energy consumption of sensing modules are summarized. Next, it is shown that probabilistic model checking has been rarely used in the context of WSNs. Finally, previous work on exploiting a sensor network ontology as a decision making mechanism in WSNs is mentioned.

5.2.1. Low-power Sensing Module

As earlier clarified, sensing modules may consume a considerable amount of energy. According to [ACFP2009], the efforts exerted to reduce the energy consumption of sensing modules have been categorized into *adaptive sampling*, *model-based active sampling*, and *hierarchical sampling*. Adaptive sampling techniques dynamically adapt the sampling rate via exploiting characteristics of data streams, such as the spatio/temporal correlation between the sensed data items. For instance, the authors in [ZZ2012] discussed scheduling sensor nodes (SNs) while maintaining the desired accuracy level. This is achieved by activating a sensor system only at the region of interest for the time needed to acquire a new set of samples and then deactivating the sensors. An alternative approach is proposed in [LAGD2008], where the sampling-related energy-cost rate and the sampling statistics are collected and analyzed to predict an optimal sampling plan. In [YRT2014], the authors developed a framework which adapts sampling rates based on battery level, energy harvesting level, and characteristics of the gathered data. A different approach in [SDC2011] considers exploiting adaptive automata to dynamically vary the interval between data logging.

In model-based active sampling, a model of the sensed phenomenon is constructed. Thus, future values can be predicted with certain accuracy and the acquisition rate can be minimized. The authors in [JC2004] have developed a distributed algorithm based on the Kalman-filter's estimation error to dynamically adjust the sampling rate within a predefined range. Another example is the adaptive sampling approach to the energy-efficient periodic data collection (ASAP) technique which employs a dynamically changing subset of the nodes as samplers [GLY2007]. The measurements of the sampler nodes are collected, whereas the values of the non-sampler nodes are predicted based on these measurements and a model. Probabilistic models that are locally and periodically constructed have been used for producing the forecast.

Although the aforementioned two categories typically reduce the sensing energy costs, they are merely applicable for periodic sampling-based application scenarios. In real-time scenarios, unavailability of the sensors due to adaptive acquisition plans, may lead to the miss of some interesting events. In other words, such methods are not suitable solutions for event-based applications such as object detection, gas leak monitoring, etc. The hierarchical sampling approach has been employed with multi-modal SNs which comprise different sensors for measuring certain phenomena [ACFP2009]. Each of those sensors is characterized by specific performance features, i.e. resolution and energy consumption. Hence, hierarchical sampling adapts the acquisition rate based on a tradeoff between accuracy and energy consumption. In this chapter, a contribution to this category is given via a novel approach, referred to as the *virtual sensing*, to decrease the energy dissipated by those sensing modules while offering minimal event-miss probabilities.

In [KcHA2006], the authors used a similar idea to virtual sensing, called *trigger sampling*, for structure health monitoring and damage detection. Two sensors with different capabilities, m-nodes and μ-nodes, are deployed. The m-nodes are equipped with accelerometers and sample the environment periodically. The μ-nodes are provided with strain gauges and are deactivated for most of the time. When m-nodes detect a problem, then first, they contact their neighbors to cross-check readings. If the check leads to a suspicious problem, then the surrounding densely deployed μ-nodes are activated to get fine-grained information and eventually report the damage.

In fact, the proposed work differentiates itself from previous work through designing the virtual sensor from a set of low-power software and hardware components. Thus, a fine-grained control over the virtual sensor characteristics – such as accuracy, energy consumption, and response time – is provided.

5.2.2. Probabilistic Model Checking

In this work, a case study of virtual gas detection is introduced and its efficiency has been analyzed formally using probabilistic model checking. Regarding the WSN's evaluation, formal methods have been successfully applied. For instance, the authors in [LSPM2012] performed a comparative analysis of clustering protocols with probabilistic model checking. They identified some basic properties of WSNs such as completeness, validity, and consistency, as well as performance-related properties such as energy consumption and lifetime. Furthermore, probabilistic model checking was used for dynamic power management in order to achieve a tradeoff between the performance and power consumption of a system and its components [NPK+2005].

In the proposed work, the hybrid sensing technique and its analysis using probabilistic model checking are concentrated. We regard the energy consumption, lifetime, and response time as costs/rewards in a Markov reward model and pursue a series of exhaustive analyzes on this model, which deal with accumulated energy consumption, lifetime of WSNs, and expected response times for the sensing module.

5.2.3. Sensor Network Ontology

In this work, quality of real and virtual sensors in the light of the changing environmental conditions is assessed. Afterward, ontology-based rules determine the best sensor module which reduces both, energy consumption and event-miss probability. The ontology is a recent method where little research has been devoted to engage it in the context of WSNs. In [APJ2004], the authors integrate an ontology-based mechanism into a WSN to capture the most important features of a sensor node that describe its functionality and its current state. In contrast to the approach stated in [APJ2004], we estimate the current state of a node through modeling the relations between all nodes in the ontology and observing the existing environmental conditions via linked nodes.

The next section discusses in more detail the concept of virtual sensing and how it can be applied to extend the WSNs lifetime.

5.3. Virtual Sensing

The main idea behind virtual sensing is to decompose an "energy-hungry" sensor $\mathcal{S}_r$ into multiple "energy-efficient" software and hardware components, referred to as a virtual sensor $\mathcal{S}_v$. As expressed in Equation 5.1, the signal measured by n hardware sensors $h_1, h_2, \cdots, h_n$ stimulates a lightweight algorithm, denoted by the function $f(\cdot)$. The output of such an algorithm Φ describes a phenomenon which can be directly measured by an "energy-expensive" sensor $\mathcal{S}_r$. The idea here is to design a virtual sensor whose power consumption $P(\mathcal{S}_v)$ is extremely smaller than that of the main sensor $P(\mathcal{S}_r)$. Figure 5.2 illustrates a flow chart of the sensing module with employing virtual sensors. After deploying the sensor motes, the main sensors are activated once to monitor the environment. If nothing is interesting, then the virtual sensor is activated in lieu of the main one. The virtual sensors are continuously polled until an interesting event is detected. Then, the main sensor is again invoked. For the sake of illustration, an example of virtual gas leak detection using a set of energy-cheap sensors is introduced below.

$$\mathcal{S}_v : \Phi \triangleq f(h_1, h_2, \cdots, h_n), \text{ where } P(\mathcal{S}_v) \lll P(\mathcal{S}_r) \tag{5.1}$$

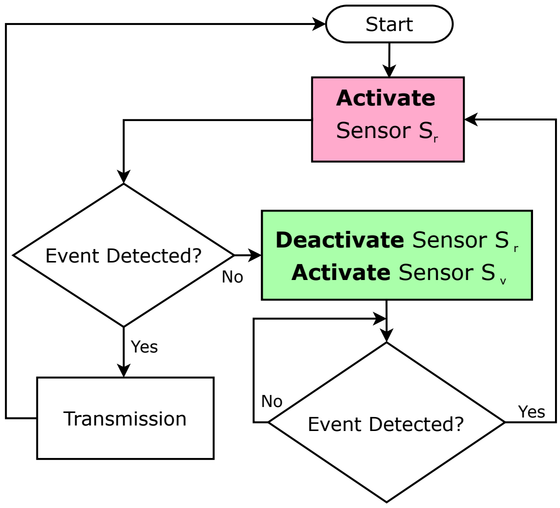

Figure 5.2.: Flowchart of the virtual sensing operations

5.3.1. Example: Virtual Gas Leak Sensors

Specifically, gas leak sensors belong to the category of energy-expensive sensors. They typically consume approximately between 500 and 800 mW on each sample. The authors in [SBS+2011] developed a WSN for detecting combustible gases inside a building. They utilized a pulse heating profile to reduce the sensor energy consumption. Thus, a pulse width modulation (PWM) signal with a 60% duty cycle resulted in 80 mW per sample. Nevertheless, the gas sensor is still "energy-hungry" due to continuous data acquisition. Hence, invoking the concept of virtual sensing here is beneficial to drastically reduce the energy consumption.

The core idea of such virtual gas detectors is to take advantage of a chemical film whose color changes whenever it is exposed to a gas. Figure 5.3 depicts the proposed sensing module where virtual sensing is incorporated. A decision-making algorithm selects between a main sensor S_r and a virtual sensor S_v. In this setup, the sensor energy consumption has been utilized as a single criterion to make a selection decision. Section 5.4 discuss an improved method for designing decision-making algorithms for virtual sensing. On the one hand, a main sensor S_r mainly consists of an "energy-hungry" hardware transducer for detecting gases, as shown in the figure. Gas detections are announced whenever the sensor readings exceed a certain threshold. On the other hand, a virtual sensor S_v consists of three "energy-cheap" sensors (light intensity, temperature and humidity) in conjunction with a Fuzzy logic controller. The light intensity sensor is directly fixed in front of a chemical responsive film for detecting gases. Color of such a film is typically altered with the presence of gases [HBLD2008].

Figure 5.4 demonstrates responses of two temperature sensors and two light intensity sensors to gas leaks during a certain period. This idea is inspired by the work done in [HBLD2008] to efficiently detect gas leaks. As shown in the figure, light intensity was reduced – below 80 lux – due to changing color of the chemical film with gas injection into the experiment chamber. It degraded continuously until an equilibrium point was reached. At equilibrium points, further color changes do not commonly affect

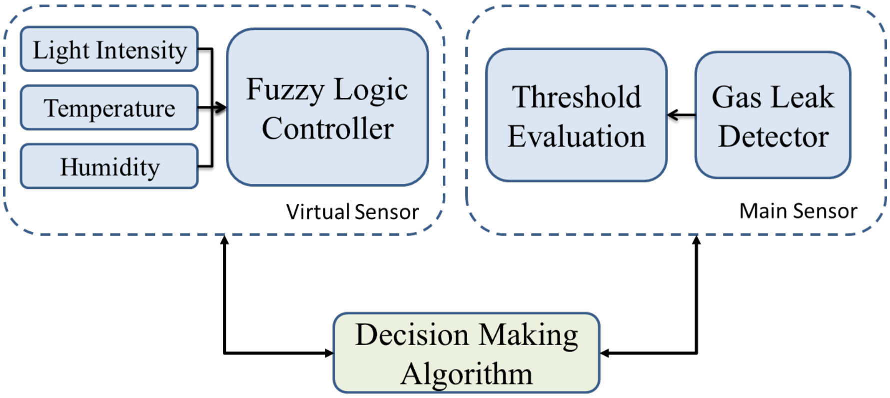

Figure 5.3.: A gas leak sensing module which includes a virtual sensor

the measured luminosity. When the chamber was opened, the confined gas was released which in turn increased the measured light intensity. Temperature was approximately fixed at 20 °C. Notably, light dimming may be due to reasons other than gas detection. Consequently, it is preferable to include temperature and humidity sensors for retaining an overview about the environment, as depicted in Figure 5.3.

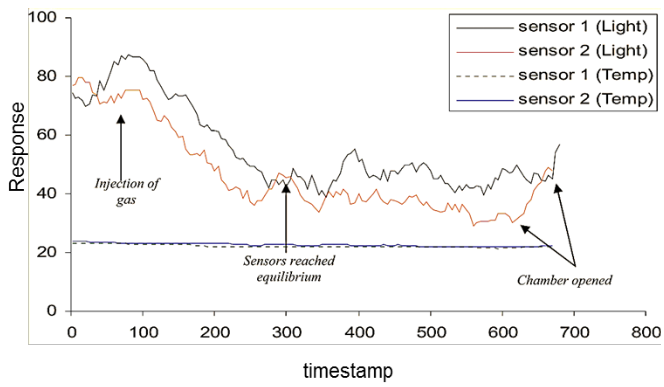

Figure 5.4.: Impact of gas leaks on light intensity and temperature readings [HBLD2008]

For compounding these measurements, a simple *Fuzzy logic controller* (FLC) is devised to estimate the probability of gas leaks. In general, Fuzzy Logic is a multi-valued logic, able to formalize reasoning when dealing with vague terms [Ros2004]. The decisions are not limited to either true or false, or as with Boolean logic either 1 or 0. Figure 5.5 shows the general architecture of a FLC controller. It comprises three main components including *the fuzzifier*, *the inference and rules evaluator*, and *the defuzzifier*. First, the fuzzifier converts the sensors readings into a set of linguistic variables, e.g. low, medium, and high, using triangular Fuzzy sets A_{in} (cf. Figure 4.3). Second, an inference is formed based on a set of empirically developed rules. The output from each rule is aggregated into one Fuzzy variable. Finally, the resulting Fuzzy output is mapped to a probability of detection P_d using Fuzzy sets A_{out}. Whenever such a probability exceeds a predefined threshold $P_d \geq H$, the main sensor is then activated to confirm a gas detection.

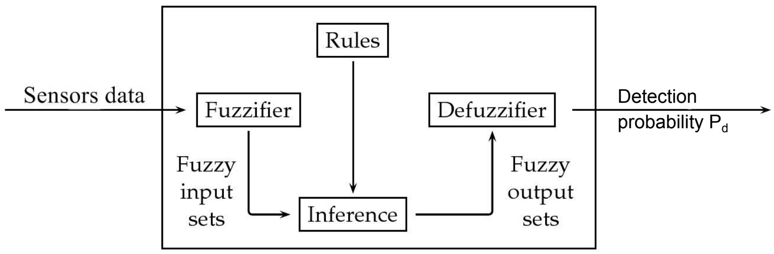

Figure 5.5.: A general architecture of a Fuzzy Logic system

To visualize the Fuzzy rules, Figure 5.6 shows an example of the FLC controller's control surfaces, utilized within the sensor $\mathcal{S}_v$. This 3D graph delineates the input/output relationships, where two certain inputs usually form the graph base, while the output is denoted by the height above each input pair. In the proposed scenario, such a FLC controller has been implemented using the Matlab software. As it can be seen in Figure 5.6, the control surface depicts the control strategy between the light intensity, the temperature, and the detection probability P_d . This strategy was empirically devised, however advanced learning methods are strongly recommended for this task. Therefore, developing realistic rules is left for further investigations. The next section presents more details about the performance evaluation. A probabilistic model checker, called PRISM, has been utilized for evaluating the advantages and disadvantages of using virtual sensors.

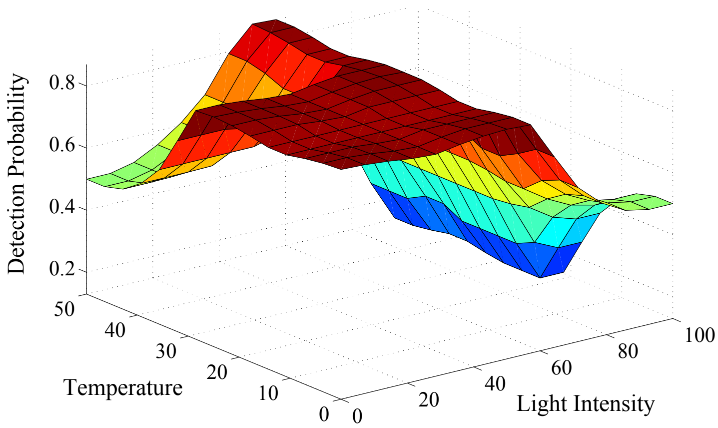

Figure 5.6.: Control surface of light intensity and temperature readings

5.3.2. Probabilistic Model Checking

Probabilistic model checking is a well-established formal technique to exhaustively and automatically examine a system. Building on a model of the system and the specifications, formulated in some precise logic language supporting quantitative probabilistic statements, probabilistic model checking determines whether a given structure (i.e. "the model") satisfies a given logical formula. While probabilistic model checking appeared within the academic community only a few years ago [KNP2011,KKZ2005], it has already been successfully used in industry to assist system design and performance analysis, with *PRISM* [KNP2011] being a mature tool. In this work, the proposed technique is evaluated via constructing a corresponding probabilistic model. Then, the properties of interest are identified and verified.

Modeling Virtual Detection in PRISM

In the proposed settings, the probabilistic model is a discrete-time Markov Chain (DTMC). A DTMC is a pair $\mathcal{M} = (S, P)$ where S is a countable set of states and $P : S \times S \to [0, 1]$ is a function called the transition probability function, such that $\sum_{s' \in S} P(s, s') = 1$. Intuitively, the states of the DTMC are the states the system can be in, and the transitions are the probabilities of switching between these states. In PRISM, models are described using a simple, state-based language. The behavior of the model is defined by commands, comprising a guard and one or more actions:

Listing 5.1: Structure of a probabilistic model in PRISM

```
[]<guard> -> <prob_1>: <action_1> +
             <prob_2>: <action_2> +
             ...
             <prob_n>: <action_n>
```

As shown in Listing 5.1, the guard is a predicate over variables in the model. If the guard is true, then an action is selected according to the probability assigned to this action. As it has been stated above, the sensors can be categorized into energy-expensive sensors such as 2D semiconductor gas leak detectors and energy-cheap sensors such as light intensity, temperature, and humidity sensors. Therefore, these components are respectively modeled in PRISM as follows.

Gas Sensor Modeling Initially, the main gas sensor is activated. Then, based on gas detection, it either activates the transceiver or is itself deactivated. Once the main sensor is switched *OFF*, the virtual sensor is activated therefor. The corresponding code snippets are shown in Listing 5.2, where the variables *gs* and *vs* represent the status of the main gas sensor and the virtual sensor, respectively. The variable *isdanger* is set to 0 if no gas leak is detected; otherwise it is set to 1.

Listing 5.2: PRSIM commands for the gas sensor

```
module sensor
    gs:[0..2] init 1;      //Gas Sensor: 0-OFF, 1-ON, 2-transmission
    vs:[0..1] init 0;  //Virtual Detector: 0-OFF, 1-ON
    isdanger :[0..1] init 0;   //0-no gas leak, 1-gas leak
    trans:[0..1] init 0;     //0-no transmission, 1-transmission
          ⋮
    [] gs=1 & vs=0 & isdanger=1 ->(trans'=1) & (gs'=0);
    [] trans=1 & gs=0 & vs=0 ->(trans'=0) & (gs'=1) & (isdanger'=0);
    [] gs=1 & vs=0 & isdanger=0 ->(gs'=0)&(vs'=1);
```

```
        ⋮
endmodule
```

Virtual Sensor Modeling Specifically, the virtual sensor consists of a FLC which generates the detection probability P_d, according to the environment. Moreover, three sensors for detecting the light intensity, temperature, and humidity are also utilized. The PRISM pseudo code snippets in Listing 5.3 describe the virtual sensor behavior.

Listing 5.3: PRISM pseudo code for the virtual detectors

```
module sensor
    // The rules for fuzzy logic
    [] Light_intensity=50 & Temperature=25 &
       Humidity=50 -> (Pd'=0.87);
        ⋮
    // The behavior of the virtual detector
    [] Pd < Thr_1 -> Sampletime doubled;
    [] Pd >= Thr_1 & Pd<Thr_2 -> Sampletime unchanged;
    [] Pd >= Thr_2 -> (gs'=1);
        ⋮
endmodule
```

Virtual Detection Property Formulation in PRISM

In this work, we concentrate on the evaluation of energy consumption, lifetime, and response time. This can be achieved in PRISM by attaching *rewards* to each transition. That is to say, if the system switches from one state to another one, some value has to be taken into account, e.g. time, response_time and energy_consumption, which are called *rewards*. In this way, a discrete-time Markov reward model is constructed, which is a variant of classical DTMC with rewards assigned to each transition. Listing 5.4 depicts a part of the rewards definition in PRISM. For example, if the main gas sensor is *ON* and no leak is detected, then the sensor consumes 80mA of energy and takes 1s to do so (cf. lines 1–2). If the virtual sensor is *ON* and generates some values that overstep a predefined threshold, then the gas leak event triggers the controller. Thence, detecting gas leak may undergo an endurable delay. A series evaluations w.r.t. the proposed case study can be performed:

Listing 5.4: Rewards definition in PRISM

```
1    rewards "energyconsumption1"
2    [] gs1=1 & vs1=0 & isdanger1=0: 80;
3    ⋮
4    endrewards
5    rewards "lifetime"
6    [] gs1=1 & vs1=0 & isdanger1=0: 1;
```

```
⋮
endrewards
rewards "responsetime"
[] vs=1 & grade>=Threshold:sampletime;
⋮
endrewards
rewards "gas_leak_events"
[] vs=1 & grade1>=Threshold:1;
⋮
endrewards
```

- $R\{“energyconsumption''\} =?[C < T]$, what is the cumulative energy consumption within T steps? where C means that we want to compute an accumulated quantity. The term T is the time bound. So, the condition means till time T, how much energy has been consumed.
- $R\{“lifetime''\} =?[C < T]$, what is the cumulative lifetime within T steps?
- $\frac{R\{“responcetime''\}=?[C<T]}{R\{“gas\_leak\_events''\}=?[C<T]}$, what is the average response time within T steps?

where T depends on the time that the energy resources of the sensor are depleted.

The probabilistic behavior of a heterogeneous WSN consisting of real and virtual sensors has been examined. As earlier mentioned, the goal of adopting virtual sensing is to extend the functionality period of the main energy-expensive sensors. On the other hand, engaging virtual sensors into the system may increase the time required for a true detection. Accordingly, both, the main sensor's lifetime and the virtual sensor's response delay constitute the evaluation metrics in the following study. Table 5.2 lists the parameter values that have been used in the evaluation.

Table 5.2.: Parameter values used in the energy model

	Description	**Value**
Gas Sensor	Current consumption of measurement	80 mA
	Current consumption of data transmission	40 mA
	Measurement time	1 sec
	Data transmission time	347 ms
	TSL13T light sensor current consumption	1.1 mA
	Temperature and humidity sensor current consumption	90 uW
Virtual Detector	sampling time	288 ms
Battery	Quantity	3
	Capacity	3000 mAh

Discussion of the Results

The results obtained from checking this model with PRISM are as follows. Figure 5.7 compares the energy consumed by a main gas sensor in two cases, (1) gas leaks are frequent, and (2) gas leaks are rare. Intuitively, the former case (depicted in blue) is seen as the worst case. In this case, the main sensor switches frequently between active and sleep modes of operation. Accordingly, its energy consumption resembles a train of square pulses, as shown in Figure 5.7. Alternatively, if the probability of gas leak is low, then duty cycle of the main sensor is drastically reduced. However, other probabilities were not considered in [SBS+2011]. Instead, the proposed strategy – in this thesis – examines all possible detection probabilities.

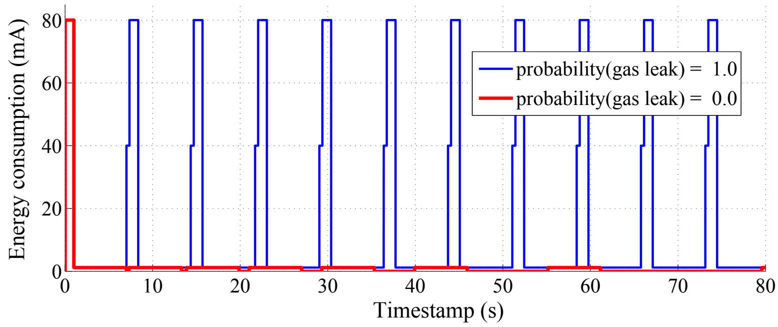

Figure 5.7.: Probabilistic behavior of the main gas sensor

Figure 5.8a demonstrates the lifetime of the main gas sensor versus different probabilities of gas leaks. The lifetime is gradually decreased as the gas leak probability increases. In fact, the resultant lifetime presented in [SBS+2011] is a deterministic value where it was empirically computed, assuming the occurrence of no gas leaks. Therefore, the lifetime in [SBS+2011] – depicted in red – is compared with the result obtained at zero probability of gas leak. As earlier expected, the proposed virtual sensing method increases the lifetime by 58 times relative to the PWM method at zero probability of gas leaks. However, the lifetime – in case of virtual sensing – is decreased below what we can get by only adopting the PWM method at probability of 0.28. This reduction occurs due to increasing the main sensor's duty cycle. Practically speaking, high probabilities of gas leaks are typically not realistic. Therefore, we can conclude that virtual sensors significantly improve the main sensors lifetime.

Aside from the lifetime extension, the system performance in terms of the incurred delay is also examined. The average response time is a QoS metric defined as the average time required for the sensor to react to a sudden change in the quantity of interest. Figure 5.8b depicts the average response delay versus different probabilities of gas leak. The response time is gradually reduced with increasing the probability of gas leaks. To further save energy, we adapt the virtual sensor's duty cycle, where it is increased as the gas leak probability increases. This adjustment leads to shrinking the average response time. In other words, response time of the virtual sensors can be significantly shortened by increasing their duty cycle. To sum up, virtual sensing is an efficient mechanism for highly reducing the energy dissipation of main sensors at the expense of a slight incurred delay, as depicted in Figure 5.8b.

Although, the lifetime has been extended via utilization of virtual sensors, a question arises about the reliability of such virtual sensors. Combining n sensors to form a virtual sensor may result in precision problems which may have a negative impact on

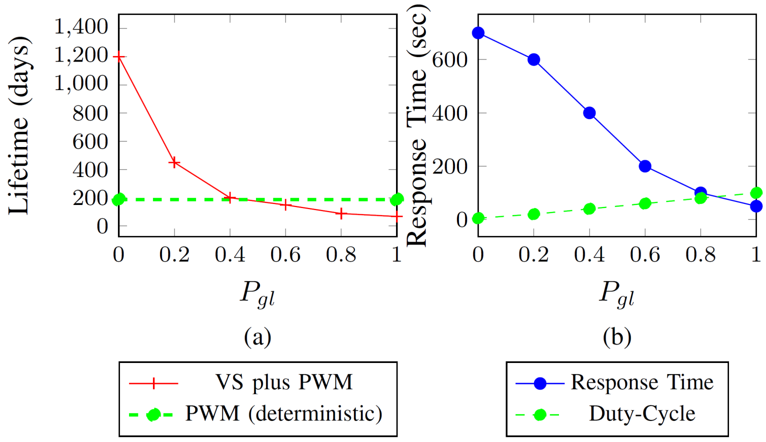

Figure 5.8.: A comparison between (a) lifetimes of the main gas sensor, (b) response time both, with and without virtual sensing

the detection probability of interesting events. Hence, the next section discusses the reliability problem and how it can be overcome.

5.4. Reliable Virtual Sensing

In this section, an ontology-based technique is proposed to improve the sensing reliability. Based on design-time knowledge, switching between real and virtual sensors during run-time is feasible via ontology-generated rules. These rules represent the relationship between sensors, their observations, and environmental conditions. To take the quality into account, the quality of one particular set of sensors is first focused on and it is shown how this set can replace an "energy-expensive" sensor under certain quality aspects. In a subsequent step, the proposed approach will be extended to show how the quality of a complex set of heterogeneous sensors can be estimated using a sensor relationship ontology.

Next, the state-of-the-art of the ontology and how to employ it for improving the reliability of virtual sensing are introduced.

5.4.1. Preliminaries: SSN Ontology

To represent the relationship between sensors, their observations and environmental conditions, the so-called Semantic Sensor Network (SSN) ontology is used [CBBea2012]. An ontology can describe sensors, the accuracy etc of sensors, observations and methods used for sensing. In addition, concepts for operating and survival ranges are included, as these are often part of a given specification for a sensor, along with its performance within those ranges. Finally, a structure for field deployments is included to describe deployment lifetime and sensing purpose of the deployed macro instrument. As shown in Figure 5.9, the SSN is divided into 10 conceptual models, represented by boxes in the figure, and consists of 41 concepts (blue boxes with round corners). The SSN was first released in 2010 by the W3C Semantic Sensor Network Incubator Group. Its main goal was to provide an additional layer for semantic compatibility. In [CBBea2012], the authors introduced the architecture of the SSN.

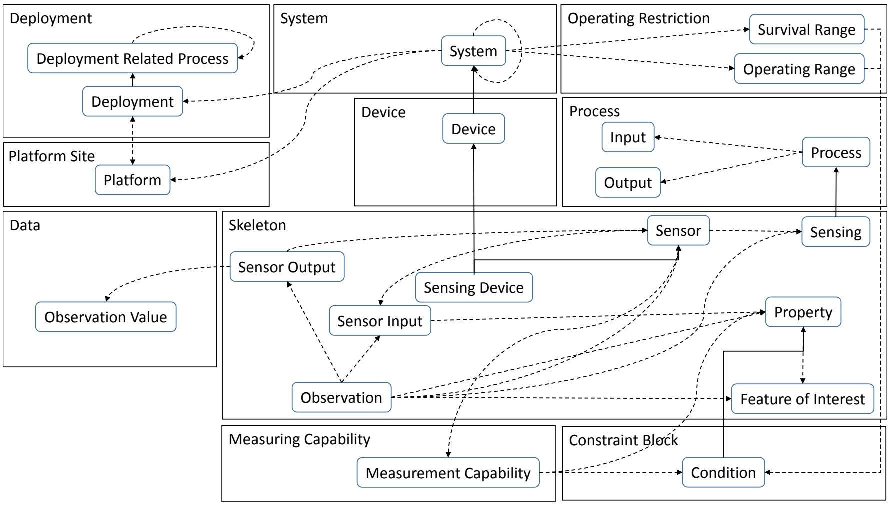

Figure 5.9.: The 10 conceptual models of the SSN ontology [CBBea2012]

In the following, the sensor perspective (see Figure 5.10) of the SSN ontology is focused on to describe the quality of sensor observations. The ontology includes multiple quality dimensions, like accuracy and selectivity, for measurement capabilities as an extension of the class *Measurement Property* that is inherited by the class *Property* (ssn:MeasurementProperty $\subseteq$ ssn:Property). A *Measurement Property* can thus be linked to any sensing device as an observable property in the real world. By doing so, the SSN ontology includes concepts to link the quality of sensor measurements with other sensors and thus, gives us the opportunity to query the ontology for appropriate sensors to estimate the current reliability of a given sensor under certain environmental conditions.

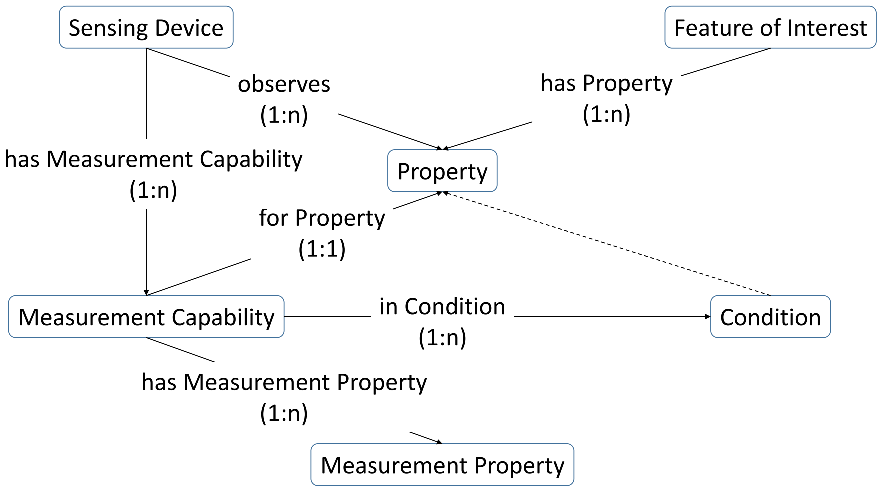

Figure 5.10.: The sensor perspective of the SSN ontology

5.4.2. Quality Estimation

To select the appropriate sensor, the current condition of each sensor has to be estimated by looking up the relationship between the sensors. Using the condition, the current qualities of the sensor measurements can be estimated and thus, the sensor with the highest qualities can be selected for a given task. To do so, a condition of a sensor has to be first defined. Let $\mathbb{D}$ be the set of all quality dimensions, $\mathbb{S}$ the set of all sensors, and $s \in \mathbb{S}$ a sensor. Furthermore, let $\mathbb{P}$ be the set of all observable properties and let $p \in \mathbb{P}$ be a property, as well as $\mathbb{M}_p^s$ the set of all measurement capabilities of a sensor s for a property p. $m \in \mathbb{M}_p^s$ is a measurement capability of the sensor s for the property p. As an example, consider a seismic sensor that has a measurement capability to measure vibrations. The quality of each measurement can be described by different immanent quality dimensions, like accuracy or selectivity.

In addition, let $name(m)$ be the name of the measurement capability in the ontology. It should be noted that the name of the measurement capability need not be unique across all $m \in \mathbb{M}_p^s$. Let furthermore be $\mathbb{C}^m$ the set of all conditions, that are defined for measurement capability m and let $c \in \mathbb{C}$ be a condition. The condition is a function $c : \omega \rightarrow$*{true, false}* that delivers either *true* if the condition is met or *false* if the condition is not met. As an example for a condition consider the aforementioned seismic sensor that has a measurement capability named “vib” to measure vibrations with a high accuracy under normal operating temperatures.

Be $\mathbb{D}^m$ the set of all quality dimensions with $\mathbb{D}^m \subseteq \mathbb{D}$ that are defined for the measurement capability m and $d \in \mathbb{D}^m$ is a quality dimension. It should be mentioned, first, that in principle several conditions can be defined using a measurement capability and thus also the associated quality dimensions. This behavior is understood hereinafter as a combination of conditions, so that the test for the quality dimension condition is to be understood as an examination of all individual conditions and the resulting condition is defined as follows:

$$\tilde{c}(\hat{e}) = \bigvee_{c \in \mathbb{C}^m} c(e) \tag{5.2}$$

where $\hat{e}$ are the measurements from a sensor. To estimate the overall quality, the qualities of each measurement capability have to be aggregated. This aggregation assumes that a higher value of the quality dimension has in principle a stronger negative impact on the calculation result as a lower value, e.g. a high latency or a larger dispersion has a stronger negative impact on the quality of a sensor measurement than a low latency or dispersion.

5.4.3. Ontology-Generated Selection Rules

In the following, it is shown how the ontology is used to generate the correct selection rules for each sensor so that – depending on the current sensor setup and the current environmental conditions – the sensor with the highest measurement quality is selected for the task at hand. To do so, the relationship between the sensors used, their measurements, and the influence of internal and environmental factors on these measurements has to be modeled in the ontology.

Using the ontology, rules are generated to switch between sensors depending on observable properties in the feature of interest. To transform this information into processing instructions, the required sensing devices have to be concatenated to have the observed property values at hand. The required sensing devices that are necessary to estimate the quality of each property, are provided by the ontology and can be retrieved using the following query:

```
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn>
SELECT * WHERE {
  ?foi rdf:type ssn:FeatureOfInterest;
  ssn:hasProperty ?p .
  ?sd rdf:type ssn:SensingDevice;
  ssn:observes ?p;
  ssn:hasMeasurementCapability ?mc .
  ?mc  rdf:type ssn:MeasurementCapability;
  ssn:forProperty ?p
}
```

The example query retrieves all sensing devices *?sd* and their measurement capabilities *?mc* able to observe a given condition *?p*. This is possible because the condition inherits from the concept *Property*. In addition, the query filters for a given feature of interest *?foi* to only select sensing devices that are able to give reasonable observations of the condition. Thus, using the resulting list of sensing devices, the quality of each sensor measurement can now be continuously evaluated and the one with highest quality can be selected.

The run-time evaluation of the sensor selection rules could result in a highly dynamic selection of the most reliable sensor. This adaptability is guaranteed even if sensors fail or new sensors are deployed. However, it would be necessary to store the complete ontology on each sensor node in a WSN. Instead, the ontology is utilized to generate static sensor selection rules. Thus, the sensor selection rules have to be propagated each time a new node is deployed through the complete WSN. Under the assumption that we have to deploy new nodes very rarely, this propagation is acceptable in the focused scenario. In the sequel, an object tracking mechanism and how virtual sensing can be beneficial in this field are discussed.

5.5. Case Study: Object Tracking Sensor Network

Generally, object tracking sensor network (OTSN) scenarios paved their path in numerous civilian and military applications. For instance, the authors in [JRL2010] proposed an OTSN for the detection of human intruders into secured regions, such as the ground floors of high security buildings, large industrial installation, or airports. Another OTSN is developed for generic (animal) target tracking in the surrounding area of wildlife passages built [GSGSLea2010]. Hence, it is easier to establish safe ways for animals to cross transportation infrastructures. Finally, the authors in [KWIea2012] proposed a μ-radar for tracking intruders with a long sensing range (approx. 6 meters omnidirectional).

A plethora of passive and active sensors such as magnetic, radar, thermal, acoustic, electric, seismic, and optical sensors are established. However, radar sensors can be an excellent solution for different scenarios and have many advantages:

- they do not require a direct line of sight to the object,
- they have a long sensing range guaranteeing covering large spaces,
- they can provide estimations of speed and location,
- they can distinguish different materials and shapes [ADBea2004].

Nevertheless, these μ-radar sensors suffer from excessive power consumption. Therefore, this case is well-suited for examining the validity of reliable virtual sensing. Specifically, the problem is interpreted as a desire to preciously detect intruders provided that minimum energy is drawn and minimum event-miss probability is pledged.

In this section, a case study of an OTSN is engineered to demonstrate the applicability of the proposed reliable virtual sensing method. According to [ADBea2004], disturbance, emerged from persons and vehicles, can be detected by seismic sensors. Thence, the proposed virtual sensor is composed of seismic sensors, a look-up table of targeted vibration signals (expected intrusions) called, *codebook*, and a pattern matching algorithm. We selected a well-known pattern matching algorithm, referred to as the dynamic time warping (DTW). It has been used to contrast the measured vibrations with the predefined codebook. Selection of the DTW algorithm is reasonable since: 1) it is well-suited for matching patterns of unequal lengths, and 2) it carries out a non-linear matching which preciously describes the correlation. Figure 5.11 depicts the structure of an object tracking system composed of real and virtual sensors. For the virtual sensor, outcomes from omni-directional seismic sensors (sequence A) are to be applied to a DTW algorithm. The key idea underlying the proposed virtual sensor $\mathcal{S}_v$ is to stretch (or compress) the seismic trace until it best matches one of the reference traces in the codebook $(B_1, .., B_m)$. Seismic sensors have long sensing ranges, no line-of-sight requirement, the ability to harvest vibration energy, and are of passive nature [ADBea2004].

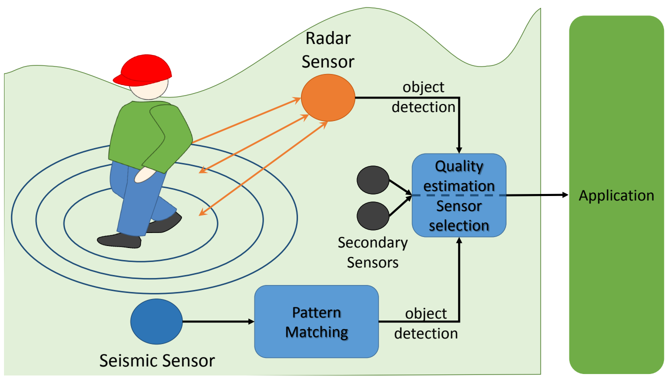

Figure 5.11.: System structure with real and virtual sensors

The quality estimation mechanism utilizes secondary sensors to monitor the quality of sensors, such as temperature and humidity sensors. Based on these qualities, the rules - generated by the ontology - determine the best-suited sensor. The switching decision between real sensor $\mathcal{S}_r$ and virtual sensor $\mathcal{S}_v$ is affected by the sensing reliability and precision. In this concrete case, the relationships between the participating sensors can be modeled as shown in Figure 5.12.

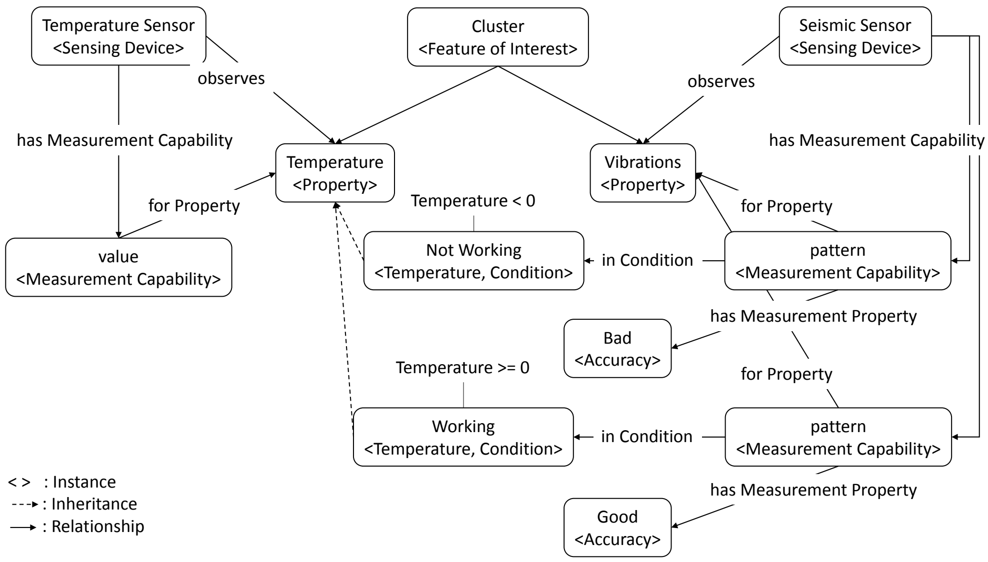

Figure 5.12.: SSN ontology for moving object tracking

In this example, the seismic sensor observes a property called *vibrations* using two measurement capabilities called *pattern*. Both measurement capabilities have different accuracies that are defined under a specific condition that is inherit from the property *temperature*. This property is observed by the temperature sensor. Thus, the temperature readings can be utilized to evaluate the current condition. This is feasible via transforming the modeled relationships into formulas to estimate the current qualities. These qualities are used within the switching decision algorithm.

Algorithm 5 demonstrates the system structure when utilizing virtual sensors. Line 2 denote sensing a vibration pattern ϕ and then using the DTW algorithm to compute the distance D_{ϕ,B_i} between ϕ and the stored patters in the codebook $B_i = \{B_1, \cdots, B_m\}$. If at least one distance D_{ϕ,B_i} exceeds a predefined threshold H, then the real sensor S_r has to be activated, as depicted in Line 4. If event detection is confirmed by the real sensor S_r, then the measured pattern is fed back to all virtual sensors to learn this new pattern. Actually, the learning procedure here is not described, as it is left as future work.

If the threshold H is not violated, the sensor's S_v reliability is estimated in the light of the environmental parameters such as rain and temperature, as given in Lines 7-8. The quality sensors, e.g. temperature and humidity sensors provide insight into the quality of the sensor readings. The quality set Q stimulates a decision making algorithm whose rules are automatically generated via an ontology.

Below, the DTW algorithm is discussed in the light of the aforementioned scenario. Additionally, accuracy of such an algorithm is examined using real vibration traces.

5.5.1. Dynamic Time Warping

Preliminaries

The pattern matching problem has been previously addressed in the context of data mining. *Euclidean distance* is a well-known linear metrics for quantifying the distance

Algorithm 5 Object tracking scenario

Require: Heterogeneous network of $\mathcal{S}_r = \{s_1, .., s_\tau\}$ & $\mathcal{S}_v = \{v_1, .., v_\omega\}$
1: **while** true **do** {
2: **Deactivate**(sensor $\mathcal{S}_r$) and **Activate**(sensor $\mathcal{S}_v$);
3: **Sample** ϕ and **Estimate** $D_{\phi,i} = \{D_{\phi,1}, \cdots, D_{\phi,m}\}$;
4: **if** $D_{\phi,i} \geq H$ **then**
5: **go to** 9;
6: **end if**
7: *Evaluate* the ontology rules;
8: **if** the sensor S_v is not reliable **then**
9: **Activate** the sensor $\mathcal{S}_r$;
10: **end if**
11: *learn*($\mathcal{S}_v$, new_pattern); }
12: **end while**

between two vectors. Assume two sequences, $A \in \mathbb{R}^n$ and $B \in \mathbb{R}^m$ to be correlated where

$$A = \langle a_1, a_2, a_3, \ldots, a_i, \ldots, a_n \rangle \quad \text{and} \tag{5.3}$$

$$B = \langle b_1, b_2, b_3, \ldots, b_j, \ldots, b_m \rangle \, . \tag{5.4}$$

The Euclidean distance $Euc(A, B)$ is as denoted as follows.

$$Euc(A, B) = |A - B| \qquad \text{iff} \;\; m = n \tag{5.5}$$

Despite its simplicity, the Euclidean distance is ill-suited for real-time tasks with sequences of unequal lengths. Moreover, it is highly sensitive to outliers. The DTW algorithm, on the other hand, is a classic time series alignment algorithm. It has been widely used for optimal alignment of two time series through warping the time axis iteratively until an optimal match (according to a suitable metrics) between the two sequences is found. In general, the DTW's non-linear behavior produces a more intuitive similarity measure compared with the Euclidean distance. As shown in Figure 5.13, the DTW measure replaces the one-to-one point comparison, used in the Euclidean distance, with a many-to-one (and vice versa) comparison. The green lines represent mappings between points of time series T and S. The main feature of this distance measure is that it enables recognizing similar shapes, even if they present signal transformations, such as shifting and/or scaling.

Figure 5.14 visualizes the matching between a reference and a test pattern arranged on the side borders of a $m \times n$ matrix where the matrix elements are the DTW distances $d_{n,m}$ as expressed in Equation 5.6. Several paths can be drawn from $(1, 1)$ to (n, m). However, the optimum alignment $P_{opt} = \langle p_1, p_2, \ldots, p_k \rangle$ minimizes the total inter-distances as denoted by Equation 5.7.

$$d_{n,m} = \begin{cases} |a_1 - b_1| & \text{if } n = m = 1 \\ |a_n - b_m| + W_{n,m} & \text{otherwise} \end{cases} \tag{5.6}$$
$$\text{with} \quad W_{n,m} = min(d_{n-1,m}, d_{n,m-1}, d_{n-1,m-1})$$

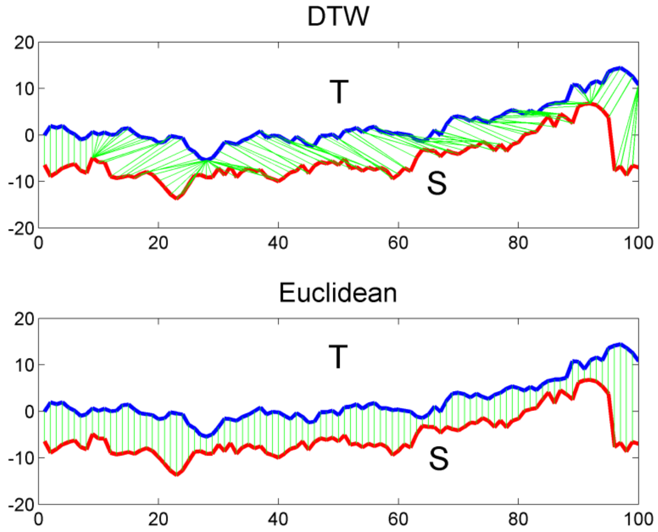

Figure 5.13.: Principle of operation of the DTW metric and the Euclidean distance [MAC+2012]

$$P_{opt} = \min_{P} \left\{ \sum_{s=1}^{k} d_{n,m} \right\} \tag{5.7}$$

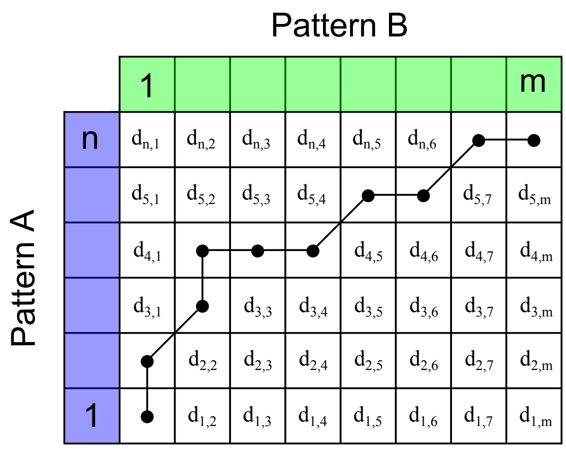

Figure 5.14.: Choosing the optimal warping path, where $k = 10$

The search space is governed by a set of design constraints summarized in Table 5.3. Firstly, the path P should continuously advance one-step at a time to avoid discarding important features. Moreover, the path should be monotonically non-decreasing to hamper feature recurrence. Finally, the start and end points should extend from $(1, 1)$ to (n, m) to align the entire sequence. In some applications, a global rule defines a warping window $R \subseteq [1 : n] \times [1 : m]$ to speed up the algorithm. Nevertheless, confining the search space to R is debatable, since the path P_{opt} may traverse cells outside the specified constraint region. Thereof, this constraint for matching optimization is deliberately ignored.

Table 5.3.: Summary of alignment path constraints

Continuity	$d_{i,j} - d_{i-1,j} \leq 1$ & $d_{i,j} - d_{i,j-1} \leq 1$
Monotonicity	$d_{i-1,j} \leq d_{i,j}$ & $d_{i,j-1} \leq d_{i,j}$
Boundary Conditions	$p_1 = d_{1,1}$ & $p_k = d_{n,m}$
Warping Window	$\lvert i_s - j_s \rvert \leq R$

Algorithm 6 states the proposed DTW implementation. Recursively, the DTW distances are horizontally and vertically determined. Afterward, Equation 5.6 evaluates the rest of the matrix. Through lines 12-18, the path P_{opt} is selected through point-to-point optimization. Finally, the DTW distance $\chi_i = D(A, B_i)$ is obtained via normalizing P_{opt} relative to the length of the optimal path, denoted by k. This enables fair comparisons. In the proposed scenario, the test pattern is matched with a codebook of size m. To determine the best match between the measured pattern and a set of codebook patterns, the distance $\chi_{min} = min(\chi_1, \cdots, \chi_i)$ is chosen.

Algorithm 6 Implementation of the DTW algorithm

Require: Reference pattern $A \in \mathbb{R}^n$, and test patterns $B \in \mathbb{R}^m$
1: **for** i such that $1 \leq i < n$ **do**
2:     **for** j such that $1 \leq j < m$ **do**
3:         $d_{1,1} \leftarrow |a_1 - b_1|$;
4:         **if** $!i$ & $j == 1$ **then** ▷ Horizontal border
5:             $d_{1,j} \leftarrow \sum_{x=1}^{j}(|a_1 - b_x|)$;
6:         **else if** i & $!j == 1$ **then** ▷ Vertical border
7:             $d_{i,1} \leftarrow \sum_{y=1}^{i}(|a_y - b_1|)$;
8:         **else** $d_{i,j} \leftarrow |a_i - b_j| + W_{i,j}$; ▷ Matrix's heart
9:         **end if**
10:     **end for**
11: **end for**
12: **while** $i < 1$ & $j < 1$ **do** { ▷ Optimal path
13:     $P_{opt} \leftarrow P_{opt} + d_{i,j}$;
14:     **if** $W_{i,j} == d_{i-1,j}$ **then**
15:         $i \leftarrow (i - 1)$;
16:     **else if** $W_{i,j} = d_{i,j-1}$ **then**
17:         $j \leftarrow (j - 1)$;
18:     **else** $i \leftarrow (i - 1)$ & $j \leftarrow (j - 1)$;
19:     **end if**
20: **end while**}
21: $D(A, B) \leftarrow \frac{P_{opt}}{k}$; ▷ Single distance normalization

Equation 5.8 defines the threshold H, utilized for initiating a new learning phase.

The threshold is determined via normalizing the χ_{min} distance relative to the distance $\chi_{max} = max(\chi_1, \cdots, \chi_i)$. where H is a margin such that $H \leq 1$. Adopting this decision rule is wholesome. It normalizes the distance $DTW(A, B)$. Hence, H is not updated in accordance with ϕ. Subsequently, it is shown experimentally below that a fixed value of H = 0.1 is robust across a wide range of patterns.

$$[\chi_{min}] \cdot [\chi_{max}]^{-1} < H \tag{5.8}$$

Important to mention that the standard version of the DTW is relatively complex and ill-suited for tiny sensing devices. Therefore, a side work has been performed to simplify the DTW algorithm. The key idea of the modified DTW algorithm, referred to as liteDTW, is to reduce the input data dimension and to linearize the procedure of evaluating the algorithm. Specifically, data abstraction highly reduces the computational overhead. Whereas, linearizing the algorithm shrinks the required memory footprint from $n \times m$ to only $2 \times m$. More details about this modification are included in Appendix A.

DTW Evaluation

In this section, the DTW precision is examined with real vibration traces prior to be incorporated into the virtual sensor. At the outset, an Arduino UNO board has been utilized to sample seismic patterns from a LDT piezoelectric vibration sensor. Different measuring scenarios of speed 0.5 m/sec have been considered and summarized in Table 5.4. Practically, the vibration records have been classified into *targeted* T and *non-targeted* NT patterns. The former results from the motion of a tracked object, whereas the latter comprises unwanted footprints. This taxonomy is rationale to facilitate making the object detection decision.

Table 5.4.: Indexing reference and test patterns

Index	Target Patterns	Index	Non-target Patterns
T_1	Straight walking (indoor)	NT_1	Background
T_2	Straight walking (outdoor I)	NT_2	Finger drumming
T_3	Straight walking (outdoor II)	NT_3	Straight walking by a mobile machine (indoor)
T_4	Circle walking (indoor)	NT_4	Straight walking by a mobile machine (outdoor)
T_5	Circle walking (outdoor)		

Figure 5.15 depict samples of precision results obtained from contrasting the codebook to some targeted and non-targeted patterns. Knowing that $DTW(A, A) = 0$, pattern A_{indoor} is matched with $A_{outdoor}$ to clarify the process of selecting the best match. The columns – in Figure 5.15a – represent the distance χ_{N1,B_i} between pattern $N1$ and other patterns B_i (excluding the pattern $N1$). The threshold H is delineated as a horizontal line (in red). Obviously, the DTW algorithm has successfully matched the indoor and outdoor pairs via adopting the minimum DTW inter-distance. A fixed margin of $H = 0.1$ is utilized to initiate a new learning phase whenever exceeded. In the proposed scenario, this learning is useful to store new vibration patterns that have been detected by the real sensor. Accordingly, the virtual sensors can detect these new pattern in the following

rounds without the need for activating the main sensor. In the sequel, the energy consumption outstrip of virtual sensors over real ones is examined. Afterward, the proposed ontology-based reliability approach is validated via a set of experiments.

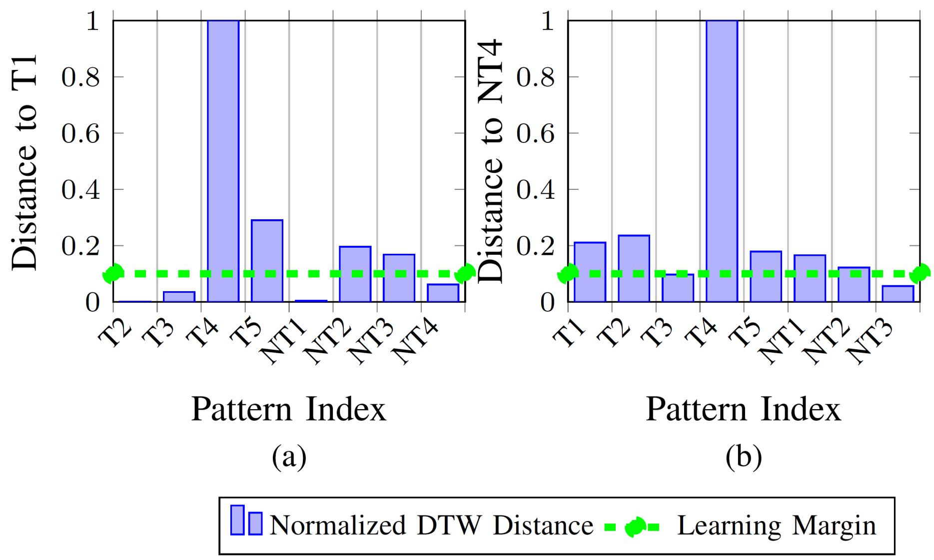

Figure 5.15.: Matching the measured patterns $T1$ and $NT4$ to other targeted and non-targeted vibration pattern stored in the codebook

5.6. Performance Evaluation

In this section, merits and flaws of utilizing virtual sensors for mitigating the burden of "energy-hungry" sensors are addressed. Two sets of experiments have been conducted. Firstly, two virtual sensors have been implemented on TelosB nodes (CM5000 MSP) where energy consumption and time/space complexity are quantified. Secondly, a large-scale heterogeneous network composed of virtual sensors and radars has been built on the WSNet simulator [CFH]. The WSNet enables large-scale monitoring with accurate insight of the energy consumption.

5.6.1. Node-level Evaluation

In this set of experiments, the DTW algorithm is ported to a sensor node reporting its output via a sink node. Both were TelosB nodes with 10K bytes of RAM. A minimum of 1000 samples should be collected for the test pattern to be distinguishable. However, it was not trivial to store this bulky pattern; hence data is collected over multiple iterations. Algorithm 7 illustrates a modified DTW algorithm, referred to as *lite*DTW, which requires a codebook whose patterns are of size $\omega = 50$. At the outset, a bundle of $\tau = 200$ samples is compressed into $\theta = 10$ samples to reduce the DTW computational overhead. The FuzzyCAT compressor (see Chapter 4) has been utilized where each compressed value is a weighted average of the original samples. Hence, the compressed samples still preserve their correlation. Furthermore, Line 9 comprises the function

2colDist which is used to the memory footprint of the DTW algorithm via iterating only over two columns of the matrix. Appendix A discusses the proposed *lite*DTW algorithm in more detail. The procedure continues until determining the warping path P_{opt}. Then, new bundles are processed, until the 1000 samples are covered as depicted in line 6. This procedure is repeated with different codebook patterns as stated in line 14.

Algorithm 7 The *liteDTW* algorithm (ported to the TelosB nodes)

Require: codebook $C = \{B_1[\omega], \cdots, B_i[\omega], \cdots, B_m[\omega]\}$
1: **repeat**
2:     **for** i such that $0 \leq i \leq \tau$ **do**
3:         **collect** $A = \langle x_1, \cdots, x_\tau \rangle$;
4:     **end for**
5:     **compress** $A[\tau]$ into $A[\theta]$;
6: **until** $(\frac{\omega.\tau}{\theta})$ samples are processed;
7: **repeat**
8:     **for** j such that $0 \leq j \leq \theta - 1$ **do**
9:         **Compute** 2colDist$(B_i[r : r + \theta], A[\theta])$;
10:         **Locate** the path P_{opt} points;
11:     **end for**
12:     **Increment** $r \leftarrow r + \theta$;
13:     $\chi \leftarrow \sum P_{opt}/k$;
14: **until** A is matched with all $B_i \in C$;
15: $\chi_{min} = min(\chi_1, \cdots, \chi_m)$;

The setup involved a network under the ConikiMAC radio duty cycling protocol with the *unicast* communication primitive in the Rime stack of the Contiki OS. For simplicity, only three patterns were included in the codebook. The instantaneous current consumption was estimated utilizing the *Energest* module as denoted in Equation 5.9 [Voi2012]. The current consumption in miliampere is a function of the activity (ψ) start and end times, the operating current (I_0) which can be obtained from a datasheet, and the runtime per each round. Two TelosB virtual sensors exchanged the minimum DTW distances for around 63 minutes.

Figure 5.16 depicts the current consumed per round for evaluating the *liteDTW* algorithm (CPU), and for transmitting the minimum distances (radio). The figure also comprises the total current consumed by the virtual sensor, and the μ-radar sensor for the same duty cycle. Despite the monotonic behavior of the various current curves, they give a clear insight about the significant difference between the current consumed by the main and the virtual sensors. In fact, the virtual sensor's radio overhead is drastically reduced in real-time applications where transmission only occurs on an event-based basis. Based on the results published in [KWIea2012], it is concluded that the virtual sensor has approximately 99.93% less energy consumption than the μ-radar sensor.

$$I_\psi = \frac{(\psi_{end} - \psi_{start}) \times I_0}{runtime} \text{ [milli-Ampere]} \tag{5.9}$$

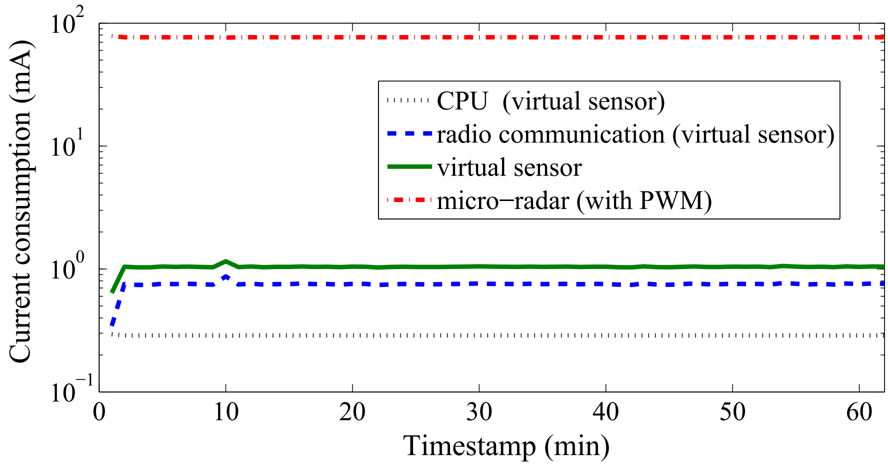

Figure 5.16.: Current consumption of the μ-radar and the virtual sensor

5.6.2. Network-level Evaluation

In this section, a combination of virtual sensing and SSN ontology utilization is presented and evaluated in terms of the radar's lifetime extension and the overall event-miss probability. The WSNet simulator has forged a circularly deployed OTSN with single real sensor S_r and eight virtual sensors S_v. The inter-distance between the scattered sensors was set to 25m. Each sensor node has an omni-directional antenna using a half-duplex radio interface with -30dBm. The simulator is configured with a Rayleigh fading and log-normal shadowing propagation model and orthogonal interferences. The IEEE 802.15.4 unslotted CSMA/CA module has been adopted as the default MAC protocol.

In this heterogeneous OTSN, each sensor node $\mathcal{S}_v$ has a seismic and a temperature sensor whereas $\mathcal{S}_r$ is represented by a Doppler μ-radar sensor. Generally, the seismic quality is typically affected by four factors:

a) *the range*, it is defined as the physical distance from a sensor $\mathcal{S}_v$ to the tracked object.

b) *the processing frequency*, it is determined by the required precision.

c) *the DTW accuracy*, it is defined by the temperature range.

d) *the selectivity*, it is delineated by vibration amplitudes. This quality describes situations like rain in which the sensor $\mathcal{S}_v$ is not capable to detect ϕ due to high background noise.

Based on the sensor-property relationships from the SSN ontology, a sensor $\mathcal{S}_v$ triggers $\mathcal{S}_r$ to perform the measuring process whenever the above margins are violated (quality is below threshold). For simple evaluation, two quality dimensions in this evaluation are focused on: accuracy and selectivity. However, other quality dimensions, defined in the ontology, should also be considered in real applications.

Specifically, the operating temperature range of the sensor $\mathcal{S}_v$ is defined in the interval $[0, \cdots, 50]$°C. Further, an accurate pattern matching is no longer guaranteed for instantaneous vibrations above 300 mV/g in the proposed setup. In addition, the selectivity is modeled as a value depending on the current vibration. The concrete conditions and their effect on the two qualities are empirically listed in Table 5.5. The placeholders *Temp* and *Vib* represent the measurement capabilities of the sensor for the environmental properties and correspond to the names of the nodes in the ontology.

Table 5.5.: Relationship between virtual sensor quality dimensions and environmental properties

Quality Dimension	Property	Expression	Value
Accuracy	Temperature	Temp $\geq 0 \wedge$ Temp ≤ 50	0.7
Accuracy	Temperature	Temp $< 0 \vee$ Temp > 50	$1.0 - \frac{0.4}{e^{(x_{min})}}$
Accuracy	Vibration	Vib > 300	0.1
Selectivity	Vibration	Vib > 200	$1.0 - \frac{0.6}{e^{(x_{min})}}$
Selectivity	Vibration	Vib ≤ 200	0.3

The environmental properties are simulated by two-dimensional sinus waves for temperature and vibration. The evaluation is performed for quality dimension margins in the range $[0.00, 1.00]$ with a step size of 0.1 for both dimensions to compare lifetime and event-miss probability depending on the quality requirements of the application. In Table 5.6, the impact of the quality thresholds on the μ-radar lifetime and the overall event-miss probability is listed. For high quality margins, e.g. accuracy and selectivity margins approach unity, the virtual sensors $\mathcal{S}_v$ have to frequently trigger the real sensor $\mathcal{S}_r$ since the virtual sensors' qualities reside below the margins. Accordingly, the event-miss probability is drastically reduced at the expense of reducing the lifetime to circa one year at both margins set to one. For low accuracy and selectivity margins, e.g. margins approach zeros, less calls are provoked thereby increasing the lifetime. However, the event-miss probability may only increase if the seismic sensor functions outside its operating environmental properties. Finally, we conclude that selecting margins in the range $[0.4, \cdots, 0.7]$, yields an acceptable compromise between the event-miss proability and the real sensor's lifetime.

5.7. Discussion

In this chapter, the energy consumed by energy-hungry sensors is targeted. The core idea was to minimize these sensors' duty cycle through switching them into sleep mode for an extended period of time. As this requires functionality substitution, another so-called virtual sensor was used to monitor the environmental process in the meantime. We demonstrated the principle on an example of gas detection and analyzed its efficiency formally using probabilistic model checking, which is able to compute probabilistic quantified properties pertaining to energy consumption, lifetime expectancy, and response time. The preliminary results confirmed significant savings in energy consumption while

Table 5.6.: Impact of varying the selectivity and accuracy margins of the virtual sensors on the event-miss probability and the lifetime

Accuracy	Selectivity	Lifetime (days)	Event-miss Probability
0.1	0.1	993.069	0.838
0.2	0.2	989.889	0.754
0.3	0.3	989.02	0.679
0.4	0.4	983.032	0.499
0.5	0.5	824.488	0.451
0.6	0.6	783.057	0.368
0.7	0.7	498.785	0.227
0.8	0.8	443.325	0.208
0.9	0.9	412.277	0.135
0.10	0.10	380.536	0.019

retaining an acceptable average response time.

Based on design-time knowledge, an ontology can generate rules for switching between real and virtual sensors during run-time. This mechanism constantly monitors the relationship between sensors, their observations and environmental conditions. The approach was illustrated by an object tracking scenario by stretching (or compressing) a seismic trace until it best matched one of the reference traces. The results of experiments over real nodes have confirmed a virtual sensor's superiority: it saves circa 99.93% of the energy needed; in the particular scenario. This amount of saved energy highly depends on the "energy-cheap" sensors and the application scenario. Moreover, a benchmark for the reliability parameters versus the lifetime and the event-miss probability has been constructed via large-scale simulation. From this benchmark, we estimate values of selectivity and accuracy margins that have to be used to switch between a real sensor and a virtual sensor in a reliable manner. Hence, we ensure that virtual sensors are only adopted whenever their qualities are high in the light of the environmental dynamics.

6. QoS Improvement with Lifetime Planning

6.1. Introduction

Wireless Sensor Networks (WSNs) have been recognized as promising tools to collect relevant, in-situ data for a wide range of application domains. Their decreasing cost and continuously improving capabilities are increasing the attention of the research and development communities toward benefiting from the spatially distributed sensors, which are capable of collecting large volumes of data about objects and spatial-temporal events of interest while operating unattended. The limited energy budget of wireless sensor node is typically the most critical challenge when designing WSNs. This constraint directly affects the network lifetime.

Consequently, the WSN technical literature is currently full of methods for solving the lifetime maximization problem [HY2008, YX2010, ZWLL2012]. However, plenty of real-world applications (tasks) were remarked which have known pre-defined lifetimes. Table 6.1 lists few WSNs applications and their lifetimes. Indeed, each application is already designed with an expected lifetime within which the network must properly function. For instance, the College of Atlantic and Berkeley University conducted field research on Great Duck Island (GDI) [MCP+2002]. Their objective was to explore the usage pattern of the nesting burrows when one or both parents alternate between incubation and feeding. The scattered WSN is designed to run for nine months. Hence, making use of this knowledge can highly improve relevant QoS parameters.

Table 6.1.: Various WSN applications [RM2004]

Project	Task lifetime	Influenced QoS metrics
GDI	7 months (breeding period)	Accuracy
ZebraNet	one year	Throughput, Accuracy
Glacier	several months	Packet delivery ratio
Ocean	4-5 years	Packet delivery ratio, real-time
Grape	several months (growth period)	Accuracy, latency
Avalanche	days (duration of a hike)	dependability
Vital Sign	days to months (hospital stay)	real-time, dependability, eavesdropping-resistance
Tracking	weeks - years (conflict duration)	stealth, tamper-resistance, real-time

This chapter discusses a novel strategy for QoS improvement, referred to as the *lifetime planning*. The core idea behind the proposed strategy is to save energy by deliberately reducing the operational lifetime beyond a given maximal required lifetime. Simultaneously, the nodes' lifetime has to meet the predefined time required for task completion. The amount of conserved energy is then utilized to improve the QoS provided. According to the application scenario, the end-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- and time-driven manner. Namely, the proposed strategy reconfigures the low-level parameters, such as transmission power, duty cycle, and sampling rate, so that event-miss probabilities are minimized along with optimizing the sampling rate for the continuous flow of information.

Figure 6.1 depicts a modified architecture of a general sensor node with the inclusion of a lifetime planner. The function of this planner is to reconfigure the system parameters in accordance with the predefined user requirements and the environmental dynamics. Based on the design-time knowledge, QoS maximum and minimum boundaries are estimated. At run time, low-level parameters are updated – within the allocated range – in the light of changes in the internal context information including the residual energy, the packets generation rate, etc. Moreover, external information – such as the environmental parameters – is also considered by the adaptation mechanism. In this work, a self-adaptive framework is exploited to dynamically adapt the low-level parameters, such as transmission power, duty cycle, processor frequency, etc. Such modulations aim at improving the high-level QoS parameters, such as latency, data fidelity, throughput, etc.

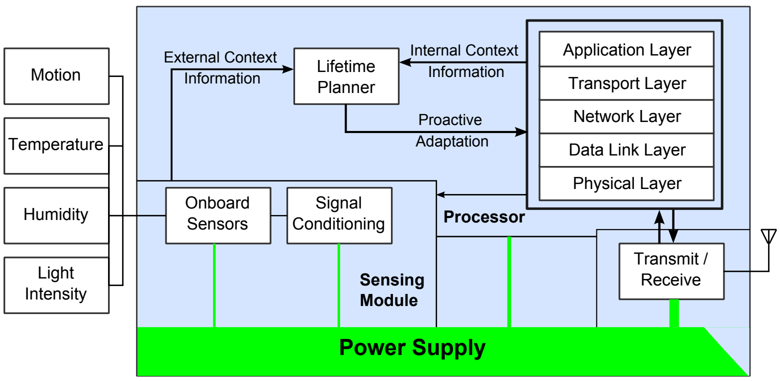

Figure 6.1.: A lifetime planner in a general sensor node architecture

The autonomic Monitor-Analyze-Plan-Execute (*MAPE*) Model [GSC2009] is the basis of the proposed technique. This architecture is used in this context for exploiting the environment dynamics. Specifically, each node has to monitor changes in the relevant context information. Afterward, a reasoning engine has to check for QoS boundary violation. Whenever a QoS deviation occurs, the low-level parameters are therefor updated. Actually, this chapter considers the cooperative nature of WSNs through introducing a hierarchical structure of the *MAPE* control loop. Throughout this chapter, a cluster-tree WSN topology is considered. Accordingly, system updates – emerging at each sensor node – are solely executed whenever they are approved by a corresponding cluster head. This limitation is intuitive to prevent any selfish strategy.

As a proof of concept, an office monitoring scenario has been engineered. Accordingly, indoor environment dynamics inherit in such scenario are exploited to validate the proposed approach. A network of *Tmote sky* sensor nodes is deployed in the *Contiki OS* [DGV2004] network simulator – *Cooja* [Öst2006]. The goal of such scenario is to determine the system performance throughout the entire lifetime.

To sum up, this chapter provides the following contributions:

- proposing a novel strategy of lifetime planning and QoS provision in WSNs,
- introducing a hierarchical lightweight *MAPE*-based self-adaptation framework,
- refining a QoS analytical model which maps between several controllable low-level parameters and the high-level service quality metrics,
- validating the new model by performing simulations in an environment that is consistent with the model, and
- evaluating the outcome of lifetime planning over static heuristics and unplanned adaptations using the network simulator *Cooja*.

The remainder of this chapter is organized as follows. Section 6.2 discusses the previous efforts for maximizing lifetime and the adaptation mechanisms. Section 6.3 introduces the proposed strategy of planning the entire lifetime. Moreover, the system model and the problem definition are given. Afterward, the analytical QoS model is introduced and validated in Section 6.4. The office monitoring scenario and its dynamics are elaborated in Section 6.5. Section 6.6 shows a comparative analysis in terms of reliability, delay, and lifetime. Finally, a conclusion and future work are given in Section 6.7.

6.2. Related Work

In this section, a motivation to the proposed solution is given via exploring the existent solutions and their disadvantages.

6.2.1. QoS with lifetime

As one of the conflicting constraints with energy consumption in WSNs, QoS provision has been essentially addressed in the recent research. For example, the authors in [RLCL2009] considered the trade-off between network performance optimization and lifetime maximization in real-time WSNs as a joint non-linear optimization problem. Based on the solution of such a mathematical optimization problem, they developed an on-line distributed algorithm to achieve the appropriate trade-off. Even though, the authors in [RLCL2009] spared no effort to balance the network performance and lifetime, yet the practical applications might require an expected lifetime and a high level of QoS metrics. Under this circumstance, always balancing the trade-off would not be sufficient per se.

The authors in [CSE2011] designed an adaptive fault-tolerant QoS control algorithm to satisfy the application QoS requirements in a query-based WSNs. They developed a mathematical model where the lifetime of the system is considered as a system parameter, to determine the optimal redundancy level that could satisfy QoS requirements while prolonging the lifetime. However, the authors in this work aimed at maximizing the lifetime while maintaining the QoS, especially based on the fault-tolerant applications. The network dynamics in their application was not fully considered. Furthermore, the authors in [JHH+2013] adopted a self-adaptation framework, referred to as *Monitor-Execute-Plan-Analyze* (MAPE), to optimize the energy consumption. They developed a probabilistic approach that estimates part of the QoS – the residual energy – to conserve the transmission energy, and thus extending the sensor lifetime. Their approach is based on a *hidden Markov chain* and *fuzzy logic*, consisting of learning and predicting steps. However, they did not pay enough attention to the QoS provision in WSNs. In addition, the relationships between the remaining power in a battery, considered as a QoS metric by the authors, and the QoS metrics on the network scale are still missing.

6.2.2. Frameworks for Self-adaptation

Several strategies of designing frameworks for self-adaptation are addressed in the literature. For instance, the authors in [MGR2009] design a tuning algorithm based on a *Markov Decision Process* (MDP). This tuning is scheduled to meet the dynamic user requirements and the environmental changes. Hence, the network is flooded with messages going back and forth. Moreover, they consider an energy consumption minimization, which in turn increases the number of reconfigurations. The MDP algorithm is evaluated in MATLAB, so that the overhead of porting it to real sensor nodes is not investigated. Alternatively, a proactive mechanism is proposed in [ASB+2014] 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.

An adaptation technique – similar to the proposed approach – is introduced in [SVS+2011]. It 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 communicate commands among sensor nodes. However, the adoption of network routing based on flooding completely contradicts with the goal of energy savings. In [JH2013], the authors focus on the self-adaptation mechanism. They adopt 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 the proposed self-adaptation mechanism, the energy draw is balanced via planning the reconfigurations at each node. Then, the cluster heads utilize their knowledge of the cluster status to approve or disallow the new reconfigurations.

In this chapter, lifetime planning is presented in which the drawbacks of the aforementioned methods are sidestepped. First, the algorithm makes use of design-time knowledge and relies on a lightweight procedure during run-time. In the next section, several definitions are briefly described including WSN lifetime, QoS boundaries, as well as the network model which we need to concisely describe the proposed approach. Moreover, the assumptions made are explained and the problem of QoS degradation in response to lifetime maximization is defined.

6.3. Lifetime Planning

In this section, the basic idea of the proposed strategy of WSN lifetime planning is explained. The main ideas are to 1) save energy by limiting the lifetime below a maximal possible lifetime, and 2) exploit the design-time knowledge for controlling application-relevant QoS metrics. Figure 6.2 depicts the idea of incorporating lifetime planning into each sensor node. 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. Moreover, it can be utilized to engineer the QoSs' lower and upper boundaries. For instance, office monitoring applications do not have to provide extremely high service quality during holidays or at night. Based on the scenario dynamics, the instantaneous service qualities are confined between their boundaries, $\bar{Q}_{upper}$ and $\bar{Q}_{lower}$.

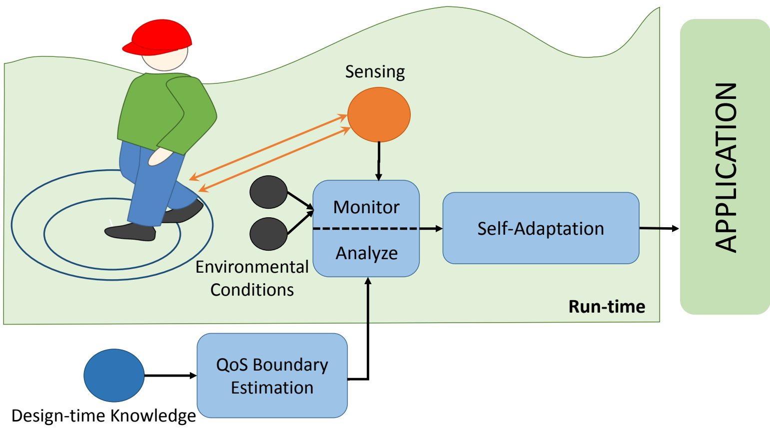

Figure 6.2.: Building blocks of lifetime planning in a typical sensor node

The autonomic *MAPE* reference model [GSC2009] is utilized to design a self-adaptive QoS controller. 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 interesting events. Accordingly, each sensor node finds a course of action to adapt the low-level parameters once a problem has been detected.

6.3.1. Comparative Analysis

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

Each cluster C_i has a 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 size. Clusters are formed based on various criteria such as communication range, number and type of sensors, and geographical location. 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.

All nodes s_{ij} are allocated an equal amount of energy E_0. Hence, the worst case where the network has neither energy harvesters nor special nodes with a larger energy budget is considered. 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 end-user can provide a set of QoS requirements $Q = \langle q_1, q_2, \ldots, q_n \rangle$ which are used for adapting the parameters of P. Moreover, the end-user has to define the task lifetime L_{task} in which the network must function with a best-effort QoS. Accordingly, the WSN lifetime is defined as the time span from deployment till the end of the intended task.

The main research question, provoked in this work, is whether lifetime maximization should be a design goal or not? To provide an answer to this question, the lifetime maximization and the lifetime planning strategies have to be modeled. For the former, the lifetime is typically maximized by "squeezing" other quality metrics. The second row of Table 6.1 lists several QoS parameters of real-world WSN applications. These parameters are severely affected by maximizing the operational lifetime. However, a maximum lifetime sometimes is beyond the planned lifetime, leading to energy waste. Such an energy waste can be modeled as an area of a rectangle, as shown in Figure 6.3. The height of this rectangular represents the provided QoS level. Accordingly, the provided QoS after the task's period (hashed in the figure) $\bar{\Delta}$ can be characterized by

$$\bar{\Delta} := \bar{Q}_{lower}(L_{max} - L_{task}). \tag{6.1}$$

where $\bar{Q}_{lower}$ is the provided QoS level over the lifetime L_{max}. L_{task} denotes the task's actual lifetime. Level $\bar{Q}_{lower}$ is considered as the worst case, where the service can be further improved. As shown in the figure, the dashed area is transferred (denoted by a red arrow) to the task time $0 < t < L_{task}$ to enhance the service quality (green rectangular). To sum up, sensor nodes, in many cases, consume superfluous energy beyond the lifetime L_{task} at the expense of relaxing other service qualities. In the sequel, the architecture of the proposed framework to overcome the aforementioned problem is discussed in more detail.

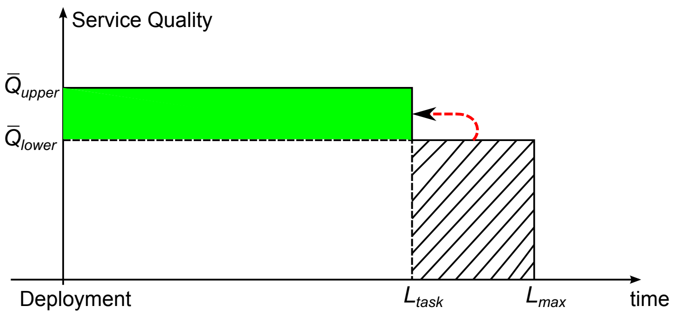

Figure 6.3.: Service quality in case of lifetime maximization and lifetime planning

The proposed lifetime planning exploits the residual energy – energy that still exists after the WSN task has been already accomplished – to improve the QoS provided in lieu of wasting it beyond the lifetime L_{task}. Simultaneously, this planning improves the provided QoS. Without planning, the average QoS provided resembles a lower boundary during the task lifetime. On the contrary, the planning strategy affords an upper level of QoS by investing the waste energy $\bar{\Delta}$ over the lifetime L_{task}. Equation 6.2 expresses the new level of the provided QoS $\bar{Q}_{upper}$ over the lifetime L_{task}.

$$\bar{Q}_{upper} := \left(\bar{Q}_{lower} + \frac{\bar{\Delta}}{L_{task}}\right) \tag{6.2}$$

In the sequel, a speedup ratio is estimated to examine the superiority of lifetime planning over lifetime maximization. A speedup ratio $\mathcal{S}$ between the naïve quality – obtained by solving the maximization problem – and the quality provided by lifetime planning is defined as given in Equation 6.3. The total power consumption is a "good" 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 power consumed during the lifetime L_{task}, where E_w is the unused energy when a sensor node dies. Similarly, the power consumed 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

$$\mathcal{S} := \frac{\bar{Q}_{upper}}{\bar{Q}_{lower}} \cong \frac{\int_{t=0}^{L_{task}} P_2(t)dt}{\int_{t=0}^{L_{max}} P_1(t)dt} \tag{6.3}$$

where $P_2(t)$ and $P_1(t)$ are the instantaneous power consumptions as functions of time. Knowing that the energy consumed in both cases is identical and that $L_{max} > L_{task}$ holds, it can be extrapolated that the speedup ratio is larger than 1 (i.e. $\mathcal{S} > 1$). Hence, it is a beneficial policy to adopt lifetime planning while designing WSN applications.

After analyzing the idea of saving energy in lieu of maximizing lifetime, the idea of globally controlling self-adaptation by CHs and by the base station is discussed. In the next section, the question of how to implement the *MAPE*-based self-adaptation algorithm within the network is answered.

6.3.2. Hierarchical Self-adaptation

In this work, the self-adaptation mechanism and system performance of each sensor node under lifetime planning are investigated. However, the self-adaptation mechanism from the network level has to be addressed. Accordingly, a novel hierarchical architecture of the self-adaptation algorithm is introduced. Figure 6.4 depicts the *MAPE* loop assignment within a clustered network. 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 depicted in the figure, both CHs and their children implement all components of the *MAPE* loop, whereas the base station can only analyze and plan for the CHs. This assignment scheme is rational since it avoids superfluous control messages.

Under these settings, modifications of the low-level parameters, generated at each sensor node, have to be confirmed by the corresponding CH, which is "data-rich" about its child nodes. This constraint primarily sidesteps the execution of plans that harm the inter-sensor cooperation. The base station, in the proposed setting, gives permission to action plans proposed on the CH level. This hierarchy of information flow can highly improve the nodes' self-adaptations with emphasis on the global network parameters.

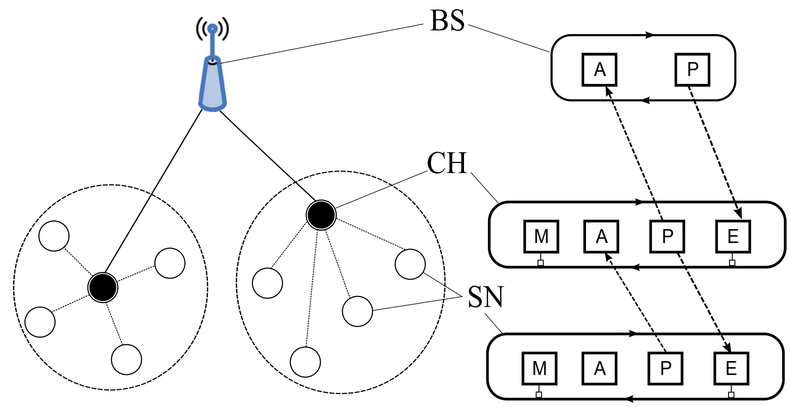

Figure 6.4.: An architecture of a proactive network

The mechanism by which cluster heads judge the proposed plans is left for further research, as this chapter concentrates on the node-level mechanism.

In general, two research questions are always raised when dealing with proactive adaptations: 1) how to select the best fitting target configuration? and 2) how to deal with conflicting objectives? However, lifetime planning 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 QoS conflicts are sidestepped through the lifetime. To confirm this hypothesis, the quality metrics have to be investigated to discover possible conflicts. In the sequel, metrics of interest such as communication reliability, delay, and energy consumption are investigated.

6.4. QoS Modeling

In this section, a novel QoS model via relating some QoS parameters to low-level parameters P is provided. The goal is to probe for possible conflicts between the various service qualities when adopting lifetime planning. For the analysis, three major service qualities, are accounted for, namely reliability, delay, and energy consumption. These mapping functions are devoted to control the node's behavior at run-time. Specifically, the proposed QoS model is based on merging the Hoes's model [HBT$^+$2007] and the probabilistic Markov chain model of the IEEE 802.15.4 MAC protocol [ISA2011], referred to as Park's model [PMFJ2013].

6.4.1. Analytical Model

Communication Reliability In the proposed 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, the model differentiates itself from Park's model via 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 metrics $\mathcal{R}(s_{ij})$ is expressed in terms of the signal to interference noise ratio *SINR* and the approximated probability of successful packet transmission $\tilde{R}(s_{ij})$ as given in Equation 6.4.

$$\mathcal{R}(s_{ij}) = \left(1 - Q\left(\sqrt{2 \times SINR}\right)\right)^b \times \tilde{R}_{mac} \tag{6.4}$$

where b is the packet size in bits and $Q(\cdot)$ is the tail probability of the standard normal distribution. The second part of Equation 6.4 represents the probability of successful packet reception. It mainly depends on the packet generation rate and the operational duty cycle. The term *SINR* has a direct relationship with the transmission power via the following formula:

$$SINR = \left(\frac{P_{rx}}{N}\right) = \frac{K_r \times P_{tx} \times (d_0/d)^r}{K \times T \times B} \tag{6.5}$$

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.

Delay In Park's model, the delay is defined as the time interval from the time instant a packet is at the head of its MAC queue and ready to be transmitted until transmission was successful and an acknowledgment has been received. In contrast, the proposed model extends this notion via considering the time span from the arrival of a nearby target until its detection as well. Equation 6.6 expresses 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 $\tilde{D}$.

$$\mathcal{D}(s_{ij}) = \left(\frac{1}{r_s} + D_s\right) + \tilde{D} \tag{6.6}$$

Energy Consumption In this work, it is assumed that the radio transceiver is in sleep mode during the backoff mechanism specified by the IEEE 802.15.4 standard [ISA2011]. Moreover, it is presumed that packet transmission and reception have identical energy consumption. Accordingly, the total energy consumption $\tilde{P}(s_{ij})$ can be expressed as given by

$$\mathcal{P}(s_{ij}) = f(s_{ij})\left(E_s \times r_s + P_{mcu}\right) + E_{radio}\left(r_o + \kappa \times r_i\right) \tag{6.7}$$

where E_s is the sensing module energy consumption, P_{mcu} is the power consumed for processing the sampled data, and $f(s_{ij})$ is the transceiver duty cycle. 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 energy consumed in sleep mode, the term E_{radio} reflects the average power consumed during channel sensing, MAC backoff state, and packet transmission including both, successful transmission and packet collision.

6.4.2. Analytical Model Validation

In this section, the validation of the proposed QoS analytical model is discussed. The aforementioned analytical model was implemented in MATLAB. For the simulation, the *Cooja* simulator of *Contiki OS* was used. The contention loss and the transmission delay in the various simulation scenarios were estimated. Simultaneously, the parameters in the MAC layer that would be applied later in the proposed analytical model in MATLAB were measured, such as the probability of the first clear channel assessment (CCA) α and the probability of the second one β. During the simulations, the *Powertrace* application [DEFT2011] was utilized to estimate the power consumption. Table 6.2 presents the general parameters for the simulations.

Table 6.2.: Simulation Parameters

Parameter	Value
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

Considering the packet error ratio (PER) and the contention loss ratio, the simulations were carried out by setting different values of parameters in *Cooja*, such as the number of nodes, the channel check rate, and the idle time interval, based on the equations stated above. For the number of nodes, the values of 4, 8, and 16 were selected. Besides, the channel check rate was set to 8 Hz, 16 Hz, 32 Hz, and 64 Hz, respectively, in each simulation. For simplicity, the idle time interval was fixed to 1000 milliseconds for all simulations. In this case, 12 ($3 \times 4 \times 1$) simulations in total were run. Taking the matter of time into account, each simulation was run 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, which means the channel is idle at the moment, then the node transmits that data packet to the receiver (i.e., cluster node).

Reliability In the proposed 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 the data packet is discarded as a result of a contention of the communication medium.

By utilizing the measurements from previous simulations, the average packet delivery ratio (PDR) of all the sensor nodes is estimated in different simulation scenarios. Then, using the reliability formula (Equation 6.4), the average PDR of the analytical model is obtained in the corresponding scenario via MATLAB. Figure 6.5 shows the deviation of the average PDRs between the analytical model and the simulation.

As Figure 6.5 illustrates, 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

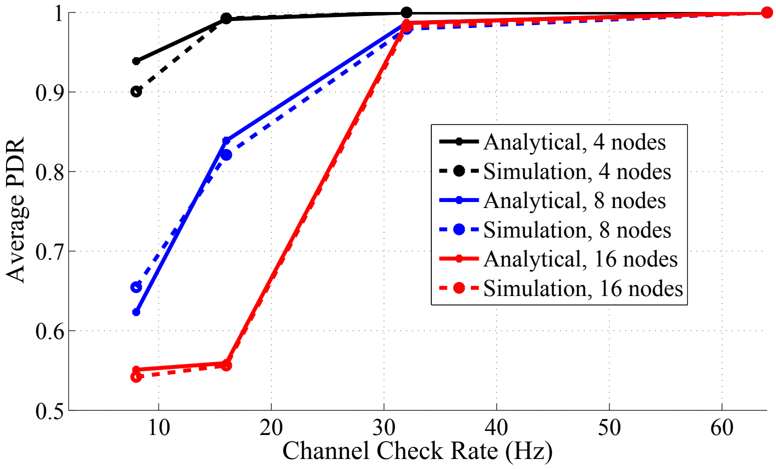

Figure 6.5.: Average packet delivery ratio versus channel check rate

rate, the difference between the analytical and the simulated results is still less than 4%.

Delay As stated in Equation 6.6, the average transmission delay in the proposed 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 and the durations of sampling and 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. Hence, the validation of the average transmission delay $\tilde{D}$ is mainly discussed.

Similarly, the average transmission delay of all the sensor nodes is obtained from previous simulations in different scenarios. By using the previous measurements of parameters in the MAC layer, the average transmission delay is obtained in the analytical model. Figure 6.6 illustrates the deviation of average transmission delay between the analytical model and the simulation.

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 16 nodes scenario, the difference between the analytical model and the simulated one is still less than 10 milliseconds. To conclude, the proposed analytical model considers several layers of the network stack. According to the validation work, the model is accurate enough to be invoked for detecting possible QoS conflicts due to lifetime planning.

Obviously, 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. Certainly, these trade-offs are implicitly overcome in lifetime planning. This fact emerges from intentionally reducing the lifetime below the maximal possible lifetime. As a proof of concept, the next section discusses the implementation of an office monitoring scenario. This case study is investigated to examine the network performance with and without lifetime planning.

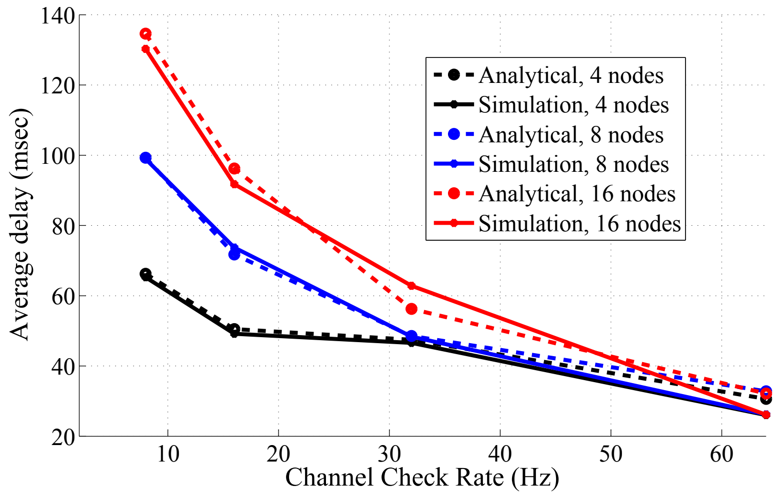

Figure 6.6.: Average transmission delay versus channel check rate

6.5. Case Study: Office Monitoring Scenario

Energy efficiency in smart buildings is an important application of WSNs. To conserve energy, indoor online data has to be collected as a basis for dynamically controlling the lighting and heating systems. Examples of such data include the location of persons and their activities. In this chapter, an active indoor localization scenario is considered. A network of *TelosB* sensor nodes has been simulated in the *Cooja* simulator of the *Contiki OS*, as depicted in Figure 6.7. Specifically, the testbed consists of eleven static sensor nodes (SSNs), which monitor the observed area, measure environmental parameters and forward detection packets to the sink. The network is divided into three clusters to avoid flooding the network of excessive control packets. Another set of mobile sensor nodes (MSNs) represents the indoor traffic. These MSNs broadcast identification packets to the nearby SSNs. To localize the mobile nodes, the sink node processes the received signal strength (RSS) values for communication links between the MSNs and the nearby SSNs.

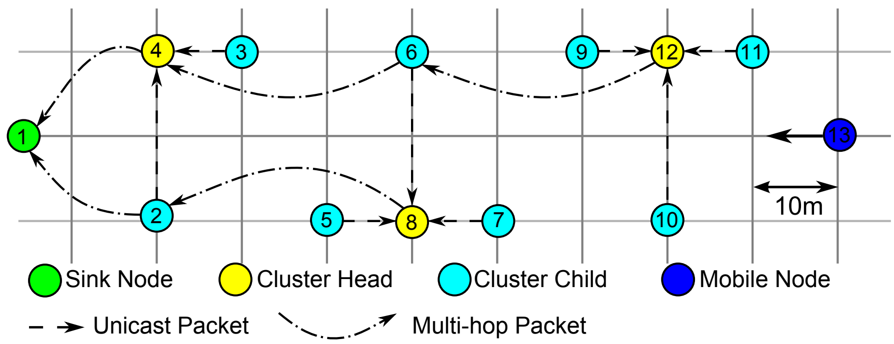

Figure 6.7.: An office monitoring testbed implemented in *Cooja* simulator

6.5.1. Scenario Dynamics

In real 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 lifetime planning. For instance, mobility of a MSN is highly affected by a person's status. For instance, detecting whether the person is "stationary" or "walking," triggers a set of reconfigurations. During the "walking" activity, the connectivity is continuously altered based on the distance between the SSNs and the MSN. Hence, the transceiver duty cycle f and the sampling rate r_s have to be quickly modulated. Moreover, the interference is waving when walking. Thus, the transmission power T_{tx} should also be modified to overcome the imposed interference.

The daily pattern of day and night is also of interest to the proposed self-adaptation mechanism. The hallway has mostly low traffic at night. Accordingly, less data has to be reported to the sink node at those times. Hence, service quality could be customized to save energy. In the next section, reasoning engines for modulating QoS parameters are discussed.

6.5.2. 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 priori knowledge of the application scenario leads to define a set of possible events. Hence, reactions can be earmarked even before deployment [SVS$^+$2011]. Nevertheless, environmental dynamics typically results in unexpected events which could also be exploited. Examples of such engines are the *Constraint-Satisfaction Problem* [GRF$^+$2012], *MDP* [MGR2009], *Fuzzy inference* [MPH2007], and *Event-Condition-Action* (ECA) rules [ST2009].

In this work, ECA rules are used to reconfigure the nodes. Such rules are simple and have low computational overheads. Formerly, ECA rules were used in active database systems [ST2009]. Afterward, they have been widely utilized as a flexible strategy to support the management, the reconfiguration and the execution of reasoning rules. In general, an ECA rule has the following form:

```
Rule <RuleName>
WHEN <EventExpression>
 <Condition 1><Action 11><Action 12> ...
 <Condition 2><Action 21><Action 22> ...
    ...
 <Condition n><Action n1><Action n2> ...
ENDRULE<RuleName>.
```

The above rule comprises 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 true, 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 parameters demands the modulation of manifold low-level parameters. Hence, an ECA rule has to execute multiple actions per single condition.

In the next section, the performance evaluation is introduced through a comparative study between lifetime planning, blind adaptation, and lifetime maximization strategies.

6.6. Performance Evaluations

An experimental study on the office monitoring scenario has been performed to evaluate the proposed lifetime planning. As explained earlier, the goal of lifetime planning is to improve the WSN's service qualities with adequate network lifetime. Hence, a proactive adaptation mechanism based on the *MAPE* framework has been adopted. To fulfill this goal, the following research questions have to be answered.

a) Does lifetime planning improve the provided service qualities compared to static heuristics and blind adaptation?

b) Does the actual network lifetime meet the application requirements?

c) Can the quality parameters be confined within the QoS boundaries throughout the entire lifetime?

As aforementioned, a scenario of office monitoring has been engineered for evaluation purposes. The inherent dynamics in such a scenario are to be exploited to show the effect of planning the service quality levels throughout the entire lifetime. Figure 6.7 shows the layout of the proposed office monitoring scenario. The simulator runs on a virtual machine with a 2.5 GHz processor and 8 GB RAM using an Ubuntu OS.

For a comparative analysis, lifetime planning has been contrasted to two different strategies, namely

- *blind adaptation* [SVS+2011], it utilizes the application scenario details to switch at run-time between different modes of operations to only reduce the energy consumption;
- *Static heuristics* They represent fixed strategies in which the controllable parameters P are assigned the minimal or maximal values, respectively. Examples of this strategy are lifetime minimization ($L < L_{task}$) and lifetime maximization ($L \geq L_{task}$).

Below, implementation details of both, blind adaptation and lifetime planning are discussed in more detail.

6.6.1. Blind Adaptation

The authors in [SVS+2011] introduced a method to exploit the design-time knowledge of application scenario dynamics in order to design a proactive run-time reconfiguration method. At design-time, modes of operation and values of the controllable parameters of the network stack have to be defined. Hence, the parameters are adapted in response to some expected events, e.g. the building occupancy during day and night. The main problem of such an approach is to only deal with energy consumption. The adaptation framework, designed in [SVS+2011], is not planned in the light of other relevant service qualities. Hence, invoking it for the comparative study, highly clarifies the advantages and disadvantages of the proposed approach.

6.6.2. Lifetime Planning

Algorithm 8 introduces the main steps of applying lifetime planning. At design-time, lower and upper boundaries of QoS metrics have to be estimated in the light of the expected task lifetime L_{task} and the initial budget E_0. For the boundaries $\left\langle \bar{Q}_{lower}, \bar{Q}_{upper} \right\rangle$, they are determined using the QoS analytical model together with simulating the application scenario, as stated in line 3. Considering only the analytical model is not practical due to the run-time loss in the upper layers such as collisions and failure of acknowledgment packet reception. Moreover, probabilities of the busy clear channel assessment (CCA) have to be used during the model evaluation. Hence, the network is simulated in *Cooja* for each sensor node, as depicted in lines 1-4. To this end, several simulation rounds have to be executed. For each sensor node S_{ij}, values of the low-level parameters $P = \langle r_s, f, P_{tx} \rangle$ are determined in a way that maximizes the operational lifetime. Corresponding values of the QoS metrics are designated as the low-level QoS boundary $\bar{Q}_{lower}$. Similarly, the upper-level QoS boundary $\bar{Q}_{upper}$ is determined when the obtained lifetime approaches the task lifetime.

Algorithm 8 The lifetime planning algorithm

Require: task lifetime L_{task}, energy budget E_0
    // *Estimate the QoS boundaries at design-time*
1: **for** i such that $0 \leq i < \mathcal{M} - 1$ **do**
2:     **for** j such that $0 \leq j < (\mathcal{N}/\mathcal{M}) - 1$ **do**
3:         **Estimate** the parameters $P \leftarrow f(E_0, L_{task})$ where $P = \langle r_s, f, P_{tx} \rangle$
4:         **determine** $\left\langle \bar{Q}_{lower}, \bar{Q}_{upper} \right\rangle$
5:     **end for**
6: **end for**
    // *Run-time processing*
7: **monitor** the building occupancy (*traffic and speed*)
8: **if** an ECA rule is fired **then**
9:     **update** the parameters P ▷ mathematical model
10: **end if**

During run-time, the ECA rules are continuously evaluated to check for environmental changes. Specifically, four rules have been designed. Two of such ECA rules monitor the environmental events Whereas, the other rules are created to limit the service qualities between the predefined boundaries. To design the ECA rules in the light of the office monitoring scenario, the events, conditions, and actions have to be clearly defined. Table 6.3 summarizes the definitions of parameters used in the ECA rules. The numbers listed in this table regarding the channel checking rate r_c and the transmission power P_{tx} are standard and are obtained from the TelosB SN's datasheet. The traffic volume and speed of mobile nodes are presumed for this scenario.

Scenario Dynamics Rules The following rule considers the case of moving traffic in the corridor, like the one delineated in Figure 6.7. The rule is designed to be fetched whenever at least a single person passes through this corridor. Then, the engine has to evaluate conditions so that the optimal action is taken. As explained above, two criteria

Table 6.3.: Parameter definitions

	low	medium	high
channel checking rate r_c (Hz)	8	32	64
transmission power P_{tx} (dBm)	-7	-3	0
traffic (number of MSNs)	1	–	4
speed (m/sec)	0.5	–	1

are considered here which in turn define four conditions. Assume linguistic control levels such as low ("L"), medium ("M"), and high ("H"). Values of such variables are defined in Table 6.3. The first rule is fired when at least one MSN is detected. Further details have to be considered. The traffic size and speed are to be used for fine-grained parameter adjustments. For instance, the first condition in the "WalkingState" rule is designed to check if single MSN is moving at a low speed. In this case, the MSN can be monitored even if the transmission power P_{tx} and the channel checking rate r_c are set to low values.

```
Rule  <WalkingState>
WHEN <at  least  one  person  is  detected>
IF  traffic == low && speed is low  THEN
P_tx is low  && r_c  is  low
IF   traffic == high && speed is low  THEN
P_tx is medium    && r_c  is  medium
IF   traffic == low && speed is high  THEN
P_tx is medium    && r_c  is  high
IF   traffic == high && speed is high  THEN
P_tx is high    && r_c  is  high
ENDRULE <WalkingState>
```

The *StationaryState* rule, shown below, is fired whenever a mobile node is frequently detected and successive RSS values are identical. In this case, the controllable parameters are relaxed based on the traffic status.

```
Rule  <StationaryState>
WHEN <a person  is  constantly detected>
IF  traffic == low   THEN
P_tx is low    && r_c  is  medium
IF   traffic == high  THEN
P_tx is medium   && r_c  is  medium
ENDRULE <WalkingState>
```

QoS Control Rules The *ReliabilityState* rule is continuously fetched to modify the packet delivery ratio in the light of the pre-assigned lower and upper boundary. As clarified in Chapter 6, two boundaries are estimated, during design-time, for each QoS parameter. These boundaries are determined through merging the QoS analytical model and also simulating the WSN scenario.

```
Rule  <ReliabilityState>
WHEN  <reliability is out of range>
IF  R =< lower_boundary    THEN
P_tx is increased    && r_c  is  increased
IF  R >= higher_boundary     THEN
P_tx is decreased    && r_c  is  decreased
ENDRULE <WalkingState>
```

Similarly, the *DelayState* rule, shown below, frequently modifies the detection delay in accordance with the residual energy consumption and with the status of the instantaneous delay. As long as the encountered delay lies between the predefined upper and lower boundaries $\left\langle \bar{Q}_{lower}, \bar{Q}_{upper} \right\rangle$, no action is taken. Otherwise, the rule modifies the transmission power P_{tx} and the channel checking rate r_c.

```
Rule  <DelayState>
WHEN  <delay is out of range>
IF  D =< lower_boundary    THEN
P_tx is decreased    && r_c  is  decreased
IF  D >= higher_boundary    THEN
P_tx is increased    && r_c  is  increased
ENDRULE <WalkingState>
```

Table 6.4 summarizes the operational mode (rows) and all possible scenarios (columns) for two static heuristics, blind adaptation, and lifetime planning. In fact, adopting general criteria – such as the traffic size and the speed of mobile nodes – mostly covers all possible events. The settings are classified in the light of a mobile node's state, namely 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 checking rate r_c, which substitutes the duty cycle. For blind adaptation, the values, indicated in the table, have been selected to reduce the power consumption, as proposed in [SVS$^+$2011]. Alternatively, the values for lifetime planning have been derived based on the required lifetime L_{task} via the mapping functions. Below, the obtained results are discussed in the context of the aforementioned research questions.

6.6.3. Evaluating the QoS improvements

In this section, the impact of applying lifetime planning, blind adaptation, and static heuristics on the service qualities is examined. Figure 6.8a delineates the lifetime of cluster heads and children for each strategy. The horizontal axes comprise the ID number of cluster heads and children, according to Fig. 6.7. The average lifetime obtained with lifetime planning is about 41.6% and 54.5% less than the other methods. Nevertheless, the achieved network lifetime (approximately 100 days) meets the planned task lifetime (green line). Remember that this task lifetime was used for estimating the QoS boundaries $\left\langle \bar{Q}_{lower}, \bar{Q}_{upper} \right\rangle$. These results are confirmed by Fig. 6.8d which shows the radio duty

Table 6.4.: Mode selection for office monitoring scenario

Local mode (MSNs)		Walking (traffic, speed)				Stationary (speed)	
Local mode (SSNs)		Static				Static	
Scenarios		(Low, Low)	(Low, High)	(High, Low)	(High, High)	(Low)	(High)
Number of MSNs		1	1	4	4	1	4
Speed (m/sec)		0.5	1	0.5	1	–	–
Max: $L \geq L_{max}$	P_{tx} (dBm)	-7	-7	-7	-7	-7	-7
	r_c (Hz)	8	8	8	8	8	8
Min: $L < L_{task}$	P_{tx} (dBm)	0	0	0	0	0	0
	r_c (Hz)	64	64	64	64	64	64
Blind: $L < L_{max}$	P_{tx} (dBm)	-7	-7	-7	-3	-7	-7
	r_c (Hz)	8	16	16	64	8	16
Planning: $L_{task} < L < L_{max}$	P_{tx} (dBm)	-7	-3	-3	0	-7	-3
	r_c (Hz)	8	32	64	64	32	32

cycle of each node as a percentage. With lifetime planning, the sensor nodes activate their transceivers for longer time than for blind adaptation. This additional energy cost contributes to the communication reliability and the detection delay.

Figure 6.8b and Figure 6.8c show a comparison between the four strategies in terms of the average *packet delivery ratio* (PDR) – representing a realistic measure of the reliability $\mathcal{R}$ – and the average delay $\mathcal{D}$ in milliseconds. The PDR is defined as a ratio between the number of received packets to the number of transmitted packets. In these experiments, we focus on the communication link between cluster heads and their children. Accordingly, quality values of the sink and the cluster heads (Node 1, 4, 8, and 12) have been zeroed in the figures. Both quality metrics have been obtained via averaging the results over several runs for the various scenarios, listed in Table 6.4.

As expected, lifetime planning achieves highly better reliability and delay than the other approaches, as can be seen in Fig. 6.8b and Fig. 6.8c respectively. Particularly, lifetime planning has approximately 9.6% and 20% higher PDR than the blind adaptation and maximization method, respectively. Similarly, lifetime planning has about 53% and 78% less delay than the other methods. This excel is reasonable due to spending more energy in case of lifetime planning.

6.6.4. Evaluating the QoS boundaries

Finally, it is crucial 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 6.9a and 6.9b, 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 delivery ratio, the lifetime planning (in blue) values are confined between the two gray thresholds, as shown in Figure 6.9a. Alternatively, blind adaptation (in red) is reduced without any restrictions to reduce the energy consumption. Figure 6.9b shows a similar behavior for the delay metrics.

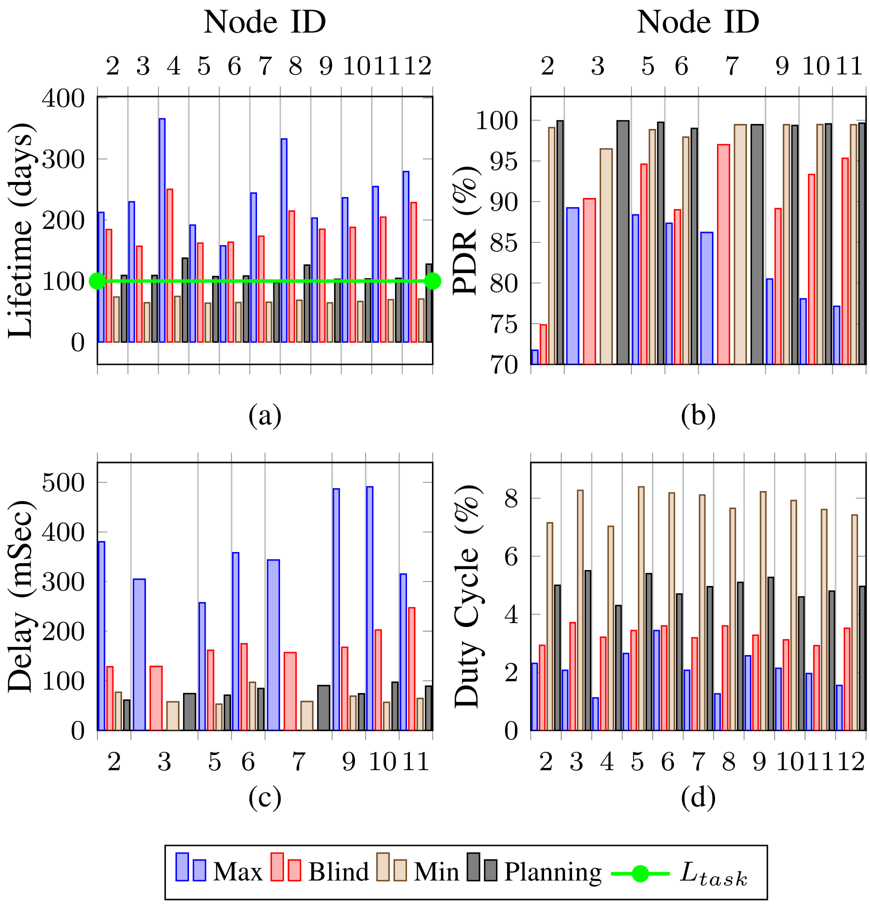

Figure 6.8.: Impact on of the strategies on (a) the lifetime, (b) the average delivery ratio, (c) the average delay, and (d) the transceiver duty cycle, for the office monitoring scenario

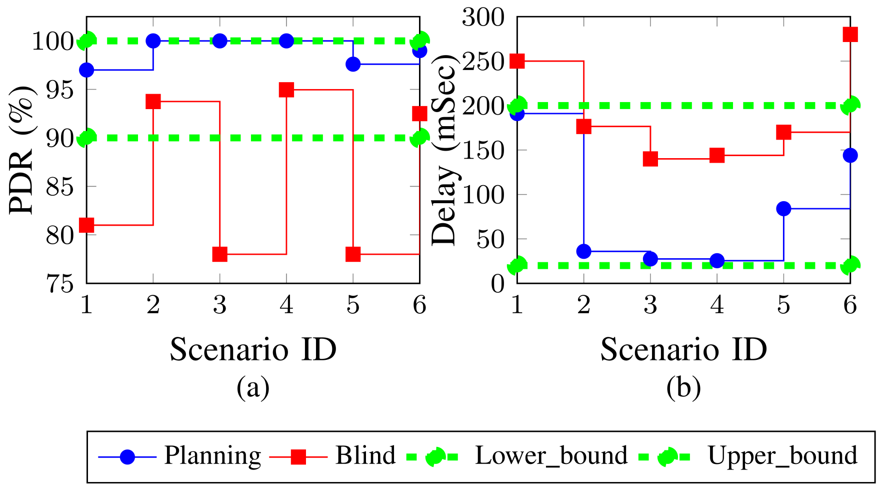

Figure 6.9.: (a) Delivery ratio and (b) average delay in case of lifetime planning and blind adaptations

6.7. Discussion

Upcoming applications require WSNs to meet application-specific performance targets such as reliability, delay, throughput, etc. In this chapter, a novel strategy of handling the QoS management in WSNs, referred to as *lifetime planning* is proposed. Different from conventional QoS designs, the proposed approach 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, lifetime planning is seen as a method for avoiding any possible conflicts between the QoS metrics. Furthermore, an office monitoring scenario was engineered and has been used to examine the proposed strategy. The results show that lifetime planning highly improves the QoS metrics. This profit came at the expense of reducing the WSN lifetime. But the new, shortened lifetime is still long enough to complete the assigned WSN task.

7. Summary & Outlook

As a conclusion, this chapter summarizes the contributions of this thesis and discusses possible future extensions.

7.1. Summary

This thesis advances the state of the art in WSN research by presenting concepts that deal with the natural trade-off between the operational lifetime and the application-relevant QoS metrics. In the following, each contribution is summarized.

As a first contribution, three extensive surveys were constructed. The first survey comprises the main energy consumers in WSNs. To understand how each energy consumer was tackled in the literature, the second taxonomy lists the main energy-efficiency methods in WSNs. The methods were classified into *data-oriented methods*, *node-oriented methods*, and *network-oriented methods*. The energy-efficiency taxonomy is unique where it comprises the most recent methods. We discuss how each energy-efficiency method deals with the QoS metrics that are significant to the WSN application. We concluded that most of these methods negatively affect the QoS metrics. This motivated toward reporting the QoS control methods in WSNs. This third survey divides the existent work into single- and multi-objective optimization methods. The multi-objective optimization methods are found to be the best-suited methods in the context of WSNs where many QoS metrics are engaged in each application.

The second contribution of this thesis is a novel strategy, referred to as an *energy-centric* strategy, for improving many QoS metrics while meeting the expected lifetime. The main goal is to avoid the computational complexity of the multi-objective optimization methods via an efficient WSN design procedure. The underlying idea behind the energy-centric strategy is to dynamically adjust the amount of saved energy – during run-time – in the light of the environmental dynamics. To resolve possible conflicts between the various QoS metrics, we utilize design-time knowledge of the application scenario to leverage self-adaptations. Implementing the energy-centric methods depends on the type of the application scenario. Therefore, classify the WSN applications into *time-based*, *event-based*, and *hybrid* application sets. As a proof of concept, the energy-centric strategy was evaluated through three different methods. Each of those methods targets a certain set of WSN application scenarios. In each method, the provided QoS were improved while offering adequate operational lifetime.

The third contribution of this thesis is a highly-precise data compression method, referred to as FuzzyCAT. Targeting the WSN time-driven applications set, the standard Fuzzy transform was successfully modified to decrease the recovery errors. FuzzyCAT compressor showed a comparable precision performance with other lossy compressors such as the wavelet transform, and the model-based methods. FuzzyCAT was designed to detect the input signal curvature and dynamically modifies the standard transform.

To resolve the delay problem associated with transform-based data compressors, we proposed to shrink the reporting delay via exploiting the interplay between compression and prediction. Hence, a base station does not have to wait for the readings, which is still processed at each sensor node. To overcome the predictors' stability problems, we proposed a cooperative prediction scheme in which all adjacent sensor nodes participate in the prediction process, given that they exhibit a high spatial correlation among each other. In addition to the significant improvement of the operational lifetime (96.07% less energy than the well-known LTC method for a fixed throughput), data accuracy and latency were also highly improved.

The fourth contribution of this thesis targets the event-driven WSN applications set. Virtual sensing is exploited being a novel method for reducing energy consumed by "energy-hungry" sensors and simultaneously reducing the event-miss probability. Sensing reliability was tackled via an Ontology-based decision-making algorithm for selection between main and virtual sensors at run-time. Two case studies were considered, including object tracking and gas leak detection. In both studies, lifetime of the main sensors is significantly extended. Moreover, virtual sensing accuracy and reliability were improved through adopting the ontology-based generated rules for sensor selection where sensing quality and environmental conditions are the selection criteria.

The last contribution of this thesis is lifetime planning which targets the hybrid WSN applications set. The core idea behind lifetime planning is to control level of the provided service qualities via exploiting the design-time knowledge. Additionally, limiting the lifetime to only the task lifetime, enables dedicating more energy for improving the QoS metrics. Both, lower and upper QoS boundaries, which are typically estimated at design-time, were used to control the QoS metrics. During run-time, a self-adaptive framework – based on the the *event-condition-action* (ECA) rules – is proposed to respond to the environmental dynamics. A case study of office monitoring WSN was employed. The results showed a significant improvement in the provided QoS, namely the communication reliability (throughput) and the reporting delay.

Generally speaking, the thesis mainly proposes an energy-centric strategy as an alternative to the complex algorithmic MOO methods. Its core idea is to enable energy-efficiency methods to deliver better service qualities. The proposed strategy relies on exploiting both, design-time knowledge and environmental dynamics together with adopting efficient self-adaptation mechanisms. The concept of the energy-centric strategy was validated via three different methods, as explained above.

7.2. Outlook

The work presented in this thesis could be extended in different directions. In the following, we have a look at two possible extensions that build upon the contributions of this thesis.

One possible extension would be a comprehensive comparative study between the algorithmic MOO methods and the proposed energy-centric strategy. This comparative analysis has to consider the diversity of WSN applications. For the time-driven WSN applications set, we select data compression as an example. However, the proposed strategy can be easily applied to other energy-efficiency methods by simply considering the application scenario and adopting an efficient self-adaptation mechanism. Furthermore, energy-efficiency methods, which target the event-based applications set, can also be examined in a similar manner. For the hybrid application set, we examined the scenario of office monitoring. However, other scenarios may be considered to prove superiority of the lifetime planning.

Another interesting research direction is the adoption of the proposed methods in this thesis to other platforms such as smartphones. Currently, smartphones are engaged in many sensing scenarios. These mobile devices have similar characteristics as the WSNs. However, more challenges emerge since smartphones are not primarily designed for sensing operations. Therefore, sensing tasks have not to consume so much energy, or users will not participate in the sensing network. Smartphones also suffer from strict constraints in their size and the allocated energy. Therefore, an emerging research trend considers both, energy-efficiency and QoS control in smartphone sensing. The ideas and concepts, presented in this thesis, are applicable in the context of smartphones. For instance, self-adapting mechanisms can highly reduce the burden of sensing and reporting the measurements and simultaneously improving the application-relevant QoS metrics.

Bibliography

[AA2013a] H. Alwan and A. Agarwal. A cross-layer-based routing with QoS-aware scheduling for wireless sensor networks. In *Proceedings of the 2013 ACS International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8, May 2013.

[AA2013b] H. Alwan and A. Agarwal. Multi-objective QoS routing for wireless sensor networks. In *Proceedings of the 2013 International Conference on Computing, Networking and Communications (ICNC),*, pages 1074–1079, Jan 2013.

[AADFR2009] C. Alippi, G. Anastasi, M. Di Francesco and M. Roveri. Energy management in wireless sensor networks with energy-hungry sensors. *Instrumentation Measurement Magazine, IEEE*, 12(2):16–23, 2009.

[AAG+2007] C. Alippi, G. Anastasi, C. Galperti, F. Mancini and M. Roveri. Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications. In *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems, 2007. MASS 2007*, pages 1–6, 2007.

[AARH2015] A. Ahmad, S. Ahmad, M.H. Rehmani and N.U. Hassan. A survey on radio resource allocation in cognitive radio sensor networks. *Communications Surveys Tutorials, IEEE*, 17(2):888–917, 2015.

[ABK+2009] 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.

[ABRL2010] Gaetano F Anastasi, Enrico Bini, Antonio Romano and Giuseppe Lipari. A service-oriented architecture for QoS configuration and management of wireless sensor networks. In *Proceedings of the 2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8. IEEE, 2010.

[ACDFP2009] Giuseppe Anastasi, Marco Conti, Mario Di Francesco and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Netw.*, 7(3):537–568, May 2009.

[ACFP2009] G. Anastasi, M. Conti, M. Di Francesco and A. Passarella. Energy conservation in wireless sensor networks: a survey. *Ad Hoc Network Journal*, 7(3):537–568, 2009.

[ADBea2004] A. Arora, P. Dutta, S. Bapat and et al. A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking. *Computer Networks*, 46(5):605–634, December 2004.

[Adv2015] Advanticsys. TelosB Sensor Nodes. `http://www.advanticsys.com/shop/mtmcm5000msp-p-14.html`, 2015. Accessed: November 2015.

[AE2008] A. Aksu and O. Ercetin. Reliable multi-hop routing with cooperative transmissions in energy-constrained networks. *IEEE Transactions on Wireless Communications*, 7(8):2861–2865, 2008.

[ALGM2009] Edgardo Avilés-López and J Antonio García-Macías. TinySOA: A service-oriented architecture for wireless sensor networks. *Service Oriented Computing and Applications*, 3(2):99–108, 2009.

[AMS2013] Mohammad Arifuzzaman, Morio Matsumoto and Takao Sato. An intelligent hybrid MAC with traffic-differentiation-based QoS for wireless sensor networks. *Sensors Journal, IEEE*, 13(6):2391–2399, 2013.

[And2000] Johan Andersson. A survey of multiobjective optimization in engineering design. *Department of Mechanical Engineering, Linktjping University. Sweden*, 2000.

[Anw2006] M. Anwander. Comparison of TDMA and contention based MAC protocols on embedded sensor boards. Master's thesis, Universität Bern, 2006.

[APJ2004] Sasikanth Avancha, Chintan Patel and Anupam Joshi. Ontology-driven Adaptive Sensor Networks. In *Proceedings of the MobiQuitous conference*, volume 4, pages 194–202, 2004.

[APM2005] I. Akyildiz, D. Pompili and T. Melodia. Underwater acoustic sensor networks: Research challenges. *Ad Hoc Networks Journal*, 3(3):257–279, 2005.

[AR2008] S. Ali and P. Roy. Energy saving methods in wireless sensor networks. Master's thesis, School of Information Science, Computer and Electrical Engineering, Halmstad University, May 2008.

[ARAI2013] 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$^+$2014] I Anaya, V. Simko, J. Bourcier, N. Plouzeau and J. 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 2014, pages 145–154, 2014.

[ASSC2002] I. Akyildiz, W. Su, Y. Sankarasubramaniam and Y. Cayirci. Wireless sensor networks: a survey. *Computer Network Journal*, 38(4):393–422, 2002.

[AY2003] K. Akkaya and M. Younis. An energy-aware QoS routing protocol for wireless sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops*, pages 710–715, 2003.

[BA2006] K. C. Barr and K. Asanović. Energy-aware Lossless Data Compression. *ACM Transactions Computer Systems*, 24(3):250–291, 2006.

[BA2012] A.O. Bicen and O.B. Akan. Energy-efficient RF source power control for opportunistic distributed sensing in wireless passive sensor networks. In *Proceedings of the 2012 IEEE Symposium on Computers and Communications (ISCC)*, pages 000738–000743, 2012.

[BD2008] N. P. Borg and C. J. Debono. The implementation of an adaptive data reduction technique for wireless sensor networks. In *Proceedings of the IEEE International Symposium on Signal Processing and Information Technology*, pages 402–406, 2008.

[BDH2013] He Ba, Ilker Demirkol and Wendi Heinzelman. Passive wake-up radios: From devices to applications. *Ad Hoc Networks*, 11(8):2605–2621, 2013.

[BPC+2007] P. Barontib, P. Pillaia, V. W. Chook, S. Chessab, A. Gottab and Y. F. Hua. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications Journal*, 30(7):1655–1695, May 2007.

[BRK2011] M. Balouchestani, K. Raahemifar and S. Krishnan. Compressive sensing in wireless sensor networks: Survey. *Canadian Journal on Multimedia and Wireless Networks*, 2(1), February 2011.

[BTS2012] S.S. Beheshtiha, H. Tan and M. Sabaei. Opportunistic routing with adaptive harvesting-aware duty cycling in energy harvesting WSN. In *Proceedings of the 2012 15th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 90–94, Sept 2012.

[CBBea2012] M. Compton, P. Barnaghi, L. Bermudez and et al. The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17:25–32, December 2012.

[CCB+2013] Li Chen, S. Cool, He Ba, W. Heinzelman, I. Demirkol, U. Muncuk, K. Chowdhury and S. Basagni. Range extension of passive wake-up radio systems through energy harvesting. In *Proceedings of the 2013 IEEE International Conference on Communications (ICC)*, pages 1549–1554, 2013.

[CDHH2006] D. Chu, A. Deshpande, J.M. Hellerstein and Wei Hong. Approximate data collection in sensor networks using probabilistic models. In *Proceedings of the 22nd International Conference on Data Engineering, 2006. ICDE '06*, pages 48–48, 2006.

[CFH] G. Chelius, A. Fraboulet and E. Hamida. Wsnet: an Event-driven Simulator for Large Scale Wireless Networks. [accessed May 2014].

[Cir2011] C. Cirstea. Energy efficient routing protocols for wireless sensor networks: A survey. In *Proceedings of the IEEE 17th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 2011.

[CKC2007] Youngjin Cho, Younghyun Kim and N. Chang. Pvs: passive voltage scaling for wireless sensor networks. In *Proceedings of the 2007 ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, pages 135–140, 2007.

[CKea2002] Selina Chu, Eamonn J Keogh and et al. Iterative deepening dynamic time warping for time series. In *Proceedings of the SDM conference*, pages 195–212. SIAM, 2002.

[CMdS$^+$2007] Luiz H.A. Correia, Daniel F. Macedo, Aldri L. dos Santos, Antonio A.F. Loureiro and José Marcos S. Nogueira. Transmission power control techniques for wireless sensor networks. *Computer Networks*, 51(17):4765 – 4779, 2007.

[CNC$^+$2014] Long Cheng, Jianwei Niu, Jiannong Cao, Sajal K Das and Yu Gu. QoS aware geographic opportunistic routing in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(7):1864–1875, 2014.

[CQXB2014] Yongrui Chen, Fei Qin, Yulong Xing and Dujdow Buranapanichkit. Cross-layer optimization scheme using cooperative diversity for reliable data transfer in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2014, 2014.

[CS2015] Xiaoyu Chu and Harish Sethu. Cooperative topology control with adaptation for improved lifetime in wireless sensor networks. *Ad Hoc Networks*, 30(C):99–114, July 2015.

[CSE2011] I. Chen, A. Speer and M. 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.

[CV2004] Dazhi Chen and Pramod K Varshney. QoS support in wireless sensor networks: A survey. In *Proceedings of the International Conference on Wireless Networks*, volume 13244, pages 227–233, 2004.

[CYRV2006] R.R. Choudhury, Xue Yang, R. Ramanathan and N.F. Vaidya. On designing MAC protocols for wireless networks using directional antennas. *IEEE Transactions on Mobile Computing*, 5(5):477–491, 2006.

[Cza2006] P. P. Czapski. A survey: MAC protocols for applications of wireless sensor networks. In *Proceedings of the IEEE Region 10 Conference (TENCON)*, 2006.

[DBF2007] Thanh Dang, Nirupama Bulusu and Wu-chi Feng. Rida: A robust information-driven data compression architecture for irregular wireless sensor networks. In Koen Langendoen and Thiemo Voigt, editors, *Wireless Sensor Networks*, volume 4373 of *Lecture Notes in Computer Science*, pages 133–149. Springer Berlin Heidelberg, 2007.

[DD2012] Q. Dong and W. Dargie. A survey on mobility and mobility-aware MAC protocols in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 2012.

[DEFT2011] 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.

[DEO2009] I. Demirkol, C. Ersoy and E. Onur. Wake-up receivers for wireless sensor networks: benefits and challenges. *Wireless Communications, IEEE*, 16(4):88–96, 2009.

[DGV2004] Adam Dunkels, Björn Grönvall and Thiemo Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, 2004*, pages 455–462. IEEE, 2004.

[Dia2009] K. Diamond. Climate change, sustainable development, and ecosystems committee newsletter. Technical report, American Bar Association, July 2009.

[DPR+2005] Flávia C Delicato, Paulo F Pires, Luiz Rust, Luci Pirmez and José Ferreira de Rezende. Reflective middleware for wireless sensor networks. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1155–1159. ACM, 2005.

[EV2006] S.C. Ergen and P. Varaiya. Optimal placement of relay nodes for energy efficiency in sensor networks. In *Proceedings of the IEEE International Conference on Communications, 2006. ICC '06.*, volume 8, pages 3473–3479, 2006.

[FK2011] M. Farooq and T. Kunz. Operating systems for wireless sensor networks: A survey. *Sensors Journal*, 11(6), 2011.

[FRWZ2007] E. Fasolo, M. Rossi, J. Widmer and M. Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *Wireless Communications, IEEE*, 14(2):70–87, 2007.

[FT2007] Konstantinos P Ferentinos and Theodore A Tsiligiridis. Adaptive design optimization of wireless sensor networks using genetic algorithms. *Computer Networks*, 51(4):1031–1051, 2007.

[GB2008] JamesM. Gilbert and Farooq Balouchi. Comparison of energy harvesting systems for wireless sensor networks. *International Journal of Automation and Computing*, 5(4):334–347, 2008.

[GD2012] B. Griffin and C. Detweiler. Resonant wireless power transfer to ground sensors from a uav. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2660–2665, 2012.

[GEH2003] D. Ganesan, D. Estrin and J. Heidemann. Dimensions: Why Do We Need a New Data Handling Architecture for Sensor Networks? *SIGCOMM Comput. Commun. Rev.*, pages 143–148, 2003.

[GHIGGHPD2007] C. Garcia-Hernandez, P. Ibarguengoytia-Gonzalez, J. Garcia-Hernandez and J. Perez-Diaz. Wireless sensor networks and applications: a survey. *IJCSNS International Journal of Computer Science and Network Security*, 7(3), March 2007.

[GLQC2011] X. Ge, J. Liu, Q. Qi and Z. Chen. A New Prediction Approach Based on Linear Regression for Collaborative Filtering. In *Proceedings of the 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, volume 4, pages 2586–2590, July 2011.

[GLY2007] B. Gedik, L. Liu and P. Yu. Asap: An adaptive sampling approach to data collection in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(12):1766–1783, 2007.

[GRF$^+$2012] N. Gamez, D. Romero, L. Fuentes, R. Rouvoy and L. 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.

[GSC2009] D. Garlan, B. Schmerl and S. Cheng. Software architecture-based self-adaptation. In *Autonomic Computing and Networking*, pages 31–55. Springer US, 2009.

[GSGSLea2010] A. Garcia-Sanchez, F. Garcia-Sanchez, F. Losilla and et al. Wireless Sensor Network Deployment for Monitoring Wildlife Passages. *Sensors*, 10(8):7236–7262, 2010.

[HBLD2008] J. Hayes, S. Beirne, K. Lau and D. Diamond. Evaluation of a low cost wireless chemical sensor network for environmental monitoring. In *Proceedings of the Seventh Conference on Sensors*, pages 530–533, October 2008.

[HBT$^+$2007] R. Hoes, T. Basten, C. Tham, M. Geilen and H. 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.

[HMV2004] Qi Han, S. Mehrotra and N. Venkatasubramanian. Energy efficient data collection in distributed sensor environments. In *Proceedings of the 24th International Conference on Distributed Computing Systems*, pages 590–597, 2004.

[HN2015] Mohammad Hammoudeh and Robert Newman. Adaptive routing in wireless sensor networks: QoS optimization for enhanced application performance. *Information Fusion*, 22:3–15, 2015.

[HSR$^+$2008] An He, S. Srikanteswara, J.H. Reed, Xuetao Chen, W.H. Tranter, Kyung Kyoon Bae and M. Sajadieh. Minimizing energy consumption using cognitive radio. In *Proceedings of the 2008 IEEE International Conference on Performance, Computing and Communications (IPCCC 2008)*, pages 372–377, 2008.

[HY2008] C. Hua and T. Yum. Optimal Routing and Data Aggregation for Maximizing Lifetime of Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 16(4):892–903, Aug 2008.

[IBRL] Intel Berkeley Research Lab. WSN dataset. `http://db.csail.mit.edu/labdata/labdata.html`. Accessed: April 2014.

[ISA2011] 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.

[Ita1975] Fumitada Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23(1):67–72, 1975.

[Jay2006] S.K. Jayaweera. Virtual mimo-based cooperative communication for energy-constrained wireless sensor networks. *IEEE Transactions on Wireless Communications*, 5(5):984–989, 2006.

[JC2004] Ankur Jain and Edward Y. Chang. Adaptive sampling for sensor networks. In *Proceeedings of the 1st International Workshop on Data Management for Sensor Networks: In Conjunction with VLDB 2004*, DMSN '04, pages 10–16. ACM, 2004.

[JDiJR2007] G. Joshi, G. Da-iict, S. Jardosh and P. Ranjan. Bounds on dynamic modulation scaling for wireless sensor networks. In *Proceedings of the Third International Conference on Wireless Communication and Sensor Networks (WCSN '07)*, 2007.

[JFD2015] Seyed Mahdi Jameii, Karim Faez and Mehdi Dehghan. Multiobjective optimization for topology and coverage control in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2015, 2015.

[JG2011] O. Jonah and S.V. Georgakopoulos. Efficient wireless powering of sensors embedded in concrete via magnetic resonance. In *Proceedings of the 2011 IEEE International Symposium on Antennas and Propagation (APSURSI)*, pages 1425–1428, July 2011.

[JH2013] A. Jemal and R. B. 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 (WETICE)*, pages 125–130, June 2013.

[JHH+2013] Ahmed Jemal, Marwa Hachicha, Riadh Halima, Ahmed Hadji Kacem, Khalil Drira and Mohamed Jmaiel. Mpaas: Monitoring values prediction as a service for energy consumption optimization purpose. *Rapport LAAS n 13449*, 2013.

[JJW2011] Hongbo Jiang, Shudong Jin and Chonggang Wang. Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(6):1064–1071, 2011.

[JRL2010] R.C. Jisha, M.V. Ramesh and G.S. Lekshmi. Intruder Tracking Using Wireless Sensor Network. In *Proceedings of the 2010 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–5, Dec 2010.

[JSB+2006] Sushant Jain, Rahul C Shah, Waylon Brunette, Gaetano Borriello and Sumit Roy. Exploiting mobility for energy efficient data collection in wireless sensor networks. *Mobile Networks and Applications*, 11(3):327–339, 2006.

[JT2014] Yunye Jin and Hwee-Pink Tan. Optimal performance trade-offs in MAC for wireless sensor networks powered by heterogeneous ambient energy harvesting. In *Proceedings of the Networking Conference, 2014 IFIP*, pages 1–9, June 2014.

[JWI2011] Jin Woo Jung, Wensi Wang and M.A. Ingram. Cooperative transmission range extension for duty cycle-limited wireless sensor networks. In *Proceedings of the 2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*, pages 1–5, 2011.

[Kal2004] P. Kalekar. Time Series Forecasting using Holt-Winters Exponential Smoothing. Technical report, Department of Computer Science and Engineering, Indian Institute of Technology Bombay, December 2004.

[KBW2013] U. Kulau, F. Büsching and L. Wolf. A node's life: Increasing WSN lifetime by dynamic voltage scaling. In *Proceedings of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 241–248, 2013.

[KCC2008] N. Kim, S. Choi and H. Cha. Automated Sensor-specific Power Management for Wireless Sensor Networks. In *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 305–314, Sept 2008.

[KcHA2006] T. Kijewski-correa, M. Haenggi and P. Antsaklis. A Multi-scale Wireless Sensor Networks for Structural Health Monitoring. In *Proceedings of the 17th Analysis and Computation Specialty Conference (ASCE)*, pages 16–18, 2006.

[KH2014] Ansgar Kellner and Dieter Hogrefe. Multi-objective ant colony optimisation-based routing in WSNs. *International Journal of Bio-Inspired Computation*, 6(5):322–332, October 2014.

[KKW2004] Evangelos Kranakis, Danny Krizanc and Eric Williams. Directional versus omnidirectional antennas for energy consumption and k-connectivity of networks of sensors. In *Proceedings of the OPODIS conference*, volume 2004, pages 357–368. Springer, 2004.

[KKZ2005] J-P Katoen, Maneesh Khattri and IS Zapreevt. A markov reward model checker. In *Proceedings of the 2nd International Conference on the Quantitative Evaluation of Systems*, pages 243–244. IEEE, 2005.

[KL2005] N. Kimura and S. Latifi. A survey on data compression in wireless sensor networks. *International Conference on Information Technology: Coding and Computing*, 2, 2005.

[KLV2006] Abtin Keshavarzian, Huang Lee and Lakshmi Venkatraman. Wakeup scheduling in wireless sensor networks. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '06, pages 322–333. ACM, 2006.

[KNP2011] M. Kwiatkowska, G. Norman and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

[KP2000] Eamonn J Keogh and Michael J Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 285–289. ACM, 2000.

[KP2001] Eamonn J Keogh and Michael J Pazzani. Derivative dynamic time warping. In *Proceedings of the SDM conference*, volume 1, pages 5–7. SIAM, 2001.

[KPC2001] Sang-Wook Kim, Sanghyun Park and Wesley W Chu. An index-based approach for similarity search supporting time warping in large sequence databases. In *Proceedings of the 2001 17th International Conference on Data Engineering*, pages 607–614. IEEE, 2001.

[KPea2013] P. Kumsawat, N. Pimpru and et al. Wavelet-based Data Compression Technique for WSNs. In *Proceedings of the World Academy of Science, Engineering and Technology*, pages 689 – 696, 2013.

[KR2005] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and information systems*, pages 358–386, 2005.

[KWIea2012] R. Kozma, L. Wang, K. Iftekharuddin and et al. A Radar-enabled Collaborative Sensor Network Integrating COTS Technology for Surveillance and Tracking Sensors. *Sensors*, 12(2):1336–1351, 2012.

[LAGD2008] S. Lin, B. Arai, D. Gunopulos and G. Das. Region sampling: Continuous adaptive sampling on sensor networks. In *Proceedings of the IEEE 24th International Conference on Data Engineering, 2008. ICDE 2008*, pages 794–803, 2008.

[Liu2016] Xuxun Liu. A novel transmission range adjustment strategy for energy hole avoiding in wireless sensor networks. *Journal of Network and Computer Applications*, 67:43–52, 2016.

[LLC$^+$2013] Huan Li, Yanlei Liu, Weifeng Chen, Weijia Jia, Bing Li and Junwu Xiong. Coca: Constructing optimal clustering architecture to maximize sensor network lifetime. *Computer Communications*, 36(3):256 – 268, 2013.

[LRL$^+$2012] Anfeng Liu, Ju Ren, Xu Li, Zhigang Chen and Xuemin (Sherman) Shen. Design principles and improvement of cost function based energy aware routing algorithms for wireless sensor networks. *Computer Networks*, 56(7):1951 – 1967, 2012.

[LSPM2012] Q. Li, P. Schaffer, J. Pang and S. Mauw. Comparative analysis of clustering protocols with probabilistic model checking. In *Proceedings of the 2012 Sixth International Symposium on Theoretical Aspects of Software Engineering (TASE)*, pages 249–252. IEEE, 2012.

[LTW2004] J.N. Laneman, D.N.C. Tse and Gregory W. Wornell. Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *IEEE Transactions on Information Theory*, 50(12):3062–3080, 2004.

[LY2006] M. Li and B. Yang. A survey on topology issues in wireless sensor network. In *Proceedings of the 2006 International Conference on Wireless Networks (ICWN)*, 2006.

[LzHbGT2011] W. Li-zhong, L. Hong-bo, Z. Gang and H. Tao. The network nodes design of gas wireless sensor monitor. *The 2nd International Conference on Mechanic Automation and Control Engineering (MACE)*, July 2011.

[MAC+2012] P. Montalto, M. Aliotta, A. Cannata, C. Cassisi and A. Pulvirenti. Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining. *Advances in Data Mining Knowledge Discovery and Applications*, 2012.

[MAJ2011] Nader Mohamed and Jameela Al-Jaroodi. A survey on service-oriented middleware for wireless sensor networks. *Service Oriented Computing and Applications*, 5(2):71–85, 2011.

[MCP+2002] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk and J. Anderson. Wireless Sensor Networks for Habitat Monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pages 88–97, 2002.

[MCW+2011] Flávio VC Martins, Eduardo G Carrano, Elizabeth F Wanner, Ricardo HC Takahashi and Geraldo R Mateus. A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks. *Sensors Journal, IEEE*, 11(3):545–554, 2011.

[MFjWM2004] D. Malan, T. Fulford-jones, M. Welsh and S. Moulton. Codeblue: An ad hoc Sensor Network Infrastructure for Emergency Medical Care. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, 2004.

[MGR2009] A. Munir and A. 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.

[MHDN+2012] M. Masonta, Y. Haddad, L. De Nardis, A. Kliks and O. Holland. Energy efficiency in future wireless networks: Cognitive radio standardization requirements. In *Proceedings of the 2012 IEEE 17th*

International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pages 31–35, 2012.

[MLWSW2007] Ye Ming Lu and Vincent W. S. Wong. An energy-efficient multipath routing protocol for wireless sensor networks: Research articles. *International Journal of Communication Systems*, 20(7):747–766, July 2007.

[MMH2011] S. Misra, N.E. Majd and Hong Huang. Constrained relay node placement in energy harvesting wireless sensor networks. In *Proceedings of the 2011 IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 25–34, 2011.

[MPH2007] M. Marin-Perianu and P. Havinga. D-fler: A Distributed Fuzzy Logic Engine for Rule-based Wireless Sensor Networks. In *Ubiquitous Computing Systems*, volume 4836, pages 86–101. Springer, 2007.

[MR2011] R. Mietz and K. Römer. Exploiting correlations for efficient content-based sensor search. In *Proceedings of the Sensors conference, 2011 IEEE*, pages 187–190, 2011.

[MRRR2013] K. Miranda, V.M. Ramos R and T. Razafindralambo. Using efficiently autoregressive estimation in wireless sensor networks. In *Proceedings of the 2013 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–5, 2013.

[Mül2007] Meinard Müller. *Information retrieval for music and motion.* Springer, 2007.

[NKVD2013] Stefanos A Nikolidakis, Dionisis Kandris, Dimitrios D Vergados and Christos Douligeris. Energy efficient routing in wireless sensor networks through balanced clustering. *Algorithms*, 6(1):29–42, 2013.

[NPK+2005] G. Norman, D. Parker, M. Kwiatkowska, S. Shukla and R. Gupta. Using probabilistic model checking for dynamic power management. *Formal Aspects of Computing*, 17(2):160–176, 2005.

[OR2007] Brian Otis and Jan Rabaey. *Ultra-Low Power Wireless Technologies for Sensor Networks.* Springer US, 1. edition, 2007.

[OR2011] L. Oliveira and J. Rodrigues. Wireless sensor networks: a survey on environmental monitoring. *Journal of Communications*, 6(2), April 2011.

[Ort2013] Carolyn M. Ortega. *Service Oriented Wireless Sensor Network.* PhD thesis, City University of New York, New York, NY, USA, 2013. AAI3601886.

[Öst2006] Fredrik Österlind. A sensor network simulator for the contiki os. *SICS Research Report*, 2006.

[OWK2014] Pouya Ostovari, Jie Wu and Abdallah Khreishah. Network coding techniques for wireless and sensor networks. In Habib M. Ammari, editor, *The Art of Wireless Sensor Networks*, Signals and Communication Technology, pages 129–162. Springer Berlin Heidelberg, 2014.

[PBD+2004] Vamsi Paruchuri, Shivakumar Basavaraju, Arjan Durresi, Rajgopal Kannan and S Sitharama Iyengar. Random asynchronous wakeup protocol for sensor networks. In *Proceedings of the First International Conference on Broadband Networks, 2004. BroadNets 2004*, pages 710–717. IEEE, 2004.

[Per2004] I. Perfilieva. Fuzzy Transforms. *Transactions on Rough Sets II*, pages 63–81, 2004.

[PH2007] L. Paradis and Q. Han. A survey of fault management in wireless sensor networks. *Journal of Network and Systems Management*, 7(3), 2007.

[PMFJ2013] P. Park, P. Di Marco, C. Fischione and K. 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.

[PP1990] Marc Parizeau and Rejean Plamondon. A comparative analysis of regional correlation, dynamic time warping, and skeletal tree matching for signature verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):710–717, 1990.

[Pre2009] The Associated Press. Minnesota: Cause of bridge collapse is questioned, 2009.

[RBC2014] Tifenn Rault, Abdelmadjid Bouabdallah and Yacine Challal. Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks*, 67:104 – 122, 2014.

[RBD2013] M. A. Razzaque, C. Bleakley and S. Dobson. Compression in wireless sensor networks: A survey and comparative evaluation. *ACM Transactions of Sensor Networks*, 10:5:1–5:44, November 2013.

[RCM+2012] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas and G. P. Picco. What does model-driven data acquisition really achieve in wireless sensor networks? In *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 85–94, 94, March 2012.

[RDBL2012] Marjan Radi, Behnam Dezfouli, Kamalrulnizam Abu Bakar and Malrey Lee. Multipath routing in wireless sensor networks: Survey and research challenges. *Sensors*, 12(1):650, 2012.

[RGG2012] R.R. Rout, S. Ghosh and S.K. Ghosh. Efficient data collection with directional antenna and network coding in wireless sensor networks. In *Proceedings of the 2012 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS)*, pages 81–86, 2012.

[RL2014] Hee-Tae Roh and Jang-Won Lee. Cross-layer optimization for wireless sensor networks with rf energy transfer. In *Proceedings of the 2014 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 919–923, 2014.

[RLCL2009] 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.

[RM2004] K. Römer and F. Mattern. The Design Space of Wireless Sensor Networks. *IEEE Wireless Communications*, 11(6):54–61, Dec 2004.

[Ros2004] T. Ross. *Fuzzy Logic with Engineering Applications.* John Wiley & Sons Ltd, 1. edition, August 2004.

[RWC2014] M. Rubin, M. Wakin and T. Camp. A Comparison of On-Mote Lossy Compression Algorithms for Wireless Seismic Data Acquisition. In *Proceedings of the 2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 111–118, May 2014.

[SBK2014] Marion Souil, Abdelmadjid Bouabdallah and AE Kamal. Efficient QoS provisioning at the MAC layer in heterogeneous wireless sensor networks. *Computer Communications*, 43:16–30, 2014.

[SBS+2011] A. Somov, A. Baranov, A. Savkin, D. Spirjakin, A. Spirjakin and R. Passerone. Development of wireless sensor network for combustible gas monitoring. *Sensors and Actuators, A: Physical*, 171(2):398–405, 2011.

[SC1978] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1):43–49, 1978.

[SC2004] Stan Salvador and Philip Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. In *Proceedings of the KDD workshop on mining temporal and sequential data.* Citeseer, 2004.

[SD2012] Kamil Staniec and Grzegorz Debita. Interference mitigation in WSN by means of directional antennas and duty cycle control. *Wireless Communications and Mobile Computing*, 12(16):1481–1492, 2012.

[SDC2011] I. Santos, M. Dota and C. Cugnasca. Dynamic Definition of the Sampling Rate of Data in Wireless Sensor Network with Adaptive Automata. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 9(6):963–968, Oct 2011.

[SG2009] R. Sugihara and R.K. Gupta. Optimizing energy-latency trade-off in sensor networks with controlled mobility. In *Proceedings of the INFOCOM 2009, IEEE*, pages 2566–2570, 2009.

[SGOea2004] T. Schoellhammer, B. Greenstein, E. Osterweil and et al. Lightweight Temporal Compression of Microclimate Datasets [Wireless Sensor Networks]. In *Proceedings of the 29th Annual IEEE LCN*, 2004.

[SK2011] S. Sudevalayam and P. Kulkarni. Energy harvesting sensor nodes: Survey and implications. *Communications Surveys Tutorials, IEEE*, 13(3):443–461, Third 2011.

[SLGT2011] Sandra Sendra, Jaime Lloret, Miguel García and José F Toledo. Power saving and energy optimization techniques for wireless sensor neworks. *Journal of communications*, 6(6):439–459, 2011.

[SM2009] S.R. Singh and K. Moez. Ultra low leakage 90nm content addressable memory design for wireless sensor network applications. In *Proceedings of the 52nd IEEE International Midwest Symposium on Circuits and Systems, 2009. MWSCAS '09*, pages 1191–1194, 2009.

[SMSC2011] S Saxena, Shailendra Mishra, AK Sharma and DS Chauhan. Efficient power utilization techniques for wireless sensor networks- a survey. *International Journal on Computer Science and Engineering*, 3(2):905–925, 2011.

[SR2006] Silvia Santini and Kay Römer. An Adaptive Strategy for Quality-Based Data Reduction in Wireless Sensor Networks. In *Proceedings of the INSS conference*, 2006.

[SS2012] R Sumathi and MG Srinivas. A survey of QoS based routing protocols for wireless sensor networks. *Journal of Information Processing Systems*, 8(4):589–602, 2012.

[SSB2014] Ricardo Silva, Jorge Sa Silva and Fernando Boavida. Mobility in wireless sensor networks - survey and proposal. *Computer Communications*, 52:1 – 20, 2014.

[SSS2010] S. Singh, M. Singh and D. Singh. Routing protocols in wireless sensor networks-a survey. *International Journal of Computer Science and Engineering Survey (IJCSES)*, 1(2), November 2010.

[ST2009] M. Salehie and L. Tahvildari. Self-adaptive Software: Landscape and Research Challenges. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 4(2):14:1–14:42, 2009.

[Ste2007] M. Stepnicka. Fuzzy Transform for Practical Problems. Technical Report 114, Institute for Research and Applications of Fuzzy Modeling, University of Ostrava, 2007.

[STS2002] C. Schurgers, V. Tsiatsis and M.B. Srivastava. Stem: Topology management for energy efficient sensor networks. In *Proceedings of the Aerospace Conference Proceedings, 2002. IEEE*, volume 3, pages 3–1099–3–1108 vol.3, 2002.

[SVS+2011] M. Steine, N. Viet, O. Serna, M. Geilen, T. Basten, G. Fohler and J. 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.

[SWLC2014] Peizhong Shi, Yun Wang, Kai Li and Alvin TS Chan. Cross-layer adaptive end-to-end delay control for asynchronous duty-cycle wireless sensor networks. In *Pervasive Computing and the Networked World*, pages 520–531. Springer, 2014.

[SYF2005] Yasushi Sakurai, Masatoshi Yoshikawa and Christos Faloutsos. Ftw: Fast Similarity Search Under the Time Warping Distance. In *Proceedings of the 24th ACM Sigmod-Sigact-Sigart Symposium on Principles of Database Systems*, pages 326–337. ACM, 2005.

[SYL2010] Ahmed K. Sadek, Wei Yu and K. J. Ray Liu. On the energy efficiency of cooperative communications in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 6(1):5:1–5:21, January 2010.

[Tex2006] Texas Instruments, Dallas, Texas 75265. *MSP430x1xx Family: User's Guide*, 2006.

[TSH2010] Wei Tuming, Y. Sijia and W. Hailong. A dynamic voltage scaling algorithm for wireless sensor networks. In *Proceedings of the Third International Conference on Advanced Computer Theory and Engineering (ICACTE)*, volume 1, pages V1–554 – V1–557, August 2010.

[TSKJ2013] Ranjana Thalore, Jyoti Sharma, Manju Khurana and Manish Kumar Jha. QoS evaluation of energy-efficient ML-MAC protocol for wireless sensor networks. *AEU-International Journal of Electronics and Communications*, 67(12):1048–1053, 2013.

[TYU2015] K. Tutuncuoglu, A. Yener and S. Ulukus. Optimum policies for an energy harvesting transmitter under energy storage losses. *IEEE Journal on Selected Areas in Communications*, 33(3):467–481, March 2015.

[VBI2010] G. Vijay, E. Bdira and M. Ibnkahla. Cognitive approaches in wireless sensor networks: A survey. In *Proceedings of the 2010 25th Biennial Symposium on Communications (QBSC)*, pages 177–180, 2010.

[VGP+2007] G. Vellidis, V. Garrick, S. Pocknee, C. Perry, C. Kvien and M. Tucker. How wireless will change agriculture. *the Sixth European Conference on Precision Agriculture (6ECPA)*, 2007.

[VIK+2005] N. Vijaykrishnan, M. J. Irwin, M. Kandemir, L. Li, G. Chen and B. Kang. *Designing Energy-Aware Sensor Systems.* Distributed Sensor Networks. CRC Press, 2005.

[Voi2012] Thiemo Voigt. Contiki cooja crash course. *The International School on Cooperative Robots and Sensor Networks (RoboSense School 2012), Hammamet, Tunisia*, 2012. [accessed May 2014].

[VRL+2012] Thiemo Voigt, Utz Roedig, Olaf Landsiedel, Kasun Samarasinghe and Mahesh Bogadi Shankar Prasad. On the applicability of network coding in wireless sensor networks. *SIGBED Rev.*, 9(3):46–48, July 2012.

[VSXea2009] Domingos Viegas, Albert Simeoni, Gavriil Xanthopoulos and et al. Recent forest fire related accidents in europe. Technical report, European Commission, Joint Research Centre, Institute for Environment and Sustainability, 2009.

[WCLD2008] Miao-Miao Wang, Jian-Nong Cao, Jing Li and Sajal K Dasi. Middleware for wireless sensor networks: A survey. *Journal of computer science and technology*, 23(3):305–326, 2008.

[WHBD2005] R. Wagner, C. Hyeokho, R. Baraniuk and V. Delouille. Distributed Wavelet Transform for Irregular Sensor Network Grids. In *Proceedings of the 2005 IEEE/SP 13th Workshop on Statistical Signal Processing*, pages 1196–1201, 2005.

[XS2007] Sandeep Neema Xenofon Koutsoukos, Manish Kushwaha, Isaac Amundson and Janos Sztipanovits. OASiS-A Service-Oriented Architecture for Ambient-Aware Sensor Network. *Lecture Notes in Computer Science*, 4888:125–149, 2007.

[XSH+2015] Liguang Xie, Yi Shi, Y.T. Hou, Wenjing Lou, H.D. Sherali, Huaibei Zhou and S.F. Midkiff. A mobile platform for wireless charging and data collection in sensor networks. *IEEE Journal on Selected Areas in Communications*, 33(8):1521–1533, 2015.

[XSHL2013] Liguang Xie, Yi Shi, Y.T. Hou and A. Lou. Wireless power transfer and applications to sensor networks. *Wireless Communications, IEEE*, 20(4):140–145, August 2013.

[XZST2007] Feng Xia, Wenhong Zhao, Youxian Sun and Yu-Chu Tian. Fuzzy logic control based QoS management in wireless sensor/actuator networks. *Sensors*, 7(12):3179–3191, 2007.

[YBo2009] B. Yahya and J. Ben-othman. Reer: Robust and energy efficient multipath routing protocol for wireless sensor networks. In *Proceedings of the Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–7, 2009.

[YIE2011] M Aykut Yigitel, Ozlem Durmaz Incel and Cem Ersoy. QoS-aware MAC protocols for wireless sensor networks: A survey. *Computer Networks*, 55(8):1982–2004, 2011.

[YRT2014] J. Yang, T. Rosing and S. Tilak. Leveraging Application Context for Efficient Sensing. In *Proceedings of the 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–6, April 2014.

[YWL2015] Yuanyuan Yang, Cong Wang and Ji Li. Wireless rechargeable sensor networks-current status and future trends. *Journal of Communications*, 10(9), 2015.

[YX2010] Y. Yun and Y. 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.

[YY2015] Sarika Yadav and Rama Shankar Yadav. A review on energy efficient protocols in wireless sensor networks. *Wireless Networks*, pages 1–16, 2015.

[ZHL+2009] Fei Zhang, S.A. Hackworth, Xiaoyu Liu, Haiyan Chen, R.J. Sclabassi and Mingui Sun. Wireless energy transfer platform for medical sensors and implantable devices. In *Proceedings of the IEEE Annual International Conference of Engineering in Medicine and Biology Society (EMBC 2009)*, pages 1045–1048, Sept 2009.

[ZLZ+2014] Degan Zhang, Guang Li, Ke Zheng, Xuechao Ming and Zhao-Hua Pan. An energy-balanced routing method based on forward-aware factor for wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 10(1):766–773, Feb 2014.

[ZOS+2009] Jinyun Zhang, Philip V Orlik, Zafer Sahinoglu, Andreas F Molisch and Patrick Kinney. UWB systems for wireless sensor networks. *Proceedings of the IEEE*, 97(2):313–331, 2009.

[ZS2012] Xinyu Zhang and Kang G. Shin. E-mili: Energy-Minimizing Idle Listening in Wireless Networks. *IEEE Transactions on Mobile Computing*, 11(9):1441–1454, Sept 2012.

[ZWLL2012] Y. Zhao, J. Wu, F. Li and S. 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.

[ZWM+2010] Liqiang Zheng, Wensi Wang, Alan Mathewson, Brendan O'Flynn and Michael Hayes. An adaptive transmission power control method for wireless sensor networks. In *Proceedings of the IET Irish Signals and Systems Conference (ISSC 2010)*, pages 261–265, June 2010.

[ZWYX2006] Wen Zeng, Hong Wang, Haibin Yu and Aidong Xu. The research and application of UWB based industrial network. In *Proceedings of The Third International Conference Ultrawideband and Ultrashort Impulse Signals*, pages 153–155, 2006.

[ZZ2012] M. Zhang and X. Zheng. An energy effective adaptive spatial sampling algorithm for wireless sensor networks. In *Proceedings of the International Conference on Computer Technology and Science (ICCTS 2012)*, volume 47, Singapore, 2012. IACSIT Press.

A. A Lightweight Dynamic Time Warping for Tiny Wireless Sensing Devices

A.1. Introduction

Dynamic time warping is a well-known technique to find an optimal alignment between two sequences under certain restrictions [Mül2007]. It has been widely used for optimal alignment of two time series through iteratively warping the time axis until an optimal match between the two data sequences is found. However, such an algorithm is complex and ill-suited for tiny sensor nodes. In this appendix, the *dynamic time warping* (DTW) implementation is refined to minimize the time/space complexity. The novel algorithm is referred to as the *lightweight dynamic time warping* (*liteDTW*). The main idea of *liteDTW* is to 1) reduce the required memory footprint, and 2) shrink the length of the input patterns. For the sake of linear implementations, the DTW matrix is evaluated over several iterations. At each iteration, solely two columns of the matrix are considered to determine the suboptimal path points. Then, shift operations are executed to iterate over the rest of the DTW matrix. The second step is to reduce the window size via using a Fuzzy transform-based compression (FTC) technique (see Chapter 4). Accordingly, the data dimension is decreased prior to the DTW execution. Finally, some experimental results are conducted to clarify the efficiency of the *liteDTW* technique compared to the classical DTW algorithm.

A.2. Related Work

In literature, several articles have discussed the DTW algorithm as in [PP1990], [KP2001], [Mül2007]. In the last decades, a lot of attention has gone to modify the classical DTW algorithm as it has a quadratic time and space complexity that limits its use to only small time series data sets. The well-established, yet still being subject of active research, techniques for making the DTW faster fall into three categories, *constraints* [SC1978,Ita1975], *data abstraction* [CKea2002,KP2000], and *indexing* [KR2005,KPC2001]. Constraints typically limit the number of cells that are evaluated in the cost matrix. In the second category, data abstraction performs the DTW algorithm on a reduced representation of the data. Finally, indexing uses lower bounding functions to shrink the number of times the DTW algorithm must be executed during time series classification of clustering. The proposed techniques in [SC2004] and [SYF2005] are promising techniques with some similarities to our work.

Salvador et al. [SC2004] proposed an approximation of the DTW algorithm based on a multilevel approach that is used for graph bisection. The algorithm uses techniques that belong to the two categories, namely constraints and data abstraction. For instance, the algorithm uses a projection operation to find the minimum-distance warp path at a lower resolution, and it uses that warp path as an initial guess for a higher resolution's minimum-distance warp path. This projection may ignore local variations in the warp path that can be very significant even after refinement. Alternatively, our method linearizes the DTW algorithm by only retaining two columns per iteration. Therefore, disadvantages of projecting the warp path from low resolution to higher resolution are avoided. Moreover, the FuzzyCAT compressor is used to lessen the data dimension due to its high speed and adequate precision. This abstraction is more efficient for tiny devices such as in WSNs than the coarsening operation presented in [SC2004].

Sakurai et al. [SYF2005] proposed a fast search method for the DTW algorithm. The core idea is to use a lower bounding distance measure with segmentation (LBS) technique. In lieu of computing the exact time warping distance for all sequences in the dataset, LBS prunes a significant number of sequences. Then, it excludes warping paths that will not lead to useful search result by using dynamic programming. Finally, the authors have used a search algorithm to enhance the accuracy of the distance approximations. This technique is significant, however our technique is much simpler and easy to implement especially in case of WSN applications. In addition, our approach has been examined on real datasets via extensive simulations over the Telosb sky sensor nodes. In the next section, the basics of pattern matching using the DTW algorithm are introduced.

A.3. Classical DTW Algorithm

A detailed explanation of the classical DTW algorithm is given in Chapter 5. Now, the cost of utilizing such a complex algorithm is empirically determined. Generally, manifold cells of a $m \times n$ matrix are filled exactly once throughout the DTW execution, and each cell is filled in constant time. This yields both a space complexity of $\mathcal{O}(n \times m)$. Initially, the window size of the contrasted patterns is a main metric which affects the algorithm speed and the memory footprint. Figure A.1 depicts the impact of increasing the window size on the distance between the pattern $T1$ and some other patterns $\langle T2, T5, NT1, andNT4\rangle$. These patters are defined in Table 5.4. As it can be seen, the pattern $T2$ has minimum distance with $T1$ for the entire range of window sizes, even with small windows.

However, the above finding is collapsed when other patterns are examined versus various window sizes to safely adopt a window size that is adequate for matching the entire patterns. Figure A.2 depicts the minimum required window size for each pattern, to enable a precise pattern matching. Each blue point has been estimated via repeating Figure A.1 on all other recorded patterns. For instance, the pattern $NT2$ can be matched easily if the window size is above 200. Obviously, sampling approximately 960 data points, ensures safe detections for all vibration patterns. However, this window size burdens the nodes in terms of space/time complexity. To sum up, the DTW-based duty cycling will be solely eligible for hardware implementation, whenever we managed to reduce the optimal window size.

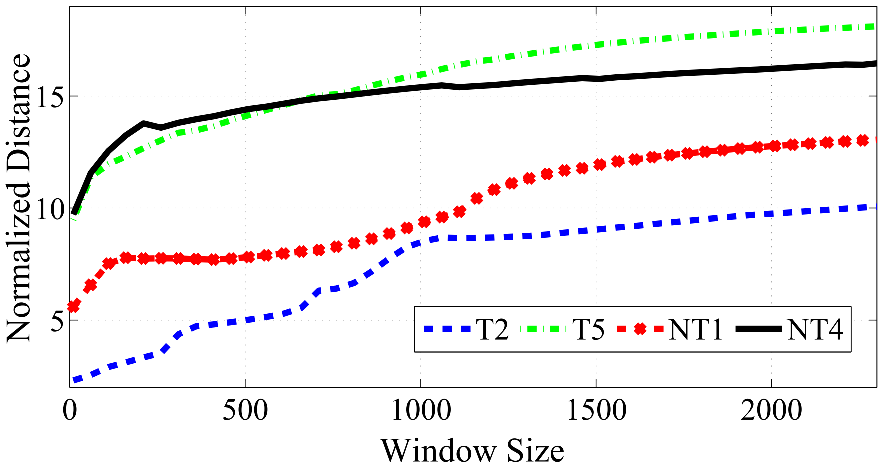

Figure A.1.: Window size versus warping distance from $T1$

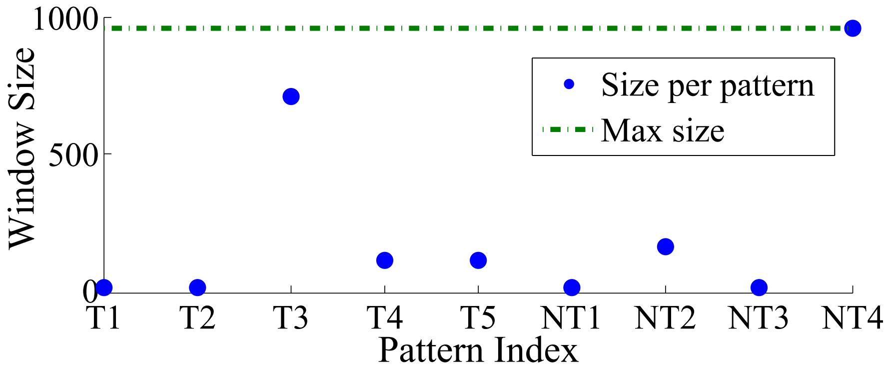

Figure A.2.: Minimum window size for the recorded patterns to be matched

A.4. liteDTW: a DTW Refinement

In this section, the proposed procedure for minimizing the time/space complexity from $\mathcal{O}(n \times m)$ to an extent viable for hardware implementation is introduced. The idea is to integrate two complementary approaches: one for reducing the code complexity and memory utilization, and the other one for reducing the window size. Both approaches, as discussed below, upgrade the standard DTW algorithm into a new version referred to as the *liteDTW* algorithm.

A.4.1. Linear DTW Algorithm

The first approach is to reduce the time/space complexity of the DTW algorithm. This is feasible through preserving only the current and previous columns in memory as the DTW matrix is filled from left to right. Figure A.3 shows a three-iterations matching process between two sequences. In each iteration, only two columns are retained and points of the optimal warp P_{opt} (colored in red) are determined. Then, the first column is discarded, while the second column is used to estimate the third column. This process is repeated until covering the entire matrix. Algorithm 9 formalizes the linearization mechanism. Lines 2-7 clarify the first two columns' processing. Afterward, the $(n \times 2)$ sub-matrix (colored green in Figure A.3) is shifted once to discard the first column and the variable ρ is set to 1 to compute only one column during the next iteration. Specifically, the linear DTW method simplifies the execution overhead from $\mathcal{O}(n \times m)$ to merely $\mathcal{O}(n \times 2)$ which highly reduces the required memory footprint.

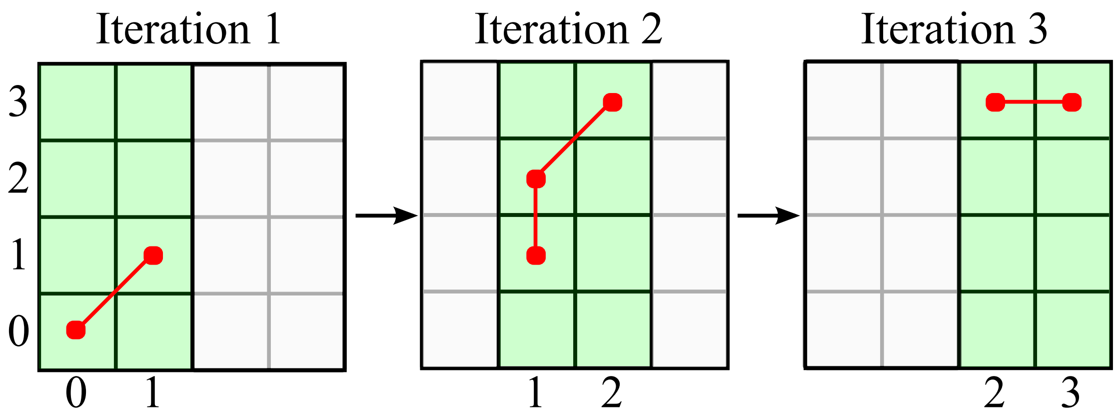

Figure A.3.: Two-columns version of the DTW algorithm

Algorithm 9 Two-columns version of the DTW algorithm

Require: Reference pattern $A \in \mathbb{R}^n$, and test pattern $B \in \mathbb{R}^m$, $\rho = 0$

1: **for** s such that $0 \leq s < m - 1$ **do** ▷ (m-1) iterations
2:     **for** i such that $0 \leq i < n$ **do**
3:         **for** j such that $\rho \leq j < 2$ **do**
4:             **Determine** $d_{i,j}$ as in Equation 5.6
5:         **end for**
6:     **end for**
7:     **Select** $d_{i,j} \in P_{opt}$;
8:     $d[n \times 2] \leftarrow$ left_shift$(d[n \times 2])$;
9:     $\rho \leftarrow 1$; ▷ Evaluating only one column
10: **end for**
11: $\chi(A, B) \leftarrow \sum(P_{opt})/k$;

A.4.2. Fuzzy Abstraction

The crux here is to lessen the data size prior to DTW execution. Various techniques have been introduced in the literature for data compression. However, the FuzzyCAT compression method was used due to its simplicity while offering adequate precision. The approximation error – introduced by the compression process – is relative and has no influence on the correlation decision since it is applied to the entire set of contrasted patterns. The full explanation of the FuzzyCAT compressor is given in Chapter 4.

A.5. Performance Evaluation

In this section, the *liteDTW* algorithm has been evaluated and compared to the recursive procedure of the DTW algorithm. Two sets of evaluations have been conducted. The simulations have been devoted to evaluate accuracy, time, and space complexity of both algorithms in a sample setting. In order to validate the profit of adopting the *liteDTW* algorithm for data fusion, a network of TelosB sensor nodes has been simulated in a *Cooja* environment. Both of DTW and *liteDTW* algorithms have been implemented using the C language. The simulator runs on a machine with 2.5 GHz processor and 8 GB RAM with Windows 7 OS. These simulations prove the superiority of *liteDTW* over the naïve algorithm (the recursive implementation) in terms of the matching accuracy and the time/space complexity.

Figures A.4 and A.5 depict two examples of performance comparison between the standard DTW algorithm and *liteDTW*. In these figures, two vibration patterns $NT4$ and $T1$, respectively, have been selected to be contrasted with other vibration patterns (see Section 5.5.1). One thousand data points of each vibration pattern have been used for the comparison. Obviously, the *liteDTW* has an identical precision as the naïve DTW although *liteDTW* solely matches fifty fuzzy-compressed samples. For instance, both algorithms generate a minimum correlation between the patterns $T1$ and $T2$ as shown in Figure A.5. The figure compares accuracy of the proposed *liteDTW* algorithm (blue points) and the standard DTW algorithm (red points). The scattered points represent distance between vibration pattern $NT4$ and other recorded patterns. As it can be seen in the figure, both, DTW and *liteDTW* produce the same match with pattern $NT3$. Aside form matching accuracy, *liteDTW* has a memory footprint of 800 bytes whereas the naïve DTW demands 7.6 MByte using the same data points. Thus, *liteDTW* is an efficient algorithm for detecting correlations.

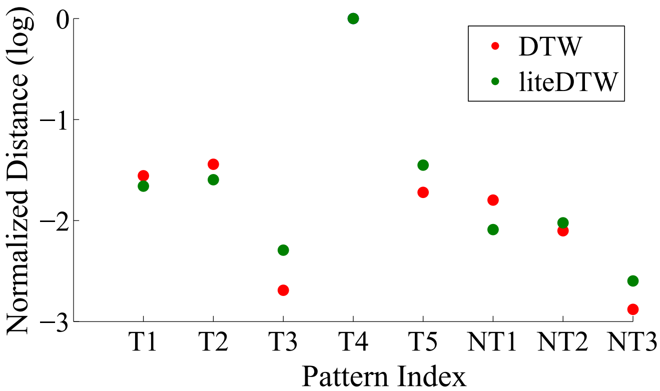

Figure A.4.: Precision of liteDTW versus DTW for $NT4$ matching

The execution time of an algorithm is another significant metrics for recognizing the algorithm's run-time complexity. This set of experiments has not been run on typical sensor nodes – due to complexity of the naïve DTW. However, it gives us a "good" indication about the difference in run-time complexity between these two implementations. Table A.1 lists the obtained results of running the two algorithms. For the *liteDTW* algorithm, it has been tested for different compression ratios including 10%, 25%, 50%,

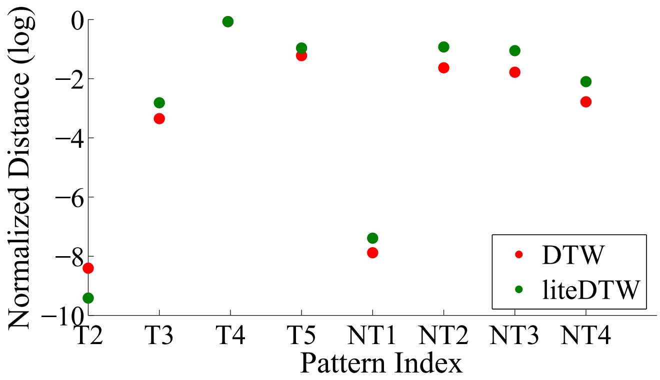

Figure A.5.: Precision of liteDTW versus DTW for $T1$ matching

and 75%. Each of these delay values represents an average of ten executions. Clearly, both algorithms have an exponential growth with increasing window size. However, the $liteDTW$ has a much smaller delay than the naïve DTW even with low compression ratio.

Table A.1.: CPU time consumption (in sec) of the DTW and the $liteDTW$

Window Size	DTW	$liteDTW$			
		10%	25%	50%	75%
200	0.43	0.01	0.02	0.03	0.06
400	0.82	0.02	0.04	0.11	0.24
600	1.88	0.03	0.06	0.26	0.55
800	3.25	0.03	0.12	0.42	0.98
1000	5.54	0.04	0.18	0.69	1.52
2000	21.77	0.13	0.69	2.63	6.11

A.6. Conclusion

In this appendix, a novel approach, referred to as $liteDTW$, has been proposed to speed up the DTW pattern matching algorithm. It utilizes the FuzzyCAT method for reducing the input pattern lengths. Hence, the number of cells in the DTW matrix is drastically reduced. Moreover, the $liteDTW$ algorithm reduces the required memory footprint via dividing the implementation process into several iterations. Thus, the complexity is reduced from $\mathcal{O}(n^2)$ to only $\mathcal{O}(2n)$. A set of experiments proved that the $liteDTW$ is an efficient method in terms of accuracy, and time/space complexity.

Biographical Sketch

Mohamed Abdelaal is working toward his Ph.D. in Computer Science – under supervision of Prof. Dr.-Ing. Oliver Theel – at the Carl von Ossietzky University of Oldenburg, Germany. The doctoral program commenced in December 2012. He got his master degree in Communications Engineering from Port Said University (Egypt) in November 2011. Earlier, he got the Bachelor degree in Communications Engineering from Suez canal University, Egypt, in July 2008. He has the following skills and relevant research experience.

- research experience – beyond five years – in the realm of information processing for wireless networks, particularly in Wireless Sensor Networks (WSNs);
- dealing with various evaluation facilities including simulations (Matlab, OMNeT++, Cooja) and real experiments (Contiki, TelosB motes, SDRs, C language);
- supervising bachelor and master students, and he has a good record of successful joint-work with other Ph.D. colleagues;
- English proficiency with emphasis on academic writing and presentations.

His current research is devoted to the design and analysis of ultra low-power architecture for QoS-aware proactive WSNs. Moreover, he seeks to develop novel ideas towards green networking without affecting system dependability. However, other areas – that interest him for future research – stem from the goal of developing improved analytical models and methods for design, evaluation, and upgrade of these distributed wireless systems. As a teaching assistant at Port Said University, he gained a valuable experience leading undergraduate discussion sections. In addition to classroom instruction, he advised students on appropriate research topics, edited, and evaluated their work. In fact, he was motivated to change the way of classical teaching into more effective methods including workshops, competitions, and interactive sections. Furthermore, he has a clear insight about the future where he believes that better education is feasible through interacting with industry.

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe.
Ebenso versichere ich, dass ich diese Disseration nur in diesem Promotionsverfahren eingereicht habe und dass diesem Promotionsverfahren keine anderen endgültig nicht bestandenen Promotionsverfahren vorausgegangen sind.

Mohamed Abdelaal, Oldenburg, den 06.06.2016