UNIVERSITÉ DE LILLE

ÉCOLE DOCTORALE ED RÉGIONALE SPI72

HEI - YNCRÉA HAUTS DE FRANCE

INRIA LILLE - NORD EUROPE

Thèse présentée par **Jad NASSAR**

Soutenue publiquement le **12 Octobre 2018**

En vue de l'obtention du grade de **docteur de l'Université de Lille**

Discipline **Informatique et Applications**

Ubiquitous Networks for Smart Grids

Composition du jury:

Rapporteurs:	**Dr. Fabrice Théoleyre** *CNRS - Université de Strasbourg, France*	**Prof. Ye-Qiong Song** *Université de Lorraine, France*
Examinateurs:	**Prof. Zied Chtourou** *ENIS, Tunisie* **Dr. Georgios Papadopoulos** *IMT Atlantique, France*	**Dr. Karen Miranda** *UAM Lerma, Mexique* **Prof. Benoît Robyns** *HEI Yncréa HdF, France*
Directeur de thèse:	**Dr. Nathalie Mitton** *Inria, France*	
Co-encadrant de thèse:	**Dr. Nicolas Gouvy** *FGES - HEI Yncréa HdF, France*	

"In memory of my father."

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude et mes sincères remerciements à ma directrice de thèse Dr. Nathalie Mitton. Sa disponibilité, sa réactivité, son immense soutien et surtout son écoute ainsi que ses judicieux conseils ont énormément contribué à alimenter ma réflexion et ont été décisifs pour la bonne réussite de cette thèse.

J'adresse de chaleureux remerciements à mon co-encadrant de thèse, Dr. Nicolas Gouvy, pour m'avoir permis de réaliser ma thèse dans les meilleures conditions et pour son attention de tous les instants sur mes travaux. Je lui suis également reconnaissant pour le temps conséquent qu'il m'a accordé et ses qualités pédagogiques et scientifiques. J'ai pris un grand plaisir à travailler avec lui.

Je remercie Dr. Thierry Lenclud, responsable du département OMI, pour son aide et son soutien au sein de HEI, ainsi que pour l'attention et l'intérêt qu'il a porté vis-à-vis de mon expertise et mes compétences.

Je remercie aussi Prof. Ye-Qiong Song et Dr. Fabrice Théoleyre pour avoir accepté d'être les rapporteurs de mon travail. Merci aux Prof. Zied Chtourou, Prof. Benoît Robyns, Dr. Karen Miranda et Dr. Georgios Papadopoulos pour avoir accepté de faire partie du jury.

Je remercie toutes les personnes avec lesquelles j'ai pu collaborer et travailler pendant ma thèse, plus particulièrement Matthieu Berthomé de l'équipe FUN, Dr. Karen Miranda pendant mon séjour au Mexique ainsi que Prof. Bruno Quoitin et Jeremy Dubrulle de l'université de Mons.

Je tiens à témoigner toute ma reconnaissance à mes financeurs : HEI – Yncréa Hauts de France et la Métropole Européenne de Lille (MEL). Ce travail n'aurait pu être mené à bien sans eux. Je remercie de même Inria Lille pour son accueil durant ma thèse.

Sur un plan plus personnel, je remercie ma famille pour m'avoir soutenu tout au long de mes études. Je désire en outre remercier tous les membres de l'équipe FUN, du département OMI, de la maison de la recherche, tous mes collègues à HEI, doctorants (surtout les Grands Docteurs), ingénieurs (écolo et "végétarien"), mes amis à Reeflex, à Lille, à Belfort et à Paris. Merci pour les discussions enrichissantes, pour votre bonne humeur et surtout pour votre support moral et intellectuel tout au long de cette aventure.

À vous tous, un grand merci !

Résumé

Les Smart Grids visent à transformer le réseau électrique actuel en un réseau "plus intelligent" où la production énergétique est décentralisée et automatisée, facilitant l'intégration des sources d'énergie renouvelables. Cette évolution est rendue possible grâce à l'utilisation d'un réseau de communication pour les multiples échanges de données hétérogènes des Smart Grids. L'objectif de cette thèse est de proposer un paradigme de communication efficace en termes de qualité de service pour les Smart Grids basé sur les réseaux de capteurs.

Dans un premier temps, on s'intéresse au protocole standard RPL. Nous proposons une évolution de celui-ci à travers une nouvelle fonction objectif. Celle-ci tire parti de l'hétérogénéité matérielle des nœuds et des liens pour introduire la qualité de service. Cela permet à RPL de satisfaire les multiples et différentes exigences en termes de fiabilité, de latence et de priorité dans l'acheminement des données. Nos résultats montrent que notre approche permet bien la différentiation du trafic tout en réduisant la latence du routage et en économisant l'énergie.

Nous proposons également d'améliorer l'utilisation du réseau de capteurs en y introduisant l'auto-organisation et la réduction des données. Le but est alors de prédire la valeur des données mesurées plutôt que de les transmettre. Une autre approche explorée est d'agréger les différents messages transitant sur le réseau tout en considérant leurs différentes exigences de qualité de service. Ces deux approches permettent ainsi de réduire la consommation d'énergie tout en respectant les exigences des différentes applications des Smart Grids.

Abstract

Smart Grids aim to transform the current electric grid into a "smarter" network where energy production is decentralized and automated, which facilitates the integration of renewable energy resources. This evolution is made possible thanks to the use of a communication network for the multiple heterogeneous data exchanges of the Smart Grids. Hence, the aim of this thesis is to propose an efficient communication paradigm in terms of quality of service for Smart Grids based on wireless sensor networks.

First, we study data routing in Smart Grids with the RPL standard. Nevertheless, RPL is not suitable for Smart Grid applications in terms of quality of service. Therefore, we propose an objective function for RPL that takes different features of both nodes and links into consideration. Results show that our approach improves network performance compared to existing solutions in terms of packet delivery ratio, network lifetime, latency and traffic differentiation.

Then, we also propose a more efficient data collection by introducing self-organization and data reduction for these wireless sensors. The goal is to predict the value of the measured data rather than transmitting them. Another explored approach is to aggregate the different messages sent across the network while considering their different requirements in terms of quality of service. These two approaches reduce the energy consumption while respecting the requirements of the different applications of the Smart Grids.

Contents

Résumé **vii**

Abstract **ix**

1 Introduction **1**

1.1 Towards A Smart Grid 1
1.1.1 Why Smart Grids? 1
1.1.2 Smart Grid architecture 2
1.1.3 Ubiquitous network for Smart Grid 3
1.2 Contribution Of The Thesis 4
1.2.1 Quality of Service routing in Smart Grids 4
1.2.2 Data reduction in Smart Grids 6
1.3 Structure Of The Thesis 8

2 State Of The Art **9**

2.1 Prerequisites 9
2.2 The Smart Grids 10
2.2.1 Smart Grid technologies: pros and cons 10
2.2.2 Smart Grid communication network 11
2.2.3 Smart Grids communication requirements 13
2.2.4 Applications 14
2.3 Routing protocols for Smart Grids 16
2.4 RPL protocol: how does it work? 18
2.5 RPL related work 21
2.5.1 SG based RPL proposed modifications 21
2.5.2 ETX & HC: main metrics in RPL objective functions . 22
2.5.3 Energy aware and load balancing metrics for RPL . . 22
2.5.4 Multipe objective metrics for RPL 23
2.5.5 Multiple instances 23
2.5.6 Observations 24
2.6 Summary 24

3 Quality of Service Routing in Smart Grids **27**

3.1 Overview 27
3.1.1 Smart Grid heterogeneous traffic 27
3.1.2 RPL multiple instances 27

3.2 Proposed Solution 28
3.2.1 *OFQS* objective function 28
3.2.2 QoS factors in *OFQS* 28
3.2.3 *mOFQS* metric 29
3.2.4 Instances classification 32
3.3 Evaluation 32
3.3.1 Performance metrics 33
3.3.2 Simulation 33
3.3.2.1 COOJA simulator 33
3.3.2.2 Simulation setup 34
3.3.2.3 Performance evaluation: Simulation 35
3.3.3 Experiment 38
3.3.3.1 FIT IoT-LAB test-bed 38
3.3.3.2 Battery level measurement 39
3.3.3.3 Network setup 40
3.3.3.4 Performance evaluation: experiment 41
3.4 Evaluation: Simulation vs Experiment 44
3.5 Discussion And Possible Improvements 45
3.6 Summary 46

4 Data Prediction in Smart Grids **47**
4.1 Related Work 47
4.1.1 Data prediction different schemes 48
4.1.2 LMS algorithm overview 49
4.1.3 Time series forecasting and LMS proposed variants . 51
4.2 LMS Limitations 52
4.3 Our Contribution: LMS_MOD 53
4.3.1 Simulation setup and parameters determination . . . 54
4.3.2 Performance Evaluation 55
4.3.2.1 root mean square error 56
4.3.2.2 Data reduction percentage 57
4.4 Discussion And Possible Improvements 58
4.5 Summary 60

5 Data aggregation in Smart Grids **63**
5.1 Related Work 64
5.1.1 Data aggregation means 65
5.1.2 Data aggregation in WSN and SGs 65
5.2 Proposed Solution 66
5.3 Expected Results 72
5.4 Summary 73

6 Conclusion and Perspectives **75**
6.1 Conclusion 75
6.2 Perspectives 76
6.2.1 Short term 76
6.2.2 Long term 77
6.3 From A Smart Grid To A Smart City 78

List Of Publications **81**

Bibliography **83**

List of Figures

1.1 Smart Grid architecture 2
1.2 Smart Grid metering data collection. 5

2.1 Smart Grid communication network [33] 12
2.2 Rank concept in RPL 19

3.1 $mOFQS$ variation with α. ETX=1, d=1 30
3.2 Network with different ETX, delay d (in ms) and PS values 30
3.3 COOJA simulation window 34
3.4 Network lifetime variation 36
3.5 Remaining energy distribution among the nodes after 24 hours 36
3.6 End-to-End delay variation with time 37
3.7 Packet delivery ratio percentage for the different OFs/metrics 38
3.8 Topology of the deployment on FIT IoT-LAB Lille's site . . . 39
3.9 Hardware of an IoT-LAB node [75] 39
3.10 End-to-End delay variation with time 42
3.11 Network lifetime variation 42
3.12 Remaining energy distribution among the nodes after 30 minutes 43
3.13 Gain comparison between simulation and experiment . . . 45

4.1 Categorization of energy saving in sensor networks [23] (data prediction highlighted) 48
4.2 Adaptive filter [28] 49
4.3 Diagram of the modified LMS algorithm process (the modification marked in blue) 54
4.4 RMSE for irradiance 57
4.5 RMSE for temperature 57
4.6 RMSE for humidity 58
4.7 RMSE for average wind speed 58
4.8 Data reduction for irradiance 59
4.9 Data reduction for temperature 59
4.10 Data reduction for humidity 60
4.11 Data reduction for average wind speed 60

5.1 Categorization of energy saving in sensor networks [23] (data aggregation techniques highlighted) 64

5.2 Aggregation scenario 68
5.3 Aggregator node algorithm 68
5.4 Lossy/Lossless queue algorithm 69
5.5 Data aggregation example in a WSN 71

List of Tables

2.1 NAN requirements in terms of reliability[12] 17

3.1 Paths values for the different metrics used 31
3.2 Parameters of the simulation 35
3.3 Parameters of the experiment 41

4.1 Data traces description 55
4.2 Data traces obtained parameters 56

List of Abbreviations

AMI	**A**dvanced **M**etering **I**nfrastructure
DER	**D**istributed **E**nergy **R**essources
DIO	**D**ODAG **I**nformation **O**bject
DODAG	**D**estination **O**riented **D**irected **A**cyclic **G**raph
ETX	**E**xpected **T**ransmission *Count*
HAN	**H**ome **A**rea **N**etwork
HC	**H**op **C**ount
LLN	**L**ow-power and **L**ossy **N**etworks
LMS	**L**east **M**ean **S**quare
MAC	**M**edia **A**ccess **C**ontrol
MRHOF	**M**inimum **R**ank with **H**ysteresis **O**bjective **F**unction
MSE	**M**ean **S**quare **E**rror
MTU	**M**aximum **T**ransmission **U**nit
NAN	**N**eighborhood **A**rea **N**etwork
OF0	**O**bjective **F**unction *Zero*
PDR	**P**acket **D**elivery **R**atio
PLC	**Power** **L**ine **C**ommunication
QoS	**Q**uality of **S**ervice
RMSE	**R**oot **M**ean **S**quare **E**rror
RPL	**Routing** **P**rotocol for **L**ow-power and lossy networks
SG	**S**mart **G**rid
WAN	**W**ide **A**rea **N**etwork
WSN	**W**ireless **S**ensor **N**etwork

Chapter 1

Introduction

In this first chapter, we firstly introduce the general context of the thesis and the necessity of Smart Grids. After that, we present our contributions. Finally, we provide an overview of the structure of this thesis.

1.1 Towards A Smart Grid

1.1.1 Why Smart Grids?

Conventional power grids are "complex systems" that generally consist of the interconnection of various power system elements such as synchronous machines, power transformers, transmission lines, transmission substations, distribution lines, distribution substations, and different types of loads. These are located far from the power consumption area and electric power is transmitted through long transmission lines. This system is characterized by a relatively simple hierarchical unidirectional flow of electricity through the grid, from a few generators to a large number of consumers.

These power grids no longer satisfy the need of energy of the twenty first century. The increased electricity demand per person is limited by the restrained electricity production and these aging and unsuitable infrastructures. This limitation is also due to inaccurate management systems, inefficient operations and maintenance processes, the need for a huge human intervention and a centralized communication system that lacks interoperability.

Besides that, the introduction into the electricity grid of multiple sporadic Distributed Energy Resources (DERs) e.g., electric vehicles, photovoltaic cells, wind farms, located in sometimes unexpected places, makes its control even more complex [1].

Smart Grid (SG) promises to solve these issues by operating with automatic control and operation in response to user needs and power availability, improving efficiency, reliability and safety, with smooth integration of numerous renewable and alternative energy sources.

1.1.2 Smart Grid architecture

Figure 1.1 shows the classical architecture of the SG (we note that some elements of this architecture may differ [2, 3] from a city or a country to another). It consists of four functional domains (bulk generation, transmission, distribution and customer [consumption and generation]), a two-way flow of electricity and a communication network that ensures bidirectional information exchange.

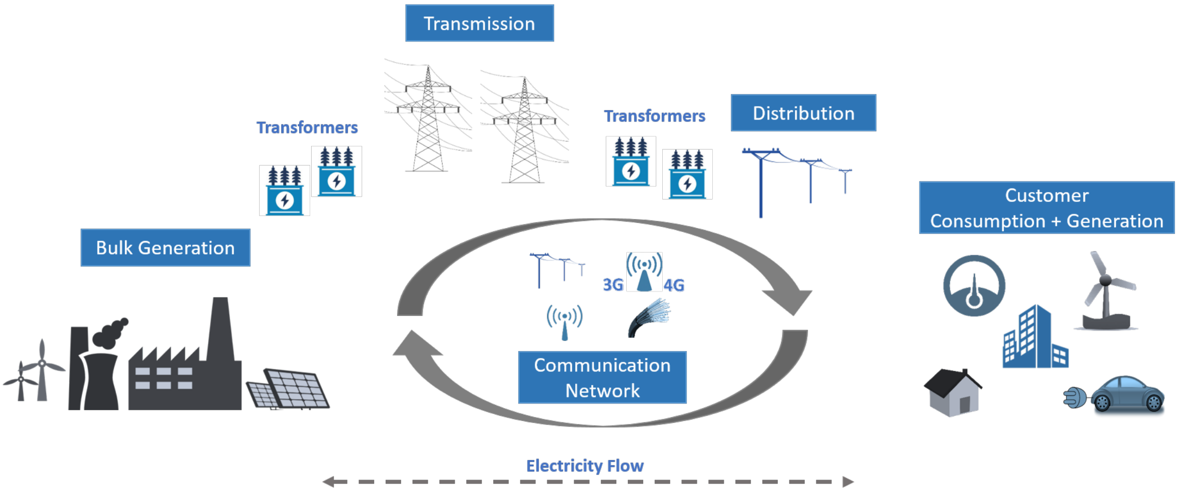


FIGURE 1.1: Smart Grid architecture

In the bulk generation, the electricity that power plants generate is delivered to customers over transmission and distribution power lines. High-voltage transmission lines, like those that hang between tall metal towers, carry electricity over long distances to where consumers need it. Higher voltage electricity is more efficient and less expensive for long-distance electricity transmission [4]. The generator voltage is usually around 15 to 25 KV (KiloVolt) [5]. This relatively low voltage is not appropriate for the transmission of energy over long distances. For that, at the generating station, a transformer is used to increase the voltage and reduce the current. The voltage is increased to 500 KV and a high-voltage line transmits the generator-produced energy to a distant substation.

Lower voltage electricity that is safer for use in the customer domain is delivered through distribution lines. For that, the voltage is then reduced to 12 KV at the distribution substation. Several distribution lines emanate from each distribution substation as overhead or underground lines and distribute the energy along streets and alleys. The distribution transformers reduce the voltage which supplies houses, shopping centers, and other local loads. In other words, transformers at substations increase (step up) or reduce (step down) voltages to adjust to the different stages of the journey from the power plant on long-distance transmission lines to distribution lines that carry electricity to homes and businesses.

In the customer domain, every household or commercial establishment can generate electricity (under certain authorizations) by installing solar photovoltaic panels or wind farms on their rooftops and become electricity producers. In that way, they can meet their electricity demand partly or fully by themselves, and even selling excess electricity to the distribution utilities which will allow bidirectional electricity flow.

Moreover, in a SG, a two way communication network sits on top of the conventional power grid to allow the required information (e.g., control and consumption messages) to be exchanged across the network. Communication technologies range between wireless and/or wired [6]. This network typically consists of a Home Area Network (HAN), which is used to gather data from a variety of devices within the household, a Neighborhood Area Network (NAN) to connect smart meters to local access points and a Wide Area Network (WAN) responsible of the decision making and connecting the grid to the utility system (more details about the communication network will be provided in Chapter 2).

It is useful to note that Enedis, the company which is in charge of the electricity distribution network in France, uses G3-PLC standard [7] which is a physical layer specification for Power Line Communication (PLC) to manage the communication network. It consists of using the power grid as a communication medium by superposing a high frequency electrical signal on the electrical signal. G3-PLC allows two-way communications that provide electricity network operators with intelligent monitoring and control capabilities. Besides that, as the power lines are managed by the operator, PLC makes it then possible to be independent of any other service provider and the associated costs. However, PLC suffers from many drawbacks such as high noise sources and interference, low data rates, open circuit problems (communication over the power lines is lost with devices on the side of an open circuit) [6].

1.1.3 Ubiquitous network for Smart Grid

In order to shift from the existing electric grid to the self-organizing and communicating SG, it appears necessary to instrument and master the high level and complex energy supervision on the electric grid. Managing the SG with a ubiquitous network to exchange regular and critical control messages all over the power network becomes then crucial. Consequently, one of the potential solutions envisioned is to equip the electrical grid with wireless sensors located at strategic measuring points to achieve remote monitoring, data collection and control of the grid [8]. These sensors are able to communicate together via the radio medium. It is the set of thousands of these communicating sensors distributed on the grid that will establish a parallel wireless data network (wireless sensor network), that will constitute the ubiquitous network of the SG, the real nervous system of it.

When compared to conventional wired communication networks, wireless communication technologies and Wireless Sensor Network (WSN) in our case have potential benefits in order to remotely control and monitor substations, e.g., savings in cabling costs and rapid installation of the communication infrastructure. They offer several advantages like their ease of deployments, scalability regarding the expandability of the network, resilience and robustness due to their ability to cope with node failures, infrastructure less and low cost in terms of material and deployment (e.g., in wireless communication, cabling cost is eliminated.) [9], etc. On the other hand, WSNs are more susceptible to electro magnetic interference and often have limitations in bandwidth capacity, autonomy and maximum distances among communication devices [6]. These issues have to be carefully considered when designing a SG network.

Furthermore, SG applications are heterogeneous in terms of requirements, criticality and delay tolerance [10, 11, 12]. These applications generate different types of traffic (real-time, critical, regular) [13]. Consequently, they require different levels of Quality of Service (QoS).

Thus, for a WSN, different criteria have to be taken into consideration in order to achieve a reliable communication with the following requirements: reliability, latency, auto-configuration, auto-adaptation, network scaling and data prioritization [6, 13, 14].

1.2 Contribution Of The Thesis

Given the SG requirements in terms of diversity of applications and traffic, the aim of this thesis is to propose a heterogeneous and QoS efficient communication paradigm for SGs based on WSNs.

At first, we address the QoS routing in SGs and traffic differentiation with the RPL (Routing Protocol for Low power and lossy networks) standard. After that, as effective as the QoS routing in SGs, and considering the WSNs challenges previously discussed, we are also interested in a more effective feedback of information by introducing self-organization and data reduction in SGs.

1.2.1 Quality of Service routing in Smart Grids

In a WSN, and in order to be able to address each sensor individually, for example to activate/disable the energy source or measure its level of production in a SG environment, the nodes cooperate to relay messages from one sensor to another across the network. Those are the algorithms called routing protocols which are in charge of determining which path to follow in the ubiquitous network of thousands of sensors to reach a given sensor and be able to communicate with it, and thus act on the SG. We understand therefore the crucial importance of routing protocols.

Among all the existing routing protocols used in the SGs, the Internet Engineering Task Force (IETF) standard RPL [15] remains the most recognized and widely used [16, 17].

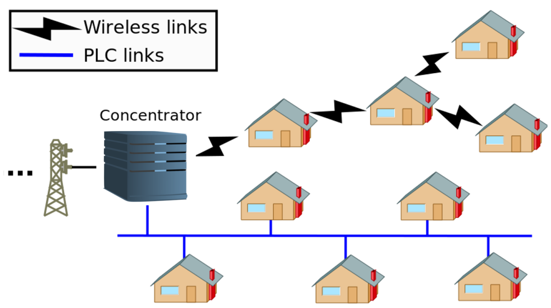


FIGURE 1.2: Smart Grid metering data collection.

As described in [18], RPL meets the scalability and reliability constraints of SG applications (e.g., Advanced Metering Infrastructure) and is recommended by the SG standards. Alongside with its support for wireless communications, RPL can be used with PLC [19]. Figure 1.2 shows how smart meters (represented by houses) can send their measurements to the concentrator via wireless or PLC links. The same Media Access Control (MAC) layer can be compatible with a physical layer using wireless or PLC communications. We note that other protocols [20] are used for routing in SGs but these protocol do not support traffic differentiation which is an important aspect for SG applications.

As a general protocol, RPL is intended to meet the requirements of a wide range of Low-Power and Lossy Networks (LLNs) application domains including the SGs ones. It allows an improvement and adaptation margin via different objective functions. They optimize the routing metrics that are used to build the routes and hence help in choosing the best route. It provides different QoS classes at the network layer through multiple logical subdivisions of the network called instances (more details in Section 2.4).

However, RPL standard objective functions do not allow traffic differentiation and do not fulfill the heterogeneous requirements of SGs in terms of QoS.

Following RPL, Request For Comments (RFC) 8036 proposes five different priority classes for the traffic in SG Advanced Metering Infrastructure

(AMI). Other work [21] classify the traffic into two levels: critical and periodic.

Based on that and since the traffic classes in the SG are not standardized, a single solution to route the traffic with different QoS may not be sufficient since the number of instances (traffic classes) vary depending on the application and the implementation. A multi-objective approach is thus essential to meet the QoS requirements of SG applications.

Therefore, we introduce $OFQS$, an RPL compliant objective function, with a multi-objective metric that considers the delay and the remaining energy in the battery nodes alongside with the quality of the links. Our function adapts to the number of instances providing a QoS differentiation based on the different SG applications requirements. Computer simulations and real test-bed experiments show that $OFQS$ provides a low packet delivery latency and a higher Packet Delivery Ratio (PDR) while extending the lifetime of the network compared to solutions in literature.

1.2.2 Data reduction in Smart Grids

In a SG, electricity and energy do exist, but connecting sensors to such high voltage with intermittent and ill-adapted energy levels is sometimes inappropriate. Moreover, self powered sensors are easier to deploy. For that, battery-powered sensors must be deployed all over the grid alongside with the main-powered ones. Thus, reporting data measurements at specific intervals has a direct effect on the sensors battery lifetime since the communication task consumes most of their available energy [22]. In such context of continuous data reporting, data changes are limited between each reading, which may cause redundant information at the destination. To mitigate these energy loses and increase the network's lifetime, data reduction approaches are used [23].

These approaches can be classified into three main categories [23]: In-network processing, data compression and data prediction:

- In-network processing consists of processing the data collected by the sensor nodes themselves between the source and the destination [24], in this way the amount of data is reduced while traversing the network.
- In data compression, data is generally reduced by performing data aggregation techniques [25] on specific nodes called "aggregators".
- Data prediction aims for reducing wireless radio transmissions in the network which in turn will reduce the amount of data sent by each sensor. This is done by predicting the measured values using specific

algorithms [26], which will require sending the predicted information to the destination (sink) only if it is shifted from the sensed one by a certain threshold.

Although data reduction techniques are widely used in literature, their adaptability is limited to specific applications. Thus, using these techniques for SG applications requires specific customization since such applications are characterized by their diversity in terms of data types and QoS requirements.

Therefore, our proposition concerning data reduction is twofold; at first, we consider data prediction techniques. This will allow us to limit redundant information at the destination nodes in a SG (based on WSN) environment where data variation is limited (e.g., photovoltaic cells monitoring). Later, we address data aggregation techniques. Here, broader applications are covered where collected data is not necessarily homogeneous. Aggregation will then enable decision making (regarding the routing) not only at the level of the sender and destination nodes as in data prediction, but potentially at every node in the network, aggregating jointly different data packets with different priorities and QoS requirements.

Both of these data reduction techniques will reduce loads on the communication links, thus achieving a better utilization of the wireless channel and reducing energy consumption. Choosing the one or the other will depend on the characteristics of the SG application and the data it generates.

For data prediction, we focus on a time series forecasting technique, called Least Mean Squares (LMS). This is an adaptive algorithm with very low computational overhead and memory consumption, that despite its simplicity, provides satisfactory performances in terms of computational speed, robustness and precision [27].

LMS main drawback is the complex task of choosing the adequate parameters. This directly impacts the stability of the algorithm specially when using it with different data types as is the case in a SG context [12].

We propose a modification for the LMS filter used for data prediction in WSN, which is introduced in [28], to adapt it to the different data types. We apply the algorithm to photovoltaic cells monitoring data set. We tune the parameters by training it offline for a certain time with the real data values of every data set and choosing the values that minimizes the Mean Square Error (MSE). Different parameters are obtained after the training process in accordance with every data type. Our simulation results show a better data prediction and a lower MSE compared to literature.

Now concerning data aggregation, and as already mentioned, in a SG different applications require different QoS priorities. Consequently, data aggregation must respect these requirements (i.e, delays caused by aggregating the packets) in order to ensure a reliable communication.

Therefore, we propose a QoS efficient data aggregation algorithm for the different traffic in a SG network. The expected results will reduce the energy consumption in the network while respecting the QoS requirements of SGs.

1.3 Structure Of The Thesis

Two main topics are addressed in this thesis: the first topic is the QoS routing in the SGs. The second one is data reduction in SGs which in turn is elaborated into two parts: data prediction and data aggregation in SGs.

In Chapter 2, we provide an outline of the SG different aspects. After that, we address the routing in the Smart Grids followed by an overview of the RPL protocol and the main contributions around it.
Chapter 3 is dedicated to the first topic - The QoS routing in the SGs. First, we present an overview on why we need multiple instances in RPL. Then, the proposed solution is provided and explained in detail. Finally, simulation and experiments results are presented and discussed.
Second topic of the thesis is discussed first in Chapter 4 - Data prediction in SGs. A brief introduction on the requirements of data prediction in SGs is provided alongside with the main work concerning data prediction in WSN and SGs in literature. After that, our proposition and simulation results are presented and discussed.
Chapter 5 tackles the second part of the data reduction topic - Data aggregation in SGs. In this chapter, we firstly provide a brief overview of data aggregation requirements in a SG environment. State of the art solutions are presented next. Last, we present our proposition, deeply analyze it with examples and discuss the expected results.
Finally, Chapter 6 concludes the work presented in the previous chapters. Future work and perspectives that could improve SG efficiency and the overall performance of the proposed solutions are drawn.

Chapter 2

State Of The Art

In this chapter, we firstly provide an overview of the Smart Grids different aspects. After that, we address the routing in the Smart Grids and provide a comprehensive overview of the RPL protocol and the main work around this standard.

2.1 Prerequisites

Before starting this chapter we briefly provide some prerequisite requirements of some useful statements/expressions:

- MAC layer: the MAC layer provides addressing and channel access control mechanisms that enable several terminals or network nodes to communicate in a network. It uses MAC protocols to ensure that signals sent from different stations across the same channel don't collide.
- Network layer: it is the layer that provides data routing paths for network communication. It selects and manages the best logical path for data transfer between nodes. This layer contains hardware devices such as routers, bridges, firewalls and switches.
- Header and payload: in a data packet, the payload is the data itself that needs to be transferred (usually the user's data). While the header identifies the source and destination of the packet and other control information, it is removed then from the packet when it reaches its destination and the payload is the only data received by the destination system.
- Routing protocol: a routing protocol also known as routing policy uses software and routing algorithms to determine optimal network data transfer and communication paths between network nodes. It specifies how these nodes communicate with each other, distributing necessary information that enables them to select routes between any two nodes on a computer network.

- Proactive, reactive and hybrid routing: : in proactive routing protocols every node stores information in the form of tables, the routes to all the destination (or parts of the network) are determined at the start up, and maintained by using a periodic route update process. In reactive protocols, routes are determined when they are required by the source using a route discovery process. Hybrid routing protocols combine the basic properties of the first two classes of protocols into one.
- Distance vector routing protocol: in distance vector routing, each router sends its neighbors a list of all known networks along with its own distance to each one of these networks. In other words, each router depends on its neighbors for information, which the neighbors in turn may have learned from their neighbors, and so on.

2.2 The Smart Grids

In this section we provide a quick overview on the SG technologies. We present then the typical architecture of the SG communication network, the corresponding communication requirements, alongside with the main SG applications.

2.2.1 Smart Grid technologies: pros and cons

Network technologies that are used for communication, distribution, transmission and customer domain in SGs are the key components of the SG infrastructure. The integration of these technologies into the electricity infrastructure will allow the exchange of a huge amount of data from the different SG entities. Hence, it will allow a greater control and vision about the grid for the utilities and provide the customers with many additional facilities, complying the new applications requirements that will come along. Choosing the adequate technology is then crucial, depending on the required data rates, security, reliability level and many other criteria. These communication technologies can be split into two main categories regarding their communication medium: wired and wireless.

- Wired communications: many wired technologies are used for SG communications, such as Power Line Communication (PLC), which uses the existing power lines for data transmissions. PLC main advantage is its existing infrastructure which decreases the installation cost of the communication infrastructure. Moreover, PLC technology is convenient for utilities from a security perspective since it will be controlling its own wired network. However, in PLC data rates are extremely variants and the communication medium suffers from high noise sources and interference [29]. Fiber optics are also a feasible wired technology in SG WAN applications mainly [30]. Data packets are transmitted through optical fibers with supported data rates between 155 Mbps and 160 Gbps [30]. Fiber optics have the ability of providing reliable, high performance and long distance communications with high data rates. However, their cost of deployment is extremely high compared to an already existing PLC infrastructure. Other wired technologies such as Ethernet, coaxial cables and Digital Subscriber Lines (DSL) communications are also used for data communications in SGs and mainly HAN and NAN [31].

With that being said, we can summarize that wired technologies are costly for wide area deployments but they may offer an increased reliability, capacity and security.

- Wireless technologies: such as Zigbee, WiFi, Bluetooth, Microwave, WSN, LPWAN [31, 32], etc. These technologies enable connecting devices in a wireless way, eliminating the cost of installation of wirelines. They offer several advantages like their ease of deployments, scalability regarding the expandability of the network, resilience and robustness due to their ability to cope with node failures, infrastructure less and low cost in terms of material and deployment. However, wireless signals are generally significantly subject to transmission attenuation and environmental interference. Moreover, they are susceptible to electro magnetic interference in SG environments.

To conclude, none of the wired or wireless technologies is suitable for all the application types, and there is always a technology that may be a best fit for a specific application.

In this thesis, our motivation for using WSNs comes from the above mentioned advantages for wireless communications, and specially their ease of deployment and expansion. This has significant benefits with the rapid growth and apparition of distributed energy resources across the electric grid. Furthermore, their ability to interoperate with other technologies makes them a good candidate in an heterogeneous SG environment.

2.2.2 Smart Grid communication network

SG promises to transform the current electric grid into a smarter network by operating with automatic control and smooth integration of renewable energy resources. For that, there is a clear need for communication networks supporting reliable information transfer between the various entities in the electric grid. Figure 2.1 shows a typical SG communication network architecture.

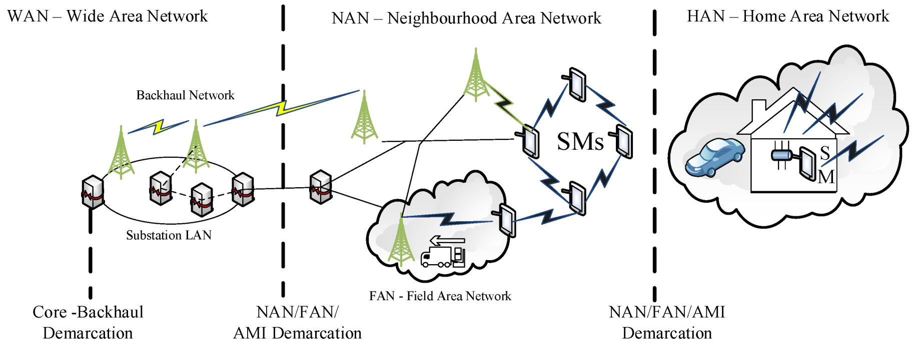


FIGURE 2.1: Smart Grid communication network [33]

It consists of a HAN, which is used to gather data from a variety of devices within the household, a NAN to connect smart meters to local access points, and a WAN to connect the grid to the utility system.

In the following, a description of the different entities is provided alongside with the corresponding communication technologies used.

- HAN is the first layer in the subsystem of the SG [34] that plays a critical role in the control of the home appliances, the proper use of electricity, and in lowering gas emission. HAN is used to gather data from a variety of heterogeneous devices within the household e.g., smart meters, in-home displays, home energy management systems. These devices have the potential to enable two-way energy flow communication between utilities and end-users. The data collected is then used by the utility company to obtain information about the global energy usage of the household. HAN connects devices using either wired technologies (e.g., coaxial cables, PLC, twisted pairs, fiber optics) or wireless technologies (e.g., WIFI, Bluetooth, LPWAN) [35, 32]. In most cases, wireless technologies are preferred due to their ease of deployments and remote control access, providing an independent or even a backup network to the current existing wired network. We note that at the same level of HAN we can find Building Area Network (BAN) and Industrial Area Network (IAN) [30] which are used for commercial and industrial customers with focus on building automation, heating, ventilating, air conditioning and other industrial energy management applications.

- NAN forms data transmission bridges between utility backbone and households/buildings in a SG [36]. It connects smart meters to local access points for remote metering applications. This can be a network of smart meters and sensors creating a mesh, as well as a part of a mesh network collecting electricity data from multiple HANs and forwarding it to the backbone through NAN gateways. The version of this network which is deployed to collect data from power lines, mobile workforce, towers, etc. for power grid monitoring is referred to as Field Area Network (FAN) [37]. Different communication and networking technologies are deployed in NANs such as WSN, Broadband Power Line Communications (BPLC), fiber optics, etc.

- WAN represents the backbone of the SG, it provides decision making in the control centers and the link between the grid and the core utility systems. WAN comprises two types of networks: Core and Backhaul. While the Core Network is used to connect the network of the utility and substations, the Backhaul Network is used to connect NAN to the core network and routes the data from the NANs to the private networks of the service providers. In most cases, the used technology in WANs is wired/optical and routing is handled by means of a public network such as the Internet or private lines [37].

  These entities, having heterogeneous communication technologies and applications, will coexist and cooperate in order to deliver the data on a "multi-technology" SG network. This raises the importance of implementing adequate routing protocols that supports diverse technologies and traffic types.

2.2.3 Smart Grids communication requirements

In a SG, the power grid infrastructure will coexist with an advanced communication system where extensive applications for consumers, manufacturers and utilities generate heterogeneous data types. In this vision, the potential promises of the SG are numerous [38], including the integration of a significant number of renewable energy resources and electric vehicles, increasing the energy efficiency and the available information for customers regarding their consumption, etc.

In order to enable such facilities, the SGs communication network necessitate an efficient communication network with numerous requirements [31, 35, 39]. In the following, we outline some of the main requirements that should be satisfied by the SG communication network.

- Reliability: One of the most important benefits of SG technologies is to improve the reliability of the electric grid [31]. Utilities will not introduce elements that could compromise reliability. Thus, the communication system should potentially guarantee data transfer according to specific requirements. Table 2.1 shows the reliability tolerance of the different SG applications where we can see that some highly critical traffic of DA, DSM, AMI, etc. require a reliability of more than 99.5%.

- Scalability: Scalability is a critical requirement for SG networks and especially NANs where a huge number of heterogeneous devices are connected in large areas [36]. Moreover, sporadic Distributed Energy Resources (DERs) are deployed and could potentially be added within time. The network must then be capable to follow the expansion of the power system without the need of a complex infrastructure deployment.

- Latency: Latency in SGs corresponds to the delay of data delivery between SG components. Table 2.1 shows the maximum allowed delays for the different SG applications, which reflects the required latency for these latter. Some critical applications may not tolerate any latency (e.g., DA critical traffic). For others latency is not critical (e.g., Network configuration traffic) [12].

- Traffic prioritization: SG applications are heterogeneous in terms of requirements and criticality. Some mission-critical applications may not tolerate any delay. Others like regular data collection could possibly have a bigger tolerance margin. Traffic prioritization is then crucial in order to route the different data and control packets within their tolerated QoS requirements.

- Interoperability: SG components are heterogeneous and diverse. Interoperability is the ability of these components to work together and exchange information cooperatively [35]. This will enable an effective two-way communication and integration among the different interconnected elements of the SG. The National Institute for Standards and Technology (NIST) provides extensive details [40] about the interoperability frameworks, protocols and standards for SGs.

2.2.4 Applications

In a SG, reliable and online information (e.g., command and control messages and instant information on the usage of the production units) becomes the key factor for efficient delivery of power from the generating units to the end users. The impact of equipment failures, capacity limitations, and natural accidents and catastrophes, which cause power disturbances and outages, can be largely avoided by online power system condition monitoring, diagnostics and protection [8]. These new applications have frequently been studied and presented in literature [10, 11, 12, 41], and many classification were made [1, 38, 39, 42]. In this section, we present five main SG applications. We study their main functionalities and advantages.

- Advanced Metering Infrastructure (or AMI) consists of an integrated system of smart meters for measuring, collecting, analyzing and communicating energy consumption of smart appliances. Enabling two-way communication between utilities and customers and providing a number of important functions that were not previously possible or had to be performed manually. Such as the ability to automatically and remotely measure electricity use, connect and disconnect to a service, identify and isolate outages and monitor voltage.

- Demand Side Management (DSM) consists of a set of interconnected and flexible programs which grant customers a greater role in shifting their own demand for electricity during peak periods, and reducing their overall energy consumption. DSM comprises two principal activities:

  - Demand Response (DR) or load shifting which aims to transfer customer load during periods of high demand to off-peak periods. The grid operator or other stakeholders influence the customers behavior mostly by monetary incentives, allowing them to participate in the energy market competition by changing their energy consumption approach instead of being passively exposed to fixed prices. This results in profits for both, the companies and the end-users.
  - Energy efficiency and conservation programs which allow customers to save energy while receiving the same level of end service, such as when they replace an old electric appliance with a more energy efficient model.

- Distribution Automation (DA) is defined as the ability of taking an automated decision to make more efficient fault detection, isolation and restoration in a grid. This is done by remotely monitoring, controlling, manipulating and coordinating distribution, improving then the reliability across the grid. DA offers new features, it incorporates alarming and automated feeder switching, which in turn will help reduce the frequency and duration of customer outages. Substation automation is achieved through Supervisory Control And Data Acquisition (SCADA) systems which are able to make these automated decisions in real time by running algorithms based on the data they receive and orchestrate adjustments to optimize voltages and self-heal any failure issue.

- Distributed Energy Resources (or DERs) such as photovoltaic cells, wind turbines and energy storage points present one of the main benefits in a SG. These DERs will be able to supply particular areas with electricity when they are isolated from the main power grid due to failure conditions or system and equipment failures. Moreover, these DERs foster the shift from a centralized power system towards a more decentralized one. This is achieved by contributing to the evolution of local grid areas served by one or more distribution substations and supported by high penetrations of DERs called microgrids. It is important to note that the introduction of these DERs located in sometimes unexpected places into the network raises challenges in managing, controlling and exchanging messages across the grid [1] due to their sporadic nature.

- Electric transport via electric vehicles (PEV: Plug-in Electric Vehicles) or hybrid electric vehicles (PHEV: Plug-in Hybrid Electric Vehicles) aims to improve or even replace traditional transport by reducing emissions produced by fossil fuels. For that, an electric vehicle uses one or more electric motors that are powered by a rechargeable electric accumulator. SGs can better manage vehicle charging so that rather than increasing peak loads, the charging can be carried out more strategically. For example when electricity demand is low or when the production of renewable electricity is high. In the long run, SGs can use electric vehicles as batteries to store renewable and other sources of electricity for later use.

However, since these applications will generate different types of traffic (real-time, critical, regular) [13], they require different levels of QoS. Table 2.1 shows the diversity of delay tolerance and reliability for the different NAN applications [12]. Thus, for a WSN, different criteria have to be taken into consideration in order to achieve a proper communication with many requirements such as reliability, latency, auto-configuration, auto-adaptation, network scaling and data prioritization [13].

2.3 Routing protocols for Smart Grids

Several routing protocols were proposed for data routing in Smart Grids and particularly in NANs [20]. In the following, we briefly present some of the latest and mostly used ones.

- LOADng (Lightweight On-demand Ad hoc Distance vector routing protocol, Next Generation) [43] is a reactive routing protocol that establishes routes towards the destination only on demand when there is some data to send. LOADng is an adapted version of the AODV protocol [44] to make it suitable for LLN. It uses flooding and RREQ/ RREP messages to establish routes within the network. LOADng

Data traffic	Maximum allowed delay	Reliability
DA - Data related to the protection of the distribution network	<3 s	>99.5%
DERs - Data related to the protection of the distribution network	<4 s	<99.5 %
Critical traffic of: DA, DSM, AMI, DERs	<5 s	>99.5%
Electric transport	<10 s	>98%
Non critical traffic of DSM & AMI	<15 s	>98%
Non critical traffic of DA & AMI	<30 s	>98%
Network configuration traffic, normal AMI traffic	<5 min	>98%
Normal AMI traffic	<4 h	>98%
Network configuration traffic	< Hours/Days	>98%

TABLE 2.1: NAN requirements in terms of reliability[12]

main drawbacks [45] are the route discovery delay (during the routing discovery process, outgoing packets are stored in buffers which may cause losses in memory constrained devices). Flooding is also an issue in networks with limited autonomy devices (e.g., WSN) causing unnecessary battery depletion and collisions of control messages.

- DADR (Distributed Autonomous Depth first Routing) [46] is a proactive distance vector routing protocol for path maintaining or path repair that adapts to link changes and minimizes control overhead in the network. It uses a lightweight mechanism to provide redundant paths for each destination and Depth First Search (DFS) guided by the routing table and backtracking mechanism for path recovery after link failures. However, loop detection false positive and false negative might occur if the Frame ID (FID) table is not well maintained (the FID table is used to store the previous sender and the next hop each time a packet is forwarded). DADR may also increase CPU and memory overhead on intermediate nodes due to additional mechanisms in the data-forwarding phase [20].

- GRACO (Geographic GReedy routing with ACO recovery strategy) [47] is a geographic routing algorithm that combines a pheromone-assisted greedy forwarding mode and an Ant Colony Optimization (ACO) based recovery mode. GRACO makes the routing decision using geographic greedy forwarding strategy. If it is not possible (if a packet arrives to a node that has no neighbor closer to destination than itself), an ACO based recovery strategy is launched to find the path.

- HYDRO [48] uses a distributed algorithm that combines local agility with centralized control. It forms a Directed Acyclic Graph (DAG) to provide multipath routing to a border router. Each node builds its default route table by adding its neighbor nodes towards a border router. The table entries are then ranked/ordered according to the link-layer packet success rate. Topology reports are sent with the periodic collection traffic, allowing border routers to build and maintain a global view of the topology.

However, although these protocols may be suitable for some SG applications, their adaptation remains limited to SG heterogeneous applications with different QoS levels and traffic differentiation (some of them like the enhanced version of GRACO, QoS-GRACO [49], may be good candidates for future research directions in SGs). Therefore, RPL, our protocol of interest in this thesis fits well, due to its design, with low power and lossy environment applications including SG ones. Particularly, the multiple instance feature allows traffic differentiation in the network level alongside with customizable objective functions for the different traffic types. Moreover, RPL remains one of the most recognized standard protocols and widely used for SGs. It is compatible with the main operating systems of the Internet of Things (e.g., RIOT [50], Contiki [51]) which makes it a hot research topic for possible improvements. Indeed, RFC 8036 [19] explains how RPL can meet the requirements of SG applications and describes the different applications in SGs that can potentially be done through RPL multiple instances. Finally, my thesis work is part of the SoMel SoConnected project[1] which has short term objectives, making RPL a good candidate in our research.

In the next section, we provide an overview of the RPL standard covering the issues related to this thesis (readers may refer to [15] for the detailed RPL description). Later, we study the main proposed modifications and metrics related to RPL.

2.4 RPL protocol: how does it work?

RPL is a proactif, distributed, distance vector routing protocol based on IPv6 for LLNs. It divides the network into multiple logical Destination Oriented Directed Acyclic Graphs called DODAGs. DODAGs are tree-like structures oriented towards the root/sink of the network built in order to avoid loops.

Rank in RPL

As we can see in Figure 2.2, each node in a DODAG has a rank that defines its relative position with respect to a sink node of the DODAG. In other

[1] http://livetree.fr/

words, it represents a distance from the root. The rank increases by going down the tree from the root.

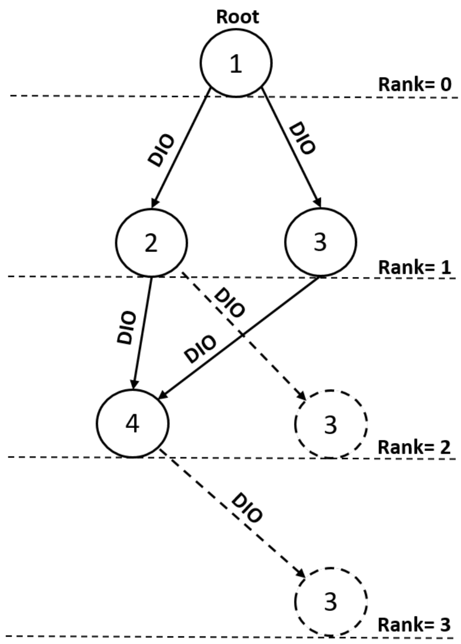


FIGURE 2.2: Rank concept in RPL

RPL instances

RPL can use multiple overlapping DODAGs over the entire network to provide different levels of QoS in the network layer. In this case, each level is called an instance. Here, few points should be mentioned:

- An RPL network contains at least one instance which may be composed of one or more DODAGs.
- A node can join only a single DODAG per instance, but it can participate in multiple instances to carry different types of traffic simultaneously.
- An RPL instance is associated with an objective function in order to optimize the topology based on several metrics/constraints that may be link or node-based such as the shortest path, nodes battery level or the quality of the links.

RPL standard objective functions

Minimum Rank with Hysteresis Objective Function ($MRHOF$) [52] and Objective Function Zero ($OF0$) [53] are the two standardized objective functions in RPL.
$MRHOF$ uses hysteresis while selecting the path with the smallest metric

value. It is designed to find the paths with the smallest path cost while preventing excessive churn in the network. It does so by finding the minimum cost path and switching to that path only if it is shorter (in terms of path cost) than the current path by at least a given threshold. $MRHOF$ uses the Expected Transmission Count (ETX) metric [54] by default.

$OF0$ uses the "step_of_rank" to compute the amount by which to increase the rank along a particular link using static (hop count) or dynamic metrics (ETX). We note that Hop Count (HC) and ETX will be explained and analyzed later on in Section 2.5.2 in order to highlight the gaps of their use for SG applications.

DODAG and upward routes construction

Whatever the used metric, a DODAG construction starts from the root by sending DODAG Information Object (DIO) messages to its neighbors (Figure 2.2). The DIO contains the metric/constraint used by the objective function and the rules to join a DODAG (e.g, DIO sending interval). Nodes will receive and process DIO messages potentially from multiple nodes (i.e. node 4 receiving DIO messages from node 2 and 3 in Figure 2.2) and make a decision to join the DODAG/graph or not according to the objective function and local policies (if existing). Once a node joins a graph, it automatically has a route towards the sink through its parent node. The node then computes its rank within the graph, which indicates its position within the DODAG. If configured to act as a root, it starts advertising the graph information with the new information to its own neighboring nodes. If the node is a leaf node, it simply joins the graph and does not send any DIO message. The neighboring nodes will repeat this process and perform parent selection, route addition and graph information advertisement using DIO messages. At the end of this process, only upward routes (i.e to the root) are built. RPL nodes can also send DODAG Information Solicitation (DIS) messages to solicit DIO messages from neighbors.

Downward routes construction

To establish downward routes, a node must send a Destination Advertisement Object (DAO) to its parent containing prefix information of the nodes in its sub-DODAG, when the DAO message arrives to the root, the prefixes are aggregated and the downward routes are then built and made available to the parents, and so on.

Here, two modes of operations are available: *non storing* and *storing* modes. In *non storing* mode, the node sends unicast DAO messages to the DODAG root which is aware of the whole topology. Routes on the way are not stored. A packet has to travel then the whole tree up to the root in order to be routed downward. In *storing* mode, each node must store routing information to reach all the destinations that are in its sub-DODAG. In this way, and unlike *non storing* mode, a packet may be routed downward through the next common parent node which is (normally) aware of the route to the concerned destination.

RPL trickle timer

RPL message generation is timer-based, it uses the trickle algorithm [55] to control the DIO messages sending rate. The main idea of the trickle timer technique is to optimize the message transmission frequency based on network conditions. This frequency is increased when an inconsistency is detected in order to enable a faster recovery from a potential failure, and decreased in the opposite case.

For example, when a node detects a loop in the network, it resets the trickle timer and send DIOs more often. Otherwise, the interval of the trickle timer increases as the network stabilizes which results in fewer DIO messages being sent in the network.

2.5 RPL related work

Being a general standard, many researches are active around RPL in order to adapt it to different Internet of Things applications. Moreover many critical analysis were made to highlight the gaps concerning reliability and adequate metrics in a SG environment [56, 18, 57].

2.5.1 SG based RPL proposed modifications

In [16], a modification of RPL is proposed to adapt it with AMI by using a new DAG rank computation based on ETX. The outward traffic (from sink to node) is implemented differently from traditional RPL by adding to each meter node a destination list containing (1) the id of the destination node; and (2) the ID of the last hop of the packet. In this way, each node will record all of its descendants in its destination list. The next-hop node ID will indicate the direction a packet has to take to reach the descendant node (in outward traffic from destination node to a source node). This produces less overhead since no DAO messages concerning downwards (outward) routes are sent anymore and that the proposed mechanism is purely based on handling the inward unicast data traffic.

RPL specification provides the means necessary for any node operating ***on the same radio channel as the root node***, to establish and maintain upward and downward routes in a tree.

In [58], the authors address the possibility of using multiple radio channels in RPL for data forwarding. A mesh radio based solution is proposed to enable smart meters to automatically discover concentrator nodes in their vicinity:

– Multiple trees were formed, each rooted at a different concentrator, with each concentrator in the radio neighborhood of each other using a different channel (as per the RPL specifications mentioned above).

– In the beginning, the nodes select random channels.

– After that, the nodes proceed to channel selection using DIO, DIS and OF to get the best channel rank to send data through it.

– Once a loss of connectivity is detected a connectivity detection procedure is launched to select the next best channel to send the data through it. It is also important to note that in order to avoid repeating the whole scanning process in case of loss of connectivity, a probing process is used for the other channels when a smart meter node is in idle periods.

RFC 6551 [59] proposes several routing metrics to be used for path calculation in LLN, i.e the Throughput, Node Energy, Latency, Link reliability with the LQL (Link Quality Level) or ETX metric. In the following sections, we outline the main metrics that were proposed in literature to be used with RPL.

2.5.2 ETX & HC: main metrics in RPL objective functions

ETX in $MRHOF$ [52] and HC in $OF0$ [53] are the two main metrics used in the objective functions. ETX finds paths with the fewest expected number of transmissions (including retransmissions) required to deliver a packet all the way to its destination [54]. Although ETX is reliable and widely used as a metric in WSNs, it does not take directly the latency into account which is critical in some SG applications [19]. ETX is not energy aware, thus for a link with few re-transmissions, ETX will keep sending packets on it without taking the decrease of battery nodes level into account. HC only takes the number of hops into consideration to calculate the best path, which is not always satisfactory in LLN.

2.5.3 Energy aware and load balancing metrics for RPL

An energy-based objective function for RPL that uses the remaining energy as the main routing metric was proposed in [60]. It achieves a better load balancing compared to ETX and increases the network lifetime but with a lower delivery ratio. In [61], two MAC aware routing metrics are proposed to be used in RPL: R-metric and Q-metric. R-metric extends ETX by considering packet losses due to the MAC contention. Q-metric provides load balancing by selecting the lightest parent in terms of traffic load by solving an optimization problem and mainly considering reliability, transmission and reception power consumption. ETT-LB is proposed in [62]. It is based on the Expected Transmission Time (ETT) metric [63], which extends ETX by considering the link transmission rate and packet size, adding to it the Expected Delay Time (EDT), which is the average link load at a node in order to achieve load balancing.

2.5.4 Multipe objective metrics for RPL

The authors in [64] propose NL-OF, an objective function based on a non linear length that constructs DODAGs from roots to nodes such that the non linear length is the smallest possible. They evaluate it using Cooja [65] simulator while considering three QoS parameters: End-to-end delay, packet loss and jitter. In [66], L^2AM metric is proposed. It is based on an combination of both data reliability (defined by ETX) and the nodes residual energy. Although their solution extends the network lifetime, it remains not adapted to a network with heterogeneous nodes and applications since it considers only one type of traffic and their model doesn't take into account powered nodes.

Metric combination is considered in several works for RPL. In [67] two combinations of two metrics are proposed: lexical and additive. In the lexical combination, the second metric is inspected if and only if the first one leads to equal paths, while in the additive combination the paths are calculated based on a different cost given to each metric. Fuzzy logic metric combination is proposed in [68, 69, 70] in order to be used for RPL. They combine several metrics like end-to-end delay, HC, link quality and battery level.

2.5.5 Multiple instances

Multiple instances in RPL and QoS were studied in many works. In [71], two distinct objective functions for traffic differentiation are implemented using ETX for critical traffic and OF0 for the regular one. They compare their implementation to a single instance scenario using ETX metric. Their simulation results show that multiple RPL instances provide a better performance in PDR and latency. Others also consider ETX and HC for the different instances [17]. In [72] two types of nodes are proposed; T1 as regular nodes for regular traffic and T2 as alarm nodes for critical traffic. The number of T1 nodes is much higher than T2's. A T2 node can choose a T1 node as its parent, and not vice versa. Regarding the DODAG structure, there are two RPL instances (RPLInstanceT1 and RPLInstanceT2). They added support for priority traffic in the MAC layer by using queuing models. Their simulation results show an improvement in Packet Reception Ratio (PRR) and latency compared to the model without priority queuing.

However, the proposed multiple instances approaches are basic, limiting the number of instances to two. Moreover, they consider ETX and HC as metrics. They do not take the drawbacks of such metrics into consideration, such as energy efficiency for ETX and the bad route selection of HC.

2.5.6 Observations

After browsing the main RPL related work we have realized that the metrics proposed for RPL could be suitable for some applications. But in a SG environment with many applications, each with different QoS requirements, it becomes quite impossible for a single existing metric to cover all of those different criteria. Same for the proposed metric combinations, they remain limited to one or two traffic types and their extensibility is either complex or impossible.

For that, a multiobjective solution is then essential to route the traffic in a SG environment. Taking into consideration the heterogeneous QoS requirements of the SG applications.

2.6 Summary

In this chapter, we have first described the different aspects of the SGs and the main routing protocols used in SG environments. After that, we have presented an overview of the RPL protocol which is the protocol of interest in this thesis used for communication and data forwarding in a WSN controlling a SG. The main works and modifications proposed for RPL are presented and evaluated. These works are not fully suitable for a SG environment. The metrics used do not fit with the SGs QoS requirements. Beside that, multiple instance implementations of the literature are limited. As a conclusion, a single routing metric cannot assure traffic differentiation in a SG since different applications require different QoS levels.

In addition, in a multiple instance environment, the chosen objective function/metric has to guarantee the QoS requirements of the concerned SG application, which to the best of our knowledge has not been proposed yet.

In the next chapter, we present our proposed multiobjective solution for SG heterogeneous traffic and evaluate it using simulation and real test-bed experiments.

Chapter 3

Quality of Service Routing in Smart Grids

In Chapter 2, we made an overview of the Smart Grid communication network architecture and the main SG applications with their requirements. After that, we explained the RPL protocol alongside with the main work around it in literature. In this chapter, we detail our contribution regarding the QoS routing in Smart Grids. We provide details concerning our simulations and experiments and discuss the obtained results.

3.1 Overview

In this section, we briefly recall the SG traffic heterogeneity characteristics and the benefits of multiple instances with RPL.

3.1.1 Smart Grid heterogeneous traffic

As we already stated, SG applications are heterogeneous in terms of criticality levels. Some tolerate delays like regular meter reading in AMI applications, others require real-time intervention and action like DA applications. These applications can be classified according to their criticality levels. RFC8036 [19] proposes five different priority classes for the traffic in SG AMI. Other work [21] classify the traffic into two levels: critical and periodic. Based on that, and since the traffic classes in the SG are not standardized, a single solution to route the traffic with different QoS may not be sufficient. The number of traffic classes vary depending on the application and the implementation and can still evolve in the future.

A multi-objective solution is then essential to meet the QoS requirements of SG applications. This is the purpose of the approach proposed in this chapter.

3.1.2 RPL multiple instances

As we saw, RPL assures QoS at the network layer in WSNs through the logical subdivision of the network in multiple instances, each one relying on a specific objective function. However, as already mentioned in the previous chapter, RPL is not optimized for SGs since its main standardized objective functions and their associated metrics do not allow QoS differentiation. Our approach takes several metrics into consideration through a multi-objective solution that adapts to the QoS requirements of various SG applications.

It is important to mention that our approach is not specific to SGs but it is mostly suitable to any context with different applications on the same physical topology with different characteristics/QoS expectations. SGs are only an example of such applications.

3.2 Proposed Solution

In this section, we present in details our objective function proposition alongside with examples. After that, we evaluate it with simulations and real test-bed experiments and discuss the results.

3.2.1 *OFQS* objective function

To overcome the drawbacks of the metrics traditionally used by RPL (ETX & HC) discussed in Section 2.5.2 and exploit the multi-instances, we introduce the tunable/parameterized multi-objective metric $mOFQS$ to be used by the objective function $OFQS$. The $mOFQS$ metric adapts to the number of instances in the network depending on their criticality level by tuning its parameters jointly. $OFQS$ is derived from $MRHOF$ as it relies on the same rank calculation mechanism, it adopts hysteresis to prevent routing instabilities by reducing parent switches under a certain threshold.

3.2.2 QoS factors in *OFQS*

$OFQS$ with its metric $mOFQS$ takes the quality of the links into consideration by calculating their ETX value. In Contiki Operating System, ETX is implemented in the $MRHOF$ objective function. ETX is updated based on callbacks from the MAC layer which gives the information whether a MAC layer transmission succeeded, and how many attempts were required. Lower ETX values mean better links quality to route the packets with less re-transmissions. Alongside with the quality of links, the delay is an important factor in SG applications as already mentioned. For that, $mOFQS$ considers the delay d between sending the packet and receiving it in the network layer between two adjacent nodes. This allows the algorithm to choose faster links especially for critical applications considering at once transmission, queuing and interference delays.

Moreover, in a SG, electricity and energy do exist, but connecting sensors to such high voltage with intermittent and ill-adapted energy levels is sometimes inappropriate or physically impossible. For that, battery-powered sensors must be deployed all over the grid alongside with the mains powered ones. Different requirements for different applications may tolerate in some cases passing by a longer route in order to preserve the remaining energy in the nodes. Hence, considering the battery level for the nodes in our metric will be beneficial in terms of traffic load balancing and network lifetime.

In order to do so, we classify the remaining energy in the nodes into three Power States (PS) [73]:

- PS=3: Full battery state (ranging between 100% and 80%) or main powered
- PS=2: Normal battery state (ranging between 80% and 30%)
- PS=1: Critical battery state (less then 30%)

By using this classification, weak nodes become unfavorable in the route selection by penalizing the ones with a smaller PS. We note that these thresholds could be adjusted for other applications depending on the network characteristics.

3.2.3 *mOFQS* metric

In order to enable RPL to consider the remaining energy of nodes, the latency and the multiple instances beside the reliability using ETX at once, $mOFQS$ includes the Power State PS, the delay d of delivering a packet within two nodes in milliseconds and two parameters α and β. $mOFQS$ formula is shown below:

$$mOFQS = \frac{\alpha(ETX \times d)}{PS^{\beta}}$$

where α and β are two tunable parameters with $\alpha = 1 - \beta$, $0 < \alpha < 1$ and $0 < \beta < 1$. $mOFQS$ is an additive metric whose values over the path is the sum of the values at each hop. It is important to note that α and β are real numbers, we can have an infinity of values, thus our metric tolerate nearly infinite different traffic classes if tuned adequately.

The idea is to multiply ETX by the delay d for every hop to get the links reliability while considering the delay of the packet delivery. We multiply then the factor $ETX \times d$ by α to foster link quality and end-to-end delay for critical applications by increasing α. d values are normalized in order to limit quick variations on the links. $\alpha(ETX \times d)$ is then divided by PS to the power of β. Increasing or decreasing β will similarly foster PS. If the application is critical, β should be decreased (resp. α increased).
For delay tolerant applications, increasing β will result in a longer route

while conserving the nodes power since the metric will weight more node energy level rather than link quality or end-to-end delay.

Figure 3.1 shows how $mOFQS$ behaves as a function of α for the different PS values (with ETX=1 and d=1). We can see that the higher α values and the more critical energy level (the worst the conditions), the higher the $mOFQS$ value to be considered i.e, higher PS with higher α will lead to a higher $mOFQS$ value which will disadvantage the node/link in question.

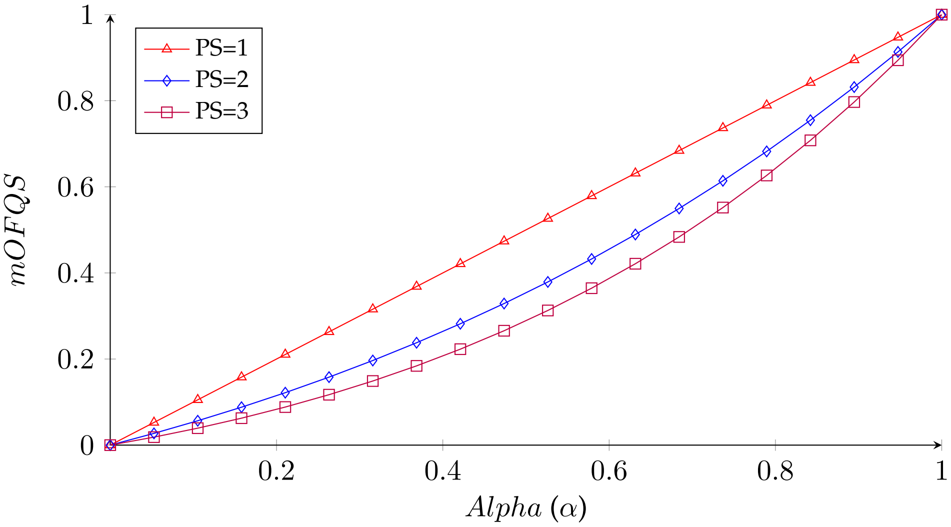


FIGURE 3.1: $mOFQS$ variation with α. ETX=1, d=1

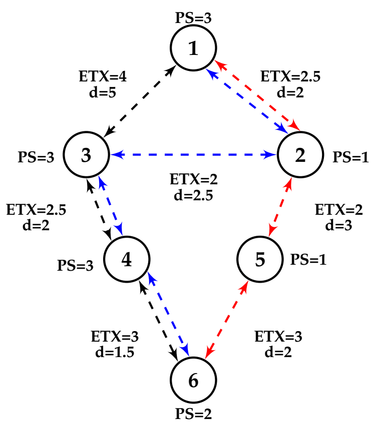


FIGURE 3.2: Network with different ETX, delay d (in ms) and PS values

Each node chooses the path upward in its DODAG with the lowest value provided by $mOFQS$ with α/β selected for the given application. The lowest value of $mOFQS$ defines the best/favorite metric value. First of all, varying α and β allow us to differentiate between instances depending on their criticality level. Less critical applications will tolerate the use of less good links. Dividing $\alpha(ETX \times d)$ by PS^{β} aims to foster routes where the nodes consumed less their batteries or are main powered. For one application, we favor α or β against the other, and since $\alpha + \beta = 1$, when one parameter increases the other decreases and vice-versa.

Figure 3.2 depicts an example of a small network of 6 nodes running RPL, considering two different applications. The first one is critical and belongs to Instance 1 and the second one is regular and belongs to Instance 2. When node 6 needs to send a packet to node 1, we consider the following paths: path 1: $6 \rightarrow 5 \rightarrow 2 \rightarrow 1$ or path 2: $6 \rightarrow 4 \rightarrow 3 \rightarrow 1$ or path 3: $6 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$.

Table 3.1 shows the different paths metric values with ETX, HC and $mOFQS$. We can thus note that each path features different QoS and can be favored by using a metric rather than another one. This is how we will achieve the multi-instance routing and QoS differentiation.

For ETX alone, path 1 is the optimal one since it is the only metric used.

	Paths		
Metrics	**Path 1** 6->5->2->1	**Path 2** 6->4->3->1	**Path 3** 6->4->3->2->1
Instance 1 ETX	7.5	9.5	10
Instance 2	-	-	-
Instance 1 ETX	7.5	9.5	10
Instance 2 HC	3	3	4
Instance 1 $mOFQS$ α=0.9 β=0.1	14.9	23.9	16.3
Instance 2 $mOFQS$ α=0.1 β=0.9	1.4	1.2	1.1

TABLE 3.1: Paths values for the different metrics used

For ETX & HC, ETX is used for the critical traffic (Instance 1) and HC for the regular one (Instance 2). As we can see Instance 2 optimal path will be 1 or 2 since they count less hops, and for Instance 1, it will be path 1 which has ETX=7.5. Neither ETX or HC take energy consumption and delay into consideration, unlike $mOFQS$ where α and β values will foster one path over the other. With $mOFQS$, in Instance 1 with critical traffic which requires minimal latency, we have to route the packets as fast as possible while guarantying a reliable link. Thus, we increment α (α=0.9) fostering $ETX \times d$ (reliability and latency), which means decreasing β (β=0.1). $mOFQS$ fosters path 1 since it has better ETX and d values than paths 2 and 3. In Instance 2, where the traffic is not critical, we increment β (β=0.9) and foster PS (in $mOFQS$), which means that we might pass by a longer and less reliable route, while guaranteeing load balancing. Consequently forcing paths where nodes consumed less their batteries (path 3 where node 3 and 4 have more than 80% energy left in their batteries unlike path 1 where nodes 2 and 5 have less than 30% energy left). We achieve then a traffic distribution along the nodes by passing by path 3 and extending the network's lifetime by advantaging the nodes with higher battery level. Unlike instance 2 with HC where path 2 or 3 are favorites (3 hops instead of 4).

3.2.4 Instances classification

Traffic classes in SG are not yet standardized. In this paper, we use the classification presented in [12] for the requirements in terms of delay and reliability in a NAN as shown on Table 2.1. The aforementioned classification sorts the traffic into 9 different classes, ranging from delays inferior than 3 seconds with reliability >99.5% for the most critical class to delays of hours/days with a reliability of >98% for the least critical class. In our model, we have gathered these 9 classes into 3 classes with 3 main instances:

- Instance 1: critical traffic with an authorized delay ranging between 1 and 30 seconds and a reliability of >99.5% packets received with α=0.9 and β=0.1
- Instance 2: non-critical traffic with an authorized delay of days and a reliability of >98% packets received with α=0.1 and β=0.9
- Instance 3: periodic traffic with an authorized delay ranging between 5 minutes and 4 hours and a reliability of >98% packets received with α=0.3 and β=0.7

In this classification, we increment α for the critical traffic thus fostering the link quality and end to end delay assured by ETX and d, which results in routing the packets in a reliable and faster path. For less critical traffic we increment β which leads to fostering paths where the nodes consumed less their batteries and then achieving a better load balancing.

We note that our model is not limited to this classification or the current tuning of α and β. For any other implementation this classification can be adjusted and α and β can ***be modified*** or ***be totally independent of each other***.

3.3 Evaluation

In this section, we first present our performance metrics. We detail then the simulation environment and setup used to pre-validate our approach in a controlled environment. After that, we provide an overview about the wireless sensor test-bed used to validate our proposition. Performance evaluation using several metrics alongside with results analysis are provided for both the simulation and experiment, with a comprehensive comparison at the end.

3.3.1 Performance metrics

In order to evaluate our approach we compare $OFQS$ with $MRHOF/OF0$ using the following performance metrics: End-to-end delay, network lifetime, load balancing and packet delivery ratio. Evaluating these metrics will allow a better understanding of the targeted factors in our proposition.

- End-to-End delay: This metric represents the sum of link latency, which is defined in [59] as an aggregated additive metric. This metric should be minimized for real-time applications. We compute it at the time taken for a packet to be transmitted across the network from source to destination.
- Network lifetime: It represents the amount of time that a network (a WSN in our case) would be fully operative. In our evaluation we compute it as the time at which a fixed percentage of nodes run out of energy to send a packet.
- Load balancing: It aims to improve the distribution of workloads across the multiple entities of the network. We compute it by calculating the percentage of remaining energy in the battery nodes after a certain time in order to evaluate the remaining energy distribution among the nodes.
- Packet delivery ratio: It represents the ratio of packets that are successfully delivered to the destination compared to the number of packets that have been sent out by the sender.

3.3.2 Simulation

3.3.2.1 COOJA simulator

COOJA simulator [65] supports cross-level simulation. It allows simultaneous simulation at the network level, the operating system level, and the machine code instruction set level. COOJA combines low-level simulation of sensor node hardware and simulation of high-level behavior in a single simulation. All levels of the system can be changed or replaced: sensor node platforms, operating system software, radio transceivers, and radio transmission models. Figure 3.3 shows a snapshot of the simulation window. Cooja's emulator MSPsim [74] is a Java-based instruction level emulator that provides accurate emulation at both cycle-level for the MSP430 micro-controller and bit-level for the CC2420 radio transceiver. This allows accurate energy estimation.

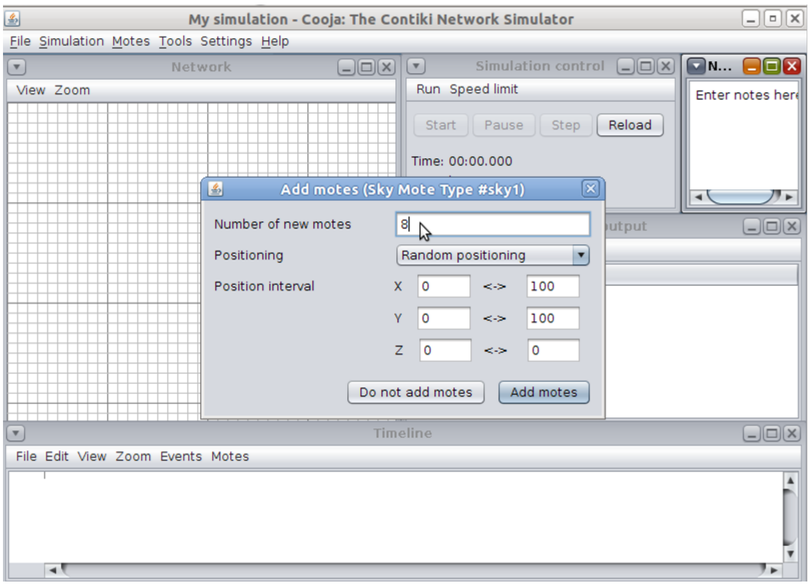


FIGURE 3.3: COOJA simulation window

3.3.2.2 Simulation setup

In order to evaluate our approach, we perform simulations on Contiki OS [51] using COOJA. Simulation parameters are detailed in Table 3.2. Zolertia Z1 motes are emulated. We use the Energest module in Contiki to estimate the battery levels by extracting the values for the energy consumption from the Z1 datasheet[1]. Z1 motes are randomly given two different battery levels at bootstrap (10 nodes have 60% of energy of the others) in order to highlight the energy consumption. The topology consists of 35 client nodes randomly positioned that send UDP packets to the server placed in the middle, randomly every 3 to 4 minutes.

Here, since nodes are emulated, it was impractical to expand the network or send packets more frequently as per SGs applications requirements. Doing so with emulated nodes would result in simulations ran slower than real time due to COOJA's limitations.

We consider a 100% transmission/reception ratio. We are aware that this is not a so realistic setting but this allows a fair comparison of ETX, HC and our metric under the best case scenario, which is from our perspective, what we aim to as a first step in this simulation evaluation.

[1]http://zolertia.sourceforge.net

Parameters	**Values**
OS	Contiki master version
Simulator	Cooja; Radio Model: Unit Disk Graph Medium
Communication protocols	CSMA, RDC contikimac, IEEE 802.15.4, channel 26, ContikiRPL, IPv6
OF	(1) OFQS with 2 instances (2) MRHOF(ETX) & OF0(HC)
Number of nodes	35 clients and 1 server
Deployment area	200m x 200m
Transmission/ Interference range	50m
Transmission/ Reception ratio	100%
Sensors	Zolertia Z1
Maximum packet size	30kb
Sending interval	1 packet every 3 to 4 minutes

TABLE 3.2: Parameters of the simulation

As multiple RPL instances are not fully implemented in Contiki, we use an implementation[2] on COOJA where multiple instances are supported and adapted it to our scenario. Only upward traffic is considered. $OFQS$ with two instances: critical and non critical (Instance 1 and Instance 2 resp.) was compared to RPL with two instances : $MRHOF$ with ETX metric for critical traffic and $OF0$ with HC metric for less critical traffic. Tests are repeated 10 times. Simulation stops when 20% of the nodes have consumed all their energy. All simulation results are measured with a 95% of confidence interval.

3.3.2.3 Performance evaluation: Simulation

In this section, we evaluate our proposition $OFQS$ on COOJA in comparison with $MRHOF/OF0$ in terms of the previously explained four performance metrics in Section 3.3.1: End-to-end delay, network lifetime, load balancing and packet delivery ratio.

Network lifetime and load balancing Figure 3.4 shows the percentage of nodes that did not exhaust their batteries during the simulation. On the one hand, we can see that for $MRHOF/OF0$ starting hour 21 battery nodes

[2]https://github.com/jeremydub/contiki

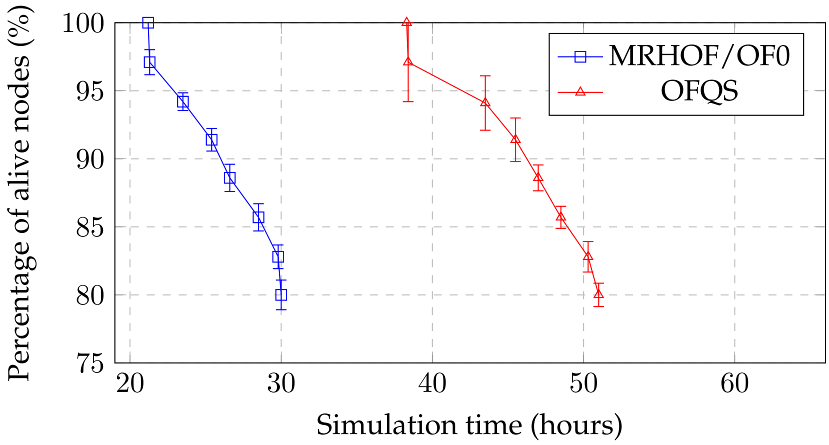


FIGURE 3.4: Network lifetime variation

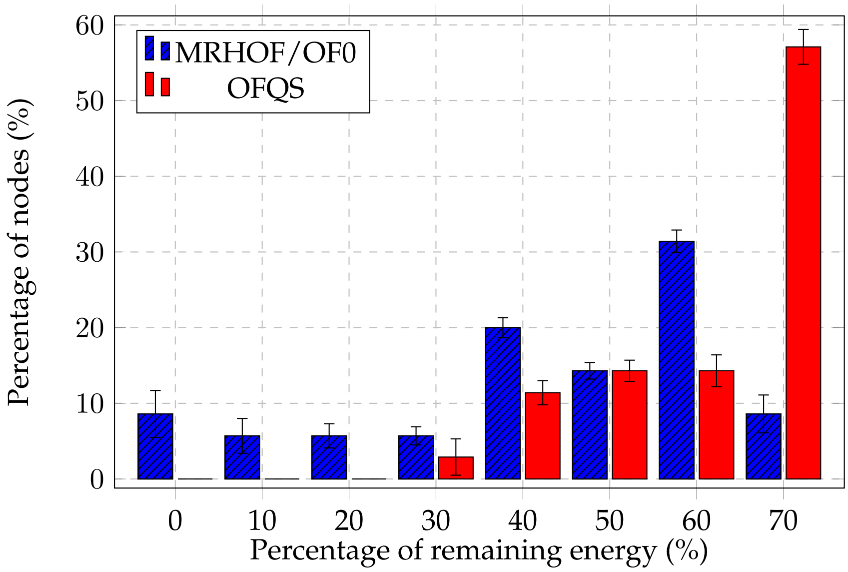


FIGURE 3.5: Remaining energy distribution among the nodes after 24 hours

start to drain and within 9 hours (after 30 hours), 20% of the nodes exhaust totally their batteries. The simulation will then stop as previously defined. On the other hand, For $OFQS$, and for the first 38 hours all nodes are still functional and none of them has consumed its total energy. Starting 39^{th} hour, the nodes batteries start to drain and the network stops after 51 hours. $OFQS$ presents a gain of 21 hours, which means a 59% improvement on the network lifetime compared to $MRHOF/OF0$. This is due to the PS factor. Indeed, after a certain period of time and when the nodes start to consume their batteries, the PS fosters the switch to other routes with better battery nodes.

In the same way, we can see in Figure 3.5 that after 24 hours of operation, 57.1% of the nodes have still 70% of their energy with $OFQS$ compared to 8.6% with $MRHOF/OF0$. Besides that, 5.7% have 20% and 10%

left energy with $MRHOF/OF0$ compared to 0% of the nodes with $OFQS$. Finally after 24 hours 8.6% of the nodes consumed their total energy with $MRHOF/OF0$ compared to 0% with $OFQS$. This is mainly due also to the PS factor which makes the choice of the path switch to nodes that consumed less their batteries achieving then a better load balancing of traffic among the nodes.

End-to-End delay Figure 3.6 shows the variation of End-to-End delay (in ms) for both $OFQS$ and $MRHOF/OF0$ within simulation time.

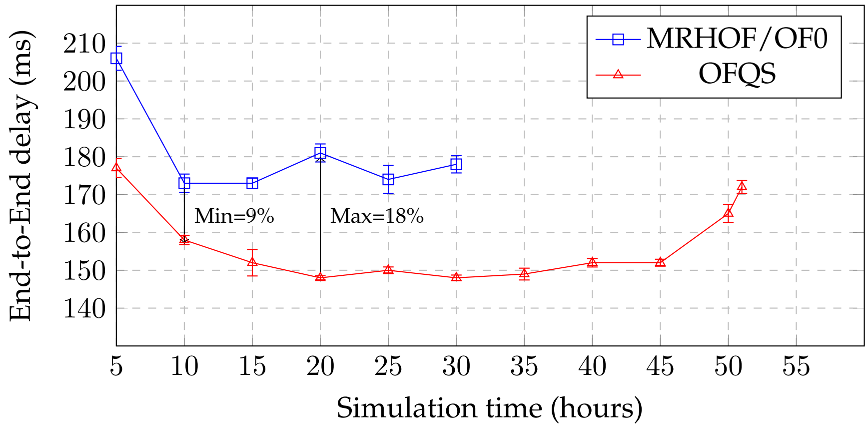


FIGURE 3.6: End-to-End delay variation with time

We can see that End-to-End delay with $OFQS$ is always below *MRHOF/OF0*. Even though that HC favors paths with fewer hops, these paths are generally longer with potential poorer connectivity. On the other hand, ETX is not also aware of the delays due to interference on the links and queuing in the nodes as long as the packets are transmitted; therefore, sending a packet with less retransmissions does not mean sending it on a faster link. $OFQS$ chooses faster routes due to the d factor in $mOFQS$ that takes the delay between two hops into consideration, which will foster faster links with less interference and congestion that ETX and HC are not aware of.

Beside that, we can see that the End-to-End delay decreases up to the first 20 hours. This is due to the fact that the battery nodes were still full and such none of them is in a critical state. Here, the d factor favors faster routes.

After that, between 20 and 45 hours, the delay is mostly stable which is due to the variation of the battery levels which is affecting choosing faster routes. Finally after 45 hours and up to the end of the simulation, we can see an increase in the End-to-End delay which is mainly due to the depletion of certain battery nodes which leads to choosing longer routes to maximize the network lifetime. $OFQS$ average End-to-End delay during the simulation makes an improvement between 9% and 18% compared to $MRHOF/OF0$, and stays within the time requirements limits previously defined in section 3.2.4.

Packet delivery ratio Figure 3.7 shows the PDR percentage for the different metrics used. We can see that for HC, the PDR is less than 60%. This is due to the route selection in this metric that only relies on the number of hops from the sink without any reliability mechanism. For ETX the PDR is 93.2% compared to 99.4% for $mOFQS$ with Instance 1 (α=0.9 & β=0.1) and 96.6% for Instance 3 (α=0.1 & β=0.9), which shows that $OFQS$ overpasses both ETX and HC in terms of reliability.

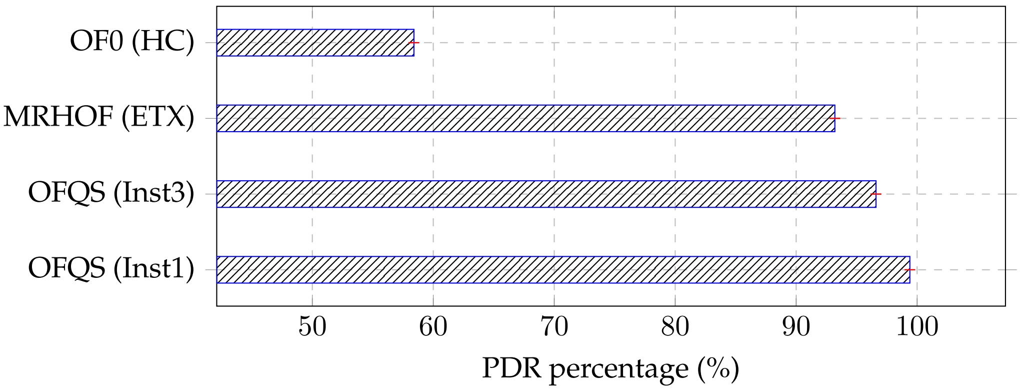


FIGURE 3.7: Packet delivery ratio percentage for the different OFs/metrics

$mOFQS$ considers besides ETX, the delay of sending a packet in one hop which reflects the interference delay on that hop, allowing more reliable routes to be chosen by multiplying those two factors (ETX & d). Finally, we note that the achieved PDR percentages with $OFQS$ are relatively close to the SG requirements that were defined in Section 3.2.4.

3.3.3 Experiment

3.3.3.1 FIT IoT-LAB test-bed

FIT IoT-LAB [75, 76] provides a large scale infrastructure facility and experimental platform suitable for testing small wireless sensor devices and heterogeneous communicating objects. It provides full control of network nodes and direct access to the gateways to which nodes are connected, allowing researchers to monitor several network-related metrics. FIT IoT-LAB features over 2000 wireless sensor nodes spread across six different sites in France. For our experiment, we chose nodes from the site of Lille[3]. These nodes are distributed inside a 200m^2 room and on the different corridors of the Inria building as seen in Figure 3.8, enabling a large-scale multihop topology.

[3]https://www.iot-lab.info/lille-new-physical-topology-released/

FIGURE 3.8: Topology of the deployment on FIT IoT-LAB Lille's site

3.3.3.2 Battery level measurement

Each node from the FIT IoT-LAB platform is composed of three parts as shown in Figure 3.9:

- The gateway that is responsible for flashing the open node and connecting it to the test-bed's infrastructure
- The open node that runs the experiment firmware
- The control node that runs radio sniffing and consumption measurement

Because we need to run scenarios with varying and restrained battery levels on different nodes, it is impractical to rely on actual lithium batteries. Instead, we rely on the real-time consumption measurement performed by the control node. The gateway collects consumption measurements every 140 μs, and write Orbit Measurement Framework (OML) files, with a μs time stamped value of the power consumption of the open node in Watts.

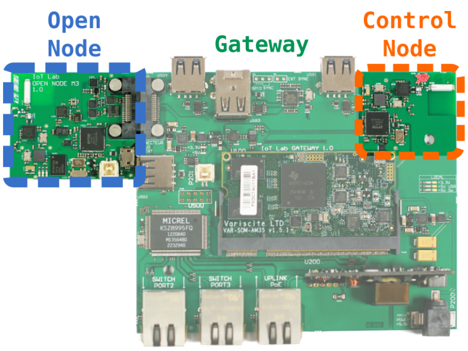


FIGURE 3.9: Hardware of an IoT-LAB node [75]

A software running inside the test-bed's user area collects then these consumption files for each node in the experiments, and numerically integrates the values through a basic rectangle sum. At the beginning of each experiment, the battery capacity of each node is decided randomly between two different values. During the experiment, when a node's consumed virtual battery exceeds the virtual battery capacity, the node is electrically shutdown by the gateway. The network must then reorganize without the missing peer. The experiment stops when at least 20% of the nodes run out of battery.

The integrated total consumed energy in Joules, as well as the battery percentage, are sent to each node through its serial port using the gateway's tooling that replicates the open node serial port on an accessible TCP socket. A Contiki process receives this information on the node, which will be used afterwards in the metric computation and route calculation. For real-life application of this work in an actual sensor network, devices would be fitted with an adequate interface to their battery controller subsystem, which would be queried by the Contiki's application through an I2C, SPI or similar link.

We note that the physical environment conditions that may influence the discharge and lifetime of the batteries [77, 78] are out of scope of this work.

3.3.3.3 Network setup

In order to evaluate our approach on FIT IoT-LAB, the experiment is performed on Contiki OS using M3 nodes. The topology consists of 67 client nodes that send UDP packets to the server repeatedly on an interval of 1 to 60 seconds between two subsequent transmissions in order to differentiate the sending rate between the two instances. Experiment parameters are presented in Table 3.3. Since multiple RPL instances are not fully supported in the embedded RPL implementation on Contiki, we use an implementation [2] [79] where multiple instances are supported. We implemented it on FIT-IoT lab in order to evaluate our proposition. In this new RPL implementation, nodes can participate in multiple instances with different objective functions and metrics. A specific instance can be set at application layer, allowing traffic differentiation. It also supports new constraints in DIO metric container object. Also, a root can now be a sink for multiple applications that have different route requirements.

For our experiments, we consider the upward traffic with two instances: $OFQS$ with critical and periodic traffic (Instance 1 and Instance 3 resp.) as presented in Section 3.2.4 compared to RPL with $MRHOF/ETX$ for critical traffic and $OF0/HC$ for periodic traffic. All experiments results are measured within a 90% of confidence interval.

Parameters	Values
OS	Contiki master version
Test-bed	FIT IOT-LAB
Communication protocols	CSMA, RDC contikimac, IEEE 802.15.4, ContikiRPL, IPv6
OF	(1) OFQS with 2 instances (2) MRHOF(ETX) & OF0(HC)
Number of nodes	67 clients and 1 server
Sensors	M3
Microcontroller Unit	ARM Cortex M3, 32-bits, 72 Mhz, 64kB RAM
Maximum packet size	30kb
Sending interval	1 packet every 1 to 60 seconds

TABLE 3.3: Parameters of the experiment

3.3.3.4 Performance evaluation: experiment

In this section, we evaluate our proposition $OFQS$ on the FIT IoT-LAB test-bed in comparison with $MRHOF/OF0$ in terms of the four performance metrics previously presented in section 3.3.1: End-to-end delay, network lifetime, load balancing and packet delivery ratio.

End-to-End delay Delay is considered when selecting the best next hop according to $mOFQS$. In order to evaluate the End-to-End delay, we calculate the difference in time between sending a packet by the client and the reception by the server. We actually ran several tests in order to check the synchronization of the clock, and we realized that clock drift is negligible.

Figure 3.10 shows the end-to-end delay variation throughout the experience time for both $MRHOF/OF0$ and $OFQS$. We can see that $OFQS$ end-to-end delay is always below $MRHOF/OF0$ with an improvement ranging from 6 to 10%. Even though HC chooses paths with the fewer hops from the sink, these paths are generally slower with a higher potential of loss since HC is not aware of links congestion and saturation. On the other hand, ETX is not also aware of the delays due to interference on the links and queuing in the nodes as long as the packets are transmitted; therefore, sending a packet with less re-transmissions does not necessarily mean sending it on a faster link. In $OFQS$, the d factor takes into account the delay of sending a packet between two adjacent nodes in the metric computation. In this way and mainly in instance 1, the metric will foster faster routes with less interference and congestion that HC and ETX are

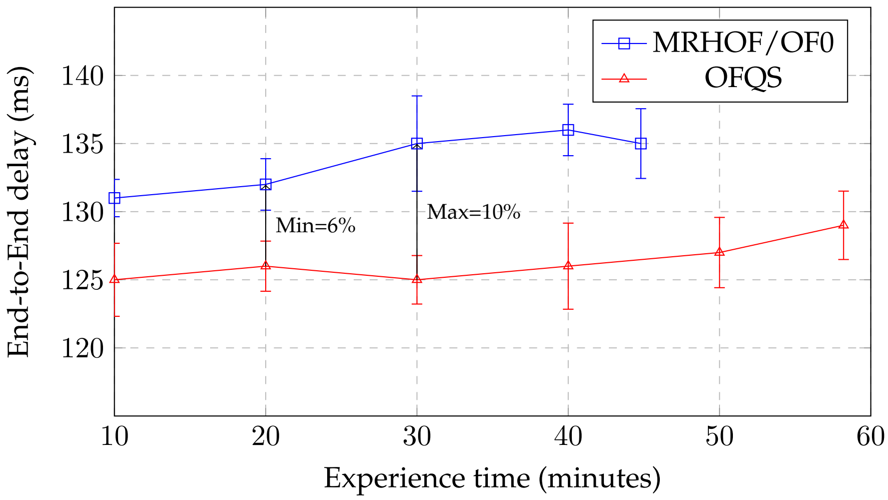


FIGURE 3.10: End-to-End delay variation with time

not aware of. Moreover, we can see that the delay variations for $OFQS$ are minimal between 20 and 40 minutes. This is due to the variation of the battery levels (PS passing to a smaller value) which affects the choice of routes with low delays. Finally, and starting from the 40th minute until the end of the experiment, we can notice that the end-to-end delay starts to increase. This is due to the depletion of the batteries of some nodes that switch to a lower PS, which means that the metric will switch from these nodes to other ones and foster sometimes longer routes in order to increase the network lifetime. We note that the experience stops after 44 minutes for $MRHOF/OF0$ compared to 58 minutes for $OFQS$ as we can see on the graph. This extension of the network lifetime will be discussed in detail in the next section.

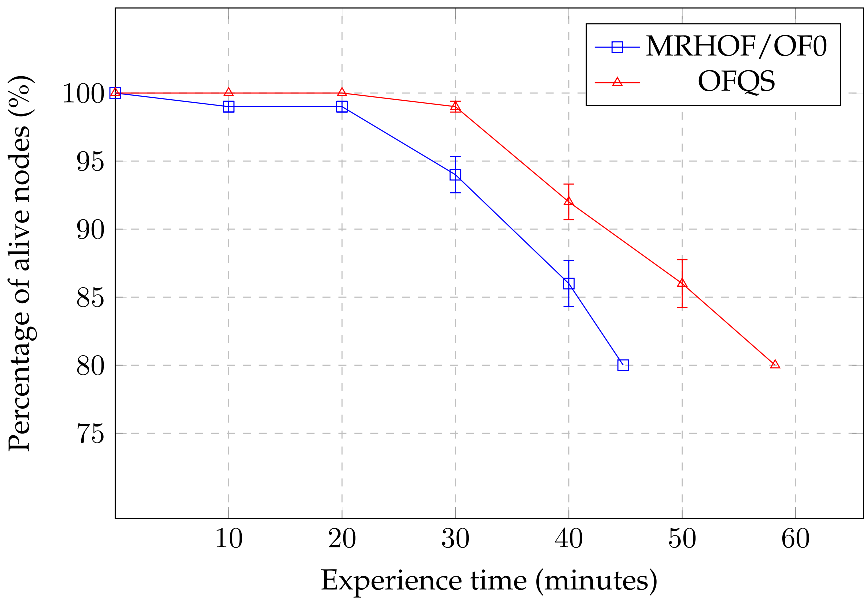


FIGURE 3.11: Network lifetime variation

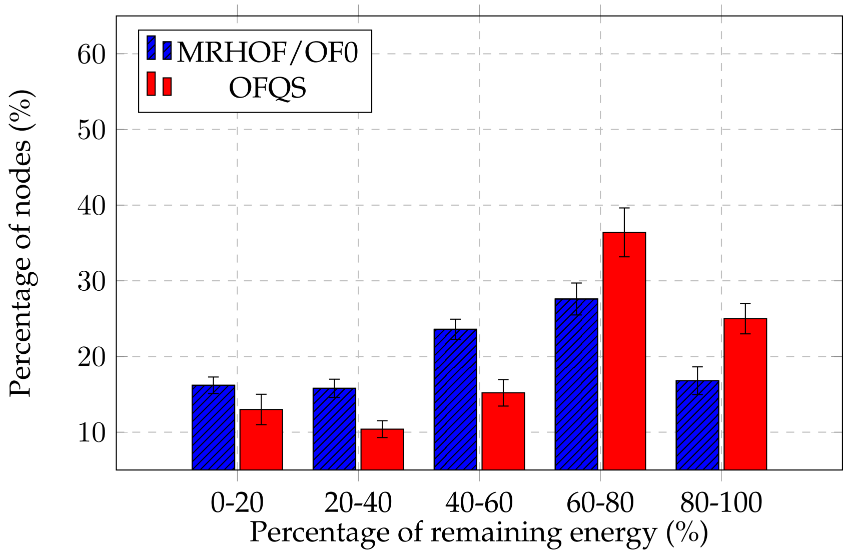


FIGURE 3.12: Remaining energy distribution among the nodes after 30 minutes

Network lifetime and load balancing Figure 3.11 shows the percentage of alive nodes for both $MRHOF/OF0$ and $OFQS$ within the experience time. We observe that for $MRHOF/OF0$ and after 10 minutes, battery nodes start to drain reaching the threshold of 20% after 44 minutes. Concerning $OFQS$ and for the first 20 minutes, all the nodes are still functional and none has consumed its total battery. After that time, the batteries start to drain reaching 20% of dead nodes after 58 minutes. $OFQS$ achieves a gain of 14 minutes of network lifetime increase which is around 25% more than the one achieved by $MRHOF/OF0$. This gain is due to the power state that is taken into consideration in $OFQS$.

In the same way, we can see in Figure 3.12 that after 30 minutes of the experiment, 16,2% of the nodes have a battery level between 0 and 20% in $MRHOF/OF0$ compared to 13% for $OFQS$. While 61,4% of the nodes in $OFQS$ have a a battery level between 60 and 100% compared to 44,4% in $MRHOF/OF0$. This shows that in $OFQS$, PS is switching to nodes that consumed less their batteries achieving then a better load balancing of traffic among the nodes.

In fact, $mOFQS$ does not take into consideration the rate of battery depletion from the beginning. In the initial state, where all batteries are fully charged, the metric will pick paths without battery level consideration since they are all fully charged. During the experience, the most loaded nodes will undergo a quicker battery drain than others and lead to power state changing (PS=3 -> PS=2). Here $mOFQS$ will react and switch to other nodes that consumed less their batteries achieving thus an extension of the network lifetime and a better load balancing.

Packet delivery ratio $OFQS$ achieves 91,8% of PDR compared to 85,7% for $MRHOF/OF0$. This shows that $OFQS$ overpasses $MRHOF/OF0$ in terms of reliability. Firstly, HC has no link reliability mechanisms in the route selection which causes packet loss by selecting congested paths. Moreover, although ETX considers the link reliability, $mOFQS$ still overpasses it by considering the delay of sending a packet in one hop which reflects the interference and the queuing delay on that hop by multiplying $ETX \times d$, allowing then more reliable routes to be chosen.

3.4 Evaluation: Simulation vs Experiment

Figure 3.13 shows a brief comparison between the simulation on COOJA and experiment on FIT IoT-LAB of the gain achieved by $OFQS$ over *MRHOF/OF0*. We note that for the simulation, Instances 1 and 2 are used and for the experiment Instances 1 and 3. Although the parameters are not the same, this comparison is still beneficial, since it clears up and allow us to discuss the difference in results between a simulation and an experiment for 2 different types of traffic in RPL. We can see that for all the evaluated metrics (maximum gain of end-to-end delay, network lifetime and PDR) the gain (of $OFQS$ over $MRHOF/OF0$) in the simulation is higher than the one in the experiment. This is mainly due to the fully controlled environment in the simulation and best case scenario parameters that we chose (100% transmission/reception ratio). In the experiment scenario, real sensor nodes distributed in a building and between the offices are used where interference with other signals i.e. WiFi is more likely. $OFQS$ delay parameter d may undergo abrupt variations on some links which may affect negatively the global efficiency of the metric (mainly the End-to-End delay and PDR). Normalizing the d parameter values is used to limit such effects.

Moreover, concerning the gain in the network lifetime, real-time measurement of actual battery consumption is conducted in the experiment. Here, the peaks of sending/receiving a packet can be slightly shadowed by the idle consumption of the sensor. Per example, sending 100 packets within 30 minutes will result in 100 peaks of 0.19 Milliwatts. These peaks can be easily shadowed if the idle consumption and message control sending of the sensor is 0.10 Milliwatts. The sending/receiving peaks will have more value in an experiment of sending heterogeneous (i.e. in size) packets more frequently between the nodes, which is the case of SG applications. We note that even though the low power mode, sending and receiving of a packet are considered in the simulation with Energest, it remains an estimation based on a software implemented on the OS alongside with theoretical consumption estimations from the sensor's datasheet (Zolertia Z1 in our case).

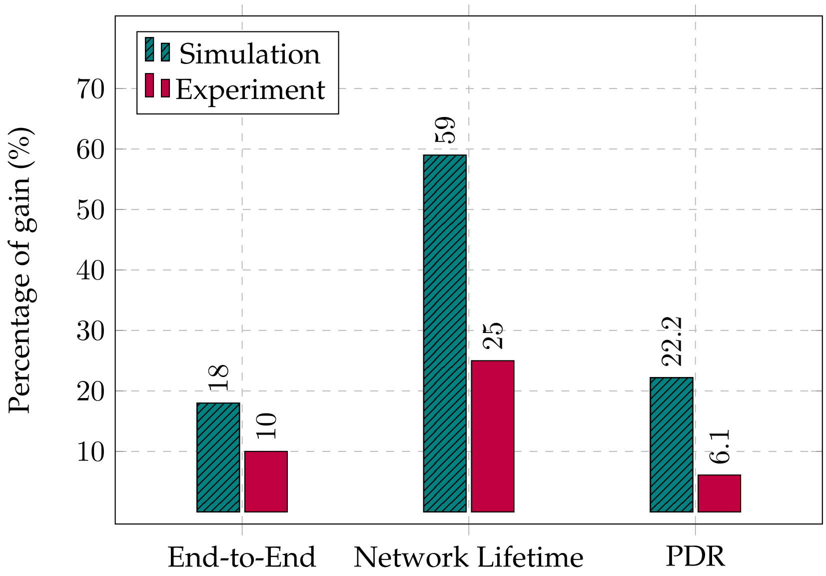


FIGURE 3.13: Gain comparison between simulation and experiment

Yet, and despite the difference in the gain between simulation and experiment, the experiment results validate our simulation results. They prove the robustness of our approach in improving the targeted metrics in $OFQS$.

3.5 Discussion And Possible Improvements

Before coming to our conclusions, we discuss some relevant issues in our proposition. While $OFQS$ proved its efficiency in the simulation and experiments, few things still need to be further investigated.

In our instances classification (Section 3.2.4), the parameters α and β are fixed for the three instances. This selection could be optimized and made dynamic using machine learning or fuzzy logic techniques in order to compute the most suitable classification for every traffic class. These techniques should respect the constraints of the WSN in terms of energy and computational limitations.

Moreover, in our simulations and experiments, although we considered two instances with different traffic, we did not target the same traffic classes as per the SG requirements presented in Section 3.2.4 and Table 2.1 in terms of packet size and data sending frequency. Doing so will require a large scale real sensor deployment for SG applications. We plan to test our approach with the SoMel SoConnected project [1], in order to validate it in a real SG scenario. We believe that expanding the network with a dense deployment of wireless sensors will result in even better results of our approach. Having a denser network will result in more backup routes and battery/main powered nodes, in that way, $OFQS$ will react accordingly achieving a better load balancing and increasing the network lifetime. Concerning the end to end delay, and as we already mentioned the delay parameter d may undergo abrupt variations on some links specially in a real scenario (e.g., a city) where we have a lot of interference sources. This need to be carefully investigated to avoid any negative impact on the global performance of the metric.

Furthermore, the multiple instances in RPL aim to differentiate the traffic in the network. In our evaluation the global performance of multiple instances was only considered. Further analysis should be made in order to study the impact of one instance on another while running together on the same network, and how many instances can we maximum run by still ensuring a proper traffic differentiation between the instances.

Finally, even though our approach is compliant with both RPL traffic (upward and downward), we tested it while considering upward traffic only. Enabling downward traffic will add more control packets and congestion to the network which will impact the metric behavior and require further investigation and tests which we aim to explore as future work.

3.6 Summary

In this chapter, we have proposed a standard-compliant objective function $OFQS$ with a multi-objective metric $mOFQS$ that considers by design the delay, the remaining energy and the quality of the links in order to be used with the standard protocol RPL in a SG environment. $OFQS$ adapts the routing to the number of instances in a network providing a differentiation based on the requirements of the SG applications. $OFQS$ is explained and detailed with examples then evaluated in simulations and experiment on the FIT IoT-LAB real sensor test-bed. The simulation and experiment results show that our approach achieves significant improvement in terms of End-to-End delay, network lifetime and PDR while insuring a load balancing among the nodes compared to standard solutions. Finally, the difference of gain between the simulation and experiment was evaluated and the obtained results were analyzed.

In the next chapter, we will address data prediction techniques for WSN and SGs. W will discuss the existing approaches and present our SG adequate solution for SG traffic.

Chapter 4

Data Prediction in Smart Grids

In Chapter 3, we highlighted the importance of having a multi-objective solution for Smart Grid traffic routing and presented our approach for traffic differentiation in RPL.

In this chapter, and as effective as the routing and traffic differentiation, we address data prediction techniques for Smart Grid applications. In fact, continuous data collection in Smart Grid applications may cause redundant information at the destination, which has a direct impact on the network lifetime in a wireless sensor network with battery powered sensors. Data prediction aims at reducing the communication task between sensors and sink nodes by predicting the next data inquiry using specific prediction algorithms. This will allow less data to be sent across the network, reducing then the battery consumption, the bandwidth utilization and increasing the network lifetime.

Existing approaches are either complex, causing significant computing overhead and load on wireless sensors with limited autonomy, either unadapted for a Smart Grid context where different applications with different requirements exist on the same network. Our proposition, which is a modified version of the Least Mean Square (LMS) algorithm [80] for data prediction, tackles these issues providing a QoS efficient adequate solution for Smart Grid applications. Our focus on LMS is due to its accuracy even when simple and lightweight models are used [23], which is beneficial in energy limited WSNs.

4.1 Related Work

In this section, we firstly present the main existing works on data prediction techniques in WSNs. After that, we provide a brief overview on the LMS algorithm and its proposed variants. We focus on time series forecasting (Figure 4.1) and mainly LMS algorithm which is the point of interest in this chapter.

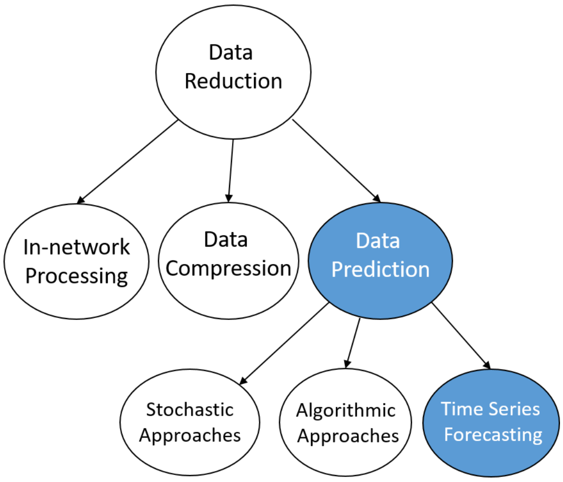


FIGURE 4.1: Categorization of energy saving in sensor networks [23] (data prediction highlighted)

4.1.1 Data prediction different schemes

Data prediction techniques focus on minimizing the number of transmitted measurements over the network by predicting future values based on previously collected data. This is generally done by predicting the measured values both at the source and the sink nodes using specific algorithms, which will require sending the predicted information only if it is shifted from the sensed one by a certain threshold.

Most of these algorithms work as follows: a model is constructed at the sensor node and sent to the sink node to keep track of the sensed phenomenon. After that, the sink node answers the user queries by using the predicted values from the model without communicating with the sensor node. This will allow reducing the energy consumption by avoiding wireless communication through the entire network up to the sink node. This operation is valid only if the model at the sensor nodes is a valid representation of the phenomenon at a given instant, e.g., monitoring the temperature variations model is different from the one of humidity. For that, the characteristics of a data prediction technique rely on the way the model is built. These techniques can be split into three categories [23] as we can see in Figure 4.1: stochastic, algorithmic and time series forecasting approaches.

Stochastic approaches consist of a characterization of the sensed phenomenon as a random process. A probabilistic model can be used for data prediction. The main drawback of these approaches is their high computational overhead, which is not suitable for sensors with limited capabilities.

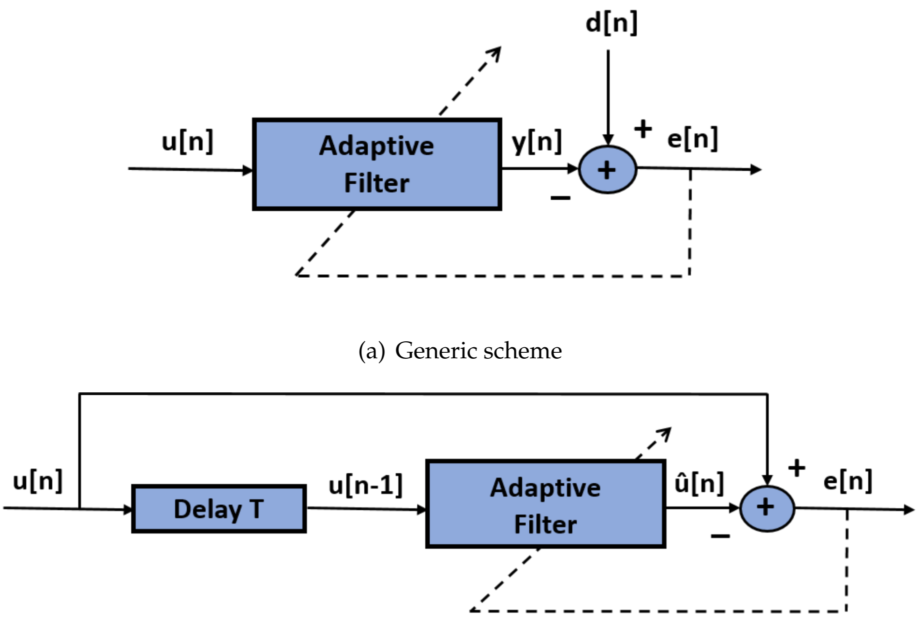
Algorithmic approaches tend to be application specific, which may not be suitable to a SG with different applications having different characteristics running on the same network.

Finally, time series forecasting consist of the use of a model to predict future values based on previously observed ones. They provide satisfactory and accurate results even when simple and lightweight models are used which is the most beneficial in energy limited WSNs.

4.1.2 LMS algorithm overview

Least Mean Square is an adaptive algorithm with very low computational overhead and memory consumption. Despite its simplicity, it provides satisfactory performances in terms of speed of computation, robustness and precision [27].

With a simple modification for the LMS filter structure (Figure 4.2), the LMS algorithm can be used for prediction by delaying the input signal by one step, using it as a the reference desired signal $d[n]$. The filter computes the estimated value $\hat{u}$[n] of the input signal at time instance n, as a linear combination of the M previous readings.



(a) Generic scheme

(b) Used as a prediction filter

FIGURE 4.2: Adaptive filter [28]

The LMS algorithm for prediction consists of building a model describing the sensed phenomenon using two instances, one at the sink and the other at the sensor node. The model at the sink can be used to answer queries without requiring any communication, thus reducing the energy consumption.

It is mainly defined by the three following equations:

1. The filter output (predicted value) [80]:

$$\hat{u}[n] = w^T[n] \times u[n] \tag{4.1}$$

   Where $u[n]$ represents the data stream generated by the sender node which consists of the previous n readings.

   û[n] is a linear combination of the previous n samples of the data stream weighted by a weight vector $w[n]$, where:

$$w(n) = [w_1, w_2, ..., w_M]^T \tag{4.2}$$

   and

$$u(n) = [u(n\text{-}1), u(n\text{-}2), ..., u(n\text{-}M)]^T \tag{4.3}$$

   where M is an integer corresponding to the memory of the filter also called filter length (how many previous samples it will use).
   We note that both w and u are of length M.

2. The estimation error:

$$e[n] = \hat{u}[n] - d[n] \tag{4.4}$$

   Which represents the error between the output and the desired signal $d[n]$ that the filter tries to adapt to. This error is given as an input for the adaptation algorithm, which will update the weight coefficients at the next instant $n + 1$ by the following weight adaptation equation.

3. The weight adaptation:

$$w[n + 1] = w[n] + \mu u[n]e[n] \tag{4.5}$$

   where μ is the step size parameter.
   The weight vector is modified at each step in order to minimize the Mean Square Error (MSE).

The step size μ and the filter length M are two important parameters that need to be defined in order to ensure the convergence and robustness of the algorithm. The former will tune the convergence of the algorithm and the latter impacts directly the computational load and memory consumption by considering more or less samples. A detailed explanation of the LMS filter can be found in [80].

The implementation of the LMS algorithm for data prediction in WSN is first presented in [28]. Here, identical filters are introduced at both the source and the sink referred as $LMS - DPS$ (dual prediction scheme). The algorithm consists of three modes of operation: ***Initialization***, ***normal*** and ***stand-alone*** modes.

– In the ***initialization mode***, the data samples are collected and reported to the sink without prediction. In this phase, the step size μ must be determined. Both the node and the sink compute the value of μ. It must satisfy the following condition[81]:

$$0 \leq \mu \leq \frac{2}{\lambda_{max}} \tag{4.6}$$

where λ_{max} is the greatest eigenvalue of the auto-correlation matrix **R** [80]

– In the ***normal mode***, both the sink and the node use the last M samples to compute the prediction for the upcoming measurement, and update the filter coefficients. When the error drops below e_{max} for M consecutive iterations, the node switches to stand-alone mode. We note that the default start values for the filter weights are assumed to be zero.

– In the ***stand alone mode***, the node still collects data and makes predictions, but as long as the error is below e_{max}, the filter is fed with the prediction $\hat{u}$[n] instead of the reading value $u[n]$, and the sink receiving no reading from the node assumes that the predicted readings are below the error threshold. If the error exceeds e_{max}, the filter switches back to normal mode and reports the readings.

It is important to note that the prediction is performed only on the sender nodes and the sink, on packets collected by the sender. Intermediate nodes do not interfere in the prediction scenario.

4.1.3 Time series forecasting and LMS proposed variants

In literature, extensive work address time series forecasting techniques for WSNs [26, 82]. For example, in [83], a couple of autoregressive mechanisms are proposed to predict sensed samples in WSNs. The authors use Yule-Walker and Lattice-based approaches to estimate the model coefficients. Similarly, several works focus on LMS algorithm as well. In [84] a gradient adaptive step size (μ) algorithm with dual LMS adaptive filters is proposed, where the gradient is measured using these two LMS filters. In [85] a new approach for updating the step size is proposed, by computing it in each iteration. The step size is dynamically re-chosen at each time point to minimize the sum of the squares of the estimation errors up to the current time, irrespective of the values of μ at all previous time points. In [28], an implementation of LMS algorithm for prediction in WSN is presented. The

LMS algorithm uses a dual prediction scheme by running the instance of the filter on both the sink and the node. In [86], a variable step size is proposed to improve the initial adaptation of the data by switching to a new step size stable value after μ has sufficiently learned what kind of data the filter receives. Many other works have addressed the variable step size of LMS [87].

However, all these proposals mostly require many adjustments of several parameters in order to optimize μ or update it on every iteration, which is not suitable for a WSN with limited computation capabilities and frequent changes.

Normalized Mean Square Error ($NLMS$) [88] is a modification of the LMS algorithm in which the step size is normalized with the power of the input data. In order to mitigate the variation of the latter the step size is updated automatically accordingly. Although $NLMS$ offers a higher stability than LMS, the base value of the step size has to be chosen carefully. Moreover, computing the step size on every iteration is a costly task for WSN with restrained energy. The Recursive Least Square (RLS) [88] adaptive filter is another algorithm that recursively finds the filter coefficients in order to minimize the weighted linear least square cost function related to the input signals. RLS algorithm has excellent performance in time varying environments and exhibits fast convergence, but this comes at the cost of high computational complexity which is also inadequate to WSNs. The readers may refer to [89] for a comparison between LMS, $NLMS$ and RLS.

Even though time series estimation techniques have been successfully used in WSN applications, it is important to note that for each individual application, the estimator parameters such as weights and order must be computed. Moreover, a single time series estimation may not fit for all different applications [90]. This is particularly noteworthy because a SG network holds in different data types with different QoS necessities. Thus, the proposed solution should handle those requirements and be as general as possible.

4.2 LMS Limitations

LMS adaptive algorithm is proven to be robust and accurate with a very low computation [27], yet showing features that perfectly fit WSN requirements. However, the choices of the step size and the filter length are essential in the convergence of the algorithm. Starting with a large step size gives a fast convergence of the filter but results in a larger MSE, and a too small step size degrades the capabilities of the algorithm. Varying the step size to a smaller value after a certain number of iterations if needed is then beneficial. Concerning the filter length, its choice will indicate the computation load of the algorithm (how many samples we will consider on every iteration). We note that increasing the filter length does not necessarily improve the performance of the filter. Choosing the right parameters is then crucial.

Many propositions to adapt and adjust these variables are proposed in literature, but having a direct mathematical analysis of the stability and steady-state performance is a very complicated task in LMS [80]. These adaptations may seem adequate for one application and kind of data set, but less efficient to other ones.

Adaptive filters perform predictions generally without requiring a priori knowledge about the statistical properties of the phenomenon of interest. But due to the very complex task of selection of the optimal step size, when to increase/decrease it and the optimal filter length, specially in the case of multiple applications running on the network, we propose a modification to LMS. It consists of collecting the data for every application for a specific time, storing them and performing a simple and straightforward training script to choose the optimal filter parameters for every application.

Our proposition enables then a reliable data prediction for the different heterogeneous applications in a SG by choosing the adequate parameters accordingly. It is mostly suitable in contexts of continuous data reporting applications where data redundancy is likely to occur.

4.3 Our Contribution: LMS_MOD

Our contribution, that we denoted by LMS_MOD (for modified LMS), consists of adding another step to the initialization phase for the LMS prediction algorithm in [28] by training the filter with enough data. Figure 4.3 shows a diagram with the new additions to the existing LMS algorithm for data prediction in WSN which are marked in blue. We vary the step size and filter length within specific intervals in order to optimize these values (by minimizing the MSE) for every specific application. We start by the upper bound of μ as per Equation 4.6.
In order to minimize the MSE we compute:

- The appropriate filter length M denoted as i,
- The optimal time to switch to a smaller value denoted as j,
- The new value of μ denoted as k.

After we obtain the three aforementioned values, we execute our prediction algorithm with these parameters for the rest of the data. In this way

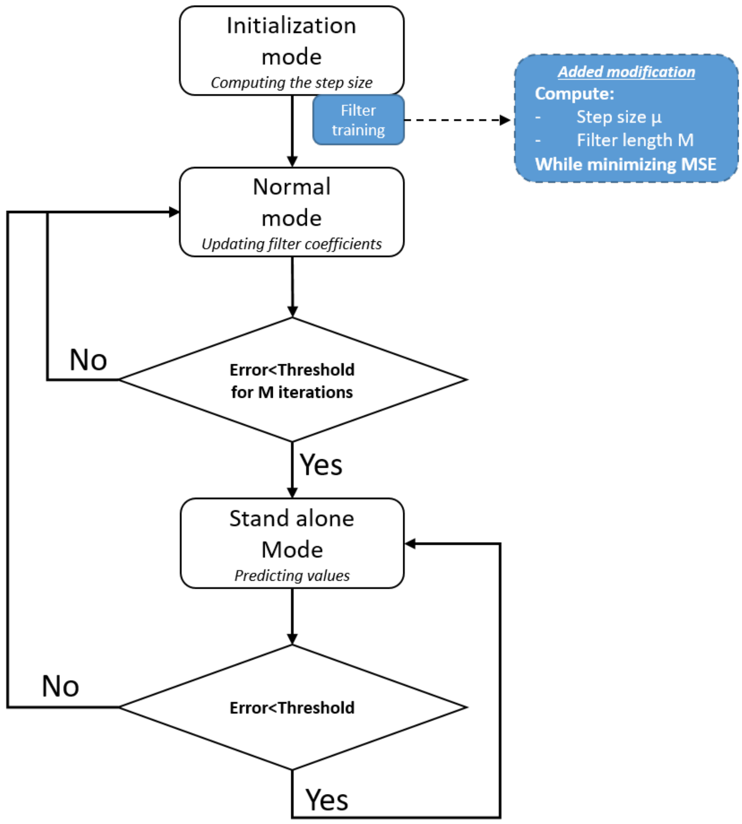


FIGURE 4.3: Diagram of the modified LMS algorithm process (the modification marked in blue)

and since the application data has different characteristics, every application will have distinct parameters achieving a minimal MSE. Concerning the energy load resulting from this adaptation, we run our adaptation script offload using numerical simulations in order to obtain the coefficients before running it on a WSN.

4.3.1 Simulation setup and parameters determination

In order to validate our proposition, we use real value traces from the NREL National Wind Technology Center [91] for photovoltaic cells. We consider the irradiance, temperature, humidity and average wind speed values collected every minute between 4 am to 8 pm from 04/06/2017 and 06/30/2018. A description of the traces characteristics is presented in Table 4.1. It is worth mentioning that each data type has different characteristics and ranges, therefore, the prediction task is even more challenging.

We calculate the upper bound of the step size λ_{max} using the first 60 values of $u[n]$, same as the number used to train the filter in the initialization phase. We consider four different thresholds for each data type (note that these thresholds can be adjusted for specific needs). We consider a one hop communication environment with no loss in order to prove the efficiency of

Data Type	Max. Value	Min. value	Std. dev.
Irradiance (w/m^2)	$1.4932 * 10^3$	0	360.41
Air Temp. (°F)	88.847	24.318	13.246
Humidity (%)	100	11.52	22.9013
Avg. Wind Speed (MPH)	54.60	0.693	6.518

TABLE 4.1: Data traces description

our proposal in an optimal case scenario. We test our algorithm by means of numeric simulation on Matlab. For the adaptation in the initialization phase, we execute a Matlab script for one day of collected data. We vary three parameters i, j and k corresponding to the filter length, the factor by which we will divide the old μ and after how many iterations simultaneously (the time we will switch to the new computed μ value) respectively. We vary i between 1 and 10, and j, k between 1 and 100 with a step of 5, and we choose the value that minimizes the MSE. We note that the choice of these intervals can be changed, but we realized after several tests that the optimal values always fall within these ranges. The obtained values are shown in Table 4.2 and then are used to feed the filter in order to predict the data for the whole previously mentioned duration. We compare LMS_MOD to LMS_VSS which is proposed in [86]. LMS_VSS respects the prediction phases as in [28] (Initialization, normal and stand-alone) but with a variable step size like our proposition. In [86] μ starts with the value:

$$\mu_{old} = 2\lambda_{max}^{-1}.10^{-2} \tag{4.7}$$

and switches to a stable value:

$$\mu_{new} = \mu_{old}/M \tag{4.8}$$

after n iterations. Where M is the filter length and n is the number of consecutive readings in stand-alone mode. They chose $n = M^{3/2}$. M is initialized to 4 in [86] and to different values in [28] chosen arbitrarily. For the sake of fairness, we chose the same filter length and λ_{max} for LMS_VSS as the one used in LMS_MOD for every data set.

4.3.2 Performance Evaluation

In this section, we evaluate our proposition LMS_MOD on Matlab in comparison with LMS_VSS in terms of two performance metrics: root mean square error and data reduction percentage.

Data Type	**Threshold**	**Filter Length**	μ **Div. Factor**	**Nbr. of Iter.**
Irradiance	1	1	96	76
(w/m^2)	3	1	51	76
	5	1	96	81
	7	1	91	71
Air Temp. $(°F)$	0.5	4	26	16
	1	4	6	21
	2	4	11	41
	2.5	4	1	1
Humidity %	1	3	96	16
	2	3	36	31
	3	3	41	16
	4	3	16	31
Avg. Wind	0.5	2	6	71
Speed (MPH)	1	2	6	96
	1.5	1	26	96
	2	1	1	1

TABLE 4.2: Data traces obtained parameters

4.3.2.1 root mean square error

In order to reflect the overall performance of our proposition we compute the Root Mean Square Error ($RMSE$) for every data trace, which corresponds to the root of the MSE that is given by:

$$MSE = \frac{1}{n}\sum_{n}^{i=1}(\hat{u}[n] - u[n])^2 \tag{4.9}$$

where n is the number of samples.The MSE reflects the overall performance in terms of prediction errors. The RMSE is used to simplify the display of the results.

Figures 4.4 $\rightarrow$ 4.7 show the $RMSE$ for LMS_MOD and LMS_VSS for the different data types. We observe that for the temperature, humidity and average wind speed LMS_MOD has a lower $RMSE$ than LMS_VSS (a lower overall error in the prediction), this is mainly due to the choice of the parameters (Table 4.2) that minimizes the MSE ($RMSE$ respectively). For the irradiance, the $RMSE$ is quite close for LMS_MOD and LMS_VSS with a slight improvement for LMS_MOD. Here, the filter length chosen is equal to one (Table 4.2). This is mostly due to the high deviation of the collected data as we observe from the high value of the standard deviation, 360.41, in Table 4.1. In this case, the step size has a relative small value (the data values for the irradiance have a strong variance between negative and positive values). Then, the dividing factor has less effect on the step size variation. Hence the relatively close values of $RMSE$ for both approaches.

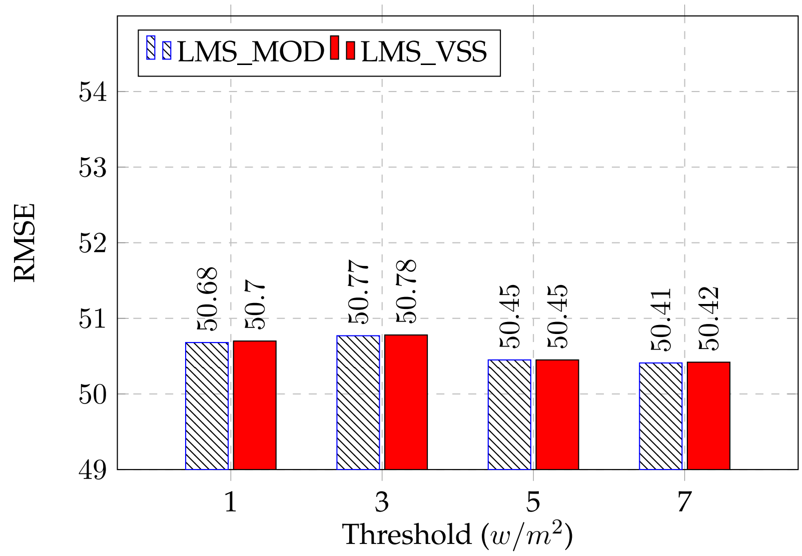


FIGURE 4.4: RMSE for irradiance

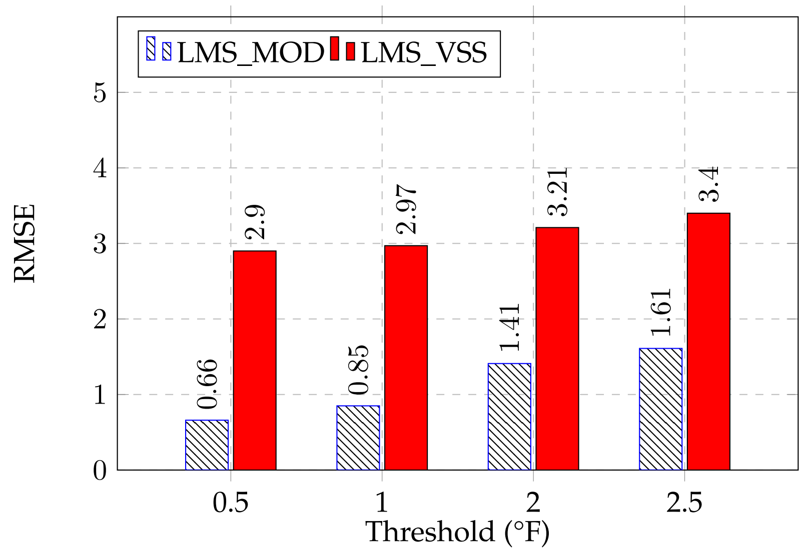


FIGURE 4.5: RMSE for temperature

4.3.2.2 Data reduction percentage

Figures 4.8 $\rightarrow$ 4.11 show the data reduction percentage achieved for both methods. This latter corresponds to the number of predicted packets whose values fall within the range of the chosen threshold, thus that were not sent to the sink. We can see that our proposition presents higher reduction percentage than LMS_VSS for the temperature, humidity and average wind speed: between 2 and 6% of packets are saved for the temperature, between 10 and 12% for the humidity and between 1 and 8% for the average wind speed. This is due to the optimal choice of the parameters during the offline training of the filter for each application and data type accordingly. Concerning the irradiance, the reduction percentage is again close between LMS_MOD and LMS_VSS with a slight improvement for LMS_MOD.

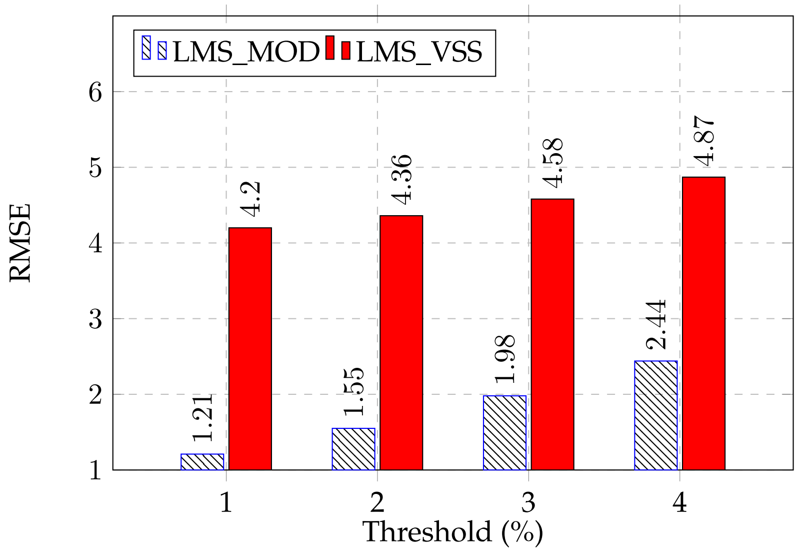


FIGURE 4.6: RMSE for humidity

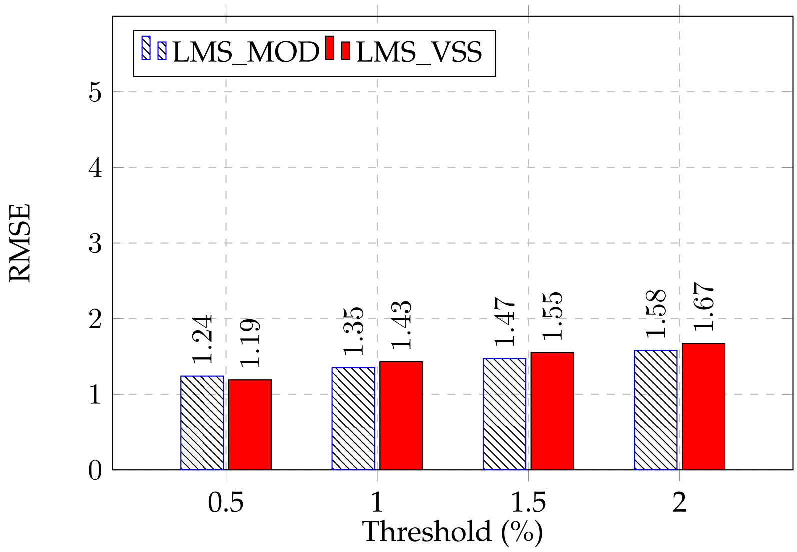


FIGURE 4.7: RMSE for average wind speed

This is due as already mentioned for the similarity of the chosen parameters between the two propositions. It is worth noting, that the existing solutions, and in particular LMS_VSS in this case may perform well in some applications but less efficiently in others, which is shown in our results. Unlike our approach that provides a benefit in every case.

4.4 Discussion And Possible Improvements

Before coming to our conclusions, we discuss some relevant issues in our proposition. While LMS_MOD proved to be efficient for several data types by reducing the MSE and ensuring a high data reduction percentage, our straightforward training may misbehave in some conditions, i.e.,

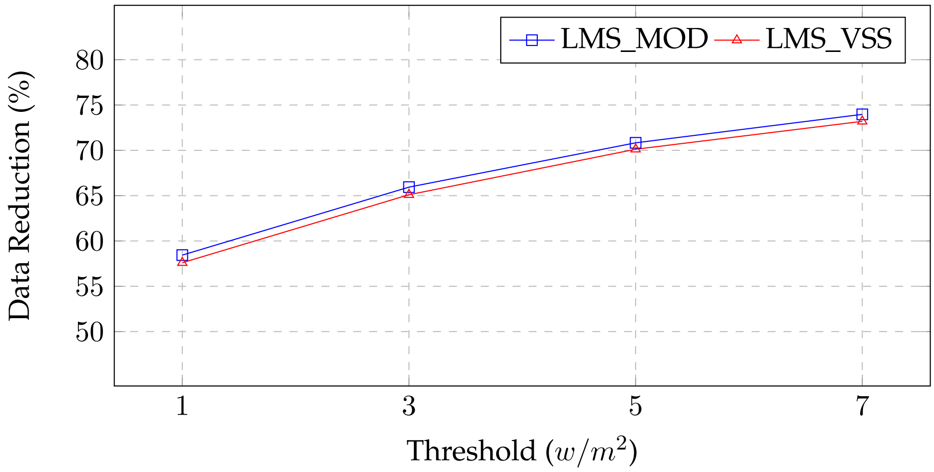


FIGURE 4.8: Data reduction for irradiance

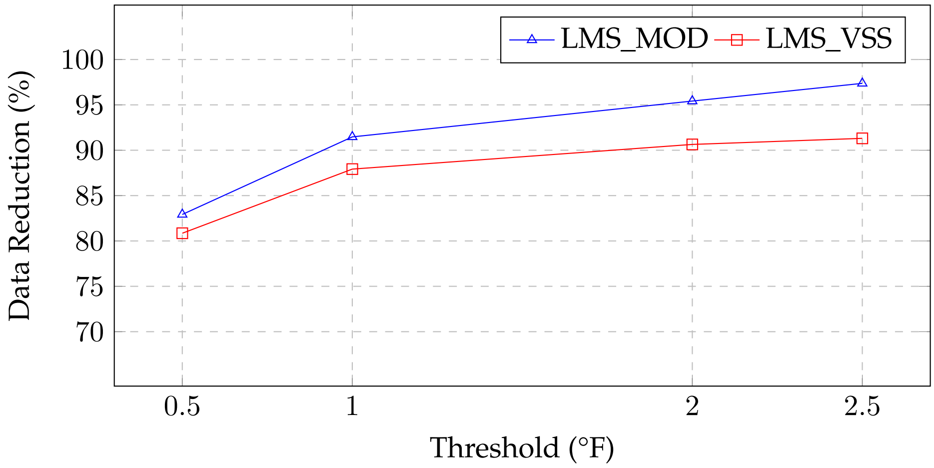


FIGURE 4.9: Data reduction for temperature

in environments where the data may become incoherent from one season to another, or when one day of data training is not enough. A possible improvement could be to investigate the variations of every data set (e.g, maximum and minimum values, standard deviation) and train the filter for every data type accordingly by taking these variations into consideration. This will allow a global yet personalized vision of every application traffic type.

Moreover, in LMS_MOD we optimize the parameters so as we minimize the MSE, which might result in a lower data reduction percentage in some cases. Same way if we train it in the opposite way. Further improvement on how to optimize these two metrics should be studied and considering new metrics as well.

Furthermore, in a real sensor network our model could rise a reliability issue; in a real WSN with interference and losses, if the message containing the reading message transmitted by the sensor is lost, the model at the destination will go apart and the algorithm will predict erroneous values.

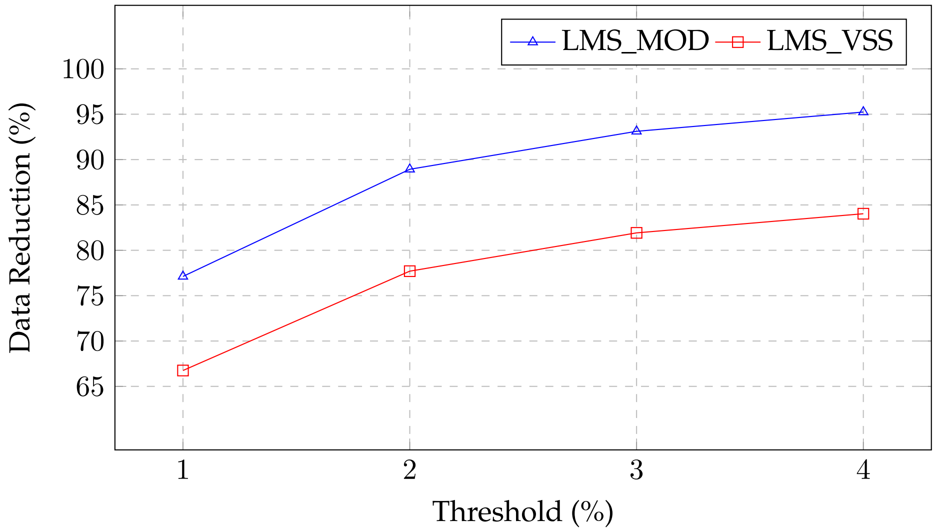


FIGURE 4.10: Data reduction for humidity

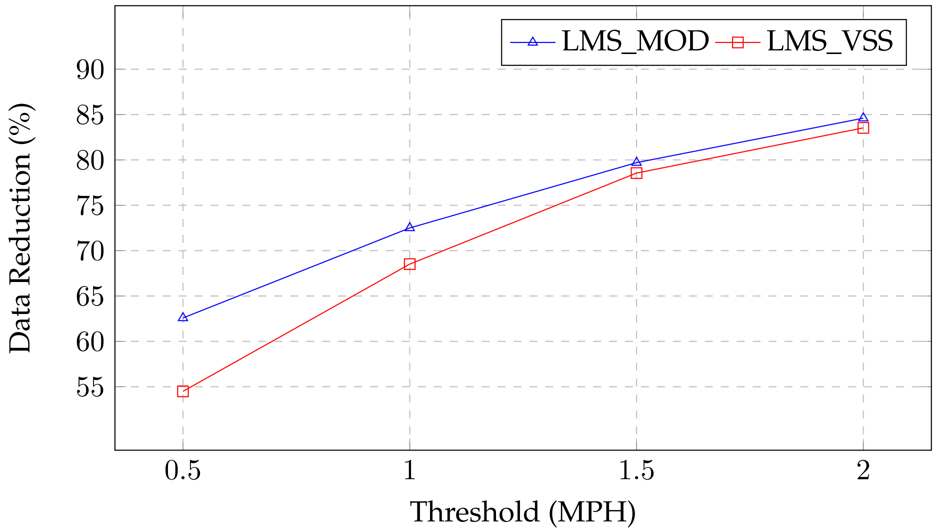


FIGURE 4.11: Data reduction for average wind speed

This should be carefully handled by sending regular control messages per example in order to maintain the synchronization between the sink and the sensor nodes.

4.5 Summary

In this chapter, we have presented an overview of the work around data prediction in WSN and mainly considering time series forecasting and LMS algorithm. We have proposed a modification of the LMS prediction algorithm for WSN to adapt it to different applications with different QoS as per a SG environment. We have tested our approach with real data traces for photovoltaic cells, and have performed simulations considering one hop communication networks. We have trained the filter offline for one day with the data traces corresponding to each application in order to optimize the parameters that minimize the MSE. Our numerical results show a better performance than LMS_VSS, a state of the art solution, in terms of $RMSE$ and percentage of data economy.

In the next chapter we will study another data reduction technique to be used for SGs which is data aggregation. We will overview the main ongoing work around data aggregation for WSN and SGs and present our proposed solution.

Chapter 5

Data aggregation in Smart Grids

In Chapter 4, we reviewed the existing work on data prediction techniques and mainly around time series forecasting and LMS algorithm in wireless sensor networks. After that, we presented our approach which consists of a modification of the LMS algorithm to fit with the heterogeneous QoS demands of Smart Grid applications. Data prediction techniques are the most beneficial in the context of continuous data collection where data variations are limited and redundancy is more likely e.g, photovoltaic monitoring applications.

In this chapter, we address another data reduction technique which is data aggregation for Smart Grids applications in a wireless sensor network. Data aggregation can be applied not only on the source and destination nodes as per data prediction, but potentially on every node in the network. It can be used with wider applications by aggregating jointly data packets generated by different applications with different QoS. However, this comes at the cost of many factors e.g., using the radio spectrum more frequently than a data prediction scenario, sending larger packets across the network which may increase the packet loss, etc. That's why more than one data reduction technique is needed, where one will be picked depending on the characteristics of the application and the data it generates.

Here, we will approach data aggregation. As presented in [23] and highlighted in blue on Figure 5.1, data aggregation can be split into two categories: in-network processing and data compression (readers may refer to Section 1.2.2 for more details).

In fact, with the extensive load of data collected from the different Smart Grid applications, it appears necessary to exploit data reduction techniques in order to mitigate the communication charge and radio spectrum usage in the wireless network. Data aggregation is a feasible process in which information is gathered and expressed in a summary form. In other words, and in a wireless sensor network scenario, it consists of combining data from multiple sensors across the network and sending the aggregated data to the base station. This will reduce loads on the communication links, thus

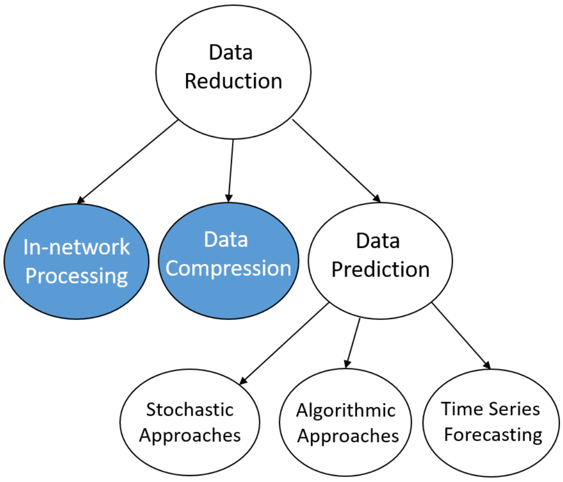


FIGURE 5.1: Categorization of energy saving in sensor networks [23] (data aggregation techniques highlighted)

achieving a better utilization of the wireless channel and reducing energy consumption. Data aggregation can be lossless, which refers to concatenating individual data packets into larger ones, thus reducing per-packet protocol overhead. In this case, no data is lost. Or lossy aggregation, where we may encounter loss of information since the payload is reduced, e.g, averaging the sensor values.

In a Smart Grid, different applications require different QoS priorities. Consequently, data aggregation must respect these requirements (i.e, delays caused by aggregating the packets) in order to ensure a reliable communication.

Existing approaches lack full suitability with Smart Grid applications in terms of QoS requirements specially in a wireless sensor network environment (e.g., delay sensitivity, heterogeneous priorities and sizes in data packets, limited autonomy). They might be suitable in some aspects but disregard others, which may not be adequate in Smart Grids.

Our proposition consists of a QoS efficient data aggregation algorithm for the different traffic in a SG network. It takes into consideration the heterogeneous traffic in terms of priority in the Smart Grids.

5.1 Related Work

In this section, we overview the different methods and categories of data aggregation in WSN and we present the main existing works on data aggregation in WSN and SGs.

5.1.1 Data aggregation means

The architecture of the sensor network may play a vital role in the performance of different data aggregation protocols. In this section, we classify these protocols from a communication perspective into three major classes of aggregation algorithms [92]: structured, unstructured and hybrid. This is done according to the characteristics of their communication pattern (routing protocol) and network topology.

- Structured communication (usually hierarchy-based) class refers to aggregation algorithms that are dependent on a specific network topology and routing scheme to operate correctly. They can be per example cluster-based, where sensor nodes are grouped into clusters, with one cluster head for each cluster. Members of a cluster send packets to their cluster head via single-hop or multi-hop communication. The cluster head is responsible for coordinating data transmission activities of all sensors in its cluster. Examples of cluster-based algorithms: LEACH [93], HEED [94], etc. Or tree-based, where sensor nodes are organized into a tree and data aggregation is performed at intermediate nodes along the tree. After that, a concise representation of the data is transmitted to the root node [95]. Examples of tree-based algorithms: EADAT [96], PEDAP [97], etc.

- Unstructured communication (usually, gossip-based) category covers aggregation algorithms that can operate independently from the network organization and structure, without establishing any predefined topology e.g., gossip-based communication protocols [98, 92]. They are strongly related to epidemics, where an initial "infected" node sends a message to a (random) subset of its "contaminated" neighbors, which repeat this propagation process "one to many". With the right parameters, almost the whole network will end up participating in this propagation scheme. Examples of gossip-based appraoches: Push-Sum Protocol [99], DRG [100], etc.

- Hybrid approaches combine the use of different communication techniques to obtain improved results from their synergy. Commonly, the use of a hierarchic topology is mixed with gossip communication [101].

5.1.2 Data aggregation in WSN and SGs

In literature, many works have addressed the data aggregation in WSN and SGs. In [102] two aggregation methods for processing data in smart meters are used: combining and manipulating. In the combining method, the concentrator removes all individual headers and includes only one single header for the large packet with no data modifications. The manipulating method consists of calculating the result of the messages thus reducing considerably the total size of the messages. In [103], two aggregation techniques are proposed as well: Quantize and average aggregation. Quantize aggregation is used for analogue signals that are sampled by sensors and average aggregation for data generated by sensors deployed on the same location. However, although these techniques reduce the size of data transmitted through the network, no QoS measures are considered concerning the delays and the diversity of data messages from different applications with different priorities.

Data packet concatenation in SGs is also addressed in [104], the authors achieve header compression on packets and formulate an optimization problem to optimally configure the sizes of the aggregated packets. They consider pre-defined message arrival distribution to the sink. However they utilize only overhead reduction, which may be insufficient alone in the presence of larger data packets with smaller headers.

Many other researches considered energy [105], delay guarantee [106] and other QoS requirements [107, 108] in data aggregation for SGs and sensor networks generally. However, none of these works addresses the challenge of having delay sensitive data traffic with different delivery priorities and sizes while reducing energy consumption and maximizing the available bandwidth.

Our proposition aims to enable QoS aggregation in the SGs for heterogeneous traffic. This is done by allowing data traffic with different priorities to be aggregated/concatenated in the WSN according to their requirements.

5.2 Proposed Solution

In our proposition, we consider a SG network consisting of several wireless sensors collecting data with different packet sizes and priorities. They can potentially act as aggregators, if they have enough resources, that receive the data and aggregate or concatenate it depending on their QoS requirements. They finally send the aggregated data across the network. The routing process is mostly left unchanged, we only add the aggregation functionality when it is possible. With that being said, this algorithm can be used with our proposition for multiple instances in RPL from Chapter 2. However, it is worth mentioning that in case of aggregating data packets inside the network (potentially on every node in the network), the routing process might get affected. The metrics might switch routes when a big packet is arriving for example. These issues will be discussed later in the chapter.

Packets are generated with classifiers in their headers considering their type and criticality. We classify them into two levels: critical and regular. These levels could be adjusted for other applications depending on the network characteristics. Two different queues are created at the aggregator level: lossy and lossless queue. The lossy queue contains delay insensitive data packets (regular) that are generally big in size [19], which will allow us to aggregate the packets with the appropriate aggregation function [25]. The lossless queue contains delay sensitive data packets with critical priorities and with a header which represents a significant overhead compared to the payload size. Header compression is thus performed on the packets. We note that in this work, we do not deal with the different aggregation techniques (e.g., average, sum), readers may refer to [25] for more details.

The proposed aggregation algorithm

In the following, we explain the main functionality of the proposed aggregation algorithm and its functions. We note that our proposition is twofold: in the first part we consider that the aggregators are situated one hop away from the sink which we refer to as data compression (Figure 5.2). In the other part, and using the same algorithm, every sensor can act as an aggregator depending on its resources wherever it is located in the network which we refer to it as in-network processing (Figure 5.5).

First part: data compression

Figure 5.2 depicts a small network consisting of 3 sender nodes (sensors) and an aggregator which is situated one hop away from the sink. The sender node sends and receives packets from other sensor nodes with different priorities included in their headers. We refer to this delay by *Maximum allowed delay*. Routes are constructed according to the existing routing protocol with no influence from the aggregation algorithm as already mentioned.

In Figure 5.3, when an aggregator receives a packet, it will firstly update its delivery time ($Update\_LD()$) corresponding to the time-stamp included in the header of the packet (*Maximum allowed delay*) minus the time T the packet spent to arrive to the aggregator. This will allow us to identify how much time the packet can stay in the aggregator before being sent to the sink. We store the value in the variable LD corresponding to Left Delay.

After that, the function $Free\_Space()$ will check whether the node can store more packets. This decision is made depending on the node's available internal storage at the time of arrival of the packet.

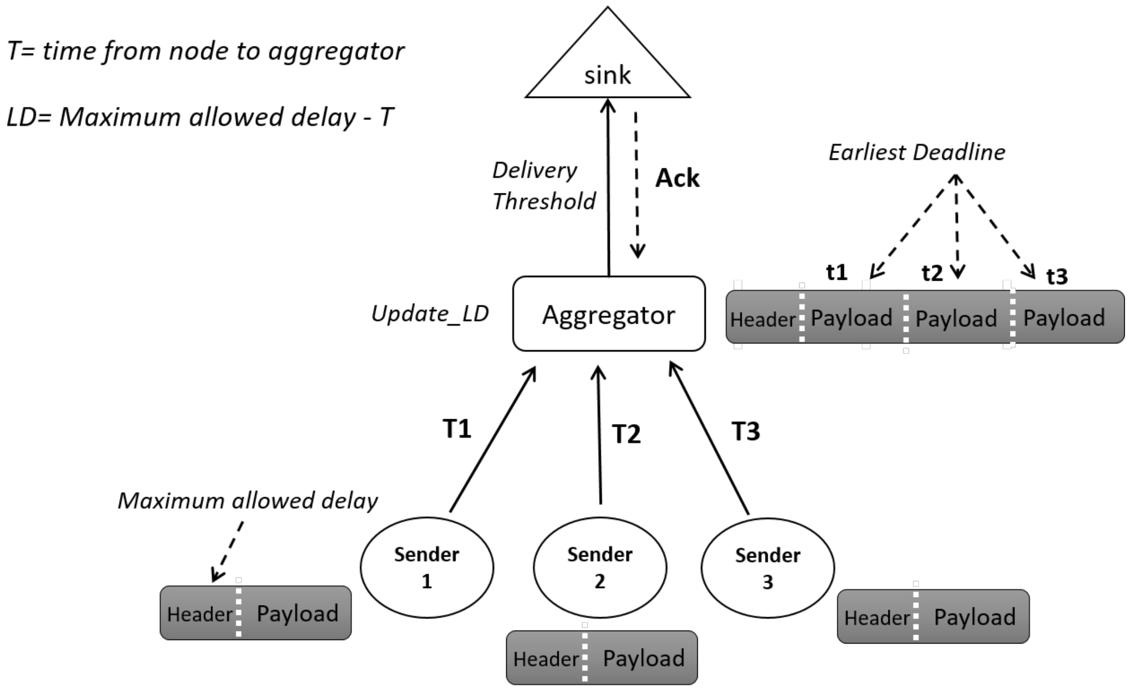


FIGURE 5.2: Aggregation scenario

Algorithm 1: Aggregator node

Update_LD() ; `/* LD= Maximum allowed delay - time from the sender node to aggregator */`
if *Free_Space()* ***AND*** *Battery_Node()* > *Threshold1* **then**
  Aggregate();
  Send_AgPacket();
else
  Send_Packet();
end

FIGURE 5.3: Aggregator node algorithm

Algorithm 2: Lossy_Queue()/Lossless_Queue()

```
TTL if Earliest_Deadline > Delivery_Threshold AND LD >
Delivery_Threshold AND AggrPktSize < MTU AND TTL > 0 then
    Lossy_Aggregation()/Lossless_Aggregation();
    Update_Earliest_Deadline() ;      /* Earliest deadline in
     the aggregated packet */
    TTL - - ;
else
    Concatenate();
end
```

FIGURE 5.4: Lossy/Lossless queue algorithm

The function $Battery\_Node()$ will check whether the node has enough energy to aggregate more packets. By enough energy we mean that the aggregator (considering a wireless sensor powered by a battery) should have enough left capacity more than a predefined threshold (e.g. 20% energy left). If these two conditions hold, we can aggregate packets and send the aggregated packets afterwards. If not, the packets are sent without aggregation. In the aggregator, the packets can be aggregated or concatenated (header compression). For the sake of simplicity, we show only header compression on Figure 5.2.

In the aggregate function, we check the packet type and send it to the corresponding queue (lossy or lossless queue). We note that the algorithm in Figure 5.4 holds for both the lossy and lossless queues. The only difference is in the function $Lossy\_Aggregation()$ or $Lossless\_Aggregation()$.

If the packet is tagged as regular, it is sent to the $Lossy\_Queue()$. If not (i.e., tagged critical), it is sent to the $Lossless\_Queue()$ (Figure 5.4), where four conditions have to be validated in order to aggregate packets:

- *Earliest_Deadline* > *Delivery_Threshold*: aggregating if the packet with the earliest deadline in the aggregated packet is still within its allowed delay. The delivery threshold is updated proactively with the acknowledgment (Ack) sent back from the sink to the aggregator that piggybacks the time spent from the aggregator to the sink. For example, we can see on Figure 5.2 3 packets in the aggregated packet with t1, t2, t3 deadlines to be sent to the sink. These deadlines are decremented with the experiment time ($Update\_Earliest\_Deadline()$). Once the packet with the earliest deadline approaches the $Delivery\_Threshold$ by a certain fixed value, the whole aggregated packet is sent to the sink. In other words, the packet with the earliest deadline must have enough time to be sent from the aggregator to the sink.

- *LD* > *Delivery_Threshold*: which means that the delivery threshold from the aggregator to the sink must be smaller than the Left (remaining) Delay LD, and always by a fixed threshold.

- *AggrPktSize < MTU*: aggregating as long the aggregated packet is smaller than the Maximum Transmission Unit (MTU) of the link.
- *TTL > 0*: even if the above conditions are valid and after a certain time we send the packet anyway on the link i.e, when the Time To Live (TTL) expires, which will avoid routing loops.

As long as these above conditions are valid, an arriving regular packet to the aggregator will undergo a $Lossy\_Aggregation()$, and all the timers are updated. Same applies for the $Lossless\_Queue()$ with a packet tagged critical. If not valid, we concatenate the incoming packet with the existing aggregated packet if possible in terms of time and available space and send it immediately to the sink.

Second part: in-network processing

In-network processing consists of processing the data collected by the sensor nodes themselves between the source and the destination reducing the amount of data while traversing the network. Our proposition relies on the hypothesis that the aggregation function is totally independent from the routing protocol. While we presented in the previous paragraph the model of aggregating the packets one hop away from the sink, our algorithm can be applied at any node in a WSN. In short, any node inside the network can choose whether to aggregate or not the incoming packets based on its resources. This will reduce the load on the wireless links, by communicating less (sending less packets) inside the network. However, and although the same algorithm (Figures 5.3 and 5.4) can be applied here, few things that may affect the routing process need to be further investigated and considered, which we will discuss in the next section.

Second part: case scenario

Figure 5.5 depicts an example of a WSN running RPL protocol with the ETX metric. RPL being a proactive routing protocol pre-establishes the routes in the network. Each node sends packets in the upward direction to its parent node noted P (for more details about RPL functionality readers may refer to Section 2.4).

Node 6 forwards packets to node 1 passing by nodes 4 then 3, nodes 7 and 8 passing by nodes 5 then 3, nodes 5 and 4 by node 3 and finally nodes 2 and 3 directly to 1.

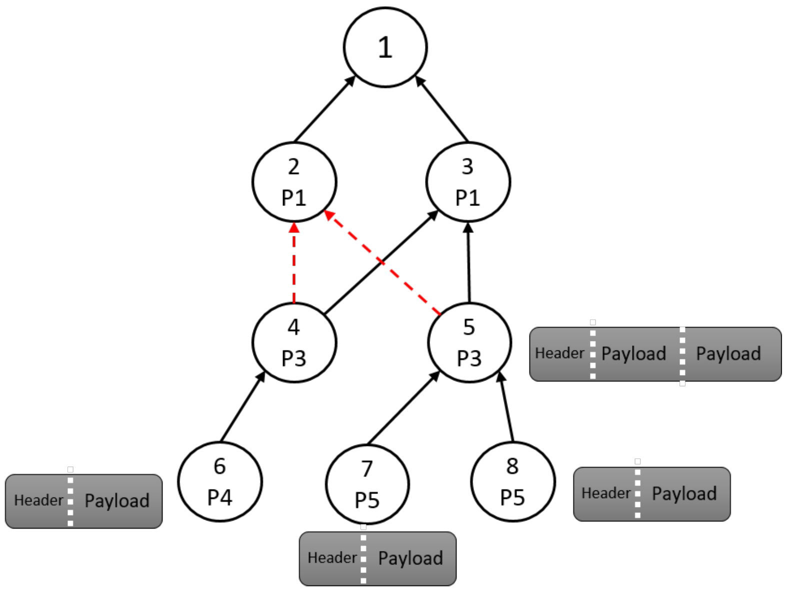


FIGURE 5.5: Data aggregation example in a WSN

First of all, the nodes will aggregate a new packet if and only if this new packet has the same destination as the one already in the queue. This will avoid disaggregating the packets on the way to the destination. We consider that node 4 is unable to aggregate packets due to its limited resources, then the routing remains unchanged through this node. Meanwhile node 5 has enough resources to aggregate packets (again, for the sake of simplicity we show only header compression as aggregated data on node 5). Node 7 and 8 forward data packets to node 5 which will aggregate these packets as described in Algorithms in Figure 5.3 and 5.4.

Here, aggregating these packets may lead to a higher percentage of packet loss (which means higher re-transmissions) since we are now sending larger packets. This will affect the ETX metric that might switch to a new parent (parent 2 in this case) if it offers a better ETX. The aggregation algorithm will then affect the choice of the routes in the network. Moreover, nodes and links near the aggregating nodes will be more saturated than the others in the network (because of sending larger packets on these links and through these nodes), which might as well alter the ETX metric value.

Now considering another example where another metric like HC for example is used. This metric has no reliability mechanism and relies only on the number of hops from the destination to deliver the packets. Aggregating packets on a network using this metric will make it more likely for routing inconsistencies to happen for the same reasons stated above.

Finally, our approach relies on timers updated regularly across the network in order to guarantee the packet delivery on time respecting each packet's QoS. Losing a packet holding a timer may cause a misbehavior of the algorithm and the routing protocol (e.g., holding a packet in the aggregator more than its tolerated delay).

As a result to the previous observations, we may realize that our algorithm, although being independent from the routing protocol, may influence and alter the choice of the routes in some cases. This is a behavior we cannot escape. But still, our algorithm can independently sit on top of any routing protocol without adding extra mechanisms to it. What should be considered carefully, is the choice of the routing metrics (if the routing protocol allows it) in case an aggregation is envisaged on the network. These metrics should take into consideration the burden that could be added by the aggregation scenario.

5.3 Expected Results

In this section, we discuss the results that we expect to obtain when validating our proposition with simulations and experiments. Our approach can be compared to any other algorithm with and without aggregation e.g., the algorithms discussed in Section 5.1.

First of all aggregating packets will lead to less packets sent across the network and less bandwidth consumed, which will result in reducing the load on the communication links and achieving energy savings since the communication task consumes most of the energy in WSNs.

Moreover, packets criticality and sizes are taken into consideration in our proposition. For that, we expect that the packets will arrive within their deadlines, thanks to the different timers and thresholds across the network. We note that in order to respect these thresholds the network and the different nodes should be perfectly synchronized. One solution could be to investigate the use of Time-Slotted Channel Hopping (TSCH) MAC protocol [109] which allows a global time synchronization between the nodes in the network.

On the other hand, the delivery delays will be longer than a non aggregation scenario where packets are not stored in the queues. This will have an impact on the total latency in the network but won't affect the QoS requirements of the packets since they will be sent within their deadlines. Packet delivery ratio might be affected also in our proposition as in any other aggregation algorithm, since aggregating means (sometimes) sending larger packets thus resulting in more losses. We will deeply investigate this issue in order to mitigate these potential losses.

5.4 Summary

In this chapter, we have overviewed the main work around data aggregation in WSN and SGs. We have proposed a work in progress solution for data aggregation in SGs networks. QoS requirements of the different applications are taken into consideration by storing the packets in two different queues depending on their quality requirements. The expected results aim to reduce the energy consumption in a SG controlled by a WSN, while respecting the corresponding delays and QoS requirements. Several tests and investigations have to be performed (i.e, computer simulations) before the completion of this work to quantify the gains. After that we will test our algorithm on a real test-bed [75] to validate our theoretical approach.

In the next and final chapter, we conclude the thesis and provide an overview of the future work and research directions we would like to consider.

Chapter 6

Conclusion and Perspectives

The Smart Grid envisages the electric grid as a distributed, flexible, automated and integrated infrastructure. It includes decentralized control, diagnosis and repair providing the various actors with major capabilities. Moreover, it faces the challenges of switching from the traditional unidirectional and centralized conventional grid to a Smart Grid where the number of producers may equalize the number of customers. With that being said, it appears necessary to instrument and control the different entities in the network with an adequate and efficient communication network. A wireless sensor network, designating the ubiquitous network of the Smart Grid, communicating and exchanging control messages through the grid will enable these aforementioned features.

The aim of this thesis consists of enabling QoS efficient heterogeneous communications for Smart Grids based on wireless sensor networks.

6.1 Conclusion

Our first objective in this thesis has been to ensure a QoS efficient routing in the ubiquitous network of the SG. Our protocol of interest is the standard RPL for routing in low-power and lossy networks that the SGs are a part of. RPL allows multiple logical instances at the network layer which will enable QoS differentiation. Moreover, being a general protocol, the main objective functions of RPL are not optimized for SG applications traffic types.

Our proposition $OFQS$ consists of a new objective function to be compliant with RPL protocol to support the multi-instance feature proposed by the standard. Our approach takes different features of both nodes and links into consideration; it considers the remaining energy of nodes, the latency and the multiple instances beside the reliability using ETX metric at once. $OFQS$ adapts the routing to the number of instances in a network providing a differentiation based on the requirements of the SG applications. We have conducted simulations and real test-bed experiments to test our approach in both best case scenario and a realistic environment respectively. Our results confirm the robustness and show the high performances of $OFQS$. It achieves significant improvement in terms of End-to-End delay, network lifetime and PDR while insuring a load balancing among the nodes compared to standard solutions.

Our second objective in this thesis has been to enable data reduction in a SG network managed by a WSN. The goal is to reduce the number of messages transmitted in the wireless network in order to save energy and bandwidth.

At first, we have considered data prediction techniques. We have highlighted the requirements that a data prediction algorithm must have in order to fulfill the SGs QoS needs. We have proposed a modification of the LMS prediction algorithm for WSN to adapt it to different applications with different characteristics as per a SG environment. We have tested our approach with real data traces of photovoltaic cells. We have trained the filter offline for one day with the data traces corresponding to each application in order to optimize the parameters that minimize the MSE. We have performed simulations considering one hop communication networks. Our numerical results show a better performance than LMS_VSS, a state of the art solution, in terms of $RMSE$ and percentage of data economy.

After data prediction, we have considered data aggregation and in-network processing. Here, we can aggregate heterogeneous data jointly from different applications.

Our proposition consists of a solution for data aggregation in SGs networks. It takes the QoS requirements of the SG heterogeneous applications into consideration by storing the packets in two different queues depending on their quality requirements. We expect to reduce the energy consumption in a SG, while respecting the corresponding delays and QoS requirements.

6.2 Perspectives

6.2.1 Short term

In this thesis, we have regularly recalled potential improvements and open perspectives concerning our work. Overall, it is noteworthy to mention that the different algorithms proposed may merit a deeper investigation and some optimization.

Starting from our objective function $OFQS$ for RPL, $mOFQS$ metric parameters α and β are fixed for the three instances. Here, further improvements should be made in order to automatically compute these parameters and optimize their values using machine learning techniques per example. One solution could be by training the network offline in order to compute the best values of these parameters. Additionally, our evaluation considered the global performance of multiple instances. However, it is important to examine each instance independently and study the impact of one instance over the other in a multiple instances network. *Are the instances going to be totally independent? Are the selected routes going to be independent from one instance to another?* Those are some of the important questions that need to be investigated and evaluated in order to quantify the impact. Moreover, the experiments should be stressed in order to investigate the impact of having a bigger number of instances running together on the routing mechanism. Furthermore, downward routing and point to point routing should also be tested using our OF in order to inspect the influence of having a two-way routing on the overall performance.

Secondly, concerning our data prediction solution LMS_MOD, our proposition is tested via numerical simulations considering a one hop communication network with no loss. It is important to test it in a realistic environment with interference and multi-hop communications. This may require few modifications in the algorithm like adding factors to consider the packet losses that may occur. Moreover, in LMS_MOD we optimize the parameters so as we minimize the MSE, further metrics should be taken into consideration and maybe combined in order to get the most out of the prediction algorithm.

Now concerning the data aggregation proposed solution, our theoretical approach has drawn the outline on how to implement our algorithm and what to expect as results. Anyhow, this algorithm should be certainly evaluated via simulations and real sensor experiment to validate this approach. Moreover, in our proposition, we left the decision of the aggregation function to the network administrator. It is useful to study and implement some aggregation functions that can be adapted to the type of traffic of a SG environment.

Hereafter, we propose one final yet important aspect to consider concerning our data reduction contribution. It consists of combining both the data prediction and data aggregation algorithms. We believe that with a fine tuning of the different parameters, this combination will be all beneficial for a SG environment with heterogeneous applications. Some applications where data redundancy is likely will enable performing data prediction and others will enable data aggregation, maximizing then the energy reduction and the available bandwidth of the wireless network.

6.2.2 Long term

As efficient as the routing of the information and data reduction techniques that we propose in the ubiquitous network of the SG, the multiplication of energy sources, storage and load control will only be possible if some level of autonomous decision is integrated into the ubiquitous network. Indeed, the massive integration of renewable energy resources (e.g., wind turbine, solar panels) with their sporadic performance, will make it difficult, even impossible, to be controlled only by human intervention. Moreover, determining the best solution to manage the network with the load control and storage possibilities will add more complexity.

In order to handle this complexity and the associated variability, the decision making must be "real-time", to avoid the interruption/failure of the electrical network. This could be done by integrating the possibility of automated decisions on the smart connected devices in the ubiquitous network of the SG.

The model that we propose to explore is that of a multi-level intelligence. Each sensor of a building would be able to decide independently the connection/disconnection from its energy source to the SG. This decision, and all the others taken by the sensors of the same logical level, would then be analyzed on the "upper" level of the network (here, the building) in order to take the appropriate decision and allow them to learn a possible better answer in the future. This cognitive schema can be applied on several levels: sensors in building, buildings in neighborhood, neighborhoods in town simultaneously and recursively.

The prediction model that we propose in this thesis, is a first step in this smart decision scenario, even though it involves the routing process more than the energy network driving. More advanced, yet lightweight models of machine learning have to be explored in order to allow "smarter" decision making. Taking into account the QoS requirements of the SGs and WSNs.

6.3 From A Smart Grid To A Smart City

The WSNs that help make up today's SGs fit right into today's Internet of Things driven economy. SGs are capable of controlling the grid and transmitting electricity more efficiently, for instance, but are also constantly gathering data about utility usages around the cities they help operate. For that, stakeholders and various actors are increasingly turning to SGs to power and manage the cities of the future [110].

Indeed, the approaches and algorithms that we proposed in this thesis are not specific to SGs. They are mostly suitable to any context with different applications on the same physical topology with different characteristics and QoS needs. This is all true for Smart Cities as well [111]. We can easily imagine that our QoS enabled proposition for SGs using RPL and multiple instances, can fit perfectly in a Smart City scenario. In fact, it is in the heart of the Smart City. Having a WSN with sensors spread across the city to collect SG related data and information can be used for Smart City applications (e.g., smart parking, smart environment). These applications may fit in one or more new instances of RPL. Similarly, for the data reduction algorithms proposed, the Smart City applications can use the same sensors and algorithms for sending their type of data simultaneously.

Consequently, a modernized grid will then offer huge advantages for the stake holders by giving them a broader vision, management and control of the "smarter" city while lowering the costs, as well as to the customers making their daily life more comfortable and convenient.

List Of Publications

Journal articles:

- **Multiple Instances QoS Routing In RPL: Application To Smart Grids**
  Jad Nassar, Matthieu Berthomé, Jérémy Dubrulle, Nicolas Gouvy, Nathalie Mitton and Bruno Quoitin
  Sensors 2018, 18, 2472.

Conference papers:

- **Heterogeneous data reduction in WSN: Application to Smart Grids**
  Jad Nassar, Karen Miranda, Nicolas Gouvy, Nathalie Mitton
  SMARTOBJECTS 2018, 4th Workshop on Experiences with the Design and Implementation of Smart Objects June 25, 2018 · Los Angeles, California, USA.

- **Prédiction de données différenciée pour les Smart Grids**
  Jad Nassar, Karen Miranda, Nicolas Gouvy, Nathalie Mitton
  CORES 2018, 3èmes Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication, May 2018, Roscoff, France.

- **QoS-compliant Data Aggregation for Smart Grids**
  Jad Nassar, Nicolas Gouvy, Nathalie Mitton
  ENERGY 2018, The Eighth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies, May 2018, Nice, France.

- **Towards Multi-instances QoS Efficient RPL for Smart Grids**
  Jad Nassar, Nicolas Gouvy, Nathalie Mitton
  PE-WASUN'17, The 14th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks. November 21–25, 2017, Miami, FL, USA.

- **Fonction objectif pour un RPL adapté aux Smart Grids**
  Jad Nassar, Nicolas Gouvy, Nathalie Mitton
  AlgoTel 2017, 19ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Quiberon, France.

Bibliography

[1] Faycal Bouhafs, Michael Mackay, and Madjid Merabti. "Links to the future: communication requirements and challenges in the smart grid". In: *IEEE Power and Energy Magazine* 10.1 (2012), pp. 24–32.

[2] Agustin Zaballos, Alex Vallejo, and Josep M Selga. "Heterogeneous communication architecture for the smart grid". In: *IEEE Network* 25.5 (2011).

[3] Xinghuo Yu, Carlo Cecati, Tharam Dillon, and M Godoy Simoes. "The new frontier of smart grids". In: *IEEE Industrial Electronics Magazine* 5.3 (2011), pp. 49–63.

[4] *U.S Energy Information Administration*. https://www.eia.gov/energyexplained/index.php?page=electricity_delivery. Accessed: 21-08-2018.

[5] *Electricity generation, transmission and distribution guides*. https://electrical-engineering-portal.com/download-center/books-and-guides/electricity-generation-t-d. Accessed: 21-08-2018.

[6] Vehbi C Gungor and Frank C Lambert. "A survey on communication networks for electric system automation". In: *Computer Networks, Elsevier* 50.7 (2006), pp. 877–897.

[7] Kaveh Razazian, Maher Umari, Amir Kamalizad, Victor Loginov, and Michael Navid. "G3-PLC specification for powerline communication: Overview, system simulation and field trial results". In: *International Symposium on Power Line Communications and its Applications (ISPLC)*. IEEE. 2010, pp. 313–318.

[8] Vehbi C Gungor, Bin Lu, and Gerhard P Hancke. "Opportunities and challenges of wireless sensor networks in smart grid". In: *IEEE Transactions on Industrial Electronics* 57.10 (2010), pp. 3557–3564.

[9] Kazem Sohraby, Daniel Minoli, and Taieb Znati. *Wireless sensor networks: technology, protocols, and applications*. John Wiley & Sons, 2007.

[10] BKW. *Smart Grid Taxonomy. A system view from a grid operator's perspective*. 2015. URL: http://www.bkw.ch/fileadmin/user_upload/3_Gemeinden_EVU/gem_smart_grid_systematik_en.pdf.

[11] Brandon Davito, Humayun Tai, and Robert Uhlaner. "The smart grid and the promise of demand-side management". In: *McKinsey on Smart Grid* 3 (2010), pp. 8–44.

[12] Mouna Rekik. "Routage géographique multi-chemin basé sur l'intelligence d'essaim pour réseaux de capteurs et d'actionneurs sans fil : application aux Smart Grids". PhD thesis. Univ. Lille 1, 2016.

[13] Nermin Suljanovic, Dzemo Borovina, Matej Zajc, Jasmin Smajic, and Aljo Mujcic. "Requirements for communication infrastructure in smart grids". In: *Energy Conference (ENERGYCON)*. IEEE. 2014.

[14] Emilio Ancillotti, Raffaele Bruno, and Marco Conti. "The role of communication systems in smart grids: Architectures, technical solutions and research challenges". In: *Computer Communications, Elsevier* 36.17-18 (2013), pp. 1665–1697.

[15] Tim Winter, Pascal Thuber, Anders Brandt, et al. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. RFC 6550. RFC Editor, 2012.

[16] Di Wang, Zhifeng Tao, Jinyun Zhang, Alhussein Abouzeid, et al. "RPL-based routing for advanced metering infrastructure in smart grid". In: *International Conference on Communications (ICC)*. IEEE. 2010.

[17] Gowdemy Rajalingham, Yue Gao, Quang-Dung Ho, and Tho Le-Ngoc. "Quality of service differentiation for smart grid neighbor area networks through multiple RPL instances". In: *Proceedings of the 11th symposium on QoS and security for wireless and mobile networks*. ACM. 2014, pp. 17–24.

[18] Emilio Ancillotti, Raffaele Bruno, and Marco Conti. "The role of the RPL routing protocol for smart grid communications". In: *Communications Magazine, IEEE* 51.1 (2013), pp. 75–83.

[19] Nancy Cam-Winget, J Hui, and D Popa. *Applicability Statement for the Routing Protocol for Low-Power and Lossy Networks (RPL) in Advanced Metering Infrastructure (AMI) Networks*. RFC 8036. RFC Editor, 2017.

[20] Nico Saputro, Kemal Akkaya, and Suleyman Uludag. "A survey of routing protocols for smart grid communications". In: *Computer Networks* 56.11 (2012), pp. 2742–2771.

[21] Kenneth C Budka, Jayant G Deshpande, Tewfik L Doumi, Mark Madden, and Tim Mew. "Communication network architecture and design principles for smart grids". In: *Bell Labs Technical Journal* 15.2 (2010), pp. 205–227.

[22] Usman Raza, Alessandro Camerra, Amy L Murphy, Themis Palpanas, and Gian Pietro Picco. "What does model-driven data acquisition really achieve in wireless sensor networks?" In: *International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2012, pp. 85–94.

[23] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. "Energy conservation in wireless sensor networks: A survey". In: *Ad hoc networks* 7.3 (2009), pp. 537–568.

[24] Elena Fasolo, Michele Rossi, Jorg Widmer, and Michele Zorzi. "In-network aggregation techniques for wireless sensor networks: a survey". In: *IEEE Wireless Communications* 14.2 (2007).

[25] Michel Grabisch, Jean-Luc Marichal, Radko Mesiar, and Endre Pap. "Aggregation functions: means". In: *Information Sciences* 181.1 (2011), pp. 1–22.

[26] Gabriel Martins Dias, Boris Bellalta, and Simon Oechsner. "A survey about prediction-based data reduction in wireless sensor networks". In: *ACM Computing Surveys (CSUR)* 49.3 (2016), p. 58.

[27] Babak Hassibi, Ali H Sayed, and Thomas Kailath. "H-infinity optimality of the LMS algorithm". In: *IEEE Transactions on Signal Processing* 44.2 (1996), pp. 267–280.

[28] Silvia Santini and Kay Romer. "An adaptive strategy for quality-based data reduction in wireless sensor networks". In: *Proceedings of the 3rd international conference on networked sensing systems (INSS)*. IEEE. 2006, pp. 29–36.

[29] Vehbi C Güngör, Dilan Sahin, Taskin Kocak, Salih Ergüt, Concettina Buccella, Carlo Cecati, and Gerhard P Hancke. "Smart grid technologies: communication technologies and standards". In: *IEEE transactions on Industrial informatics* 7.4 (2011), pp. 529–539.

[30] Murat Kuzlu, Manisa Pipattanasomporn, and Saifur Rahman. "Communication network requirements for major smart grid applications in HAN, NAN and WAN". In: *Computer Networks, Elsevier* 67 (2014), pp. 74–88.

[31] Wenye Wang, Yi Xu, and Mohit Khanna. "A survey on the communication architectures in smart grid". In: *Computer networks* 55.15 (2011), pp. 3604–3629.

[32] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. "A comparative study of LPWAN technologies for large-scale IoT deployment". In: *ICT Express* (2018).

[33] Bashar Alohali, Kashif Kifayat, Qi Shi, and William Hurst. "Group Authentication Scheme for Neighbourhood Area Networks (NANs) in Smart Grids". In: *Journal of Sensor and Actuator Networks* 5.2 (2016), p. 9.

[34] Matthew NO Sadiku, Mahamadou Tembely, and Sarhan M Musa. "Home Area Networks: A Primer". In: *International Journal of Advanced Research in Computer Science and Software Engineering* 7.5 (2017).

[35] Ye Yan, Yi Qian, Hamid Sharif, and David Tipper. "A survey on smart grid communication infrastructures: Motivations, requirements and challenges". In: *IEEE communications surveys & tutorials* 15.1 (2013), pp. 5–20.

[36] Weixiao Meng, Ruofei Ma, and Hsiao-Hwa Chen. "Smart grid neighborhood area networks: a survey". In: *IEEE Network* 28.1 (2014), pp. 24–32.

[37] Nico Saputro, Kemal Akkaya, and Suleyman Uludag. "A survey of routing protocols for smart grid communications". In: *Computer Networks, Elsevier* 56.11 (2012), pp. 2742–2771.

[38] US DoE. "Communications requirements of Smart Grid technologies". In: *US Department of Energy, Tech. Rep* (2010), pp. 1–69.

[39] V Cagri Gungor, Dilan Sahin, Taskin Kocak, Salih Ergut, Concettina Buccella, Carlo Cecati, and Gerhard P Hancke. "A survey on smart grid potential applications and communication requirements". In: *IEEE Transactions on industrial informatics* 9.1 (2013), pp. 28–42.

[40] *NIST Framework and Roadmap for Smart Grid Interoperability Standards*. Accessed 09 July 2018. URL: http://dx.doi.org/10.6028/NIST.SP.1108r3.

[41] U.S. DoE. *Advanced Metering Infrastructure and Customer Systems*. Accessed 09 July 2018. 2016. URL: https://www.energy.gov/sites/prod/files/2016/12/f34/AMI%20Summary%20Report_09-26-16.pdf.

[42] Daphne Mah, Peter Hills, Victor OK Li, and Richard Balme. *Smart grid applications and developments*. Springer, 2014.

[43] Axel Verdiere, Yuichi Igarashi, Thierry Lys, Cedric Lavenu, Jiazi Yi, Ulrich Herberg, Hiroki Satoh, Afshin Niktash, Thomas Clausen, and Justin Dean. *The lightweight on-demand ad hoc distance-vector routing protocol-next generation (LOADng)*. Internet-Draft. IETF Secretariat, 2016.

[44] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. *Ad hoc on-demand distance vector (AODV) routing*. RFC 3561. RFC Editor, 2003.

[45] Mališa Vučinić, Bernard Tourancheau, and Andrzej Duda. "Performance comparison of the RPL and LOADng routing protocols in a home automation scenario". In: *Wireless Communications and Networking Conference (WCNC)*. IEEE. 2013, pp. 1974–1979.

[46] Tadashige Iwao, Kenji Yamada, Masakazu Yura, Yuuta Nakaya, Alvaro A Cárdenas, Sung Lee, and Ryusuke Masuoka. "Dynamic data forwarding in wireless mesh networks". In: *International Conference on Smart Grid Communications (SmartGridComm)*. IEEE. 2010, pp. 385–390.

[47] Mouna Rekik, Nathalie Mitton, and Zied Chtourou. "Geographic greedy routing with aco recovery strategy graco". In: *International Conference on Ad-Hoc and Wireless Networks*. Springer. 2015, pp. 19–32.

[48] Stephen Dawson-Haggerty, Arsalan Tavakoli, and David Culler. "Hydro: A hybrid routing protocol for low-power and lossy networks". In: *International Conference on Smart Grid Communications (SmartGridComm)*. IEEE. 2010, pp. 268–273.

[49] Mouna Rekik, Nathalie Mitton, and Zied Chtourou. "QoS-aware routing for real-time and reliable wireless sensor network based Smart Grid NAN communications". In: *Smart World Congress*. IEEE. 2017.

[50] Emmanuel Baccelli, Oliver Hahm, Mesut Gunes, Matthias Wahlisch, and Thomas C Schmidt. "RIOT OS: Towards an OS for the Internet of Things". In: *Computer Communications Workshops (INFOCOM Workshops)*. IEEE. 2013, pp. 79–80.

[51] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. "Contiki-a lightweight and flexible operating system for tiny networked sensors". In: *International Conference on Local Computer Networks (LCN)*. IEEE. 2004.

[52] Omprakash Gnawali and Philip Levis. *The Minimum Rank with Hysteresis Objective Function*. RFC 6719. RFC Editor, 2012.

[53] Pascal Thubert. *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)*. RFC 6552. RFC Editor, 2012.

[54] Douglas SJ De Couto, Daniel Aguayo, John Bicket, and Robert Morris. "A high-throughput path metric for multi-hop wireless routing". In: *Wireless networks* 11.4 (2005), pp. 419–434.

[55] Philip Levis, Thomas Clausen, Jonathan Hui, Omprakash Gnawali, and J Ko. *The trickle algorithm*. RFC 6206. 2011.

[56] Emilio Ancillotti, Raffaele Bruno, and Marco Conti. "RPL Routing Protocol in Advanced Metering Infrastructures: an Analysis of the Unreliability Problems". In: *Sustainable Internet and ICT for Sustainability (SustainIT)*. IEEE. 2012, pp. 1–10.

[57] Hyung-Sin Kim, Jeongyeup Paek, and Saewoong Bahk. "QU-RPL: Queue utilization based RPL for load balancing in large scale industrial applications". In: *12th International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2015.

[58] Parag Kulkarni, Sedat Gormus, Zhong Fan, and Benjamin Motz. "A self-organising mesh networking solution based on enhanced RPL for smart metering communications". In: *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE. 2011, pp. 1–6.

[59] Jean-Philippe Vasseur, Mijeom Kim, Kris Pister, Nicolas Dejean, and Dominique Barthel. *Routing metrics used for path calculation in low-power and lossy networks*. RFC 6551. RFC Editor, 2012.

[60] Patrick Olivier Kamgueu, Emmanuel Nataf, Thomas Djotio Ndié, and Olivier Festor. *Energy-based routing metric for RPL*. Research Report. Inria, 2013.

[61] Piergiuseppe Di Marco, Carlo Fischione, George Athanasiou, and Prodromos-Vasileios Mekikis. "MAC-aware routing metrics for low power and lossy networks". In: *International Conference on Computer Communications (INFOCOM)*. IEEE. 2013.

[62] Sooyeol Yang, Youngmi Baek, Junhyung Kim, Keuchul Cho, and Kijun Han. "A routing metric for load balance in wireless mesh networks". In: *International Conference on Advanced Communication Technology. (ICACT)*. IEEE. 2009, pp. 1560–1565.

[63] Richard Draves, Jitendra Padhye, and Brian Zill. "Routing in multi-radio, multi-hop wireless mesh networks". In: *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM. 2004, pp. 114–128.

[64] Walid Khallef, Miklos Molnar, Abderrahim Benslimane, and Sylvain Durand. "Multiple constrained QoS routing with RPL". In: *International Conference on Communications (ICC)*. IEEE. 2017, pp. 1–6.

[65] Fredrik Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. "Cross-level sensor network simulation with cooja". In: *31st IEEE conference on Local computer networks*. IEEE. 2006, pp. 641–648.

[66] Silvia Capone, Riccardo Brama, Nicola Accettura, Domenico Striccoli, and Gennaro Boggia. "An Energy Efficient and Reliable Composite Metric for RPL Organized Networks". In: *International Conference on Embedded and Ubiquitous Computing (EUC)*. IEEE. 2014.

[67] Panagiotis Karkazis, Helen C Leligou, Lambros Sarakis, Theodore Zahariadis, Panagiotis Trakadas, Terpsichori H Velivassaki, and Christos Capsalis. "Design of primary and composite routing metrics for RPL-compliant wireless sensor networks". In: *Int. Conf. on Telecommunications and Multimedia (TEMU)*. IEEE. 2012.

[68] Olfa Gaddour, Anis Koubâa, Nouha Baccour, and Mohamed Abid. "OF-FL: QoS-aware fuzzy logic objective function for the RPL routing protocol". In: *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE. 2014.

[69] Patrick-Olivier Kamgueu, Emmanuel Nataf, and Thomas Ndie Djotio. "On design and deployment of fuzzy-based metric for routing in low-power and lossy networks". In: *40th Local Computer Networks Conference Workshops (LCN Workshops)*. IEEE. 2015, pp. 789–795.

[70] TG Harshavardhana, BS Vineeth, SVR Anand, and Malati Hegde. "Power control and cross-layer design of RPL objective function for low power and lossy networks". In: *10th International Conference on Communication Systems & Networks (COMSNETS)*. IEEE. 2018, pp. 214–219.

[71] Mai Banh, Hieu Mac, Nam Nguyen, Kieu-Ha Phung, Nguyen Huu Thanh, and Kris Steenhaut. "Performance evaluation of multiple RPL routing tree instances for Internet of Things applications". In: *International Conference on Advanced Technologies for Communications (ATC)*. IEEE. 2015, pp. 206–211.

[72] Nguyen Thanh Long, Marie-Paule Uwase, Jacques Tiberghien, and Kris Steenhaut. "QoS-aware cross-layer mechanism for multiple instances RPL". In: *International Conference on Advanced Technologies for Communications (ATC)*. IEEE. 2013, pp. 44–49.

[73] Jovan Radak, Nathalie Mitton, and David Simplot-Ryl. "Using Battery Level as Metric for Graph Planarization". In: *International Conference on Ad Hoc Networks and Wireless (AdHocNow)*. 2011, pp. 58–71.

[74] Joakim Eriksson, Adam Dunkels, Niclas Finne, Fredrik Osterlind, and Thiemo Voigt. "Mspsim–an extensible simulator for msp430-equipped sensor boards". In: *Proceedings of the European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session*. Vol. 118. 2007.

[75] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne. "FIT IoT-LAB: A large scale open experimental IoT testbed". In: *2nd World Forum on Internet of Things (WF-IoT)*. IEEE. 2015, pp. 459–464.

[76] Eric Fleury, Nathalie Mitton, Thomas Noel, and Cédric Adjih. "FIT IoT-LAB: The largest iot open experimental testbed". In: *ERCIM News* 101 (2015), p. 4.

[77] Christian Rohner, Laura Marie Feeney, and Per Gunningberg. "Evaluating battery models in wireless sensor networks". In: *International Conference on Wired/Wireless Internet Communication*. Springer. 2013, pp. 29–42.

[78] Chulsung Park, Kanishka Lahiri, and Anand Raghunathan. "Battery discharge characteristics of wireless sensor nodes: An experimental analysis". In: *2nd Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*. IEEE. 2005, pp. 430–440.

[79] Jeremy Dubrulle. *Master thesis*. https://jeremydubrulle.com/master_thesis.pdf. Accessed 21-08-2018.

[80] Simon Haykin and Bernard Widrow. *Least-mean-square adaptive filters*. Vol. 31. John Wiley & Sons, 2003.

[81] Hans-Jürgen Butterweck. "A steady-state analysis of the LMS adaptive algorithm without use of the independence assumption". In: *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 2. IEEE. 1995, pp. 1404–1407.

[82] Siddhartha Bhandari, Neil Bergmann, Raja Jurdak, and Branislav Kusy. "Time Series Data Analysis of Wireless Sensor Network Measurements of Temperature". In: *Sensors* 17.6 (2017), p. 1221.

[83] Karen Miranda, Tahiry Razafindralambo, and Victor Ramos. "Using efficiently autoregressive estimation in wireless sensor networks". In: *International Conference on Computer, Information and Telecommunication Systems (CITS)*. IEEE. 2013, pp. 1–5.

[84] Yuzhong Jiao, Rex YP Cheung, Winnie WY Chow, and Mark PC Mok. "A novel gradient adaptive step size LMS algorithm with dual adaptive filters". In: *35th International Conference of Engineering in Medicine and Biology (EMBC)*. IEEE. 2013, pp. 4803–4806.

[85] Peijie Wang, Pooi Yuen Kam, and Meng Wah Chia. "A novel automatic step-size adjustment approach in the LMS algorithm". In: *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)*. IEEE. 2009, pp. 867–871.

[86] Biljana Stojkoska, Dimitar Solev, and Danco Davcev. "Data prediction in WSN using variable step size LMS algorithm". In: *Proceedings of the 5th International Conference on Sensor Technologies and Applications*. 2011.

[87] Dariusz Bismor, Krzysztof Czyz, and Zbigniew Ogonowski. "Review and comparison of variable step-size LMS algorithms". In: *International Journal of Acoustics and Vibration* 21.1 (2016), pp. 24–39.

[88] S.S. Haykin. *Adaptive Filter Theory*. Prentice-Hall information and system sciences series. Prentice Hall, 2002. ISBN: 9780130901262. URL: https://books.google.fr/books?id=eMcZAQAAIAAJ.

[89] Jyoti Dhiman, Shadab Ahmad, and Kuldeep Gulia. "Comparison between Adaptive filter Algorithms (LMS, NLMS and RLS)". In: *International Journal of Science, Engineering and Technology Research (IJSETR)* 2.5 (2013), pp. 1100–1103.

[90] Karen Miranda and Victor Ramos. "Improving data aggregation in Wireless Sensor Networks with time series estimation". In: *IEEE Latin America Transactions* 14.5 (May 2016), pp. 2425–2432.

[91] David Jager and Afshin Andreas. *NREL National Wind Technology Center (NWTC): M2 Tower; Boulder, Colorado (Data)*. Tech. rep. DA-5500-56489. National Renewable Energy Lab.(NREL), Golden, CO (United States), 1996. URL: http://dx.doi.org/10.5439/1052222.

[92] Paulo Jesus, Carlos Baquero, and Paulo Sérgio Almeida. "A survey of distributed data aggregation algorithms". In: *IEEE Communications Surveys & Tutorials* 17.1 (2015), pp. 381–404.

[93] Wendi Beth Heinzelman. "Application-Specific protocol architectures for wireless networks". PhD thesis. Massachusetts Institute of Technology, 2000.

[94] Ossama Younis and Sonia Fahmy. "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks". In: *Transactions on mobile computing* 3.4 (2004), pp. 366–379.

[95] Ramesh Rajagopalan and Pramod K Varshney. "Data aggregation techniques in sensor networks: A survey". In: *IEEE Communications Surveys and Tutorials* 8 (2006), pp. 48–63.

[96] Min Ding, Xiuzhen Cheng, and Guoliang Xue. "Aggregation tree construction in sensor networks". In: *58th Vehicular Technology Conference (VTC)*. Vol. 4. IEEE, 2003, pp. 2168–2172.

[97] Hüseyin Özgür Tan and Ibrahim Körpeoğlu. "Power efficient data gathering and aggregation in wireless sensor networks". In: *ACM Sigmod Record* 32.4 (2003), pp. 66–71.

[98] Márk Jelasity. "Gossip-based Protocols for Large-scale Distributed Systems". PhD thesis. szte, 2013.

[99] David Kempe, Alin Dobra, and Johannes Gehrke. "Gossip-based computation of aggregate information". In: *44th Annual IEEE Symposium on Foundations of Computer Science*. IEEE. 2003, pp. 482–491.

[100] J-Y Chen, Gopal Pandurangan, and Dongyan Xu. "Robust computation of aggregates in wireless sensor networks: distributed randomized algorithms and analysis". In: *Fourth International Symposium on Information Processing in Sensor Networks (IPSN)*. IEEE. 2005, pp. 348–355.

[101] Laukik Chitnis, Alin Dobra, and Sanjay Ranka. "Aggregation methods for large-scale sensor networks". In: *ACM Transactions on Sensor Networks (TOSN)* 4.2 (2008), p. 9.

[102] Toshichika Shiobara, Peter Palensky, and Hiroaki Nishi. "Effective metering data aggregation for smart grid communication infrastructure". In: *41st Annual Conference of the Industrial Electronics Society (IECON)*. IEEE. 2015, pp. 002136–002141.

[103] Faycal Bouhafs and Madjid Merabti. "Managing communications complexity in the smart grid using data aggregation". In: *7th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE. 2011, pp. 1315–1320.

[104] Babak Karimi, Vinod Namboodiri, and Murtuza Jadliwala. "Scalable meter data collection in smart grids through message concatenation". In: *IEEE Transactions on Smart Grid* 6.4 (2015), pp. 1697–1706.

[105] Forkan Uddin. "Energy-Aware Optimal Data Aggregation in Smart Grid Wireless Communication Networks". In: *IEEE Transactions on Green Communications and Networking* 1.3 (2017), pp. 358–371.

[106] Ting-Chu Lee and Zsehong Tsai. "On the Capacity of Smart Grid Wireless Backhaul With Delay Guarantee and Packet Concatenation". In: *IEEE Systems Journal* (2015), pp. 2628–2639.

[107] Peyman Teymoori, Mehdi Kargahi, and Nasser Yazdani. "A real-time data aggregation method for fault-tolerant wireless sensor networks". In: *Proceedings of the 27th Annual Symposium on Applied Computing*. ACM. 2012, pp. 605–612.

[108] Tarek Abdelzaher, Tian He, and John Stankovic. "Feedback control of data aggregation in sensor networks". In: *43rd Conference on Decision and Control (CDC)*. Vol. 2. IEEE. 2004, pp. 1490–1495.

[109] Thomas Watteyne, M Palattella, and L Grieco. *Using IEEE 802.15. 4e time-slotted channel hopping (TSCH) in the internet of things (IoT): Problem statement*. RFC 7554. RFC Editor, 2015.

[110] B Morvaj, L Lugaric, and S Krajcar. "Demonstrating smart buildings and smart grid features in a smart energy city". In: *3rd International Youth Conference on Energetics (IYCE).* IEEE. 2011, pp. 1–8.

[111] Eiman Al Nuaimi, Hind Al Neyadi, Nader Mohamed, and Jameela Al-Jaroodi. "Applications of big data to smart cities". In: *Journal of Internet Services and Applications* 6.1 (2015), p. 25.