

A System-Level Methodology for the Design and Deployment of Reliable Low-Power Wireless Sensor Networks

by

Oussama BRINI

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR A MASTER'S
DEGREE WITH THESIS IN ELECTRICAL ENGINEERING
M. A. SC.

MONTREAL, APRIL 2ND, 2019

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC



Oussama Brini, 2019



This Creative Commons licence allows readers to download this work and share it with others as long as the author is credited. The content of this work can't be modified in any way or used commercially.

BOARD OF EXAMINERS (THESIS M.Sc.A.)
THIS THESIS HAS BEEN EVALUATED
BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Frederic Nabki, Thesis Supervisor
Department of Electrical Engineering at École de technologie supérieure

Mr. Dominic Deslandes, Thesis Co-supervisor
Department of Electrical Engineering at École de technologie supérieure

Mr. Ricardo Izquierdo, President of the Board of Examiners
Department of Electrical Engineering at École de technologie supérieure

Mr. Georges Kaddoum, Member of the jury
Department of Electrical Engineering at École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED
IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC
ON MAI 1ST, 2019
AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGMENT

First, I would like to thank my advisors Prof. Frederic Nabki and Prof. Dominic Deslandes for their support, guidance, time, and ideas throughout all the period of the program. Both advisors empowered me to push the limits of my knowledge and helped me to shape my ideas.

Special thanks to my parents and two brothers for their advices, constant support, patience and encouragement over the years. I would also like to take this opportunity to thank all my big family members and friends for their amazing support throughout my studies.

Finally, I wish to thank each member of the committee for agreeing to devote time to examining this work and for their helpful comments and suggestions.

Méthodologie pour la conception et le déploiement de réseaux de capteurs sans fil fiables et de faible puissance

Oussama BRINI

RÉSUMÉ

Les applications innovantes de l'internet des objets (*Internet of Things*, IoT) avec des exigences strictes en termes de performances et de consommation d'énergie et pour lesquelles la collecte agile de données est primordiale sont stimulantes. Les réseaux de capteurs sans fil (*Wireless Sensor Networks*, WSN) représentent une solution prometteuse car ils peuvent être facilement déployés pour détecter, traiter et transmettre des données. Le grand nombre de nœuds de capteurs (*Sensor node*, SN) composant un réseau de capteurs sans fil devrait être autonome, la durée de vie d'un nœud étant dictée par la taille de la batterie. La taille du nœud étant essentiel dans divers cas d'utilisation tels que l'automatisation de bâtiments et d'industries, il est prioritaire de réduire la consommation d'énergie tout en garantissant la disponibilité du réseau. De plus, les techniques de récupération d'énergie (*Energy Harvesting*, EH) sont de plus en plus considérées comme une solution prometteuse pour construire un nœud entièrement autonome et prolonger sa durée de vie. Dans le processus de construction d'un nœud et en l'absence d'une méthodologie claire et complète, le concepteur peut facilement prendre des décisions non fondées sur les bons composants matériels, leur configuration et les techniques de communication de données fiables telles que la requête automatique de répétition (*Automatic Repeat Request*, ARQ) et correction d'erreur directe (*Forward Error Correction*, FEC). Dans cette thèse, une méthodologie pour mieux optimiser la conception, la configuration et le déploiement de WSN fiables à très basse consommation est proposée. Des modèles complets et réalistes d'énergie et d'affaiblissement de propagation (*Path Loss* PL) du nœud de capteur sont également établis. À l'aide d'estimations et de mesures, il est montré que, conformément à la méthodologie proposée, le concepteur peut explorer en profondeur l'espace de conception, sélectionner et configurer de manière plus optimale les composants commerciaux (*Commercial Off-The-Shelf*, COTS) disponibles sur le marché et déployer efficacement un réseau de capteurs sans fil.

Mots clés: réseaux de capteurs sans fil, méthodologie de conception, modèle énergétique, affaiblissement de propagation, faible consommation, conception système, couche de liaison, requête automatique de répétition, correction d'erreur directe.

A System-Level Methodology for the Design and Deployment of Reliable Low-Power Wireless Sensor Networks

Oussama BRINI

ABSTRACT

Innovative Internet of Things (IoT) applications with strict performance and energy consumption requirements and where the agile collection of data is paramount are rousing. Wireless sensor networks (WSN) represent a promising solution as they can be easily deployed to sense, process, and forward data. The large number of Sensor Nodes (SNs) composing a WSN are expected to be autonomous, with a node's lifetime dictated by the battery's size. As the form factor of the SN is critical in various use cases such as industrial and building automation, minimizing energy consumption while ensuring availability becomes a priority. Moreover, energy harvesting techniques are increasingly considered as a viable solution for building an entirely green SN and prolonging its lifetime. In the process of building a SN and in the absence of a clear and well-rounded methodology, the designer can easily make unfounded decisions about the right hardware components, their configuration and data reliable data communication techniques such as automatic repeat request (ARQ) and forward error correction (FEC). In this thesis, a methodology to better optimize the design, configuration and deployment of reliable ultra-low power WSNs is proposed. Comprehensive and realistic energy and path-loss (PL) models of the sensor node are also established. Through estimations and measurements, it is shown that following the proposed methodology, the designer can thoroughly explore the design space and make most favorable decisions when choosing commercial off-the-shelf (COTS) components, configuring the node, and deploying a reliable and energy-efficient WSN.

Keywords: wireless sensor networks, design methodology, energy model, path-loss, link layer, low-power, system-level design, automatic repeat request, forward error correction.

TABLE OF CONTENTS

	Page
CHAPTER 1 INTRODUCTION	1
1.1 Applications of Wireless Sensor Networks and Their impact.....	1
1.1.1 Wireless Sensor Networks Uses Cases in Smart Cities	2
1.1.2 Economic and Social impact.....	4
1.2 Motivation.....	5
1.3 Main Contributions	5
1.4 Thesis Outline	6
CHAPTER 2 THEORETICAL BACKGROUND.....	7
2.1 Design Challenges	7
2.1.1 Energy Consumption and Network Lifetime Maximization	8
2.1.2 Meeting quality of service (QoS) requirements.....	9
2.2 Bit and Packet Error Mitigation Techniques	11
2.2.1 Packet Retransmission Techniques.....	11
2.2.1.1 Blind retransmissions (BR).....	11
2.2.1.2 ARQ-Based Retransmissions.....	12
2.2.2 Forward Error Correction (FEC)	14
2.3 Popular Low-Power Wireless Communication Standards.....	15
2.3.1 Bluetooth.....	15
2.3.2 IEEE 802.15.4.....	17
CHAPTER 3 LITERATURE REVIEW	21
3.1 Energy Consumption Modeling and Estimation in WSN.....	21
3.2 Path Loss Modeling	23
3.3 Energy cost versus QoS trade-off	25
CHAPTER 4 SYSTEM-LEVEL DESIGN METHODOLOGY	29
4.1 Sensor Node Energy Model.....	29
4.1.1 Modeling Framework.....	30
4.1.2 Energy Model Parameters.....	31
4.1.3 Analytical Energy Model.....	32
4.1.3.1 Micro-Controller Unit (MCU)	33
4.1.3.2 Wireless Transceiver and Sensor	39
4.1.4 Sensor Node Energy per Measurement.....	42
4.2 Outdoor Measurements and Wireless Link Characterization	44
4.2.1 Ambient Noise Density Measurements	46
4.2.2 Communication Range Outdoor Measurements.....	49
4.2.2.1 Suburban Area	49
4.2.2.2 Urban Area.....	50
4.2.2.3 Experimental Results	51
4.2.3 Path Loss Model	52

4.3	Data Rate Maximization	54
4.4	Energy-Reliability-Latency Trade-Off	55
4.4.1	Current Consumption Profile of Different Data Transfer Schemes.....	56
4.4.2	Packet Success Probability	57
4.4.2.1	Convolutional FEC	58
4.4.2.2	Blind Retransmissions (BR)	60
4.4.2.3	ARQ Retransmissions.....	62
4.5	Resulting Design Methodology and Case Studies	65
4.5.1	Design Flow and Methodology	65
4.5.2	Case Studies.....	66
4.5.2.1	99% Reliability Target.....	67
4.5.2.2	99.999% Reliability Target.....	70
4.6	Conclusion and Discussion	72
CHAPTER 5	CONCLUSION AND FUTURE WORK	73
5.1	Conclusion	73
5.2	Future Work.....	75
ANNEX I	PUBLISHED CONFERENCE PAPER.....	77
ANNEX II	C CODE: BME280 SENSOR DATA COMMUNICATION.....	83
ANNEX III	SENSOR NODE MODEL USING STATEFLOW	87
ANNEX IV	NOISE DENSITY MEASUREMENT PROCEDURE	89
BIBLIOGRAPHY	91

LIST OF TABLES

	Page
Table 2.1	Latency requirement of WSN applications.....9
Table 2.2	Reliability requirement of smart grid applications10
Table 4.1	Energy model parameters32
Table 4.2	Current consumption of two different MCUs at 26MHz.....33
Table 4.3	A survey on low-power COTS microcontrollers34
Table 4.4	Accurate current consumption estimation using CM.....34
Table 4.5	A survey on low-power COTS Sub-1 GHz transceivers40
Table 4.6	Example explaining how to estimate the achievable data rate66

LIST OF FIGURES

	Page
Figure 1.1	WSN applications in smart cities2
Figure 1.2	WSN connection to data base and cloud platforms3
Figure 2.1	WSN protocol stack7
Figure 2.2	Packet stream example of the BR operation12
Figure 2.3	Packet stream example of the SAW-ARQ operation.....12
Figure 2.4	Packet stream example of the <i>L</i> -REP-ACK.....13
Figure 2.5	Measured packet error rate (PER) curves for a convolutional (FEC) code of rate=3/4, 1/2 and 1/3 compared with uncoded packets' transmission15
Figure 4.1	The three main modeled components of an SN29
Figure 4.2	A Power/Energy consumption assessment framework based on Stateflow/Simulink30
Figure 4.3	A general sensor node energy model using Stateflow charts31
Figure 4.4	Current consumption measurement setup of the CC1310 wireless MCU while performing a point-to-point communication of an internal temperature sensor data and running the TI 15.4 network stack35
Figure 4.5	Clear representation of the connections35
Figure 4.6	Current consumption profile of the CC1310 wireless MCU on the transmitter's side36
Figure 4.7	Processing-time estimation using the CM/MHz figure (a) when the FPU is disabled and (b) when it is enabled38
Figure 4.8	Measurement setup for transceiver and sensor current measurement39
Figure 4.9	Current consumption breakdown of the sensor, transmitter, and receiver during one measurement41
Figure 4.10	Estimated versus measured SN's energy per measurement.....43

Figure 4.11	Energy consumption per measurement at 50 kbps	43
Figure 4.12	Energy consumption per measurement at 500 kbps	44
Figure 4.13	Communication range measurement setup	44
Figure 4.14	Background noise samples measured in three distribution environments. (a) Power grid distribution substation. (b) Low-voltage transformer. (c) Distribution lines in a residential area	46
Figure 4.15	Relative frequency distributions of the background noise strengths measured in three distribution environments. (a) Power grid distribution substation (b) Low-voltage transformer. (c) Distribution lines in a residential area	47
Figure 4.16	Measurement setup of the ambient noise.....	48
Figure 4.17	Packet format used in outdoor measurements	49
Figure 4.18	Google satellite image of the field measurement setup in the suburban area.....	50
Figure 4.19	Google satellite image of the field measurement setup in the urban area ..	50
Figure 4.20	BER, PER, and RSSI field measurements in (a) the urban and (b) the suburban areas.....	51
Figure 4.21	The measured path-loss versus the free-space and estimated ones in the (a) urban and (b) suburban areas.....	53
Figure 4.22	BER performance of GFSK.....	55
Figure 4.23	Current consumption profile of the transmitter and the receiver when using different data transfer schemes: (a) simple transmissions, (b) FEC, (c) two BR, and (d) a SAW-ARQ protocol (also referred to as 1-Rep- ACK in this work).....	56
Figure 4.24	MCU energy consumption per measurement for different data transfer schemes	57
Figure 4.25	Impact of the free distance dm on the packet success probability	58
Figure 4.26	Impact of the β_{free} parameter on the packet success probability.....	60
Figure 4.27	Impact of the number of blind transmission attempts on the packet success probability	61

Figure 4.28	Impact of the number of 1-Rep-ACK transmission attempts on the packet success probability.....	63
Figure 4.29	Packet success probability at the j th packet.....	64
Figure 4.30	SN design flow graph.....	65
Figure 4.31	Meeting 99% reliability target by using FEC, BR, and 2-Rep-ACK retransmissions ($L = 2$)	68
Figure 4.32	Meeting 99% reliability target by using FEC, BR, and 1-Rep-ACK retransmissions ($L = 1$)	68
Figure 4.33	(a) Energy consumption per measurement and (b) system latency for 99% reliability target.....	69
Figure 4.34	Meeting 99.999% reliability target by using BR and 4-Rep-ACK retransmissions.....	70
Figure 4.35	Meeting 99.999% reliability target by using BR and 3-Rep-ACK retransmissions.....	71
Figure 4.36	(a) Energy consumption per measurement and (b) system latency for 99.999% reliability target.....	71

LIST OF ABBREVIATIONS

IoT	Internet of Things
WSN	Wireless Sensor Network
SN	Sensor Node
ARQ	Automatic Repeat Request
SAW-ARQ	Stop-And-Wait Automatic Repeat Request
FEC	Forward Error Correction
CM	CoreMark
EEMBC	Embedded Microprocessor Benchmark Consortium
PL	Path Loss
QoS	Quality of Service
RTOS	Real Time Operating System
MAC	Medium Access Control
LOS	Line-Of-Sight
NLOS	Non-Line-Of-Sight
COTS	Commercial Off-The-Shelf
MCU	Micro-Controller Unit
CPU	Central Processing Unit
I2C	Inter-Integrated Circuit
RF	Radio Frequency
CRC	Cyclic Redundancy Check
DUT	Device under Test
MIPS	Million Instructions per Second

XX

FPU	Floating Point Unit
BLE	Bluetooth Low Energy
WMCU	Wireless Micro-Controller Unit
BER	Bit Error Rate
PER	Packet Error Rate
RSSI	Received Signal Strength
SINR	Signal-to-Interference-plus-Noise Ratio
GFSK	Gaussian Frequency Shift Keying
BSC	Binary Symmetric Channel
ACK	Acknowledgement
NACK	Non-Acknowledgement
SC	Smart City
E2E	End-to-End
LL	Link Layer
UAV	Unmanned Aerial Vehicle
TRPL	Two-Ray Path Loss
URLLC	ultra-reliable and low-latency communication
ISM	Industrial, Scientific and Medical

LIST OF SYMBOLS

V_{WMCU}	Wireless MCU operating voltage
V_{SENS}	Sensor operating voltage
D_R	Data rate
f_{MCU}	MCU operating frequency
I_{TRX}^{STATE}	Transceiver current consumption in each state
I_{MCU}^{STATE}	MCU current consumption in each state
I_{MCU}^{PHP}	MCU peripheral current consumption
l	Packet length
S_{REF}	Reference CM score
f_{REF}	Reference operating frequency
t_{PROC_REF}	Reference processing time
S_{MCU}	MCU CM score
T_{OVS}	Temperature oversampling factor
H_{OVS}	Humidity oversampling factor
P_{OVS}	Pressure oversampling factor
I_{SENS}^{STATE}	Sensor current consumption in each state
$t_{PROCESS}$	MCU processing time
E_{MCU}	MCU energy consumption per measurement
$I_{CoreMark}$	MCU current consumption when running CM
E_{TRX}	Transceiver energy consumption
I_{SENS}	Sensor current consumption
E_{SENS}	Sensor energy consumption
t_{TRX}	Transceiver active time
I_{TX}	Transmitter active current consumption
I_{RX}	Receiver active current consumption
t_{SENS}	Sensor active time
I_{DDT}	Temperature measurement current consumption
I_{DDP}	Pressure measurement current consumption

I_{DDH}	Humidity measurement current consumption
N_p	Total number of packets
N_{pNOK}	Corrupted packets
N	Number of payload bits
$RSSI_i$	Received signal strength indicator
$RSSI_{avg}$	Average received signal strength indicator
i	Packet index
N_A	Ambient noise
k_b	Boltzmann constant
T	Ambient temperature
ΔN	Noise density due to interfering emissions in the same frequency band
Hz	Hertz
dB/Hz	Decibel per hertz
B	Bandwidth
P_{NOISE}	Noise power
log	Decimal logarithm
%	Percentage
A	Ampere
CM/MHz	CoreMark per Mega Hertz
GHz	Giga Hertz
s	Second
Kbps	Kilobit per second
dB	Decibel
J	Joule
PL	Path-loss
PL_{exp}	Measured path-loss
d	Distance
λ	Wave length
π	Pi
h_t	Transmitter height

h_r	Receiver height
l_f	Fitting coefficient
P_t	Transmitter output power
P_r	Received power
N_{sys}	System's noise power density
N_A	Ambient noise density
N_{fg}	Noise figure of the receiver
E_b	Energy per bit
P	Packet success probability
P_e	Bit error probability
N	Payload length
n	Encoded payload length
k	Original payload length before coding
K	constraint length
d_m	Free distance
r	Code rate
G_{FEC}	Coding gain
β_{free}	The total number of non-zero information bits of all paths with a weight of d_m
p_e^{FEC}	Bit error probability when using FEC
P_{FEC}	Packet success probability when using FEC
t_{TRX}^{FEC}	Active transceiver time when using coded packets
l^{FEC}	Packet length when using FEC
E_{TRX}^{FEC}	Transceiver energy consumption when using FEC
R	Transmission times
h	Successful packet transmissions
S	Random variable
\mathbb{N}	Set of natural numbers
f	Probability mass function
pr	Probability

$t_{\text{TRX}}^{\text{R}}$	Active transceiver time when using blind retransmissions
$E_{\text{TRX}}^{\text{R}}$	Transceiver energy consumption when using blind retransmissions
I_{id}	Idle mode current
t_{id}	Time spent in idle mode
P_{ef}	Bit error probability of the feedback channel
$P_{\text{R}}^{\text{ARQ}}$	SAW-ARQ packet success probability
$P_{\text{OUT}}^{\text{ARQ}}$	SAW-ARQ outage probability
j	Transmission attempt index
P_{f}	Packet success probability of ACK/NACK packets
N_{f}	ACK/NACK packets payload length
$P_{\text{j}}^{\text{ARQ}}$	SAW-ARQ packet success probability at the j^{th} transmission attempt
$t_{\text{TRX}}^{\text{ARQ}}$	Active transceiver time when using SAW-ARQ retransmissions
$E_{\text{TRX}}^{\text{ARQ}}$	Transceiver energy consumption when using SAW-ARQ retransmissions
I_{sb}	Standby mode current
t_{sb}	Time spent in standby mode
t_{sys}	System's latency
E_{sys}	System's energy consumption per measurement

CHAPTER 1

INTRODUCTION

1.1 Applications of Wireless Sensor Networks and Their impact

Innovative internet of things (IoT) applications with strict performance and energy consumption requirements and where the agile collection of data is paramount are arising. Wireless sensor networks (WSN) represent a promising solution as they can be easily deployed to collect, process, and forward data at a sampling rate required by the application (i.e. every 500 ms). As WSN are low cost and composed of easy to deploy battery-operated devices, they are increasingly being deployed in a broad range of applications such as smart cities (SC) (G. Daniel Costa, 2019; A. Adonay Veiga, 2019), home automation (Oscar Blanco-Novoa, 2018; C. Leech, 2017), industrial automation (F. Dobslaw, 2016; P. Kong, 2017; L. P. I. Ledwaba, 2018; P. Sommer, 2018; W. Sun, 2018; L. D. Xu, 2014), and precision agriculture (M. Bacco, 2018; C. Konstantopoulos, 2016; M. Srbinovska, 2017). This study focuses on the SC use case where WSN are deployed in order to generate data and supply information that is useful for an optimal management of assets and resources in both urban and suburban areas. Therefore, the methodology presented in this thesis can be applied to design wireless sensor nodes which are intended to operate in a duty-cycled manner where the main components would switch between active and sleep states in order to save energy.

In 2014, the administration of the City of Montreal has expressed its desire that Montreal become a leader among other smart cities such as Oslo, Barcelona, and Singapore. To this end, the City of Montreal has decided to develop the "Montréal, Smart City and Digital 2017" strategy and created the Office of the Smart and Digital City. Moreover, as envisioned in (Zoya Sodhi, 2018), smart cities can and should be built through the collaboration of the civil society, academics, the private sector, and public officials. This work is a step towards achieving that goal.

1.1.1 Wireless Sensor Networks Uses Cases in Smart Cities

In 2014, the Office of the Smart and Digital City of Montreal started holding civic chats in order to hear different points of views about feasible applications and where integrated and networked intelligent systems can be deployed and create value. Figure 1.1 illustrates the smart city concept and where WSN can be deployed.



Figure 1.1 WSN applications in smart cities
Adapted from József Balázs-Hegedűs (2019)

The (<http://villeintelligente.montreal.ca>) website contains a long list of the suggested ideas and use cases of the aforementioned systems including, but not limited to:

- Establishment of an autonomous and reliable emergency communications network: In emergency situations, information should be accessible to citizens in the shortest possible time.
- Crowdsourcing data on the habits of public transport in order to enhance transportation fluidity.
- Smart parking: Monitoring of parking spaces availability in the city.

- Sending real time pollution and smog alerts to residents in a specific area through text messages or visual indicators.
- Smart road: displaying real time warning messages for drivers according to climate and road conditions.
- Structural health monitoring of buildings, bridges, historical monuments, etc.
- Water or gas leakage monitoring.
- Real time urban noise and sound monitoring.

Clearly, WSN play a key role in achieving all of these goals. They are being increasingly deployed in a broad range of smart urban and suburban applications (G. Daniel Costa, 2019). This is due to the fact that WSNs are low cost and composed of easy to deploy battery-operated devices allowing the collection and forwarding of data in a reliable manner. In a next step, software and big data analytics have the potential to provide finer-grained, wider-scale, real-time understanding and control of urban and suburban environments (Rob Kitchin, 2014).

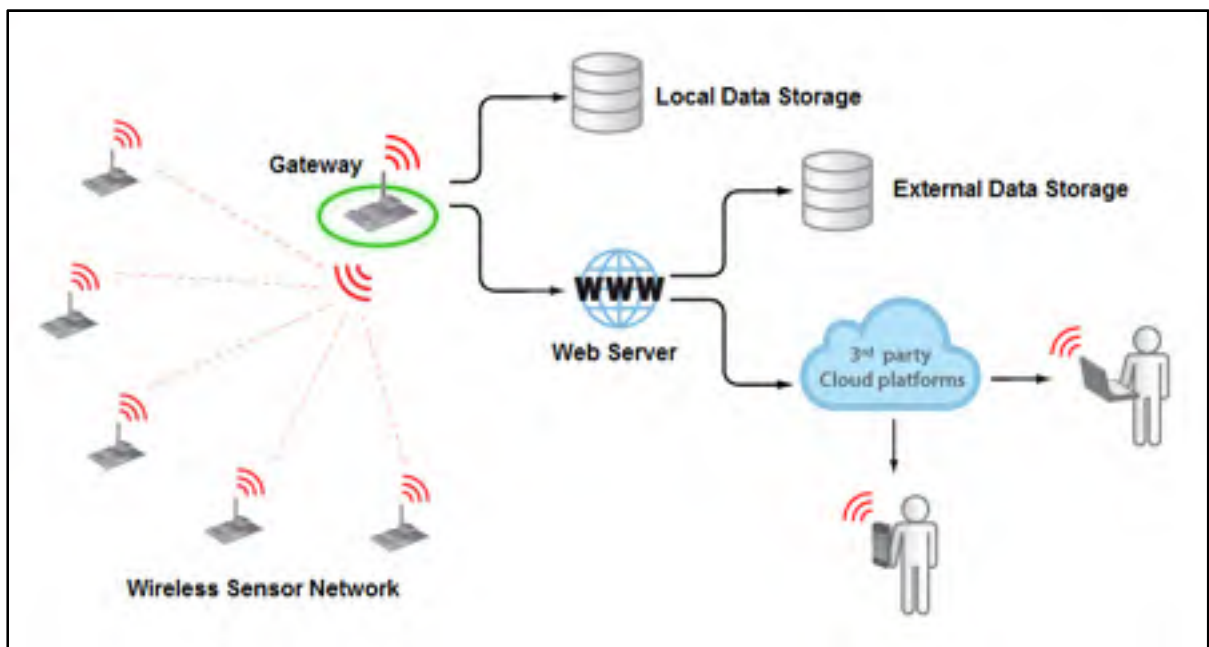


Figure 1.2 WSN connection to data base and cloud platforms
Adapted from Libelium (2012)

Figure 1.2 demonstrates how WSN can be connected to data bases and the cloud. A central node called gateway collects the sensor data from the wireless network and forwards them to a cloud application or a data storage device. A cloud platform service provider supports the cloud application services for cloud connectivity and visualization of the sensor node data.

1.1.2 Economic and Social impact

During the last two decades, and mainly due to substantial technological advancements, socio-economic requirements, and the arising of new environmental challenges, metropolitan city-regions are increasingly devoting efforts and resources to develop information and communication technology systems to find better solutions (Rob Kitchin, 2014). The smart city concept has the potential, once implemented, to foster sustainability, social inclusion, and the efficient use of natural resources and infrastructure (A. Adonay Veiga, 2019). Moreover, from a quality of life point of view, citizens and local communities can become more satisfied by the governance of their city when its development is in line with their needs.

Montreal city is a large metropolitan area and is Canada's second-most populous city. Moreover, the great majority of the civil society of Montreal is immersed in technology. Furthermore, the city is bursting with a wealth of useful data that can, once made available publicly and used to their full potential, simplify the daily life of citizens. Moreover, the city is well-known for its vibrant digital technology sector as it houses many recognized IT companies and local start-ups which can propose more effective ground-breaking solutions and services by exploiting the data generated by WSN and bring those solutions to the global market. Therefore, the city can greatly benefit from data driven and networked intelligent systems both economically and socially. In fact, cities that have introduced digitization, IoT, and SC solutions were able to benefit from an increment of gross domestic product (GDP) by reducing the final operational cost (OPEX) of previous less intelligent systems (Will Serrano, 2018).

1.2 Motivation

The motivation of this work comes from the observation that different COTS (e.g. transceivers and MCUs) can be used to design WSN for IoT applications. Moreover, an exhaustive literature survey has shown that a hands-on, fast, and well-rounded system-level design methodology of low-power, real-time, and reliable WSN was still missing. In the process of building a WSN, the designer needs to optimally select the main components out of a myriad of COTS products and account for all the software and hardware solutions concurrently. Moreover, high-level decisions need to be taken early in the design process while bearing in mind the key factors that are not directly comparable (e.g. energy per measurement and distance between nodes) and explore the limitations of the chosen hardware, configuration, and techniques.

1.3 Main Contributions

The main contribution of this thesis is the development of a clear and well-rounded methodology for the design and deployment of reliable low-power wireless sensor networks. To the best of the author's knowledge, a similar step-by-step and cross-layer design methodology has not been covered in the literature. The design steps that are introduced and explained in this thesis to form the final and well-established methodology are:

- A high level of abstraction energy modeling framework using Simulink/Stateflow is introduced. It allows the creation of energy consumption models of configurable COTS components based on finite state machines and ensures a high degree of modeling flexibility;
- A valid analytical energy model of the sensor node is proposed. It gives a good estimation of the overall and component-level energy consumption;
- Outdoor measurements are carried out in both urban and suburban areas in the city of Montreal in order to characterize the wireless link. Moreover, an empirical path loss model is proposed;

- The quality of service performance and energy efficiency of difference error mitigation techniques, namely forward error correction (FEC), blind retransmissions (BR), and the feedback-based automatic repeat request (ARQ) protocol, and a modified version of the latter are modeled and compared.

1.4 Thesis Outline

The remainder of this thesis is organized into 4 chapters. In chapter 2, a theoretical analysis as well as the design challenges are introduced. Firstly, the requirements in terms of energy efficiency and QoS are explained. Secondly, the theoretical background of three error mitigation techniques, namely FEC, BR, and ARQ is covered. Lastly, two popular wireless communication standards are studied.

Chapter 3 summarizes the literature review and the recent developments of reliable and low-power WSN design methodologies. Different models and frameworks will be presented and their advantages and drawbacks will be discussed.

Chapter 4 is based on a submitted manuscript for publication in a peer reviewed journal. It includes all of the aforementioned thesis contributions. Firstly, the modeling framework is introduced and the analytical energy models are presented. Secondly, the outdoor measurements are showcased and discussed and an empirical path loss model is proposed. Thirdly, the error correcting capability of the previously mentioned techniques is investigated and modeled. Finally, the resulting design methodology is summarized and supported by case studies.

Chapter 5 concludes this thesis and includes recommendations about future research, investigations, and enhancements.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Design Challenges

WSNs represent a promising solution for monitoring and closed loop control applications as they can be easily deployed to sense, process, and forward data in a flexible and cost-effective way. The large number of Sensor Nodes (SNs) composing a WSN are expected to be autonomous, with a node's lifetime dictated by the battery's capacity.

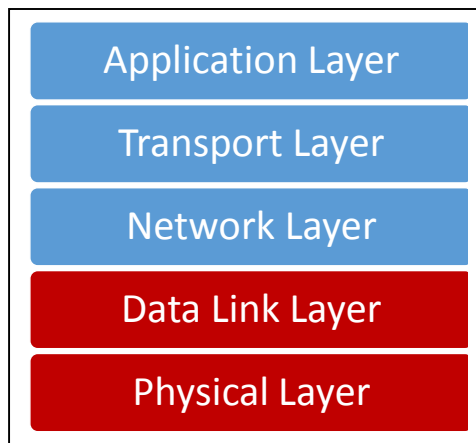


Figure 2.1 WSN protocol stack

As the form factor of the SN is critical in various use cases, minimizing energy consumption while ensuring reliability and latency requirements becomes a priority. Moreover, energy harvesting techniques are increasingly considered as a viable solution for building an entirely green SN and prolonging its lifetime. Furthermore, since a SN is intended to operate as part of a big network of other SNs, the energy consumption is considerably affected by the wireless channel condition and the distance between the nodes when strict QoS requirements, namely, reliability and latency need to be met. In the process of building a SN and in the absence of a clear and well-rounded methodology, the designer can easily make unfounded and suboptimal decisions about the right hardware components, their configuration and reliable data communication techniques, such as ARQ and FEC.

As a result, all of the aforementioned challenges are most adequately addressed if the communication layers presented in Figure 2.1 are optimized concurrently. In this work, both the physical and data link layers are addressed. This technique is referred to as cross-layer optimization in the literature (S. Biswas, 2018; F. Dobsław, 2016; N. Michelusi, 2015; F. Rosas, 2016; Batoul Sarvi, 2017; X. Zhao, 2015) where two or more protocol layers are designed while taking into account the interrelated parameters. As the WSN becomes denser, the used medium access and multi-hop routing protocols and the broadcast nature of WSN make the design problem of guaranteed reliability under strict latency and energy consumption requirements multidimensional and hard to solve. In this work, a point-to-point communication link is considered.

2.1.1 Energy Consumption and Network Lifetime Maximization

Minimizing the energy consumption of WSN while meeting strict performance demands is one of the most challenging goals to achieve. A communication that requires very low latency and high reliability is called ultra-reliable and low-latency communication (URLLC) (K. Lee, 2018; G. Pocovi, 2018). Moreover, as a small form factor is also an important requirement, the use of small batteries hinders the operation of WSN for several years without replacing or recharging them. Accordingly, various energy harvesting (EH) techniques are considered a viable green solution for powering a SN (Abdul Hafiz Alameh, 2018; Sebastian Bader, 2014; Y. K. Tan, 2011; Ljubomir Vračar, 2016; Fan Wu, 2017). In this case, rechargeable energy buffers (e.g. supercapacitors) are used for energy storage (R. Chai, 2015; R. G. Cid-Fuentes, 2014). Renewable energy sources, such as vibration, light or heat, can be considered for powering a SN when several harvesters are used and a proper dimensioning of the energy buffer is carried-out.

Therefore, the research community's first goal has always been to find both hardware (Sadok Bdiri, 2018; Bdiri Sadok, 2018; D. Selvakumar, 2015) and software (Morin É, 2017; Alexander W. Min, 2012; Xavier Vilajosana, 2014) solutions to decrease the depletion rate of the aforementioned limited energy sources. Furthermore, the actual energy consumption and the

node's lifetime need to be estimated early in the design process in order to choose the right components and node configuration.

2.1.2 Meeting quality of service (QoS) requirements

Meeting low latency requirements and high data transfer reliability targets are emerging as an important issue in URLLC IoT applications (R. Abreu, 2018; K. Lee, 2018). This is because the two metrics are tightly interrelated and have a considerable impact on energy consumption. Table 2.1 lists the required end-to-end (E2E) latency requirements of different WSN applications.

Table 2.1 Latency requirement of WSN applications
Taken from V. C. Gungor (2013); K. Lee (2018)

Application	Latency (ms)
Factory and process automation	1
Substation Automation	15 – 200
Overhead Transmission Line Monitoring	15 – 200
Wide-Area Situational Awareness System	15 – 200
Demand Response Management	500 – few minutes
Outage Management	2000
Distribution Automation	20 – 200
Distribution Management	20 – 2000
Asset Management	2000
Meter Data Management	2000
Distributed Energy Resources and Storage	300 – 2000
Vehicle to Grid	2000 – 5000

For example, in an automated factory, a machine misbehaviour or malfunction needs to be detected quickly so that a controller would be able to take action in time and prevent a serious

damage. The required system latency is generally less than 1 ms. However, in case of the vehicle-to-grid communication application (X. Hu, 2018), if the vehicle is informed of the charging and discharging times with a delay of a few seconds, the received information would still be useful and acceptable.

Table 2.2 Reliability requirement of smart grid applications
Taken from V. C. Gungor (2013); K. Lee (2018)

Application	Required reliability (%)
Asset Management	99.0
Meter Data Management	99.0
Demand Response Management	99.0
Outage Management	99.0
Overhead Transmission Line Monitoring	99.0 – 99.99
Wide-Area Situational Awareness System	99.0 – 99.99
Substation Automation	99.0 – 99.99
Distribution Automation	99.0 – 99.99
Distribution Management	99.0 – 99.99
Distributed Energy Resources and Storage	99.0 – 99.99
Vehicle to Grid	99.0 – 99.99
Factory and process automation	99.999

In this work, and as in (Saeed R Khosravirad, 2017), the data transfer reliability is directly linked to the packet success probability. As presented in Table 2.2, the required reliability target varies from one WSN application to another. It mainly depends on the type of data that is being transferred.

By taking into account the factors that can affect the signal of interest such as path loss, noise, data rate, and transmit power, the bit error rate (BER) can be estimated for a given modulation scheme and the packet success probability can be mathematically calculated. The required

reliability target is usually met using techniques such as FEC, blind or ARQ-based retransmissions which will be thoroughly investigated in this thesis.

2.2 Bit and Packet Error Mitigation Techniques

To ensure reliable data communication, both open and closed loop retransmission protocols (R. Abreu, 2018; Saeed R Khosravirad, 2017; M. Zorzi, 1997) and coding techniques (Mohammad Rakibul Islam, 2010; D. Wang, 2017), or a combination of both (J. C. Fricke, 2009; F. Rosas, 2016; M. C. Vuran, 2009) are mostly used. Since both techniques share the same goal of improving data reliability, a question arises as to which combination of these two strategies is optimal from an error recovery and energy efficiency perspective.

2.2.1 Packet Retransmission Techniques

The current and future WSN applications envision reliable communication with efficient use of the limited channel and SN resources. To this end, both open-loop and closed-loop prominent retransmission protocols were adopted. Practically, an open-loop retransmission protocol is where the transmitter blindly repeats each packet R times (R. Abreu, 2018). However, in closed-loop protocols such as the widely-known ARQ protocol (Teerawat Issariyakul, 2006), a feedback channel is used to provide the transmitter with acknowledgement (ACK) or negative acknowledgement (NACK) messages. At the receiver's side, the decision is made by using a cyclic redundancy check (CRC) code to detect errors. If the transmitted sequence is completely error-free, an ACK packet is transmitted to the data sender. Otherwise, a NACK packet is transmitted.

2.2.1.1 Blind retransmissions (BR)

The BR scheme is an attempt to avoid possible packet errors, delays and complexity caused by feedback ACK / NACK frames. Therefore, the transmitter would blindly send a data packet a predefined number of times without waiting for a feedback as illustrated in Figure 2.2.



Figure 2.2 Packet stream example of the BR operation
Adapted from Saeed R Khosravirad (2017)

Blind retransmissions are mostly interesting when the feedback channel is very unreliable which makes feedback-based retransmissions unable to meet high reliability targets. The downside of this scheme's simplicity is the fact that it hinders SN as well as channel resources since that statistically, the biggest portion of successful packet transmissions are observed in the first and second retransmission attempts (Saeed R Khosravirad, 2017). The authors in (R. Abreu, 2018) propose a scheme that allows to limit the channel capacity drawback by dynamically granting shared channel resources. However, the latency and energy consumption disadvantages are still unresolved.

2.2.1.2 ARQ-Based Retransmissions

Different varieties of closed-loop retransmission protocols are used in today's wireless systems (worldwide interoperability for microwave access (WiMAX), long term evolution (LTE), Bluetooth, etc.) to reduce packet loss (Saeed R Khosravirad, 2017; M. Zorzi, 1997). In stop-and-wait (SAW) ARQ (Saeed R Khosravirad, 2017), if a data packet is corrupted or lost, it is retransmitted until it is without any bit errors by the receiver as depicted in Figure 2.3.

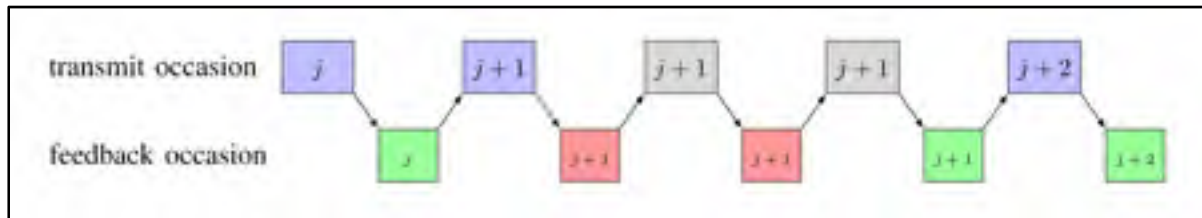


Figure 2.3 Packet stream example of the SAW-ARQ operation
Adapted from Saeed R Khosravirad (2017)

Generally speaking, feedback-based retransmissions can achieve a better wireless channel and resources utilization than BR as the number of the required retransmissions can differ from a data packet to another depending on channel conditions. This is achieved by limiting the number of repetitions to only when the previous attempt has failed. But, it is important to mention that, in addition to the data channel, the reliability of ARQ protocols also depends on the feedback channel's state (Z. Ahmad, 2018; Derya Malak, 2018; H. Shariatmadari, 2017) as a packet is dropped when a NACK packet is falsely perceived as an ACK for example. This makes ARQ-based protocols unable to meet ultra-reliable data communication requirements when the feedback channel also suffers from bit errors. A simple and straightforward solution to increase feedback channel reliability would be to retransmit ACK / NACK packets. This technique is referred to as L -Rep-ACK approach in (Saeed R Khosravirad, 2017) where $L > 1$ is the number of ACK / NACK packet transmissions as depicted in Figure 2.4. Therefore, a packet is declared as delivered only if all L observances of feedback are ACKs.

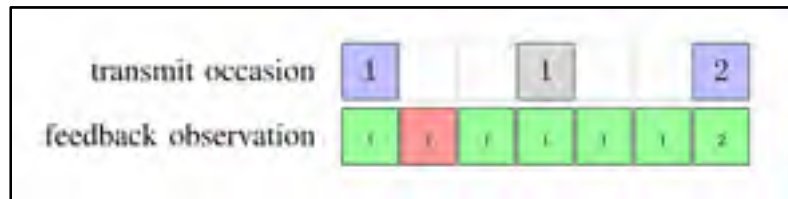


Figure 2.4 Packet stream example of the L -REP-ACK
Operation for $L = 3$
Taken from Saeed R Khosravirad (2017)

Consequently, energy and latency overheads are incurred. This approach is further studied in this work and compared to other delivery error mitigation techniques. Moreover, the authors in (Saeed R Khosravirad, 2017) propose a novel and more complicated packet delivery acknowledging method that is able to further decrease the energy and latency overheads while considering an unreliable feedback channel. The proposed scheme relies on the collaboration between transmitter and receiver nodes to provide ultra-reliable communication of packets even in poor feedback channel conditions.

2.2.2 Forward Error Correction (FEC)

FEC is a technique that makes possible the recovery of a limited number of lost bits in a data packet. This is achieved by adding redundancy bits from the transmitter's side using an encoder and according to certain rules. At the receiver's side, when the same rules are known, errors in the transmitted sequence can be detected and corrected with a given upper bound limit. Consequently, FEC codes incur energy and latency overheads which are the downside of the achieved coding gain. In general, two types of FEC codes can be used: linear block codes (such as BCH, Reed-Solomon, etc) and convolutional codes (Robin Hoel, 2007).

In this work, only convolutional codes are of interest. Fundamentally, a convolutional encoder (n, k, K) is implemented by adding $n - k$ redundant bits to the actual k bits of data. The number of bits upon which the encoder's output depend K is called the constraint length or depth of the code. Usually, decoding is performed by the Viterbi algorithm (Robin Hoel, 2007; B. Sansoda, 2013). The latter compares the received sequence to all of the possible encoded ones and keeps comparing the hamming distance. The sequence presenting the minimum hamming distance is made available at the output of the decoder.

Convolutional FEC can achieve the highest possible coding gain when the flipped bits are evenly spaced throughout the received sequence (Robin Hoel, 2007). However, in a real world wireless application, bursts of errors (i.e. a group of consecutive erroneous bits) are usually observed (Derya Malak, 2018). To tackle this issue, a technique called interleaving (Y. Cai, 2019; Robin Hoel, 2007; R. Swaminathan, 2016) can be performed at the transmitter's side after encoding the input sequence and, prior to decoding, de-interleaving is performed at the receiver's side. This technique ensures that bursts of errors in the received sequence are truncated and spread out in the sequence. Therefore, the decoder would be more capable of correcting bit errors (Robin Hoel, 2007).

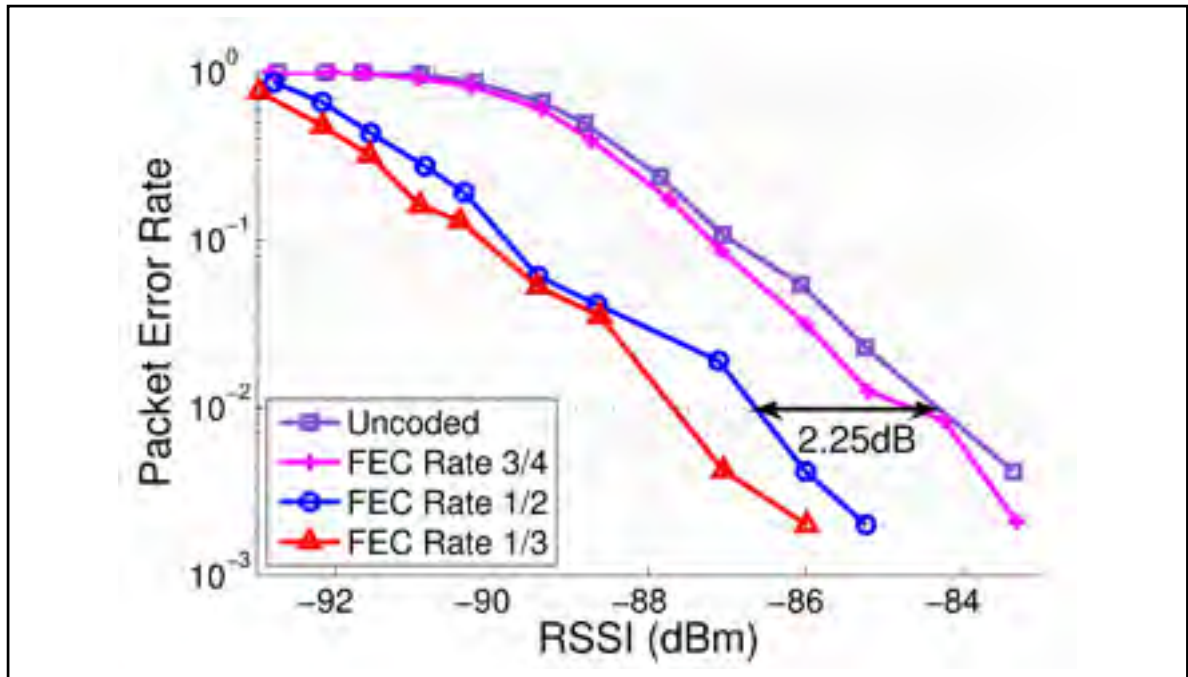


Figure 2.5 Measured packet error rate (PER) curves for a convolutional FEC code of rate=3/4, 1/2 and 1/3 compared with uncoded packets' transmission
Taken from G. Angelopoulos (2013)

Yet, it should be noted that FEC cannot ensure a very high level of reliability (e.g. 99.999%). This will be demonstrated and explained in chapter 4. Figure 2.5 shows the achievable coding gain at 1% PER when using different FEC rates. For a $\frac{1}{2}$ rate convolutional FEC code and a constraint length $K = 4$, a coding gain of 2.25 dB can be achieved.

2.3 Popular Low-Power Wireless Communication Standards

2.3.1 Bluetooth

Bluetooth (Bluetooth Special Interest Group, 2016) was primarily designed to fill the need for a wireless alternative to RS-232 data cables. It is managed by the Bluetooth Special Interest Group (SIG) and it has evolved considerably since it was first initiated to eventually become the most widely used wireless technology for short range communications. Bluetooth 1.1 was adopted by the Institute of Electrical and Electronics Engineers (IEEE) 802.15 working group in 2002 to define the specification of the IEEE 802.15.1 open standard. Subsequent versions

were released to overcome several performance and coexistence issues and the standard was updated accordingly. Most importantly, special attention was given to higher data transfer speeds, security and lower power consumption levels.

Bluetooth operates in the 2.4 GHz of the industrial, scientific and medical (ISM) band. In order to make it prone to interference and signal fading phenomena, the Frequency Hopping Spread Spectrum (FHSS) technique is adopted. Ranging from 2.402 to 2.480 GHz, Bluetooth uses 79 channels with a hop rate of 1600 hops per second. This technique also assures a more secure wireless communication and makes it difficult for intruders to intercept the link. Three possible output power classes are supported by Bluetooth (i.e. 100 mW (20 dBm), 2.5 mW (4 dBm), and 1 mW (0 dBm)) (Luiz Oliveira, 2019) in order to be able to achieve different communication ranges which are in this case 100 m, 10 m, and 10 cm respectively.

As mentioned earlier, higher data rates have been achieved in the subsequent versions. While Versions 1.1 and 1.2 allowed a maximum of 1 Mbps data rate, Versions 2.0 and 2.1 were able to achieve 2 Mbps and 3 Mbps data rates. Bluetooth supports point-to-point and point-to-multipoint communications and a Bluetooth pico-networks can support up to seven active slaves at a time. By combining various pico-networks, a scatter-network can be formed where each pico-network uses a different hop sequence. In order to be within the power budget of applications with very limited energy resources such as embedded low-power wireless sensor networks, the Bluetooth Low Energy (BLE) protocol was introduced. It coexists with the classic Bluetooth technology, enables devices to operate in sleep mode and allows accessing the medium and transmitting data within at least 3ms with data packet lengths of 10 to 47 bytes. BLE also differs from classic Bluetooth in that it supports a subset of 40 channels and a higher operating range. BLE-compliant platforms were made available by several manufacturers during the last few years. In addition, a Gaussian frequency shift keying (GFSK) with a modulation index between 0.45 and 0.55 is used. Data rates of 1Mbps and 2Mbps can be achieved with BLE 4.2 and BLE 5, respectively. Furthermore, the maximum output power for BLE 4.0, BLE 4.1, and BLE 4.2 is 10 mW.

In order to cope with packet errors, BLE optionally uses FEC, the ARQ scheme, or both. However, the header is always protected by a 1/3 rate FEC as it contains critical link information (Bluetooth Special Interest Group, 2016). In BLE, for example, coding can be performed in two stages; convolutional FEC using a non-systematic, non-recursive rate 1/2 code with constraint length $K = 4$ for example, and spreading by using a pattern mapper (Bluetooth Special Interest Group, 2016).

A Link Layer (LL) packet uses a 24-bit CRC to detect bit errors in the payload. If the CRC verification detects the presence of one or more bit errors in the packet payload, the packet is not acknowledged by the receiver and retransmitted by the sender using a go-back-n repeat mechanism. It is possible to modify this scheme in order to meet the requirements of time-critical applications. For example, the maximum number of retransmissions can be modified. When that number is reached without succeeding to deliver the packet, the link is disconnected.

Moreover, in Bluetooth, broadcast links are possible between one source device and zero or more receivers and where the traffic is unidirectional. Data can be sent at any time through this type of links without any feedback nor prior connection. To ensure an acceptable data reliability, a packet is blindly (i.e. without feedback) retransmitted on a configurable number of occasions.

2.3.2 IEEE 802.15.4

IEEE 802.15.4 (IEEE STANDARDS ASSOCIATION, 2015) was first introduced in 2003 and subsequently revised in 2006, 2011 and 2015. It specifies and standardizes the physical layer and the MAC sub-layer of networking devices used in Low-Rate Wireless Personal Area Networks (LR-WPANs). Since it enables very low-cost and low-power communications, IEEE 802.15.4 witnessed a great success and was adopted by several wireless sensor networking technologies such as ZigBee in 2006. Several frequency bands can be used, namely the sub-1 GHz and the 2.38 – 2.45 GHz bands. The 2.4 GHz band suffers from interferers such

as Wi-Fi and Bluetooth and a low communication range. However, the sub-1 GHz band can ensure a longer range and a better link robustness.

The IEEE 802.15.4 standard defines two different types of devices in a LR-WPAN. The first type is Full Function Device (FFD). An FFD device can act as a coordinator or a router and thus can be responsible for managing the network or just expanding it by finding routes and forwarding data packets. FFD devices would therefore embed more memory and CPU power resources, sleep for relatively short times or continuously listen to the channel and consume more energy. The second type is Reduced Function Device (RFD). An RFD device, on the other hand, burns less power since it only transfers application packets with no routing capabilities and can remain in a low power mode for relatively long times.

From 2003 until 2015, several physical and data link layers' specifications were adopted by the IEEE 802.15.4 standard such as the IEEE 802.15.4e and IEEE 802.15.4g amendments in 2012. The latter defines an alternate physical layer specification for outdoor low rate wireless smart metering utility network. Moreover, as the needs for more performance in dense networks and IoT applications increase, contention-based Medium Access Control (MAC) protocols no longer present a viable solution, the IEEE 802.15.4e improves the MAC layer by adopting new mechanisms such as the Time-Slotted Channel Hopping (TSCH) scheme (Glenn Daneels, 2018). TSCH was designed for applications with stringent reliability and power consumption constraints and where measurement and data transmission are performed periodically and quickly became the de facto MAC scheme for reliable and ultra-low power Industrial Internet of Things. It is important to note that TSCH does not bring any changes to the physical layer specified in the IEEE 802.15.4 standard, meaning that it can be implemented using a standard-compliant radio. Networks implementing TSCH as their MAC protocol have their nodes synchronized according to a continuously repeating frame which is divided in time to form a given number of time slots. A time slot's minimum length is actually the time needed for the longest transmission period and its acknowledgement which would be in this case around 15ms. Each slot is assigned a channel-offset that changes from a super-frame to another. This

channel hopping technique can reduce the impact of external interference and multipath fading phenomena especially in the unlicensed 2.4-GHz band, shared with Wi-Fi, Bluetooth, etc.

In the IEEE 802.15.4, a $1/2$ rate convolutional FEC code with constraint length $K = 4$ is optionally. Moreover, interleaving of code-bits can be applied only when FEC is used in order to improve the code's error correction capability by breaking the correlation of consecutive bits.

In order to detect errors, a 16-bit or 32-bit CRC is used. The transmitter optionally requests a feedback from the receiver. If requested, the transmitter waits for the reception of the corresponding ACK frame. When the latter is received within the expected period of time, the transmission is a success. Otherwise, if the feedback is negative, the transmission attempt is a failure. The transmitter would thus repeat the process of transmitting the frame and waiting for the acknowledgment up to a configurable maximum number of times. If no acknowledgment is received after by when that number is reached, the MAC sublayer assumes that the transmission has failed. Moreover, when a device encounters a transmission failure in a shared link (i.e. no ACK reception), it initializes the back-off exponent and wait for a random period of time before attempting a retransmission. However, on a dedicated link, a retransmission can take place at any time. For each failure, the device increases the exponent until a maximum value is reached and the transmission attempt is permanently considered as a fail. Upon a transmission success and ACK reception, the exponent is reset to its minimum value.

CHAPTER 3

LITERATURE REVIEW

In this chapter, a thorough literature review will cover different energy modeling and optimization frameworks as well as different techniques ensuring stringent QoS requirements in modern WSN. Moreover, the key strengths and limitations and drawbacks of the proposed solutions will be highlighted.

3.1 Energy Consumption Modeling and Estimation in WSN

A significant amount of research work has been undertaken in order to estimate the power energy consumption of WSNs at an early stage of the design process.

For instance, a power and energy estimator is presented in (Nicolas Ferry, 2011) to predict a WSN's autonomy in order to evaluate the economic benefits of replacing an existing wired network with a wireless one. The hardware power consumption models are determined using the functional level power analysis (FLPA) methodology (Johann Laurent, 2004). In addition, multiple energy harvesting systems are considered (i.e., solar, wind, and thermal). Dynamic Power Management (DPM) is performed according to a finite state machine (FSM) where the transitions are dictated by the energy saving levels and weather forecasts. However, the energy model parameters are not clearly showcased and thoroughly studied. Therefore, an accurate energy consumption estimation was not achieved nor compared to measurements. Moreover, the modeling framework is based on LabVIEW and does not allow the designer to quickly make a flexible energy model.

Also, the authors in (K. Virk, 2005) present an abstract modelling framework for both sensor-network-level and sensor-node-level modelling, and apply a hardware / software co-design approach. The framework is based on SystemC, and is neat to be used to model almost all of the aspects from sensors' modes of operation to radio signal propagation. Moreover, at the sensor-node-level, the model is split into two different but tightly dependent and related

sections (i.e., software and hardware section). The hardware section helps the estimation of the sensor node's overall power consumption by monitoring the significant parameters of the model, while the software section, on the other hand, comprises tasks models such as processing, I/O tasks, services and schedule of a real time operating system (RTOS) model. It helps simulating the functional side such as the behavior of contention-based medium access control (MAC) protocols (Paulo Bartolomeu, 2016). At the sensor-network-level, the authors model the physical phenomenon of the environment where the sensor node's hardware model will be integrated. It should be noted that in this case, the energy consumption estimation is very time consuming and relatively complicated for the accuracy that can be achieved.

Another contribution is the Powersim C++ class library presented in (Simone Orcioni, 2016). It monitors the C++ operators during the simulation of a high-level of abstraction model developed using SystemC in order to estimate a given hardware's power consumption when provided with an energy model. An energy model represents a set of simulated or hardware power consumption measurements of different operators, and it is possible for the designer to choose the modules and operators to be monitored by adding a configuration file. The energy model contains a list of energy granularities of each arithmetic and logic operation supported by a given MCU. This way, Powersim can calculate the overall energy consumption of the algorithm. The same code was then ported to an MCU and measurements showed that the simulation results with Powersim present an error of 15.8%. Again, this approach is extremely costly in terms of time and complexity. For each hardware platform, the designer needs to develop a SystemC model that will be used to estimate the energy.

In addition, work presented in (A. K. Anwar, 2010) uses the Stateflow graphical modeling environment (Mathworks, 1997–2018) which is a component of Simulink, to develop a model-based design framework of an energy-optimized protocol stack for WSNs. It allows the simulation and code generation of WSN applications intended for a variety of implementation platforms. The framework can also be used in very flexible way to build energy models while omitting the code generation capability.

In (Glenn Daneels, 2018), the authors present an energy consumption model for devices running IEEE 802.15.4e TSCH protocol. The results report an accurate energy consumption prediction as all network-related CPU and radio state periods and transitions are well modeled. They are also validated by comparing the estimated values with hardware measurements where a 3% error margin is observed. Another similar model is presented in (X. Vilajosana, 2014). It should be noted that the two models are specific to the TSCH protocol and cannot be easily adjusted and used to estimate the energy consumption of other protocols.

3.2 Path Loss Modeling

Before designing a sensor node and deploying the network of nodes, the deployment site needs to be characterized and studied. Given that a SN is intended to operate within a large group of other SNs, having a realistic model for path loss (PL) in order to estimate the operating range is of interest. This step needs to be taken early in the design process as it has a great impact on the system's performance and optimal configuration.

Extensive research has been conducted in order to estimate the received signal strength and characterize the effects of signal attenuation while it propagates in a given environment and under known circumstances. Basically, PL models attempt to capture the effect of signal attenuation in line-of-site (LOS) or non-line-of-site (NLOS) communication links (C. Phillips, 2013). Moreover, signal propagation models can be classified based on whether they are meant to capture the effects of large-scale path loss or small-scale fading (W. Sun, 2018). Small-scale fading takes place when rapid fluctuations over very short travel distances (i.e. a few wavelengths) or durations (i.e. a few seconds) are observed. As in the case of WSN applications the large-scale path loss phenomenon is dominant (W. Sun, 2018), small-scale fading is omitted in this survey.

The authors in (M. Bacco, 2018) model the signal PL between a sensor node deployed in a smart farming application and an unmanned aerial vehicle (UAV) such as a drone. The two-ray path loss (TRPL) allows to make good estimation of signal attenuation between two

endpoints acting in an open space such as a rural field in general by taking into account the line-of-sight (LOS) component and the reflected signal due to the ground. The TRPL is expressed as follows:

$$PL = 20 \log \left(\frac{4\pi d}{\lambda} \right) - 20 \log \left[2 \sin \left(\frac{2\pi h_t h_r}{\lambda d} \right) \right] \quad (3.1)$$

where d is the distance between the two nodes, λ is the wavelength, and h_t and h_r are the distances that separate the transmitter and the receiver from the ground respectively.

In (Hicham Klaina, 2018), using the received signal strength indicator (RSSI) data generated by commercial SNs, a three-slope log-normal PL model was proposed in order to model a narrowband radio channel in rural scenarios where the radios operate under near-ground conditions such as the case for smart agriculture applications. Tests were carried out in three different environments, namely in short and tall grass fields and when there is no grass.

Moreover, a survey of LOS and NLOS wireless PL models is presented in (C. Phillips, 2013). For LOS links, a fitting factor is added to the free-space PL in an attempt to avoid underestimations. It is given by:

$$F_f = l_f \log(50d) \quad (3.2)$$

where l_f is a fitting coefficient.

In this thesis, a LOS communication link is considered for WSN applications and an empirical PL model is proposed. It is an attempt to use the proposed models in the literature and come up with an adjustable one that can be used to estimate the received power from a transmitter deployed in different types of environments.

3.3 Energy cost versus QoS trade-off

In recent past, research on limiting energy consumption while meeting stringent QoS requirements has taken a considerable leap and steadily been growing especially in industrial automation applications (F. Dobsław, 2016; H. Shariatmadari, 2017; W. Sun, 2018).

The authors in (D. Singh, 2018) investigate the adaptive data rate (ADR) algorithm implemented in Long Range Wide Area Network (LoRaWAN) and its theoretical bounds of link and network capacity. They explain how the data rate is dynamically adjusted such that a node close to the gateway would use a small spreading factor in order to increase the raw data rate and be able to decrease latency and radio output power. Therefore, the closest node to the gateway transmits with the maximum data rate and lowest output power. In LoRaWAN compliant devices, the adaptive rate-power allocation is based on RSSI and signal-to-noise (SNR) readings of the last received packets of static devices. As it will be demonstrated in this work, the studied power/data rate allocation technique used in LoRaWAN can drastically optimize the wireless link performance and energy efficiency.

In (B. Makki, 2014), the authors investigate the effect of an error-prone feedback channel on the performance of ARQ protocols and study the impact of using adaptive rate and power allocation. Therefore, the throughput, the outage probability, and the feedback load of different ARQ protocols are considered. They demonstrate that optimal power and rate allocation is crucial for ensuring a good performance of noisy ARQ schemes in terms of reliability and latency with respect to the open-loop communication setup (i.e. blind retransmissions) and especially when the number of retransmissions or SNR increase. The missing part in this work is the lack of measurements and real-world case studies in order to realistically showcase the benefits.

In (K. Hedayati, 2010), in addition to optimizing transmissions scheduling in multi-access communication links, the authors propose a mathematical programming model and algorithm to perform simultaneous adaptive allocation of physical layer parameters, namely transmit

power levels and data rates across active links, while meeting required Signal-to-Interference plus-Noise Ratio (SINR) levels at intended receivers. Therefore, an energy efficient rate-power combination is achieved. The algorithm is based on the construction of a Power Controlled Rate adaptation Interference Graph yielding a 20% better throughput performance than prior algorithms using fixed transmit power and fixed rate link scheduling. However, in this case, the energy consumption and QoS requirements are not of paramount importance.

While admitting that in harsh industrial application scenarios, factors such as transmission power level, communication range, and random ambient noise affect radio link quality, a network-level reliability model for estimating and optimizing the reliability performance and deployment parameters of industrial WSN is presented in (W. Sun, 2018). They suggest a new approach where nodes measure and estimate link parameters such as the packet reception ratio (PRR) and received signal strength (RSS) and then optimize the lower-bound reliability value. To this end, an alpha-stable distribution to accurately model the background noise and a modified log-normal path loss model to estimate the RSS are introduced. A mapping function between PRR, background noise, and RSS is then proposed. Through a case study, the authors demonstrate the feasibility of their solution and optimize the reliability by computing the maximum deployment distance between sensor nodes. Yet, the energy consumption cost is not quantitatively evaluated and optimized as the authors mention that the nodes are energy-limited devices. Moreover, the latency is not clearly addressed in this study.

After optimizing the wireless link parameters, bit and packet errors can still occur. Therefore, as previously mentioned, FEC, ARQ schemes, or both are used. As FEC has a limited ability to correct errors, the authors in (M. Patil, 2017) propose a dynamic error control scheme based on link parameters such as BER and ambient noise in WSN. Through simulations, it is reported that throughput and retransmission probability are improved. An energy model that showcases the efficiency of the proposed technique is actually missing in this study.

As ARQ-based protocols suffer from feedback error, the authors in (Saeed R Khosravirad, 2017) study different approaches allowing to increase feedback channel time diversity and

attain different reliability regions with respect to feedback channel error rate such as the *L*-REP-ACK scheme. Then, they propose a new method of acknowledging packet delivery for retransmission protocols which is based on backwards composite acknowledgment from multiple packets while relying on collaboration between transmitter and receiver nodes. Therefore, depending on channel quality, the scheduler of the wireless channel would be able to configure ultra-reliable communication when needed. The proposed solution does not require increasing the time diversity order of the feedback channel and thus does not incur energy consumption and latency overheads. Moreover, they investigate the advantages and disadvantages of BR and show that in extremely unreliable feedback channel conditions, an open-loop solution is viable in terms of reliability while noting the energy consumption downside. However, in none of the studied solutions do the authors clearly and quantitatively evaluate the energy consumption nor the latency overheads. As in most cases a good compromise between reliability, latency, and energy consumption need to be found, the evaluation of the three metrics needs to be carried out in a simultaneous way.

In addition to adjusting link parameters and applying bit and packet error mitigation techniques, the authors in (P. Kong, 2016) present an intelligent starting point for network planning for an optimal sensor and sink nodes deployment. Using an analytical model, they determine the minimum concentrator nodes (i.e. sink nodes) density and locations that are required to support a given smart grid application QoS requirements in terms of packet delay, packet error probability, and outage probability. A network of sensors and smart meters deployed in a neighborhood area in a densely populated urban area and based on the IEEE 802.15.4g standard is considered. It is reported that less than ten concentrators per km² are needed to support a density of 500 sensor nodes, an end-to-end latency of less than 1s, a packet error probability below 0.005, and an outage probability below 0.01. Similarly, the authors in (F. Dobslaw, 2016) propose a framework capable of finding valid cross-layer solutions and optimizations to meet stringent end-to-end QoS requirements of industrial WSN applications. Also, an algorithm that identifies the required number of sink nodes in order to meet the requirements is introduced. For example, they report that a network of 50 sensor nodes requires an average of 8.1 sinks when four channels are used and an end-to-end reliability demand

99.999% is needed. However, both studies completely omit the impact of the proposed solutions on energy consumption which is considered to be one of the main design challenges in WSN. The placement of the sink nodes will directly affect link parameters such as the data rate, the transmission power level, and the choice of bit errors mitigation techniques.

In (R. Abreu, 2018), the authors study the possibility of achieving a 0.99999 packet success probability within a 1ms latency while bearing in mind the capacity of the network. To this end, they avoid the reliance on imperfect and error-prone feedback channels and propose a novel scheme based on blind retransmissions and coupled with successive interference cancellation to receive the remaining non-decoded data with a low latency penalty when compared with the feedback-based retransmission schemes. Finally, it is reported that depending on the number of users sharing the resources, the novel scheme can be more resource efficient than a conservative single shot transmission. However, the authors assume fast processing and transmitting/receiving times, both from the transmitter and receiver side. This assumption can also lead to reducing the communication range in a drastic way when using high data rates.

Contrarily to the reviewed works and studies in this chapter, the methodology presented in this thesis covers the three major design challenges (i.e. energy efficiency, reliability, and latency) in a simultaneous way in order to achieve a good trade-off. The next chapter includes the theory behind the methodology, the measurements and discussions, as well as the case studies.

CHAPTER 4

SYSTEM-LEVEL DESIGN METHODOLOGY

This chapter is based on a submitted manuscript for publication in a peer reviewed journal. Here, the proposed energy and path loss models are presented. The novel contribution of the proposed energy model is the accurate estimation of MCU current consumption and execution time. Moreover, a detailed study of different error control techniques in terms of reliability, energy consumption, and latency is included. Furthermore, it covers all the measurements and tests which were conducted in order to validate the proposed models. The measurement setups, tools, and methodologies are also explained. This chapter will be closed with case studies highlighting the possible design improvements in terms on energy consumption, reliability, and latency when the proposed methodology is applied. In addition, the limitations of the models and experiment as well as the viable solutions and improvements which can be used to mitigate these issues are discussed.

4.1 Sensor Node Energy Model

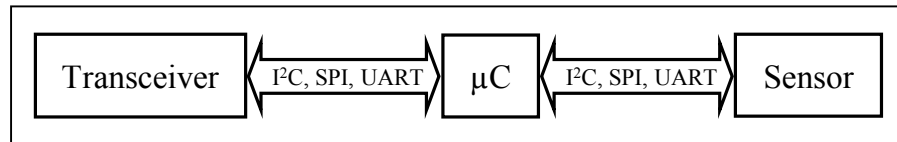


Figure 4.1 The three main modeled components of an SN

In this section, the analytical energy estimation models and the modeling framework of a functional sensor node are introduced. The main components of a SN that are modeled in this work using a high-level of abstraction framework are depicted in Figure 4.1. The considered sensor allows measuring temperature, pressure, and humidity with a given sampling rate depending on the application (i.e. every 500 ms). At the end of the section, a comparison between the estimated and the measured energy per measurement is included.

4.1.1 Modeling Framework

The system-level design framework should support multiple levels of abstraction, make possible the integration of hardware and software models in an intuitive way and allow making fast estimations. In this thesis, the proposed modeling framework (Brini Oussama, 2018) is based on Simulink / Stateflow and allows the creation of energy consumption models of configurable COTS components based on FSMs. The main components of a SN taken into consideration are the microcontroller unit (MCU), the transceiver, and the sensor. A Stateflow chart functions as a FSM within the Simulink model. Therefore, the sensor node model can be fed by different types of inputs that would change model parameters or state transitions inside the chart. Then, model outputs can be plotted or stored in an external file. In addition, it is possible to integrate MATLAB functions that can reside anywhere in a Stateflow chart, state or sub-chart.

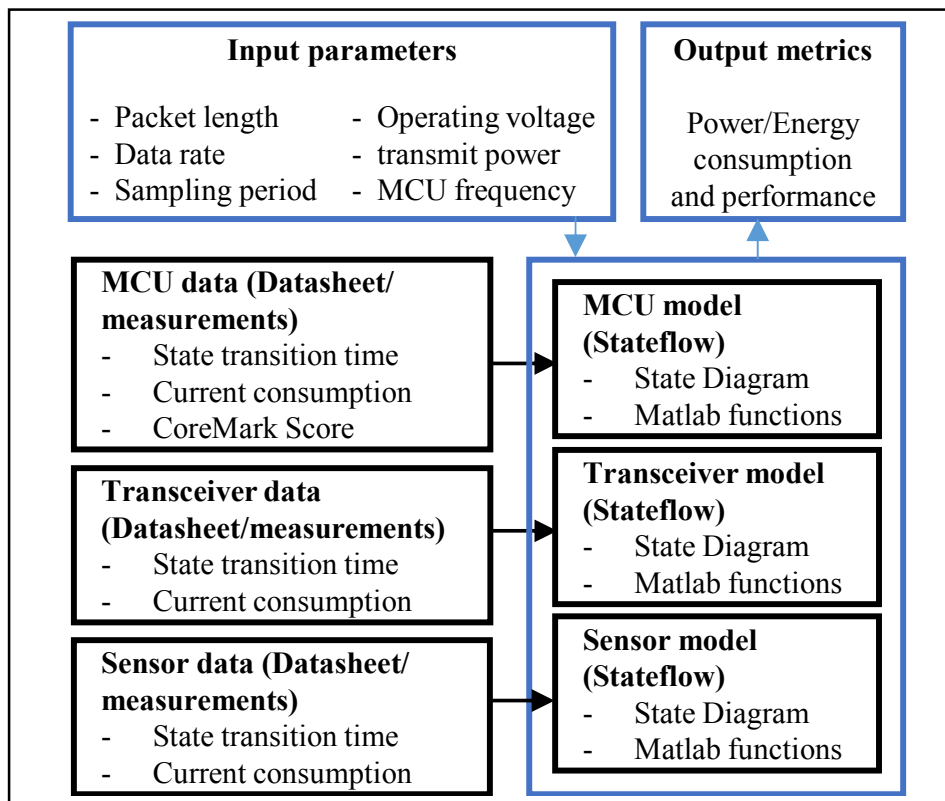


Figure 4.2 A Power/Energy consumption assessment framework based on Stateflow/Simulink

In addition, the framework ensures a high degree of modeling flexibility, as shown in Figure 4.2. Moreover, as depicted in Figure 4.3, stateflow represents multiple levels of subcomponents in a system which makes multilevel state complexity like it is the case for an SN more manageable. Also, states within a chart can be executed exclusively where one state can be active at a time or in parallel where states are active at the same time. All the aforementioned advantages allow the designer to significantly reduce the modeling time of the sensor node which can be achieved in a few days using the proposed modeling framework. Furthermore, after the models are built, it takes a few seconds to get an acceptable estimation of the energy and the average power consumption, as well as the latency.

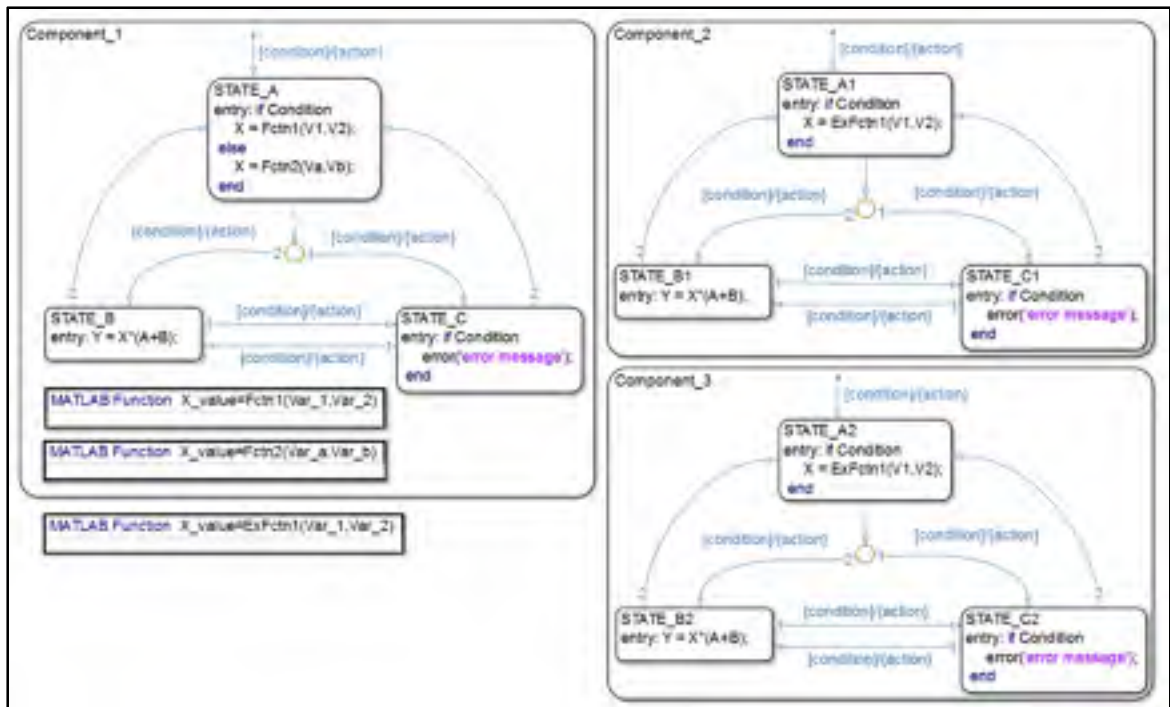


Figure 4.3 A general sensor node energy model using Stateflow charts

4.1.2 Energy Model Parameters

In Table 4.1, all the parameters that have an impact on the overall energy consumption of the studied SN based on the CC1310 wireless MCU (WMCU) system-on-chip (SoC) (Texas Instruments, 2016) are listed. It should be noted that only the parameters linked to the sensor

used in this work (i.e., Bosch Sensortec BME280 (BOSCH, 2016)) are specific to this SN. Otherwise, the list of parameters can be used and adjusted to estimate the energy consumption of approximately any wireless sensing system.

Table 4.1 Energy model parameters

Parameter	Description	Purpose
V_{WMCU}	Operating voltage of the SoC	Power / Energy consumption
V_{SENS}	Operating voltage of the sensor	Power / Energy consumption
D_R	Data rate	TX and RX active times
f_{MCU}	MCU operating frequency	Active current / Processing time
I_{STATE}^{TRX}	Current consumption in each state	Power / Energy consumption
I_{STATE}^{MCU}	Current consumption in each state	Power / Energy consumption
I_{PPH}^{MCU}	Peripheral current consumption	Power / Energy consumption
l	Packet length	Transceiver active time
S_{REF}	Reference CoreMark score	Processing time / System latency
f_{REF}	Reference operating frequency	Processing time / System latency
$t_{PROC\ REF}$	Reference processing time	Processing time / System latency
S_{MCU}	Selected MCU's CoreMark score	Processing time / System latency
T_{ovs}^*	Temperature oversampling factor	Current consumption / System latency
H_{ovs}^*	Humidity oversampling factor	Current consumption / System latency
P_{ovs}^*	Pressure oversampling factor	Current consumption / System latency
I_{STATE}^{SENSOR}	Current consumption in each state	Power / Energy consumption

* These parameters are specific to the BME280 sensor used in this work

4.1.3 Analytical Energy Model

In this section, the proposed analytical energy models of a SN's main components (i.e. MCU, transceiver, and sensor) are presented and validated by measurements.

4.1.3.1 Micro-Controller Unit (MCU)

An MCU's central processing unit (CPU) core speed and current consumption can be assessed by running a benchmark algorithm. Several benchmarking algorithms have emerged such as Fibonacci, Dhrystone, and CoreMark (CM). The latter was developed by the EEMBC consortium in 2009 and then quickly became the de facto standard for CPU core performance ratings. Most MCU manufacturers specify the current consumption of their products when running one or more benchmark algorithms, notably the industry standard CM which is the benchmark considered in this work.

Table 4.2 Current consumption of two different MCUs at 26MHz

Parameter	MCU	While loop	Fibonacci	CoreMark
Current [mA]	SAMD21	2.2	2.27	2.44
	STM32L4	1.7	2.42	2.9

This is important as the designer needs a guideline for estimating a specific application's power consumption which can vary considerably from one benchmark algorithm to another as reported in Table 4.2 (Brini Oussama, 2018). Furthermore, a survey of COTS MCUs along with the corresponding CM scores is presented in Table 4.3.

Moreover, CM is an open-source portable program allowing designers to extract the current consumption of any MCU on the market when it is not provided by the manufacturer. It's source code is written in C and implements list processing which manipulates the memory system using pointers to find and sort variables, matrices using common math operations such as the multiply and accumulate instruction, a state machine to evaluate data-dependent branch logic and a CRC mechanism to operate XOR gates, shifters, etc.

Table 4.4 shows a comparison of the average current measurements in (Joakim Lindh, 2017) to the predictions using CM, showing that the estimation of the average current consumption when using CM results in a 4% error margin, which is acceptable.

It should be noted that for the Bluetooth low energy (BLE) current, the contribution of some peripherals has been taken into account and subtracted from the actual measured current in order to determine the 2.825 mA number.

Table 4.3 A survey on low-power COTS microcontrollers

Chip	STM32L433	MSP432P401R	CC1310 (WMCU)	SAM D21
Manufacturer	STMicroelectronics	Texas Instruments	Texas Instruments	Microchip
Processor	Cortex-M4F	Cortex-M4F	Cortex-M3	Cortex-M0+
Score[CM/MHz]	3.42	3.41	2.96	2.46
Voltage supply [V]	1.71 to 3.6	1.62 to 3.7	1.8 to 3.8	1.62 to 3.63
CM current [μ A/MHz]	103	160	52	106*freq+136
Clock speed [MHz]	Up to 80	Up to 48	Up to 48	Up to 48
Lowest current ¹ [nA]	8	25	185	2700
Sleep current ² [μ A]	0.28	0.63	0.7	4.06
Wake-up time ¹ [ms]	0.26	1.1	1.097	1.0196
Wake-up time ² [μ s]	12.2	700	174	19.6

¹ Deepest low-power mode where all the clocks are disabled and memory is lost

² Deepest low-power mode where the Real Time Clock (RTC) is available

Table 4.4 Accurate current consumption estimation using CM

MCU	Software	Measured current [mA]	CM current [mA]	Error [%]
CC1310	TI 15.4 stack	3	2.88	4
CC2650/CC2640R2	BLE stack	2.825	2.938	-4

This is because CM only evaluates the MCU core. Therefore, it is safe to say that the CM benchmark represents a fairly close workload of a SN and is a reliable indicator of the power consumption of different MCUs. The measurement setup presented in Figure 4.4 was used to measure the current consumption of the CC1310 WMCU. Moreover, in order to showcase exactly the N6705A DC power analyzer was used, figure 4.5 shows the connections.

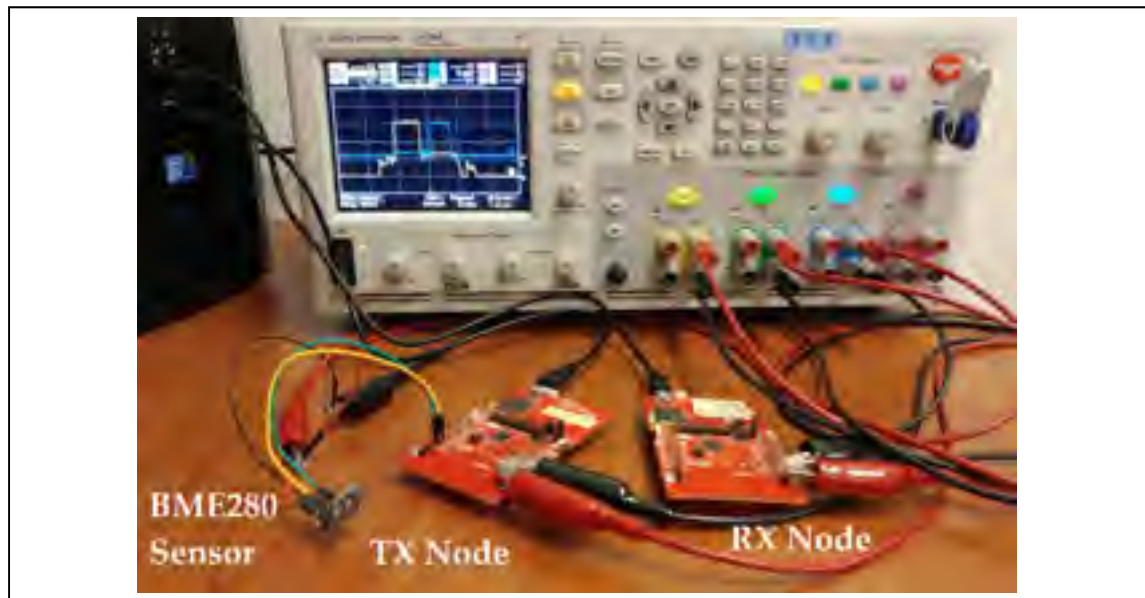


Figure 4.4 Current consumption measurement setup of the CC1310 wireless MCU while performing a point-to-point communication of an internal temperature sensor data and running the TI 15.4 network stack



Figure 4.5 Clear representation of the connections

The CC1310 is assumed to be running the TI 15.4-Stack from Texas Instruments that implements the standard IEEE 802.15.4e and 802.15.4g specification for wireless star-topology-based networking solutions. The stack also provides a real-time operating system (RTOS) with a real-time multitasking kernel. Figure 4.6 shows the processing period, the transmission of the data packet where the highest amount of current is observed, a standby period, and the reception of an ACK.

By using the N6705A DC Power Analyzer from Agilent Technologies, multiple test instruments and external circuitry to analyze the energy requirements of the device under test (DUT) can be omitted. In addition, the “Agilent 14585A Control and Analysis Software” tool is used to control the Agilent N6705A for a better display and control over the equipment.

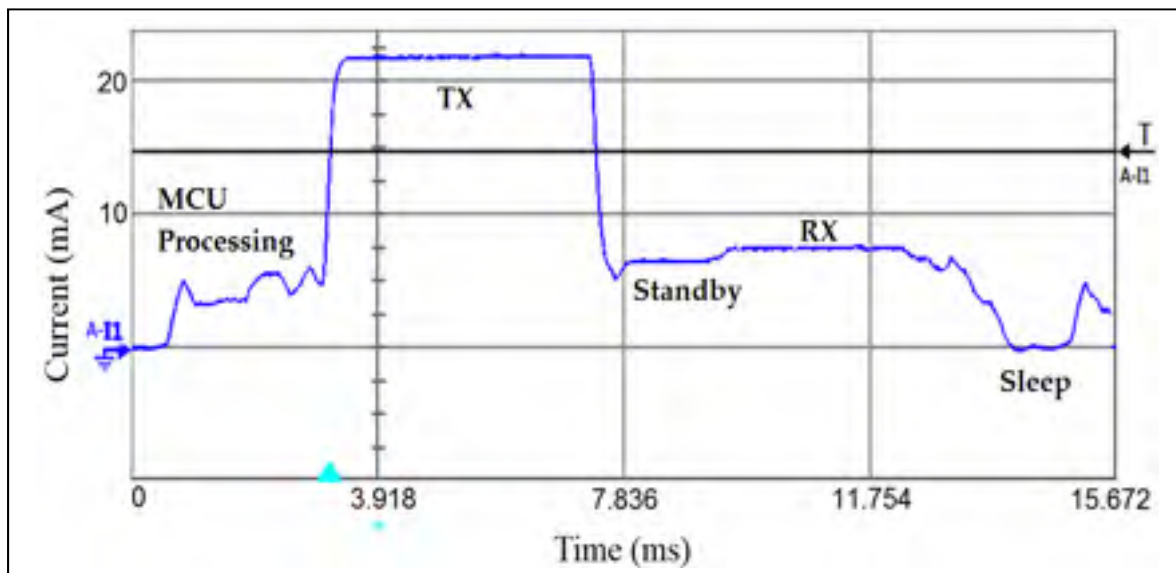


Figure 4.6 Current consumption profile of the CC1310 wireless MCU on the transmitter's side

After estimating the current consumption, an accurate estimation of the processing time is also required in order to evaluate the energy consumption. Metrics such as the million instructions per second (MIPS) are only an approximation as to how a set of processors' performance would vary since different amounts of work can be done in one cycle for each processor. Even when using the same intellectual property (IP) core such as the ones provided by ARM, each MCU

or system-on-chip (SoC) manufacturer has the freedom to decide whether or not to implement advanced features (e.g. memory accelerators, longer bus fetch widths, floating point unit (FPU)).

Therefore, rapidly comparing the speed of different MCUs that are becoming more and more complex is not a trivial task. As the CM benchmark became an industry standard, the CM / MHz figure is increasingly provided in data sheets. It is judged to be accurate enough to estimate the time it takes different MCUs to process the same workload (ARM Limited, 2013) using:

$$t_{PROCESS} = t_{PROC_REF} \frac{f_{REF} S_{REF}}{f_{MCU} S_{MCU}} \quad (4.1)$$

where t_{proc_ref} and S_{ref} are the reference time and reference CM score respectively, extracted from the reference MCU. f_{MCU} and S_{MCU} are the operating frequency and CM score of the studied candidate MCU, and f_{ref} is the reference operating frequency.

It is important to note that Equation (4.1) can be used only when the execution time of the workload on a reference MCU is known. In addition to the aforementioned advantages, CM is judged to be a reliable benchmark because it ensures that compilers would not be able to pre-compute results to completely optimize the work away unlike the Dhrystone benchmark (ARM Limited, 2013). Another important characteristic about CM, is the fact that results reporting is done following a standard format so they can be eventually certified by EEMBC.

In Figure 4.7, the STM32F070RB Cortex-M0 MCU was used as a reference to estimate the processing time of running the room occupancy estimation algorithm (C. Leech, 2017) on the STM32F401RE Cortex-M4 MCU and vice versa. Also, the STM32F051R8 Cortex-M0 MCU was used as a reference to estimate the processing time of the ECDSA cryptography algorithm for IoT applications (L. P. I. Ledwaba, 2018) on the STM32F100RB Cortex-M3 MCU and vice versa. For the four MCUs, the processing time is estimated with an error between 9.4% and 11.5%. Therefore, the estimates are judged to be accurate enough to help make high level

decisions. For a better visualization of the data, in Figure 4.7b, 15000 iterations of the Temperature-Dependent Kinetic Battery Model (T-KiBaM) algorithm (M. Leonardo Rodrigues, 2017) used in battery-powered WSN are assumed to be running on the SAMG55 32-bit ARM Cortex-M4 MCU. The SAMR21G18A 32-bit ARM Cortex-M0+ MCU studied in (M. Leonardo Rodrigues, 2017) was used as a reference.

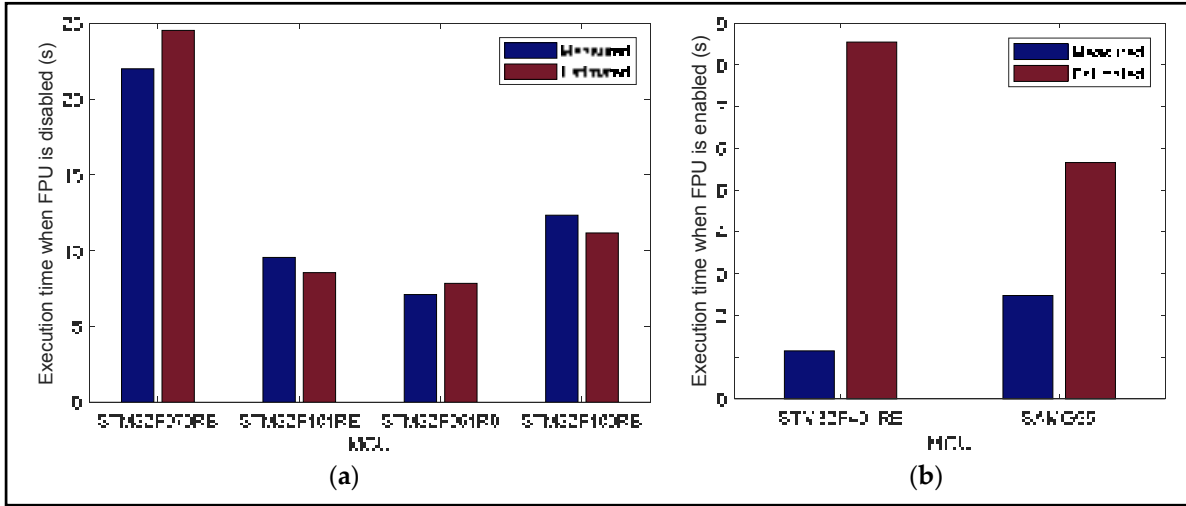


Figure 4.7 Processing-time estimation using the CM/MHz figure (a) when the FPU is disabled and (b) when it is enabled

However, when the FPU is enabled, the estimation is no longer acceptable for both MCUs, especially for the STM32F401RE. Figure 4.7a shows that when the FPU was disabled, the time was estimated with a 10.5% error. This is due to the fact that CM primarily focuses on integer operations commonly used in embedded systems and neglects features like the FPU. Moreover, the STM32F401RE is running the room occupancy estimation algorithm (C. Leech, 2017) which uses a significant number of floating-point operations.

In this work, The MCU's energy consumption is estimated using:

$$E_{MCU} = V_{WMCU} (I_{CoreMark} + I_{PPH}^{MCU}) t_{process} \quad (4.2)$$

where $I_{CoreMark}$ is the MCU's current consumption when running CM, I_{PPH}^{MCU} is the current consumption of other peripherals (e.g. peripheral power domain, RF core, I2C, and timers), and V_{WMCU} is the operating voltage.

4.1.3.2 Wireless Transceiver and Sensor

To evaluate the energy consumption per measurement of the wireless transceiver, both the transmitter and the receiver are considered. The current measurement setup is shown in Figure 4.8. The studied transceiver uses a sub-1 GHz carrier which has the capability to respond to the needs and concerns for long-range and low-power wireless connectivity (P. Sommer, 2018). Table 4.5 covers a survey of COTS sub-1 GHz transceivers.

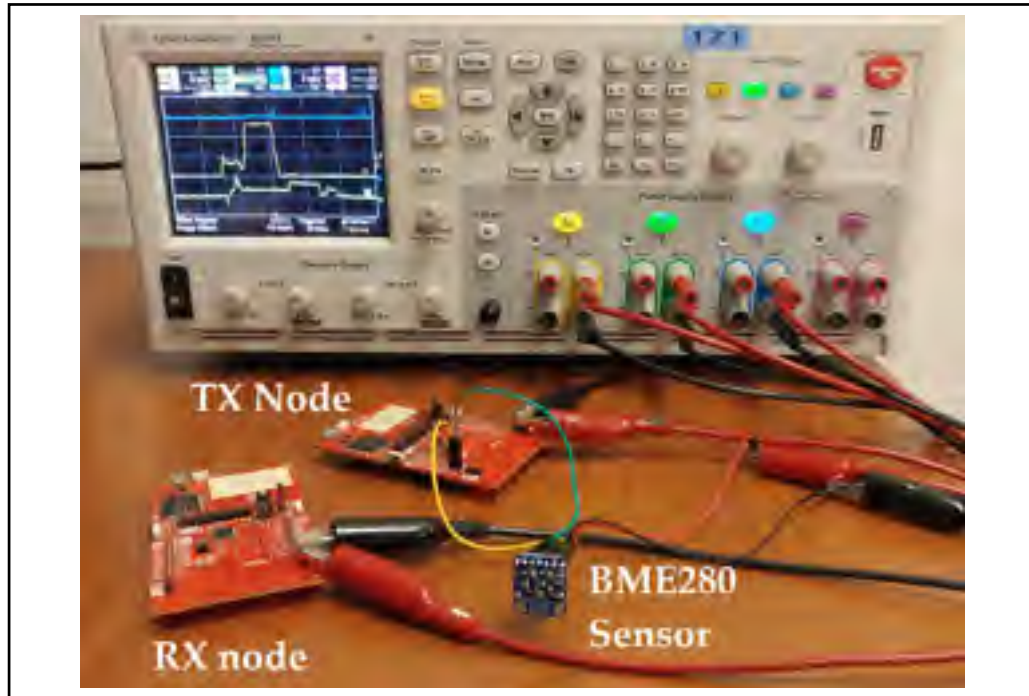


Figure 4.8 Measurement setup for transceiver and sensor current measurement

Computing the energy consumption per measurement during the active period is quite straightforward in this model. It can be estimated using:

$$E_{TRX} = V_{WMCU} \frac{l}{D_R} (I_{TX} + I_{RX}) \quad (4.3)$$

where l is the packet length, D_R is the data rate, I_{TX} and I_{RX} are the transmitter's and the receiver's currents during active mode respectively as shown in Figure 4.9.

Table 4.5 A survey on low-power COTS Sub-1 GHz transceivers

Chip	Si446x	S2-LP	CC1310 (WMCU)	AX5243
Manufacturer	Silicon Labs	STMicroelectronics	Texas Instruments	On Semi
RF bands [MHz]	119 to 1050	430 to 940	315 to 920	27 to 1050
Voltage supply [V]	1.8 to 3.8	1.8 to 3.6	1.8 to 3.8	1.8 to 3.6
Tx current (at +10 dBm) [mA]	18 (Si4460)	10	13.4	16
Rx current [mA]	10 to 13	7	5.5	9.5
Sensitivity [dBm]	-133	-130	-124	-138
Tx power [dBm]	Up to +20	Up to +14	Up to +15	Up to +16
shutdown current [nA]	50	2.5	185	50
Data rate [Kbps]	0.1 to 1000	0.3 to 500	0.625 4000	0.1 to 125
Down to Idle [us]	440	500	923	500
Idle to Active [us]	126	145	174	190

In this study, the combined digital humidity, pressure and temperature BME280 sensor from Bosch Sensortec was chosen. It is housed in a compact package allowing the reduction of the overall sensor node's form factor. Both the measurement time and the current consumption depend on the oversampling mode of the three physical quantities. This means that, on the sensor's level, noise can be traded-off against latency and current consumption.

Figure 4.9 clearly shows the different measurement phases (i.e. temperature, pressure, and humidity respectively) and current consumption on the BME280 sensor where oversampling factors of 4, 2, and 1 were selected respectively (BOSCH, 2016). It also shows the current consumption profile on the transmitter and receiver. As expected, the figure shows that the transceiver is the most energy-consuming component of a SN.

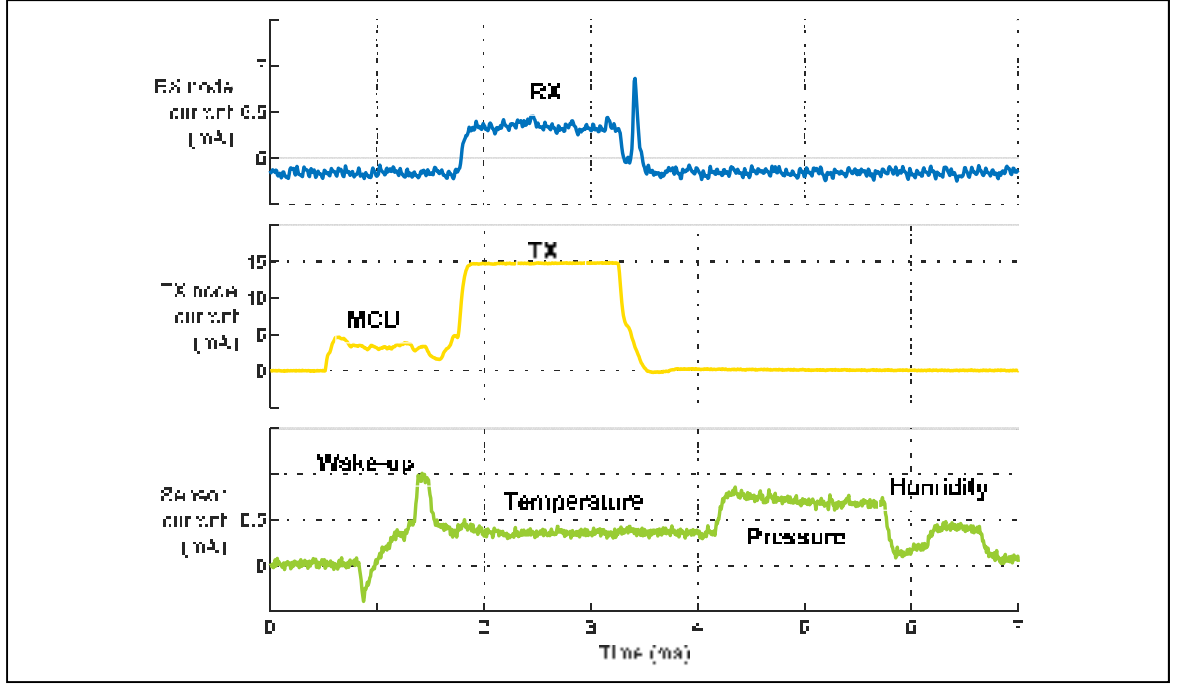


Figure 4.9 Current consumption breakdown of the sensor, transmitter, and receiver during one measurement

It should be noted that, in this case, the receiver is continuously listening as depicted in figure 4.9. However, the estimations in this work consider it to be duty-cycled (i.e. only active during the reception period). This is because a simple point-to-point communication is assumed and implemented in this work. Yet, in WSN applications, communication protocols such as time division multiple access (TDMA) are used in order to allow the receiver to go into a power-saving mode when not communicating. Moreover, it can be seen from the same figure that the measurement starts right after the MCU's wake-up and takes a long time to finish. This means that the measurement done in the i^{th} period, is actually transmitted in the $(i + 1)^{th}$ period. From an energy consumption point of view, the results remain the same.

According to the datasheet (BOSCH, 2016), the sensor's measurement time can be calculated using:

$$t_{SENSOR} = 2 (T_{OVS} + P_{OVS} + H_{OVS} + 1) \quad (4.4)$$

The average current consumption during measurement can be calculated using:

$$I_{SENS} = \frac{I_{DDT}(1 + 2 T_{OVS}) + I_{DDP}(2 P_{OVS} + 0.5) + I_{DDH}(2 H_{OVS} + 0.5)}{t_{SENS}} \quad (4.5)$$

Therefore, the sensor's energy consumption is given by:

$$E_{SENS} = V_{SENS} I_{SENS} t_{SENS} \quad (4.6)$$

where V_{SENS} is the sensor's supply voltage.

4.1.4 Sensor Node Energy per Measurement

In practice, the suitable transmission parameters (e.g. transmission power level and data rate) for a better system performance and energy efficiency can be determined offline (i.e. during system design and before network deployment) where the appropriate parameters are collected and used during data transmission. Moreover, when the channel state follows a specific pattern and only the long-run statistics change after several packet periods, an optimization algorithm can be utilized by the transmitter for adaptive and dynamic parameter setting (B. Makki, 2014). In this section, the first technique is addressed.

In order to see the impact of using different bit rates and output power levels and to evaluate the accuracy of the energy consumption models, measurements of the energy consumption per measurement (i.e. temperature, pressure, and humidity using the BME280) figure were performed at 50 kbps and 500 kbps with an output power level going from -10 to 12 dBm. The results are depicted in Figure 4.10. The estimated energy was obtained using Equation (4.6) for the sensor, Equation (4.3) for the transceiver, and Equation (4.2) for the MCU. At this point, it can be assumed from Figure 4.10 that a better energy efficiency can be achieved when using high data rates allowing the transceiver to go into a power-saving mode more quickly. However, this remains an assumption as it comes at the cost of a lower communication range.

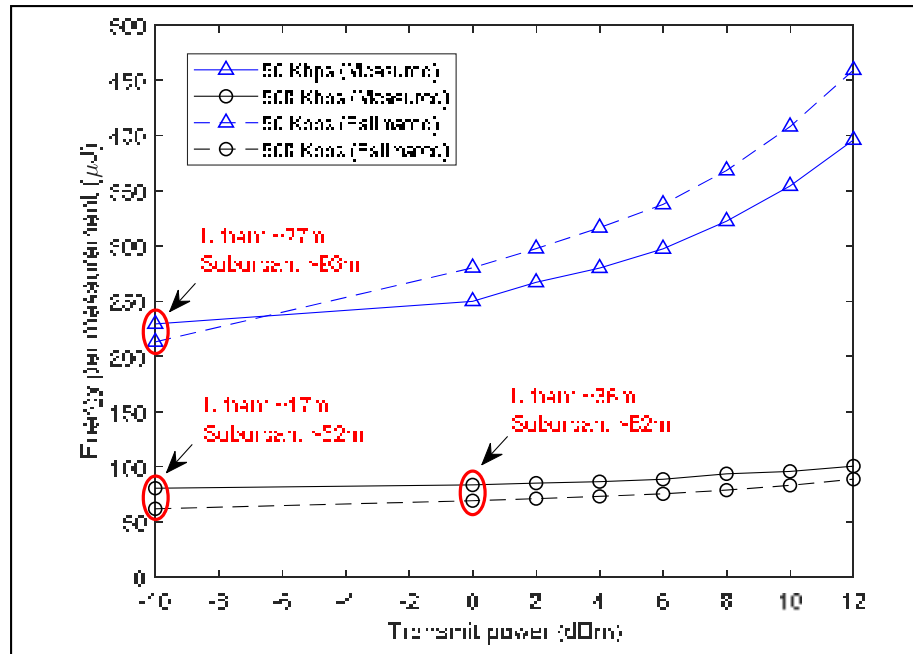


Figure 4.10 Estimated versus measured SN's energy per measurement

Therefore, the goal is to use the highest achievable data rate that can sustain the desired communication range. The latter is determined for a given BER or reliability target of the application. To this end, a realistic path loss model is indispensable in order to estimate the received signal power. It is introduced in Section 4.2.3 of this thesis.

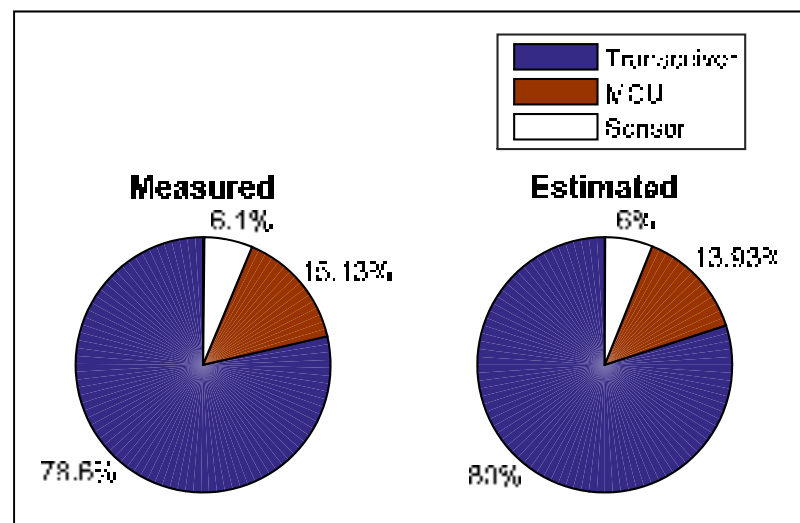


Figure 4.11 Energy consumption per measurement at 50 kbps

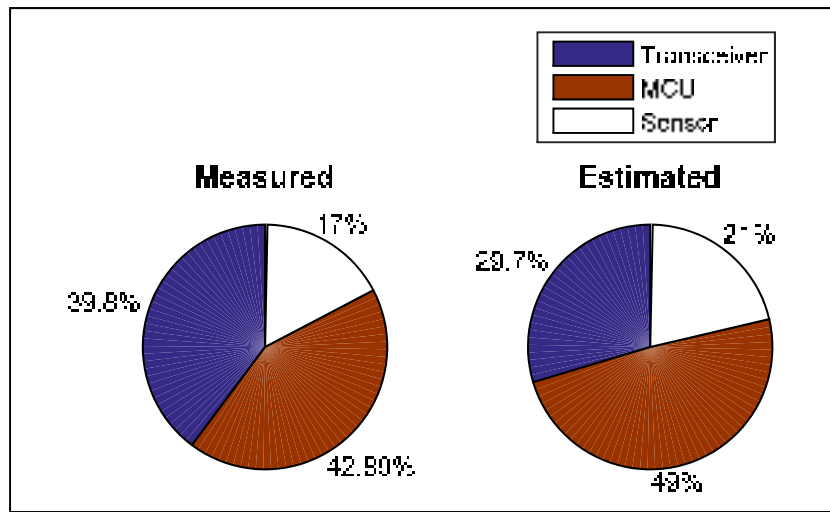


Figure 4.12 Energy consumption per measurement at 500 kbps

Moreover, Figure 4.11 also demonstrates that the biggest portion of the energy (i.e. around 80%) is dissipated by the wireless transceiver at 50 kbps. However, at 500 kbps, figure 4.12 shows that the transceiver energy per measurement is considerably reduced to become comparable to the MCU's. It is therefore very important to carefully choose the data rate.

4.2 Outdoor Measurements and Wireless Link Characterization



Figure 4.13 Communication range measurement setup

Since a SN is intended to operate as part of a big network of other SNs, the energy consumption is considerably affected by the wireless channel condition and the distance between the nodes. Therefore, a realistic model for PL is of interest in order to estimate the received power at the receiver and determine the communication range for a given reliability requirement. To this end, in the following measurements, in order to have a transmitter and a receiver, a pair of Sub-1 GHz CC1310 WMCU LaunchPad development kits operating at 915 MHz and two laptops running the SmartRF Studio application from Texas Instruments are used as depicted in Figure 7. Starting at a 1 meter distance, the three following metrics (i.e. PER , BER , and $RSSI_{avg}$) are collected in steps of 5 or 10 meters by keeping the transmitter at the same place and moving the receiver in order to evaluate the radio link quality. The packet error rate (PER) is given by:

$$PER = 100 \frac{N_{P_{NOK}} + N_{P_{LOST}}}{N_P} \quad (4.7)$$

where $N_{P_{NOK}}$ represents the number of packets received in error, $N_{P_{LOST}}$ is the number of completely lost packets (i.e. the receiver knows that it has to receive a given number of packets) and N_P represents the total number of packets which is in this case 400.

The BER is also taken into account and estimated using:

$$BER = 100 \left(1 - \left(1 - \frac{PER}{100} \right)^{\frac{1}{N}} \right) \quad (4.8)$$

where N is the number of bits per packet.

Lastly, the RSSI is also considered and calculated using:

$$RSSI_{Avg} = \sum_{i=1}^{N_P} \frac{RSSI_i}{N_P} \quad (4.9)$$

where $RSSI_i$ is the received signal strength indicator of the i^{th} packet.

4.2.1 Ambient Noise Density Measurements

In addition to thermal noise, as the sub-1 GHz ISM band is used, interference can considerably affect the wireless link quality and needs to be investigated.

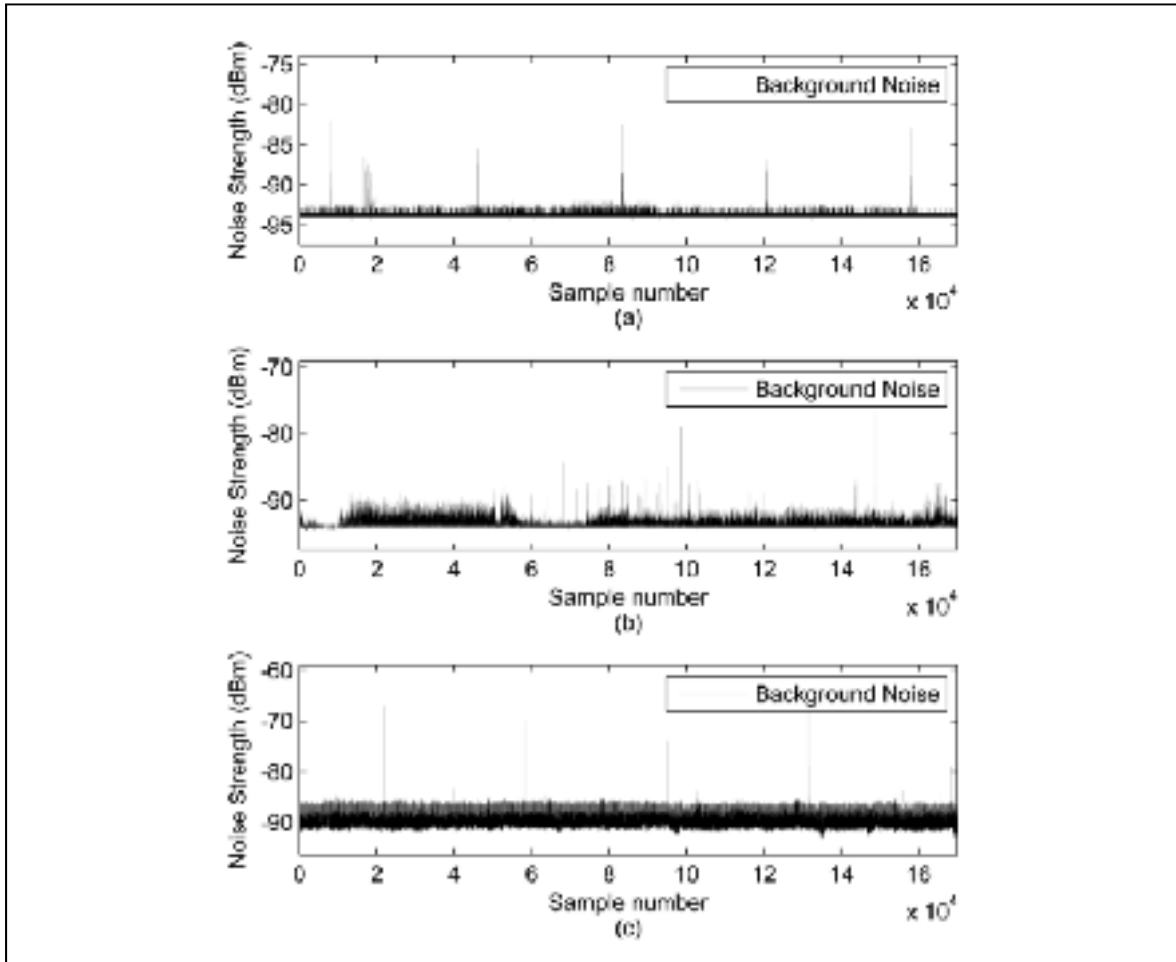


Figure 4.14 Background noise samples measured in three distribution environments. (a) Power grid distribution substation. (b) Low-voltage transformer. (c) Distribution lines in a residential area

Taken from W. Sun (2018)

As reported in the measurement results in (W. Sun, 2018) and in Figure 4.14, the ambient noise power density in a given frequency band does vary over time and from one environment to

another. However, as shown in Figure 4.15, the relative frequency distributions of the measured background noise in three different environments show a clear narrow peak with a 2-3 dB deviation.

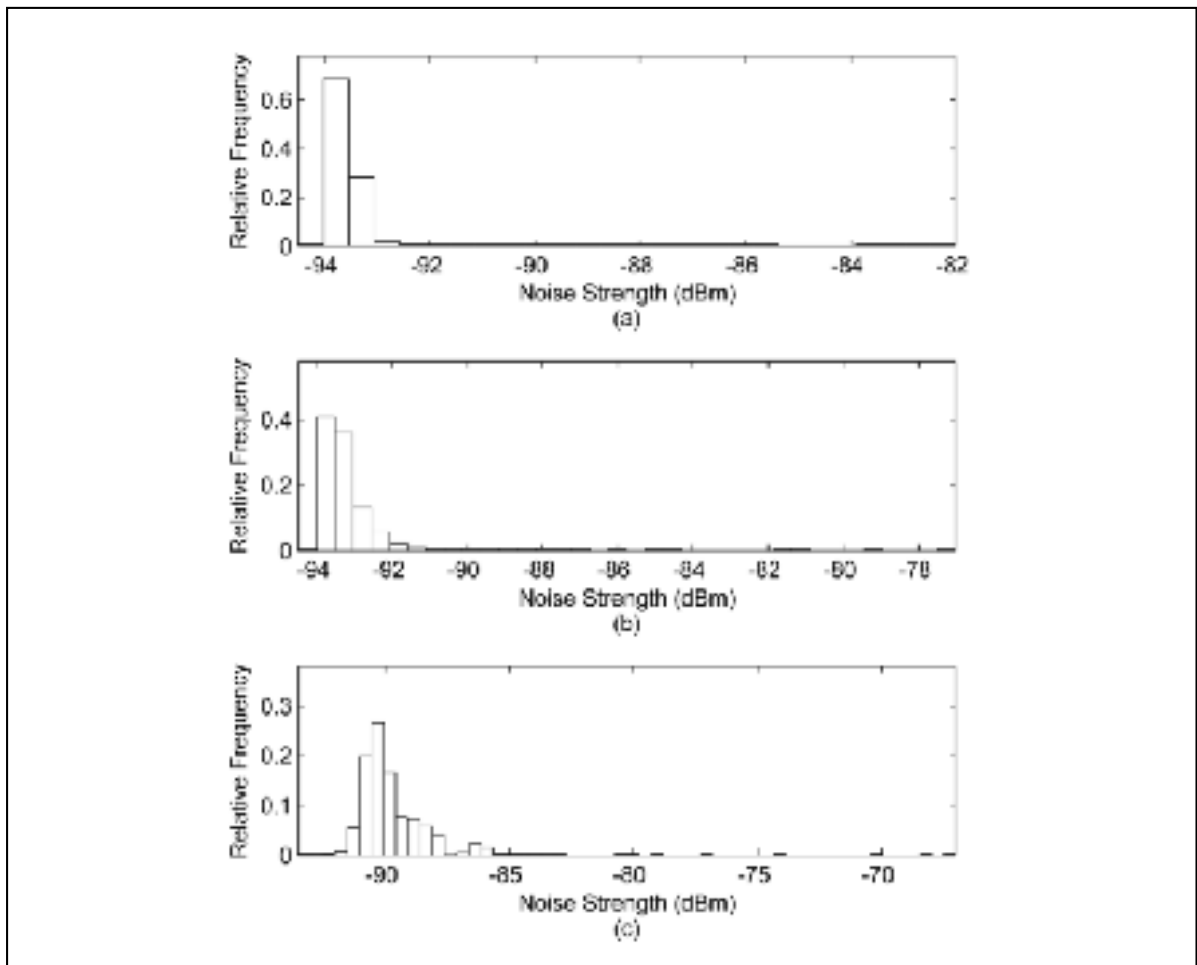


Figure 4.15 Relative frequency distributions of the background noise strengths measured in three distribution environments. (a) Power grid distribution substation (b) Low-voltage transformer. (c) Distribution lines in a residential area
Taken from W. Sun (2018)

Therefore, knowing the ambient noise power density in the deployment site can allow the designer to make more founded decisions about the right hardware and node configuration especially when a realistic model is available (W. Sun, 2018).



Figure 4.16 Measurement setup of the ambient noise

Figure 4.16 shows the measurement setup of the ambient noise density N_A by using the MS2721A spectrum analyzer. It is given by:

$$N_A = k_b T + \Delta N \quad (4.10)$$

where k_b is the Boltzmann constant, T is the ambient temperature, and ΔN is the noise density arising from other interfering emissions in the same frequency band. The spectrum analyzer's noise marker functionality was used to get a 1 Hz resolution bandwidth measurement. Although the measurement setup is different, the reader can refer to (Robert Leck, 2006) to better understand the measurement methodology. Noise densities of -151.11 dBm/Hz and -154.27 dBm/Hz were measured in the urban and suburban areas respectively. The 3.16 dB difference can be explained by the fact that the urban area is more industrialized and densely populated area. A 3 dB difference between the two environments is also reported in (A. U. H. Sheikh, 1983). Moreover, similar noise power measurements were carried out in (W. Sun, 2018) and the results are presented in Figure 4.14. It shows that in a residential area (i.e. suburban), noise power is around -90 dBm. Knowing that a CC2530 SoC and a 384 kHz

bandwidth was used to measure the signal strength, the noise density is therefore around -146 dBm/Hz. If a receiver noise figure of 7 dB is assumed as it is the case for the CC1310 (Sivan Toledo, 2018), the ambient noise density would become -153 dBm/Hz which further validates the measurements reported in this thesis (-154.27 dBm/Hz).

4.2.2 Communication Range Outdoor Measurements

In this section, the link quality characterization of a point-to-point communication link deployed in different outdoor environments (i.e. urban and suburban areas) is presented. Two different output power levels on the transmitter's side (i.e. -10 dBm and 0 dBm) and two different data rates on the transmitter's and receiver's sides (i.e. 50 Kbps and 500 Kbps) are used in order to analyze how link reliability, latency, and energy consumption can be traded-off against each other.

4 Bytes	4 Bytes	1 Byte	20 Bytes	2 Bytes
Preamble	Sync word	Packet length	Data	CRC

Figure 4.17 Packet format used in outdoor measurements

The total packet size was set to 31 Bytes as depicted in Figure 9 and for each measurement, 400 packets are sent.

4.2.2.1 Suburban Area

It should be noted that the communication range outdoor measurements were carried out while making sure that no objects nor people interrupted the LOS wireless link. Moreover, while taking the measurements, the ambient temperature was within the range of 10 – 20 degrees Celsius. These measurements were conducted in a residential area in the city of Montreal. Specifically, inside an open baseball field of a public park. The field measurement setup in the suburban area is presented in Figure 4.18.



Figure 4.18 Google satellite image of the field measurement setup in the suburban area

4.2.2.2 Urban Area

As a second step, and in order to capture the impact of the deployment site on the performance of the WSN, measurements were also conducted in a more dense and industrialized area in the same city. Specifically, beside a canal not far from downtown. The field measurement setup in the urban area is presented in Figure 4.19.



Figure 4.19 Google satellite image of the field measurement setup in the urban area

4.2.2.3 Experimental Results

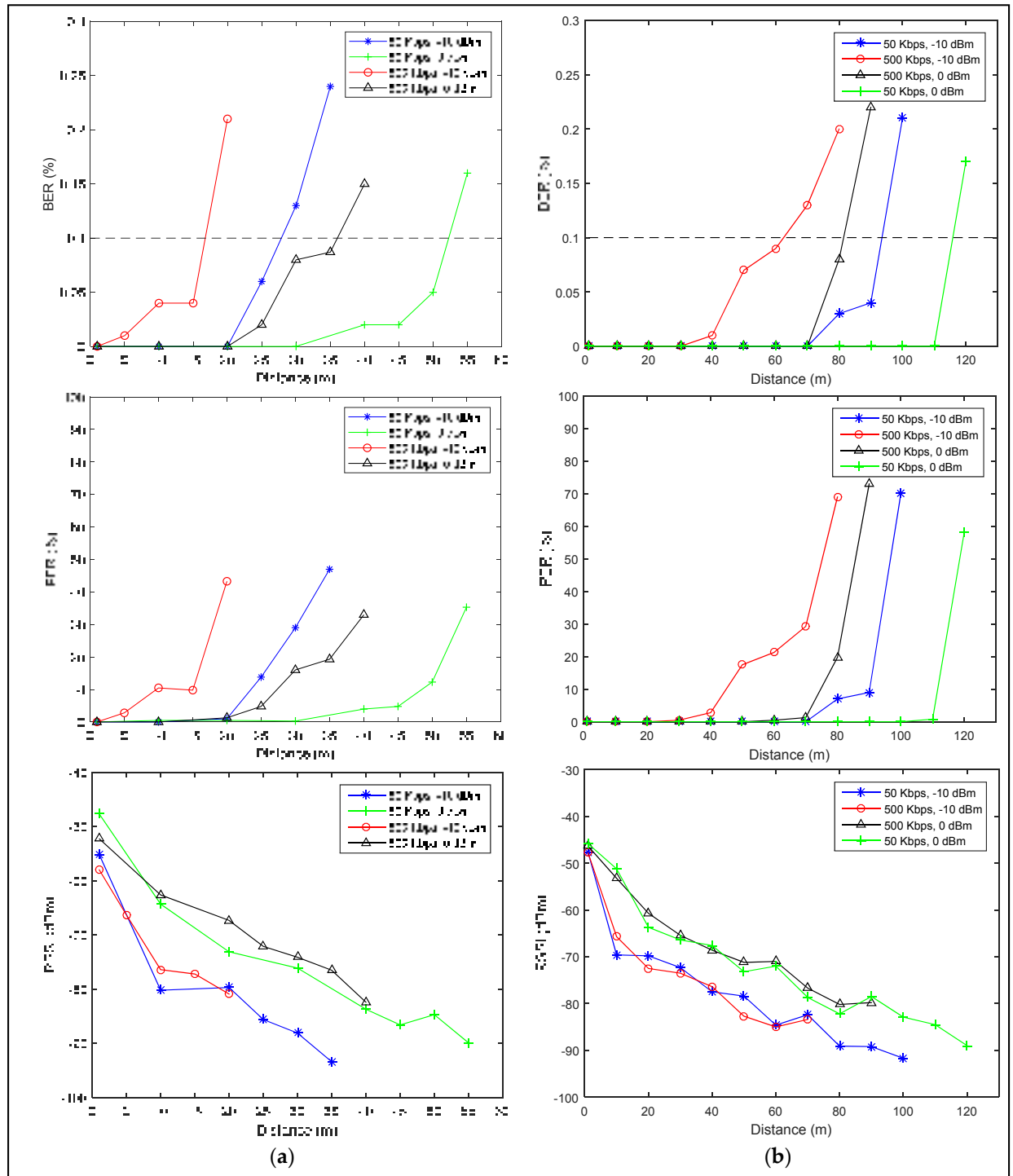


Figure 4.20 BER, PER, and RSSI field measurements in (a) the urban and (b) the suburban areas

Figure 4.20 shows the results of the measurements done in the urban and suburban areas. It can be seen that increasing the bandwidth B in the urban area to achieve a higher data rate has a more pronounced impact on range (i.e. $BER \leq 0.1$) when compared with the suburban area, which can be explained by the previously measured higher noise density. The noise power in a given bandwidth is expressed by:

$$P_{NOISE} = N_A + 10 \log(B) \quad (4.11)$$

It was previously mentioned that using high data rates would result in a better energy consumption as depicted in Figure 4.10. However, the achievable communication range was yet unknown. Figure 4.20 shows that, at 0.1 % BER as required by Bluetooth applications, similar communication ranges can be achieved by increasing the output power and data rate (i.e. 500 Kbps and 0 dBm) or decreasing them (i.e. 50 kbps and -10 dBm). However, Figure 4.10 shows that the energy consumption per measurement when the first configuration is used is three times less than when using the second. Therefore, the previously made assumption of the need to increase data rate in order to achieve a better energy efficiency while covering the desired communication range is judged to be logical and valid.

4.2.3 Path Loss Model

For outdoor applications, the most common multipath signals are caused by ground reflections (M. Bacco, 2018). The fitting factor proposed in (C. Phillips, 2013) for LOS communications systems is also used with a variable fitting coefficient l_f . The proposed path-loss model is expressed as follows:

$$PL = 20 \log\left(\frac{4\pi d}{\lambda}\right) - 20 \log\left[2 \sin\left(\frac{2\pi h_t h_r}{\lambda d}\right)\right] + l_f \log(50d) \quad (4.12)$$

where d is the distance between the two nodes, λ is the wavelength, h_t and h_r are the distances that separate the transmitter and the receiver from the ground respectively, and l_f is a fitting coefficient.

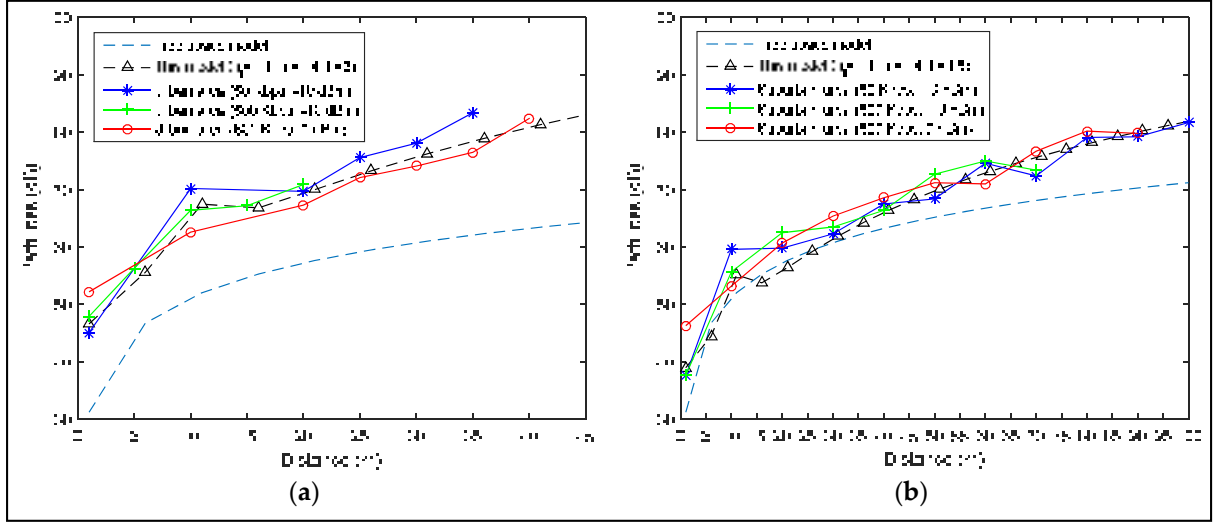


Figure 4.21 The measured path-loss versus the free-space and estimated ones in the (a) urban and (b) suburban areas

Figure 4.21 shows that the proposed model can accurately estimate the power at the receiver when compared with the free space path loss (FSPL) model. The latter is given by:

$$FSPL = 20 \log \left(\frac{4\pi d}{\lambda} \right) \quad (4.13)$$

In the urban area, a fitting coefficient $l_f = 6$ was used. However, in the suburban area, $l_f = 1.5$. Moreover, the ground reflection is well-modeled at a distance $d \approx 10m$ when the transmitter and the receiver are 1.1m and 1.4m away from the ground respectively. As no antenna gains were considered on the transmitter and receiver sides, the experimental path-loss PL_{exp} was calculated using:

$$PL_{exp} = P_t - RSSI_{avg} \quad (4.14)$$

where P_t is the output power level. Figure 4.21 also shows that the FSPL can provide an acceptable estimation of the received power up to 30 m in the suburban area. For longer distances, the FSPL presents an unacceptable error. However, in the urban area, the FSPL model drastically deviates from the experimental results and the proposed model fits.

4.3 Data Rate Maximization

Now that the received power has been accurately estimated using the proposed path-loss model, the noise detected at the receiver also needs to be accounted for to determine the maximum achievable rate. The SINR needs to be kept sufficiently large to ensure correct demodulation with a given bit error probability threshold required by the application. To this end, the system's noise power density N_{sys} (i.e. the sum of all unwanted signals that contaminate the signal of interest) in the deployment site needs to be determined. The SINR is given by:

$$SINR = \frac{P_r}{N_{sys}} = \frac{RSSI}{N_A + N_{fg}} \quad (4.15)$$

where P_r and RSS denote the received power and N_{fg} is the noise figure of the receiver. In this case, for the used CC1310 WMCU, $N_{fg} = 7dB$ (Sivan Toledo, 2018).

Using the BER performance curve (Hiroshi Harada, 2017; K. Mhlanga, 2015) of the used modulation scheme (i.e. GFSK in this case with a modulation index $m = 1$) as depicted in Figure 4.22, the required energy per bit to noise ratio can be determined to achieve the target BER. Then, using:

$$\left[\frac{P_r}{N_{sys}} \right]_{dB} = \left[\frac{E_b}{N_{sys}} \right]_{dB} + [D_R]_{dB} \quad (4.16)$$

The achievable data rate can be determined. An example is included in Section 4.5.2.

It should be noted that in WSN applications, D_R and P_t are dynamically changed during operation (D. Singh, 2018; P. Sommer, 2018). However, in this work, it is also important to make estimations early in the design process in order to make high-level decisions about the hardware and node configuration to use.

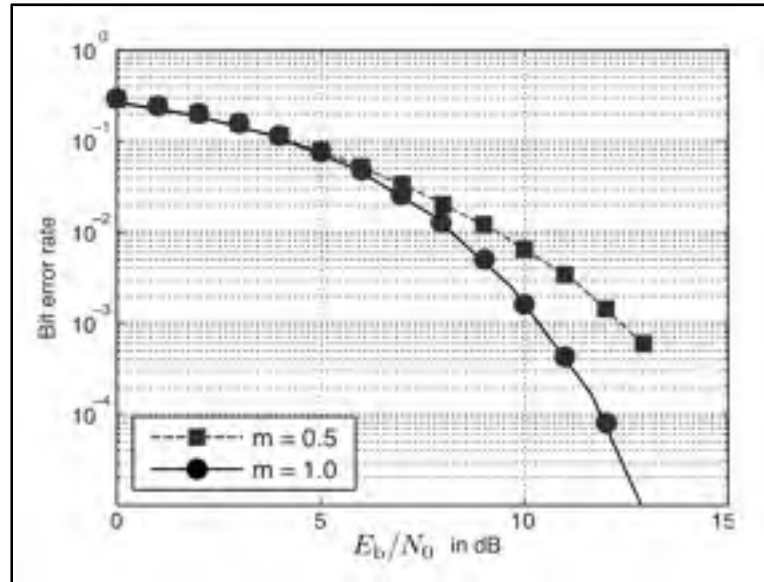


Figure 4.22 BER performance of GFSK
Taken from Hiroshi Harada (2017)

Now that the link is optimized for energy consumption, communication range, and throughput, packet loss needs to be mitigated in order to achieve the required reliability target. Moreover, the latter must be met with latency and energy costs in mind.

4.4 Energy-Reliability-Latency Trade-Off

In WSNs, finding the energy-latency-reliability trade-off is very important since it captures the interdependence of key parameters from a QoS point of view. The goal of this section is to determine the successful reception probability of a packet when using different data transfer schemes and when at most R transmissions are allowed for each packet.

4.4.1 Current Consumption Profile of Different Data Transfer Schemes

In this section, the current consumption profile of the transmitter and the receiver are considered when using simple transmissions, convolutional FEC coding, BR, and ARQ-based retransmissions. Figure 4.23b shows a considerable transmission time and energy overheads when using FEC as the payload is doubled due to coding when compared with Figure 4.23a. Similarly, transmitting the packet twice as shown in Figure 4.23c or waiting for an ACK as depicted in Figure 4.23d has more pronounced energy and time overheads.

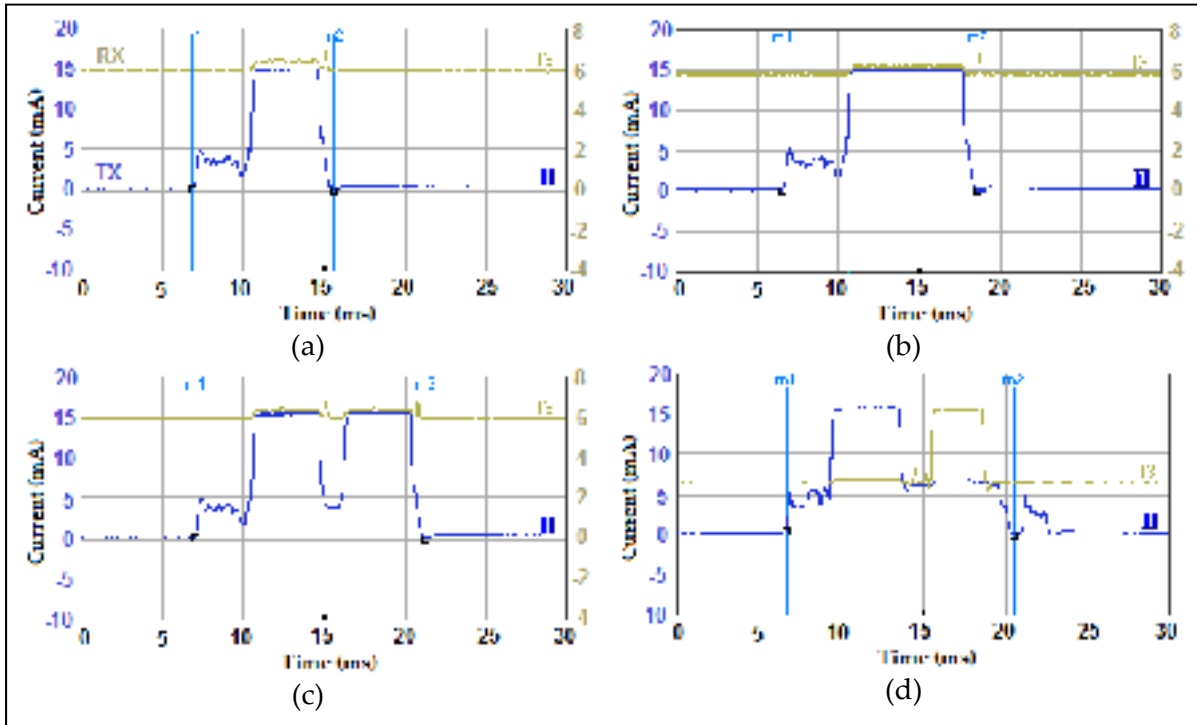


Figure 4.23 Current consumption profile of the transmitter and the receiver when using different data transfer schemes: (a) simple transmissions, (b) FEC, (c) two BR, and (d) a SAW-ARQ protocol (also referred to as 1-Rep-ACK in this work)

It should be noted that, apart from the transceiver overhead, Figure 4.24 reports that the MCU energy consumption per measurement does vary when using different data transfer schemes. An energy overhead can be observed when using FEC and the SAW-ARQ protocol due to additional data encoding and packet processing, respectively. However, when compared to the

transceiver's contribution which is investigated in two case studies presented in Section 4.5.2, it is safe to say that the MCU's energy consumption does not make a big difference.

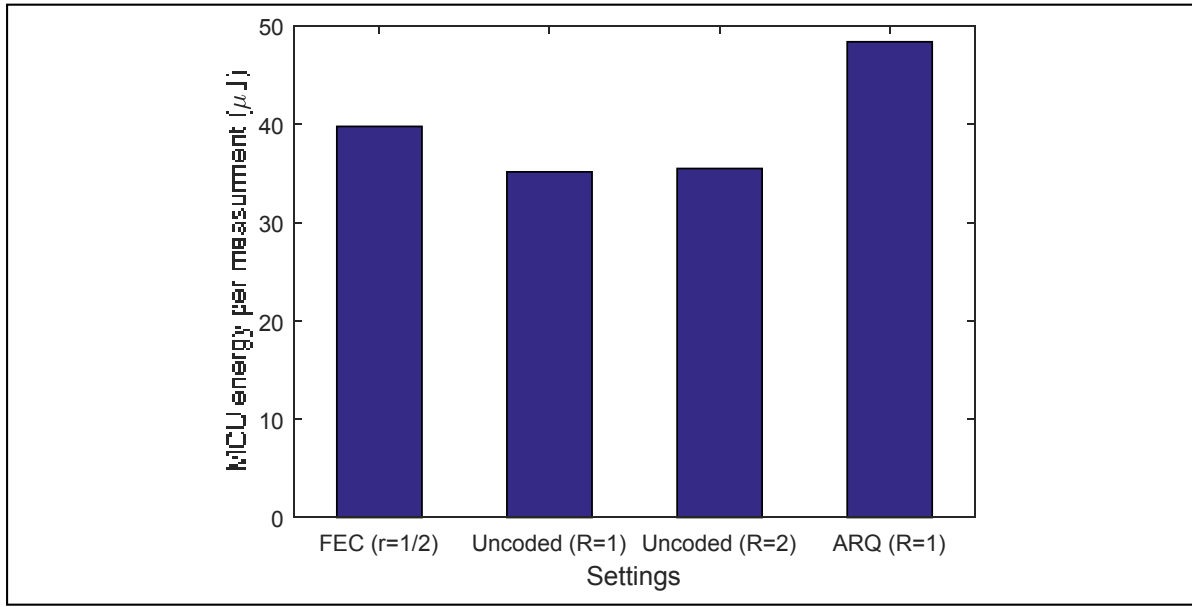


Figure 4.24 MCU energy consumption per measurement for different data transfer schemes

All of the previously mentioned techniques (i.e. FEC, BR, and SAW-ARQ) can manage bit errors. Yet, the packet success probability needs to be determined for each technique in order to objectively investigate their usefulness.

4.4.2 Packet Success Probability

In order to fairly compare the three different data transfer schemes from an energy, latency, and reliability point of view simultaneously as will be discussed in Section 4.5.2, the packet success probability must be estimated first. In the probabilistic binary symmetric channel (BSC) model, a bit is independently flipped with a bit error probability P_e during transmission. If node A is sending packets to node B through a BSC, where all packets have the same size of N bits, the probability that a packet from A arrives error-free at B is given by:

$$P = (1 - P_e)^N \quad (4.17)$$

This probability can be enhanced by using error correction techniques. Namely, cases of FEC, BR, and SAW-ARQ retransmissions are investigated in this work.

4.4.2.1 Convolutional FEC

FEC is being widely used to cope with the random erroneous bits in a data packet. As an example, a convolutional encoder (n, k, K) is implemented by adding $n - k$ redundant bits to the actual k bits of data. The number of bits upon which the encoder's output depend K is called the constraint length or depth of the code.

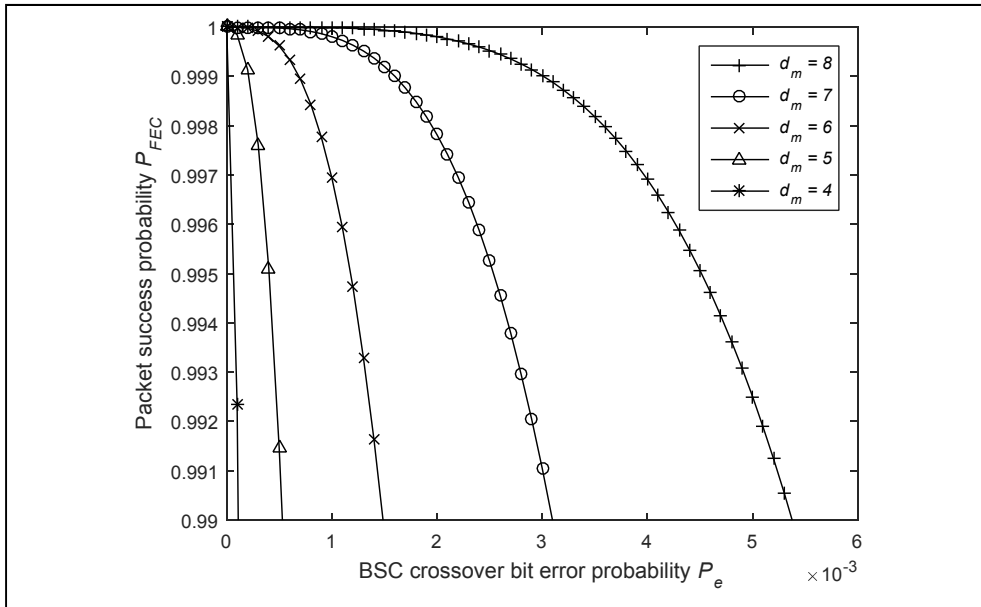


Figure 4.25 Impact of the free distance d_m on the packet success probability

The ratio $r = \frac{k}{n}$ is called the code rate. The performance of a convolutional code is also characterised by its free distance d_m which is the minimal hamming distance (R. W. Hamming, 1950) between different encoded sequences. This means that changing one bit in the message sequence will change at least d_m bits in the coded output sequence (Robin Hoel, 2007). Figure 4.25 shows the impact of the free distance on the performance of the code. The asymptotic coding gain that can be achieved can be expressed as follows (Robin Hoel, 2007):

$$G_{FEC} = 10 \log(d_m r) \quad (4.18)$$

In this work, a convolutional code with a rate $r = \frac{1}{2}$, a constraint length $K = 4$, and free distance $d_m = 6$ is considered (Robin Hoel, 2007). Therefore, the used code has a theoretic asymptotic coding gain of 4.77 dB. However, the real achievable gain is considerably less and it is generally between 2 and 3 dB (G. Angelopoulos, 2013). Moreover, the error correcting ability of a code depends on the decoding method. The probability that a bit will be received in error when using convolutional FEC and Viterbi decoding without considering the influence of an interleaver (Y. Cai, 2019; R. Swaminathan, 2016) to cope with burst errors can be approximated using (D. Wang, 2017):

$$P_e^{FEC} \approx \frac{\beta_{free}}{k} \left(2\sqrt{P_e(1-P_e)} \right)^{d_m} \approx \frac{\beta_{free}}{k} 2^{d_m} P_e^{\frac{d_m}{2}} \quad (4.19)$$

where β_{free} is the total number of non-zero information bits of all paths with a weight of d_m . It should be noted that β_{free} depends on the data pattern and is totally random. Figure 4.26 shows the impact of β_{free} on the performance of the error correcting code. For the sake of simplicity, in this study, β_{free} was fixed at 200 which is logical when using payloads of 160 bits. In this case, the probability that a packet from A arrives error-free at B is given by:

$$P_{FEC} = (1 - P_e^{FEC})^N \quad (4.20)$$

When FEC is used, the required time to send a packet is given by:

$$t_{TRX}^{FEC} = \frac{l^{FEC}}{D_R} \quad (4.21)$$

where l^{FEC} is the packet length when using FEC.

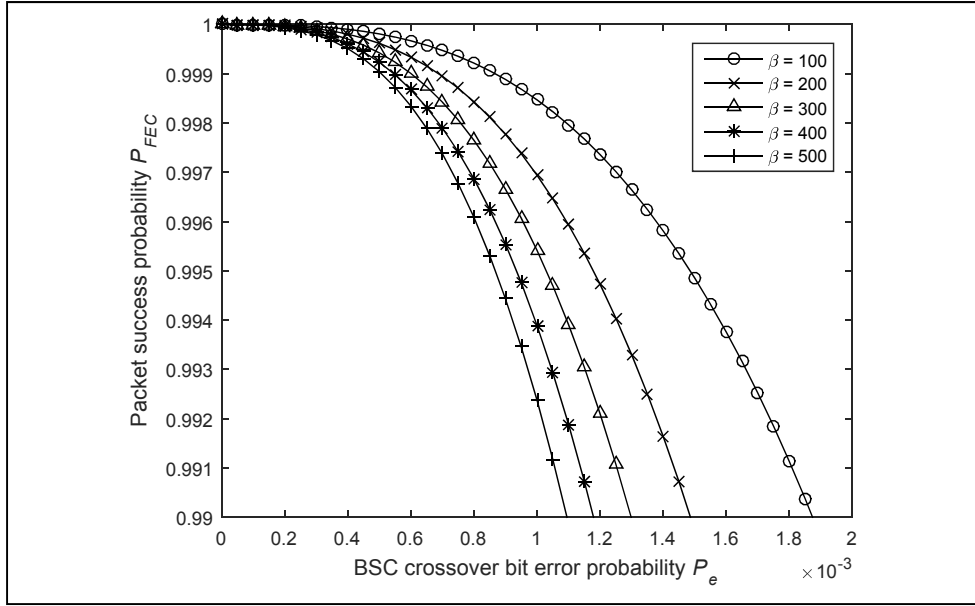


Figure 4.26 Impact of the β_{free} parameter on the packet success probability

Therefore, the energy consumption of the transceiver can be expressed as follows:

$$E_{TRX}^{FEC} = V t_{TRX}^{FEC} (I_{TX} + I_{RX}) \quad (4.22)$$

4.4.2.2 Blind Retransmissions (BR)

In this case, node A blindly transmits R times the packet with a packet successful reception probability P over each transmission. The number of successful packet transmissions h after R independent transmission trials is a random variable S that follows the binomial distribution with parameters $R \in \mathbb{N}$ and $P \in [0,1]$ and given by the probability mass function:

$$f(h, R, P) = pr(S = h) = \binom{R}{h} P^h (1 - P)^{R-h} \quad (4.23)$$

In accordance with (Teerawat Issariyakul, 2006; V. P. Mhatre, 2006), the probability that at least one packet is received successfully within R independent attempts can be calculated using:

$$P_R = 1 - pr(S = 0) = 1 - (1 - P)^R = 1 - (1 - (1 - P_e)^N)^R \quad (4.24)$$

Figure 4.27 shows the impact of R on the packet success probability.

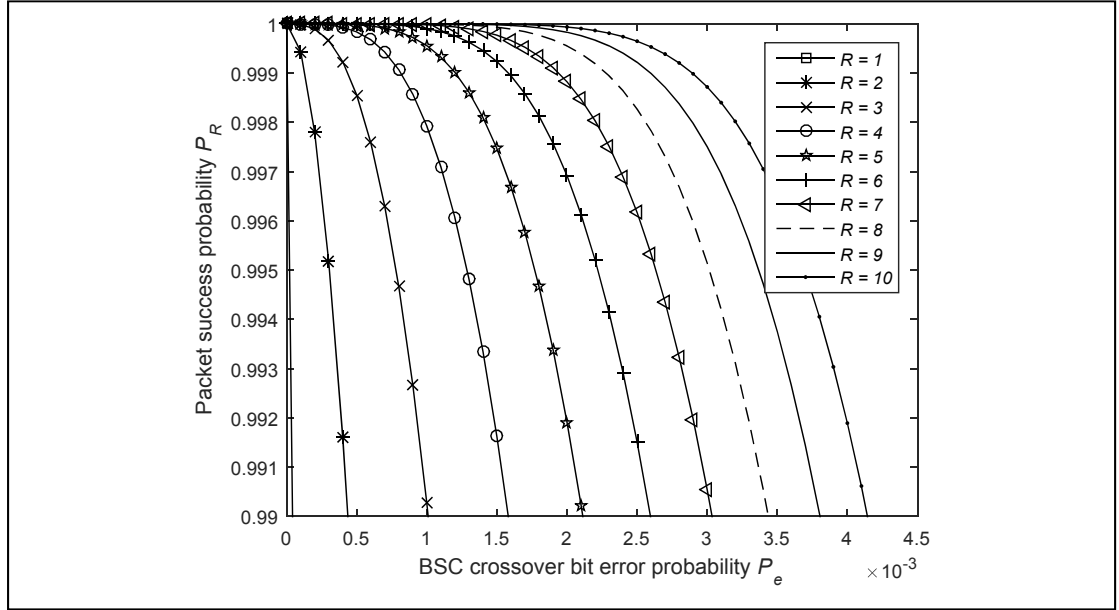


Figure 4.27 Impact of the number of blind transmission attempts on the packet success probability

When using BR, the required time to send a packet is given by:

$$t_{TRX}^R = R \frac{l}{D_R} + t_{id}(R - 1) \quad (4.25)$$

where t_{id} is the time spent in idle mode between two packet transmissions. Therefore, the energy consumption of the transceiver can be expressed as follows:

$$E_{TRX}^R = V \left[R \frac{l}{D_R} (I_{TX} + I_{RX}) + 2I_{id}t_{id}(R - 1) \right] \quad (4.26)$$

4.4.2.3 ARQ Retransmissions

In this section, the notation \bar{z} is used to denote $\bar{z} = 1 - z$ where the real valued variable $z \in [0,1]$. In this analysis, the L -Rep-ACK (Saeed R Khosravirad, 2017) protocol which achieves a better feedback channel reliability by retransmitting ACK and NACK packets is considered. Case studies presented in 4.5.2 include examples and discussions about the matter. The transmission is considered successful only if the transmitter receives L ACK packets for one data packet where $L > 1$. It should be noted that the particular case where $L = 1$ is the regular SAW-ARQ protocol. The same bit error probability P_e during transmission of ACK and NACK packets of $N_f = 40$ bits through the feedback channel is considered. The probability of successfully receiving a data packet at B after a maximum of R L -Rep-ACK transmissions is given by:

$$P_R^{L-Rep-ACK} = 1 - P_{out}^{L-Rep-ACK} \quad (4.27)$$

where L is the time diversity order (i.e. number of ACK/NACK transmissions) and P_{out}^{ARQ} is the outage probability (Saeed R Khosravirad, 2017) of the L -Rep-ACK protocol and given by:

$$P_{out}^{L-Rep-ACK} = \sum_{j=1}^{R-1} \bar{P}^j P_{pe}^L (1 - P_{pe}^L)^{j-1} + \bar{P}^R (1 - P_{pe}^L)^{R-1} \quad (4.28)$$

where j is the index of the transmission attempt and $P_{pe} = \bar{P}_f = 1 - (1 - P_e)^{N_f}$ is the failure probability of an ACK packet. Figure 4.28 shows the impact of the maximum allowed number of 1-Rep-ACK transmission attempts R on the packet success probability.

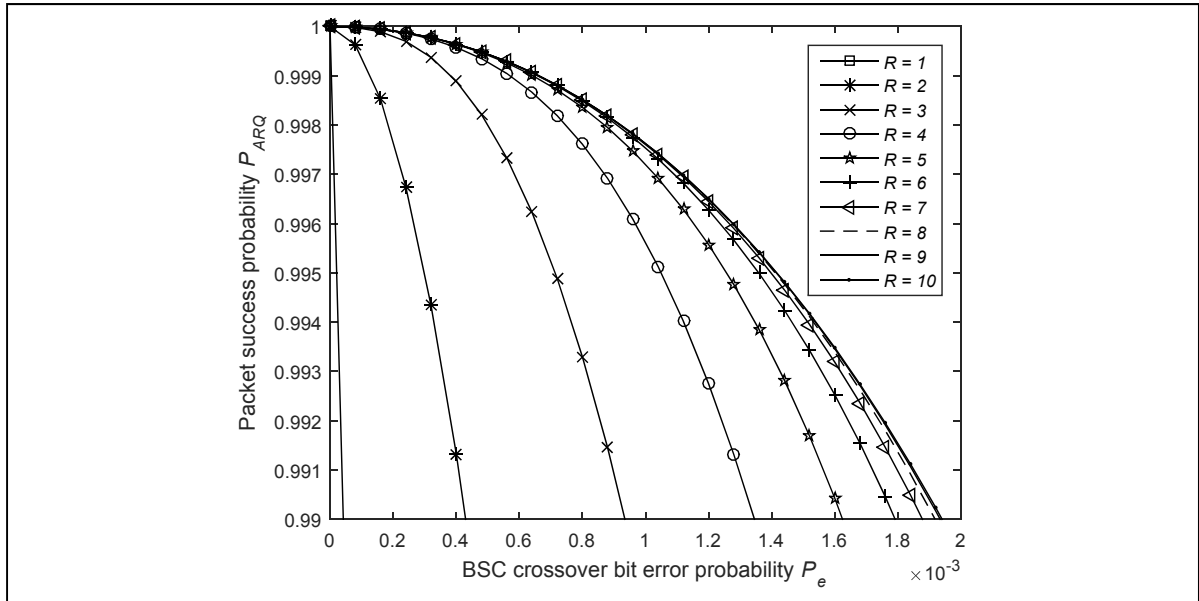


Figure 4.28 Impact of the number of 1-Rep-ACK transmission attempts on the packet success probability

Moreover, as illustrated in Figure 4.29, using a maximum of R ARQ transmissions does not mean that R is always reached and that is the reason why it is possible to achieve better energy and latency efficiencies when using acknowledgements. It is noticeable that for a bit error probability $P_e = 10^{-3}$, a packet is received at the first, second, or third transmission attempt with a probability of 0.85, 0.13, 0.025 respectively.

The probability that a packet will be received successfully at the j^{th} packet transmission attempt is given by:

$$P_j^{L-Rep-ACK} = P_R^{L-Rep-ACK} \times (1 - P_R^{L-Rep-ACK})^{j-1} \quad (4.29)$$

When using L -Rep-ACK retransmissions, the required time to send a packet is given by

$$t_{TRX}^{ARQ} = \sum_{j=1}^R P_j^{ARQ} \left[j \left[\left(\frac{l_{data}}{D_R} + \frac{L l_{ack}}{D_R} \right) + L t_{sb} \right] + t_{id}(j-1) \right] \quad (4.30)$$

where t_{sb} is the time spent in standby mode after sending the packet and before receiving the acknowledgement and l_{ack} is the length of the acknowledgement packet.

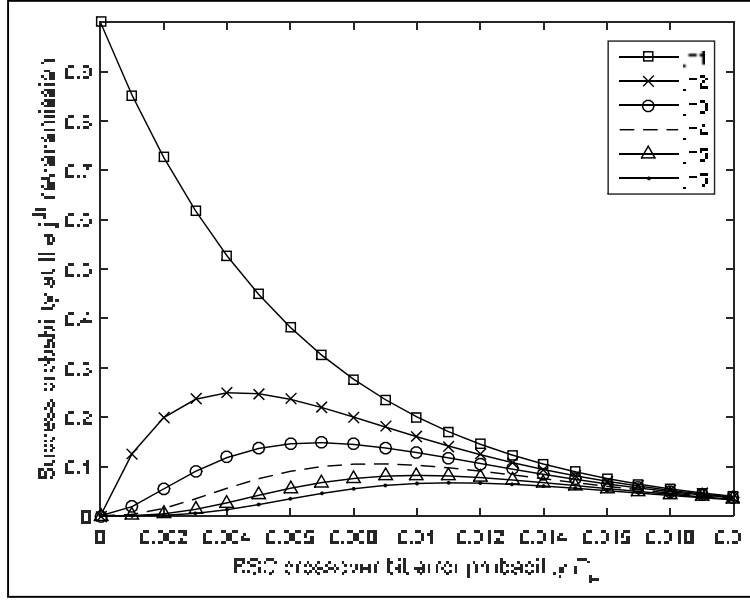


Figure 4.29 Packet success probability at the j^{th} packet

Therefore, the energy consumption of the transceiver can be expressed as follows:

$$E_{TRX}^{ARQ} = V \sum_{j=1}^R P_j^{ARQ} \left[j \left[\left(\frac{l_{data}}{D_R} + \frac{L l_{ack}}{D_R} \right) (I_{TX} + I_{RX}) + 2 L I_{sb} t_{sb} \right] + 2 I_{id} t_{id} (j - 1) \right] \quad (4.31)$$

At this point, using the previous analytical results, the three reliable data transfer schemes can be objectively compared while taking into account the achievable reliability the energy and latency costs.

4.5 Resulting Design Methodology and Case Studies

This section explains how all of the previous steps are logically linked together to form the system-level design methodology. Moreover, case studies and examples further demonstrate how the methodology can be applied.

4.5.1 Design Flow and Methodology

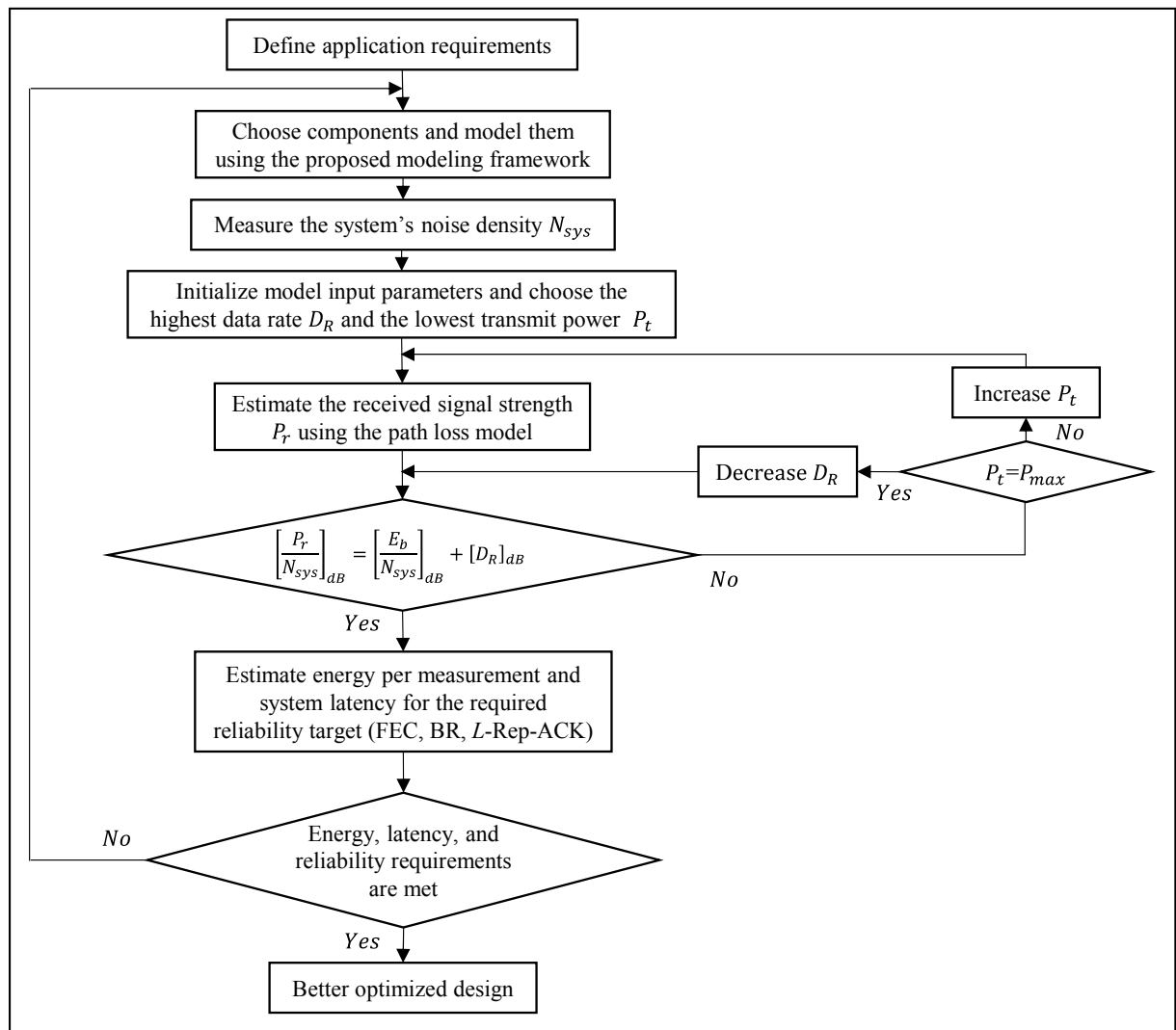


Figure 4.30 SN design flow graph

Early in the design process, a SN designer would follow the steps shown in the design flow graph presented in Figure 4.30 in order to make high-level decisions about the right COTS components to use and the possibility to meet the reliability, energy, and latency requirements. At this point, all of the model inputs which are outlined in Table 4.1 should be available. Then, the designer needs to determine the system's noise density by measuring the background noise in the deployment site and taking into account the receiver's noise figure. Before estimating the energy consumption per measurement, it should be made sure that an acceptable communication range can be achieved. Therefore, starting by using the highest data rate and the lowest output power level on the transceiver, the designer estimates the range and iterates between the two parameters while always maximizing the data rate.

4.5.2 Case Studies

Considering the measurements reported in Sections 4.2, the example illustrated in Table 4.6 shows how to estimate the maximum achievable data rate when the input parameters are known.

Table 4.6 Example explaining how to estimate the achievable data rate

	Parameter	Value
Inputs (based on data sheet information, measurements, and application requirements)	N_A	-154.27 dBm/Hz
	$N_{fg}(\text{CC1310})$ (Sivan Toledo, 2018)	7 dB
	BER (requirement)	0.1 %
	P_t	-10 dBm
	d (requirement)	95 m
	$\frac{E_b}{N_{sys}}$ (at 0.1 % BER) (Hiroshi Harada, 2017; K. Mhlanga, 2015)	10.3 dB
	PL (This model at 95 m)	80.5 dB
Results	P_r (Equation 14)	-90.5 dBm
	$SINR$ (Equation 15)	56.77 dBm
	D_R (Equation 16)	44.668 kbps

First, the received power P_r is estimated using the PL model considered at the targeted range. Second, the SINR is determined. Finally, the achievable data rate is calculated. In this case, the latter is estimated to be around 44.668 kbps. However, when compared to the measurements reported in Figure 12b, an error of 10.66% is observed as the actual data rate is 50 kbps. This is acceptable as noise and signal attenuation do vary over time. Therefore, in WSN applications, D_R and P_t are dynamically changed during operation (D. Singh, 2018; P. Sommer, 2018). Theoretically, increasing P_t by 10 dBm (i.e., $P_t = 0 \text{ dBm}$) would result in achieving the same range at 500 kbps and therefore lowering the energy per measurement from 213 μJ to 62 μJ as shown in Figure 4.10. However, practically, the measurements in figure 4.20 show that the range becomes around 80 m which is 15 m shorter (i.e., a 15.8% range reduction) than the theoretically estimated range of 95 m. This can be tackled by further increasing P_t which will always be a better compromise for the CC1310 radio chip in terms of latency and energy consumption as shown in Figure 4.10. Moreover, it is safe to say that this is also true for a wide range of current generation and similar wireless transceivers in the market.

4.5.2.1 99% Reliability Target

Like in (P. Kong, 2017), the reliability in this work is directly linked to the packet success probability P . Therefore, in order to evaluate the data transfer reliability of a point-to-point link, assume an application's requirement of 0.99 in terms of packet success probability is assumed, which corresponds to a 99% reliability figure of merit.

This level of reliability is required by applications such as utility-to-consumer real time pricing, outage management, and automated feeder switching in smart grid (V. C. Gungor, 2013; W. Sun, 2018). Also, a forward and feedback channel bit error probability $P_e = 10^{-3}$ is assumed.

Figure 4.31 Shows that the requirement is met by the studied FEC code. When using BR, only two retransmissions are needed ($R = 3$). However, the 2-Rep-ACK approach which is also affected by the imperfect feedback channel, requires three retransmissions ($R = 4$) as two

retransmissions ($R = 3$) are not enough. In fact, for a feedback channel having the same packet error probability as the forward's ($L = 1$), a large number of 1-Rep-ACK transmissions (i.e. $R > 20$) is required as shown in Figure 4.32. As such for identical forward and feedback error probabilities, it is appropriate to consider $L > 1$ for the ACK transmissions.

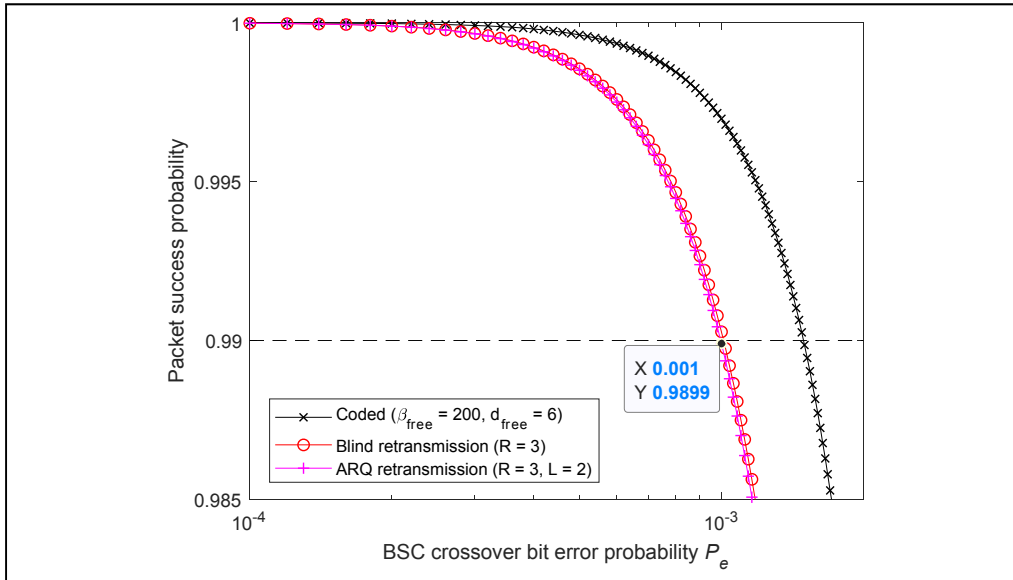


Figure 4.31 Meeting 99% reliability target by using FEC, BR, and 2-Rep-ACK retransmissions ($L = 2$)

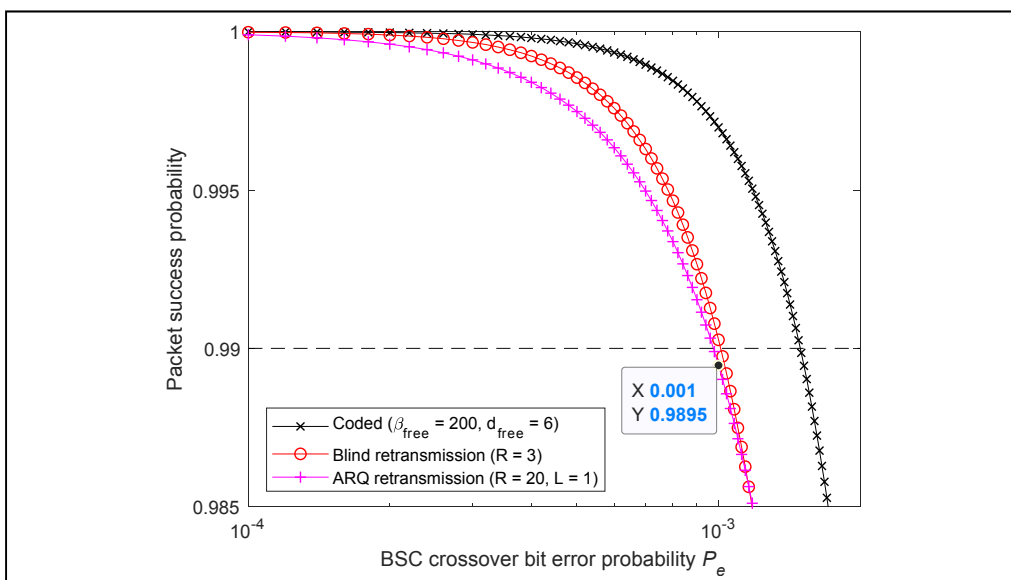


Figure 4.32 Meeting 99% reliability target by using FEC, BR, and 1-Rep-ACK retransmissions ($L = 1$)

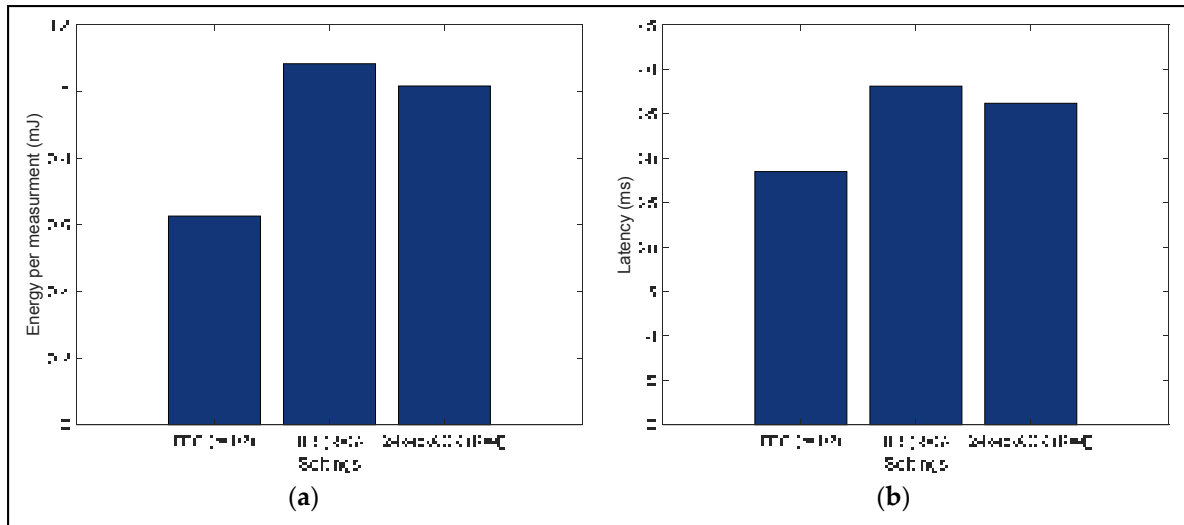


Figure 4.33 (a) Energy consumption per measurement and (b) system latency for 99% reliability target

The overall energy consumption per measurement E_{sys} is given by:

$$E_{sys} = E_{TRX} + E_{MCU} + E_{SENSOR} \quad (4.32)$$

Moreover, the latency of the wireless link t_{sys} can be calculated using:

$$t_{sys} = t_{TRX} + t_{PROCESS} + t_{SENSOR} \quad (4.33)$$

The energy per measurement results presented in Figure 4.33a were obtained using Equation (4.32). It is assumed that the sensor's and MCU's energies are not changed from one data transfer scheme to another. Moreover, the results in Figure 4.33b were obtained using Equation (4.33). Both figures show that when the required reliability is around 99%, using FEC is the least expensive choice in terms of energy consumption and latency.

Therefore, by way of example, without applying this methodology, a designer could have used the regular SAW-ARQ and not met the required reliability. Moreover, if BR were arbitrarily used ($R=3$) to ensure the 99% reliability requirement, the energy consumption and latency would increase by around 75% and 32% respectively when compared with opting for FEC.

4.5.2.2 99.999% Reliability Target

A requirement of 0.99999 in terms of packet success probability is now assumed which corresponds to a 99.999% reliability figure of merit.

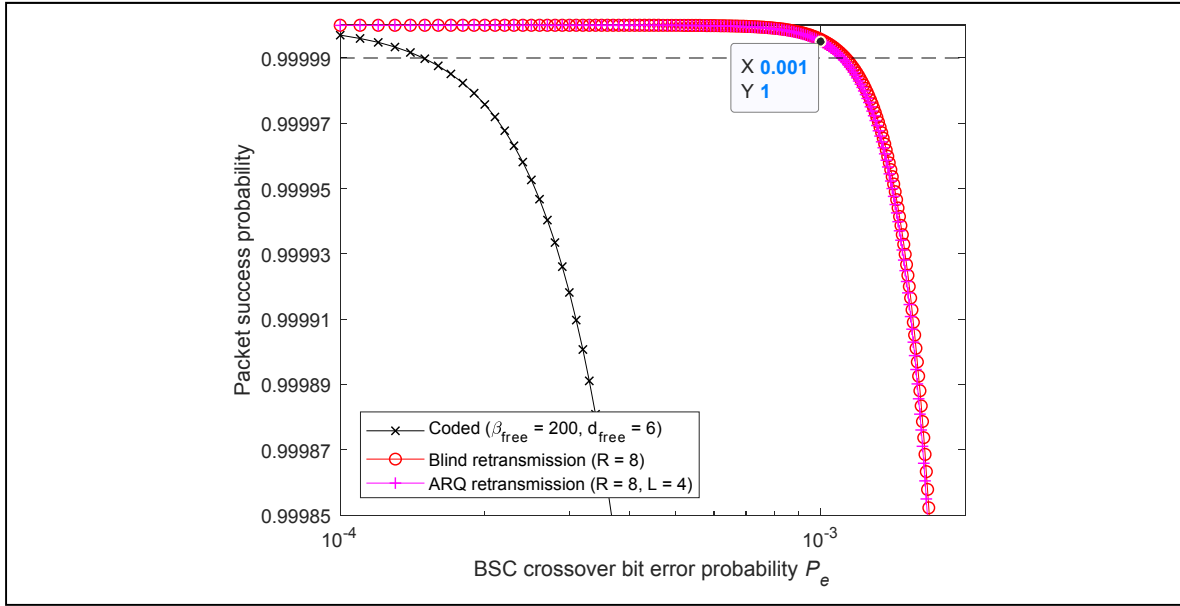


Figure 4.34 Meeting 99.999% reliability target by using BR and 4-Rep-ACK retransmissions

This level of reliability is required by applications such as industrial IoT (Matthias Herlich, 2018; K. Lee, 2018; G. Pocovi, 2018) and wide area situation awareness (P. Kong, 2017). Also, a forward and feedback channel bit error probability $P_e = 10^{-3}$ is assumed.

Figure 4.34 and figure 4.35 show that the requirement is no longer met by the studied convolutional FEC code. Therefore, a code presenting a longer free distance d_m is required. Furthermore, it can be concluded from Figure 4.34 that when using BR, seven retransmissions are needed ($R = 8$) and when using 4-Rep-ACK, seven retransmissions are required ($R = 8$). However, for the latter, that can be achieved only when $L \geq 4$. Figure 4.35 shows that with $L = 3$, a large number of 3-Rep-ACK transmissions (i.e. $R > 20$) are required making this L value unviable.

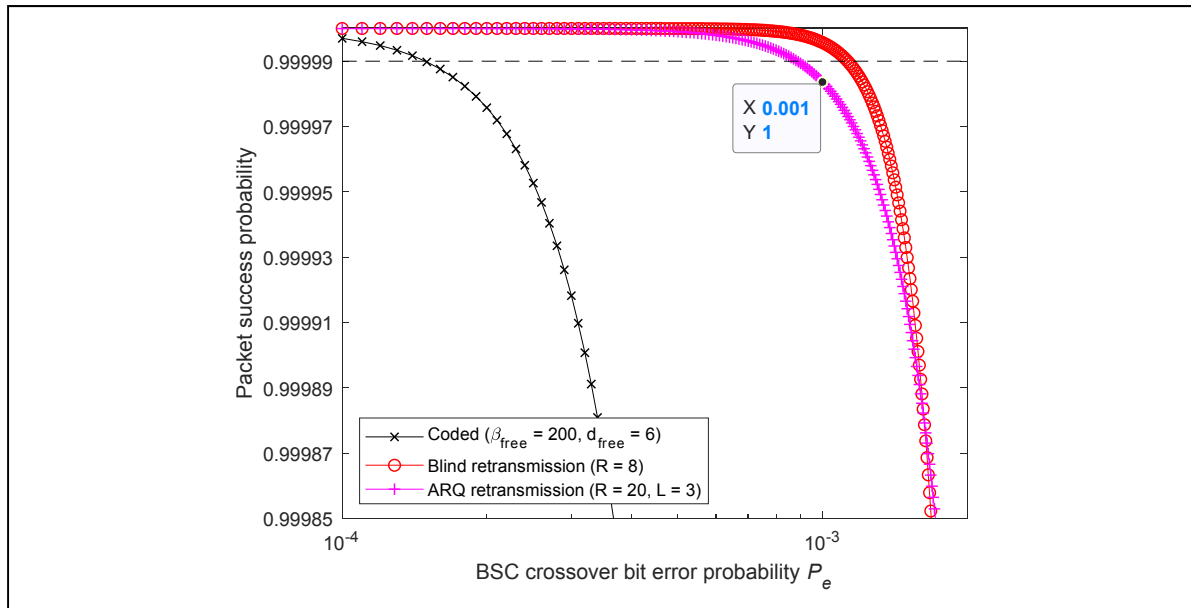


Figure 4.35 Meeting 99.999% reliability target by using BR and 3-Rep-ACK retransmissions

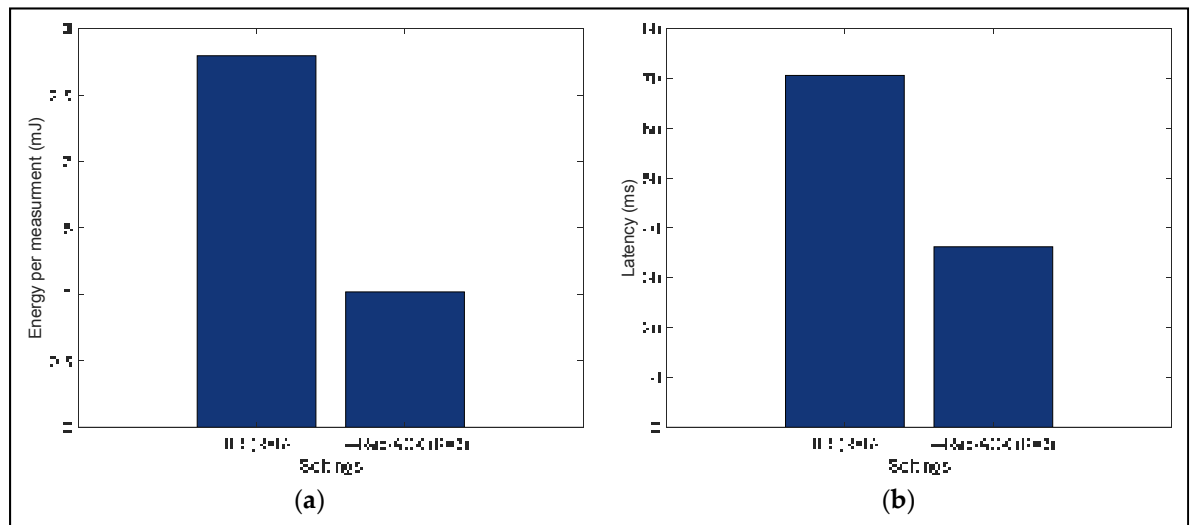


Figure 4.36 (a) Energy consumption per measurement and (b) system latency for 99.999% reliability target

Figure 4.36 shows that in this case, using 4-Rep-ACK retransmissions is the least expensive choice in terms of energy consumption and latency provided that the feedback channel is significantly more reliable (i.e., $L = 4$).

4.6 Conclusion and Discussion

It can be concluded from these case studies that, when the ambient noise density and received signal strength are accurately estimated, a good estimation of the achievable data rate can be determined for a given BER performance and at the required communication distance. However, as previously stated, after network deployment, the noise density will vary in time. Therefore, like in LoRaWAN (D. Singh, 2018), adaptive rate and transmission power allocations during operation are also of interest.

Moreover, with regard to error mitigation techniques, when feedback-based retransmissions (i.e. SAW-ARQ in this case) are used, a noisy feedback channel can severely degrade system performance and make it worse than an open-loop system (i.e., BR approach) (B. Makki, 2014). However, when the feedback channel is much less error-prone, the SAW-ARQ protocol can be as reliable as when using the time and energy inefficient BR. However, in WSN applications, this is not the case and the feedback channel is also error-prone with similar probabilities (Z. Ahmad, 2018; B. Makki, 2014). For this reason, ACK-NACK responses have to be sent with a stricter reliability requirement in mind, leading to a larger value of L requirement for the ACK approach (Derya Malak, 2018; P. Wu, 2011). Still, using the probabilistic model in Equation (4.29), it was clear that the maximum number of retransmissions is hardly reached and that a packet has a great chance of being successfully transmitted within the first three transmission attempts.

It was also demonstrated that FEC is both energy and time efficient when the required reliability is not very high. Since a limited number of bit errors can be corrected, FEC cannot achieve ultra-reliable communication by itself. Therefore, a combination of both techniques can also be of interest such as in (J. C. Fricke, 2009).

CHAPTER 5

CONCLUSION AND FUTURE WORK

This chapter concludes the thesis by providing a summary of the contributions and novelties that were covered mainly in chapter 4. Moreover, it gives some important future research directions beyond the work presented in this thesis.

5.1 Conclusion

URLLC coupled with energy consumption efficiency has the potential to change our way of living in a connected wireless world. Next generation WSN for IoT applications, in smart cities, industrial environments, and agriculture to name a few, are an inevitable path to autonomous, reliable, and real-time data collection. WSN are an interesting solution towards achieving these goals because of the following example benefits:

- 1) Compact wireless devices have the potential to drastically minimize the size of the total system;
- 2) Cable connections can be avoided or drastically reduced which makes the overall system considerably less costly;
- 3) Possibility to deploy tens or hundreds of interconnected nodes to wirelessly monitor and control a wide area of harsh and hostile in a timely and reliable manner;
- 4) Last but not least, WSN are easy to deploy.

However, as it was shown in this thesis, the design process of such systems is a multidimensional challenge. Therefore, in order to achieve that goal, continuous work and research need to be carried out to develop techniques and methodologies that address the interlinked design challenges. With this goal in mind, in this thesis, a thorough literature review highlighted the recent advancements in the energy modeling and optimization of WSN while ensuring the required QoS requirement. Moreover, the main strengths and limitations of these contributions were identified. Mainly, the following challenges were addressed:

- 1) Accurately modeling the energy consumption of sensor nodes;

- 2) The development of a path loss model that can be used to accurately estimate the received signal strength at the receiver;
- 3) Meeting QoS requirements by finding the favorable rate/power configuration and using different error mitigation techniques.

In this thesis, a methodology to better optimize the design, configuration, and deployment of reliable ultra-low power WSNs was proposed. Therefore, a comprehensive analytical energy model of the sensor node along with a high-level of abstraction modeling framework were presented and validated through measurements. The novel contribution of the model is the accurate estimation of MCU current consumption and execution time. A comparison of the average current measurement with the predictions using CM, shows that the estimation results in a 4% error margin. Moreover, for the execution time, it is estimated with an error between 9.4% and 11.5%.

In addition, an empirical and adjustable PL model was presented for both urban and suburban areas and based on field measurements. To this end, a variable fitting coefficient was used. Also, the ambient noise was measured in both areas and around 3 dBm/Hz difference is reported. Then, the achievable data rate was determined in order to ensure a better energy efficiency and a shorter latency.

Lastly, in order to mitigate wireless transmission bit errors, three error correcting techniques were studied and compared in terms of energy consumption, latency, and reliability. FEC, blind retransmissions, and feedback-based retransmissions were studied. Therefore, it was demonstrated that FEC cannot achieve ultra-reliable communication by itself. Moreover, when feedback-based retransmissions rely on a noisy feedback channel, the latter can severely degrade system performance. In WSN applications, this is usually the case and the feedback channel is also error-prone. Consequently, feedback responses were assumed to be sent with an increased reliability and evaluated. It was mathematically demonstrated that retransmitting feedback messages can considerably decrease energy consumption and latency of ARQ-based protocols when compared with blind retransmissions while meeting the reliability requirement.

After, the resulting methodology which logically links the comprehensive design and deployment steps of an ultra-low power and reliable WSN was also presented and detailed. By using estimations and measurements, it was shown that following the proposed methodology, the designer can thoroughly explore the design space, make most favorable decisions when selecting SN components and efficiently configure and deploy a WSN while taking into account the energy-reliability-latency trade-off of different error correction techniques and rate/power allocation. Through case studies, it was demonstrated how energy, latency, and reliability are interrelated and traded-off against each other, notably with respect to successful packet transmission probability metrics.

Therefore, the outcomes of this thesis can have a significant impact on the design of WSN in a wide range of energy and latency conscious applications. Whether in smart city, precision agriculture, or other monitoring and control applications, the proposed models and methodology can lead to a substantial improvement of a network's lifetime while recognizing and meeting QoS requirements.

Published conference paper:

Brini, O., Deslandes, D., & Nabki, F. (2018). A Model-Based Approach for the Design of Ultra-Low Power Wireless Sensor Nodes. Dans *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS)* (pp. 248-251). doi: 10.1109/NEWCAS.2018.8585492

Submitted manuscript:

'A System-Level Methodology for the Design of Reliable Low-Power Wireless Sensor Networks' submitted to Sensors MDPI journal (March 2019). The manuscript was reviewed and a revision has been submitted.

5.2 Future Work

Energy efficient WSN and URLLC designers and researchers are encouraged to approach the challenges imposed by today's string application requirements in a simultaneous way. In this work, the proposed methodology allows the exploration of the design space of WSN and optimization in terms of energy, reliability, and latency. However, a point-to-point

communication link was assumed. Moreover, the proposed modeling framework based on Simulink/Stateflow, only covers the energy models of the sensor node. Although, it is strong and flexible enough to allow the implementation of the whole methodology to form a unified and complete modeling framework.

Therefore, based on the central ideas introduced in this thesis, future work needs to be undertaken to extend this work and completely shape a complete methodology along with a unified modeling framework. The following research topics are then recommended:

- 1) Extend the presented energy modeling framework in order to have a unified design environment covering all the aspects discussed in this work. For instance, the complete methodology design flow can be implemented using Simulink/Stateflow;
- 2) A network of nodes should be considered in the future (e.g. start network). Therefore, the impact of the used medium access protocol will need to be taken into account;
- 3) A combination of FEC and feedback-based and blind retransmissions can also be investigated as a hybrid approach can yield significant improvements if applied intelligently and depending on channel characteristics;
- 4) Other state-of-the-art feedback channel reliability enhancement techniques in the literature are also very promising and should be investigated.

ANNEX I

PUBLISHED CONFERENCE PAPER

A Model-Based Approach for the Design of Ultra-Low Power Wireless Sensor Nodes

Oussama Brini, Dominic Deslandes and Frederic Nabki

Department of Electrical Engineering
Ecole de Technologie Supérieure
Montreal, Canada

oussama.brini.1@ens.etsmtl.ca, dominic.deslandes@etsmtl.ca and frederic.nabki@etsmtl.ca

Abstract—This paper presents a model-based approach for the design of ultra-low power wireless sensor nodes along with a high-level of abstraction modeling framework based on Simulink/StateFlow. This leads to a fast and effective method of designing low-power wireless sensing systems by serving as a guideline for choosing the right commercial off-the-shelf (COTS) components and node configuration. Through simulations, the impact of using different configurations on energy and power consumption metrics is determined, and the models capture the energy consumption contributions of each of the studied components.

Keywords—Power model, low-power design, wireless sensor networks (WSN), system-level methodology

I. INTRODUCTION

Wireless sensor networks (WSN) are an important part of the Internet of Things (IoT), as they enable the agile collection of data which is paramount in many IoT applications. This is motivating researchers to explore the design space of WSNs and optimize their performance. WSN are often deployed in places where human intervention is difficult or impossible. Therefore, sensors are expected to be autonomous, with a node's energy source usually provided by batteries that limit the energy available. In fact, even when human intervention is possible, it is laborious to service sensor batteries when operating networks with hundreds of nodes.

To design a truly ultra-low power sensor node, the designer first has to apply a methodology that will allow him to optimally select the main components out of a myriad of COTS components and account for all the software and hardware components concurrently. At the beginning of the design process, judicious reading of a component's datasheet is mandatory. This step can be time consuming since part-by-part or feature-for-feature comparisons are almost impossible. Thus, the designer can easily make unfounded decisions when a clear methodology is not applied.

There is no doubt that power consumption assessment at the system-level is less accurate than other gate-level and cycle-accurate techniques [1]. However, simulations can be much faster and efficient at the system-level, and can yield important design insights. In addition, it should be noted that initially, achieving extremely low levels of energy consumption should not deteriorate the quality of service (QoS) or responsiveness of

the system. Moreover, the system-level design tool should support multiple levels of abstraction, make possible the integration of hardware and software models in an intuitive way, and enable fast performance estimations.

Significant research has been conducted in order to develop a methodology for designing an energy-efficient WSN and estimate power consumption at an early stage of the design process. An interesting contribution is the C++ class library presented in [1] and called Powersim. Powersim monitors the C++ operators during the simulation of a high-level of abstraction model developed using SystemC in order to estimate a given hardware's power consumption when provided with an energy model. The latter is essentially a set of simulations or real hardware power consumption measurements of the different operators.

Equally important is the model-based evaluation and validation framework for WSN presented in [2]. The model is based on the multidomain simulation tool MLDesigner. The tool associates each sensor node with a function-based finite state machine (FSM), where an event-triggered mechanism captures the sensor node's operating state to estimate the energy consumption. In each state, the sensor node uses a predetermined and fixed amount of energy extracted from the data sheet of the widely used TelosB [3, 4] platform.

In [5], the authors present an energy consumption model of the time slotted channel hopping (TSCH) medium access technique introduced in the IEEE802.15.4e amendment in 2012. That work validates the model by implementing the OpenWSN protocol stack [6] on two different nodes made of COTS components, namely OpenMoteSTM and GINA [7].

All of the aforementioned works lead to a thorough analysis of system-level power consumption. However, the decrease of simulation time remains of interest, while maintaining model accuracy to ensure a reasonable error range in the estimations. Accordingly, the presented modeling framework is based on Simulink/Stateflow and the system-level model allows the consideration of interrelated specifications at the hardware level to achieve the right trade-offs.

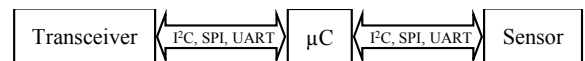


Fig. 1. The three main modeled components of an SN.

This paper is organized as follows. Section II presents the modeling tool and the model design. Section III describes the simulation results, and is followed by a conclusion.

II. MODEL DESIGN

A. Simulink / Stateflow

The presented modeling framework is based on Simulink/Stateflow and allows the creation of energy consumption models of existent configurable COTS components in the form of FSMs. The main components of a sensor node (SN), modeled in our framework, are depicted in Fig. 1. The Simulink and Stateflow models work together in a seamless way such that a Stateflow chart functions as an FSM within the Simulink model. In addition, the possibility of integrating MATLAB functions which can reside anywhere in a Statflow chart, state or sub-chart, assures a high degree of modeling flexibility, as shown in Fig. 2. The tool represents multiple levels of subcomponents in a system, making multilevel-state complexity of a SN more manageable. The states within a chart can be executed exclusively, where one state can be active at a time, or in parallel, where the states are active at the same time.

B. Sensor Node Workload Benchmark (SensBench)

In order to objectively assess the power consumption of a wireless sensing system across different component combinations, a simple yet effective application workload benchmark depicted in Fig. 3 is proposed to mimic a realistic general type of workload on the system. Three main activity periods can be identified. First, one or more sensors will perform a measurement of one or more physical quantities and send data to the MCU. Then, the latter will process the data and forward them to the wireless transceiver for transmission. Finally, the activity period of the transceiver is of particular interest, as three different modes of operation are specified. First, the transceiver enters the transmit mode during time period t_{Tx} . Then, during t_{stb} , it goes into the lowest achievable power mode where the crystal is kept running. This mode is important and needs to be considered especially when acknowledgements are used in the link layer. The last mode of operation is the receive mode lasting for t_{Rx} . The sum of the three components' activity periods is

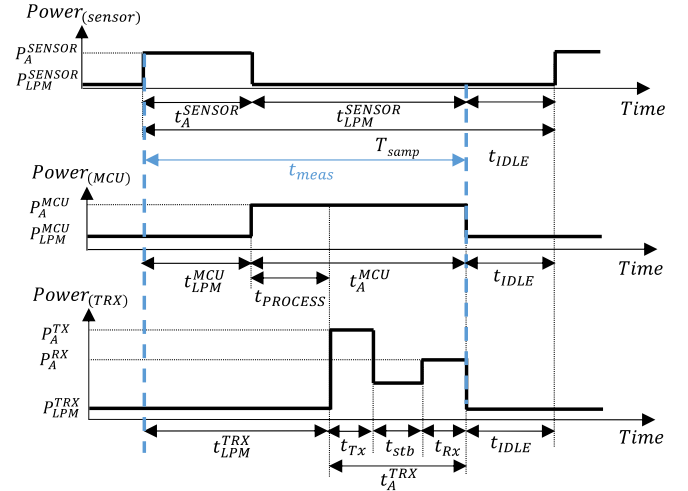


Fig. 3. Sensor node workload benchmark (SensBench).

called the measurement period and denoted by t_{meas} . The period of time where all the components enter the lowest achievable low-power mode is denoted by t_{IDLE} . $t_{PROCESS}$ is the processing time. The sampling period is denoted by T_{samp} and represents the time between the start of consecutive sensor activity times. The average power consumption of each component is calculated using

$$P_{avg} = (t_{LPM} P_{LPM} + \delta_{LPM} P_{\delta} + t_A P_A) / (t_A + \delta_{LPM} + t_{LPM}), \quad (1)$$

where t_{LPM} , δ_{LPM} and t_A are the times spent in a low-power state, wake-up from the same low-power state and active time, respectively. P_{LPM} , P_{δ} and P_A , are the power consumption in low-power mode, transition and active states, respectively.

C. Microcontroller Unit (MCU) Benchmark

A simple starting point in the assessment of an MCU's CPU core performance and current consumption can be achieved by running a benchmark algorithm that is representative of a wireless sensing application's workload in the studied case. A few performance benchmarking algorithms have emerged such as Fibonacci, Dhrystone, Whetstone, LINPACK and CoreMark [8]. The latter was developed by the EEMBC consortium in 2009 and then quickly became the de facto standard for CPU core performance ratings.

1) *Reference MCU*: This MCU is used for the validation of the representativeness of the CoreMark workload of an actual WSN application. Moreover, its score and operating frequency are used as baseline values described in Table I for a realistic estimate of the processing time using

$$t_{PROCESS} = t_{proc_ref} (S_{ref} / S_{MCU}) (f_{ref} / f_{MCU}), \quad (2)$$

where t_{proc_ref} and S_{ref} are the reference time and reference CoreMark score respectively, extracted from the reference MCU. f_{MCU} and S_{MCU} are the operating frequency and CoreMark score of the studied candidate MCU, and f_{ref} is the reference operating frequency (i.e. 48MHz).

The CC1310 wireless microcontroller (MCU) from TI [9] was chosen for this matter. It combines a Sub-1 GHz radio and a 32-bit ARM Cortex-M3 running at 48 MHz as the main processor on a single chip. It has a CoreMark score of 2.47

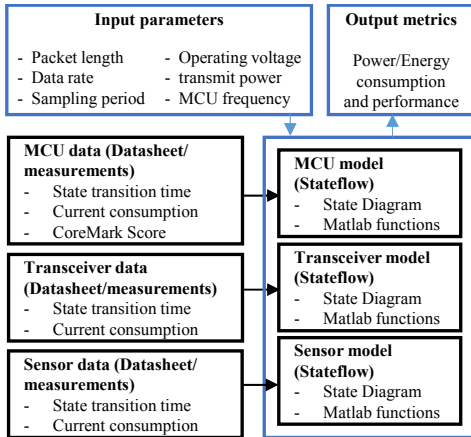


Fig. 2. A Power/Energy consumption assessment framework based on Stateflow/Simulink.

TABLE I. SPECIFICATIONS OF THE REFERENCE MCU

Parameter	Description	Value
S_{ref} [CoreMark/MHz]	CoreMark score at 48 MHz	2.47
f_{ref} [MHz]	Operating frequency	48
t_{proc_ref} [ms]	Processing time of TI 15.4-Stack	2.5
I_{active} [mA]	Active current	3

CoreMark/MHz. The CC1310 is assumed to be running the TI 15.4-Stack [10] that implements the standard IEEE 802.15.4e and 802.15.4g specification for wireless star-topology-based networking solutions. The stack also provides a real-time operating system (RTOS) with a real-time multitasking kernel.

2) *Active Current*: Most of MCU manufacturers specify the current consumption of their products when running one or more benchmark algorithms, notably the industry standard CoreMark. This is very important as the designer needs a guideline for estimating a specific application's power consumption which can vary considerably from one benchmark algorithm to another, as depicted in Table II for two MCUs [11, 12]. As shown in Fig. 4, the current consumption after wake-up of the CC1310 wireless MCU while running the TI 15.4-stack is around 3 mA in boost mode and after subtracting the contribution of some peripherals (e.g. DMA, timers and RF core idle currents). According to the datasheet, an increase of 15% is observed when the boost mode is selected. Therefore, the current consumption when running CoreMark becomes 2.88 mA instead of the 2.5 mA as mentioned in the data sheet. In comparison to the measurement in Fig. 4, this results in a 4.2 % error margin in the estimation of the active current using CoreMark, which is acceptable. Therefore, the latter represents a fairly close workload of a SN and is a reliable indicator of the power consumption of different MCUs. In this work, only the MCU core is evaluated and the impact of the peripherals are not considered. Nonetheless, extra steps need to be taken in order to determine the energy consumption of each MCU. The time required for an MCU to process the workload needs to be determined in order to estimate its energy impact.

3) *Processing Time*: After a measurement is received from the sensor, the MCU will process the data and run a protocol stack to encapsulate them and forward the packet to the corresponding neighboring node. In order to fairly compare the time it takes for different MCUs to process the same workload, again, a representative benchmark is needed to capture the influence of different instruction set architectures (ISA),

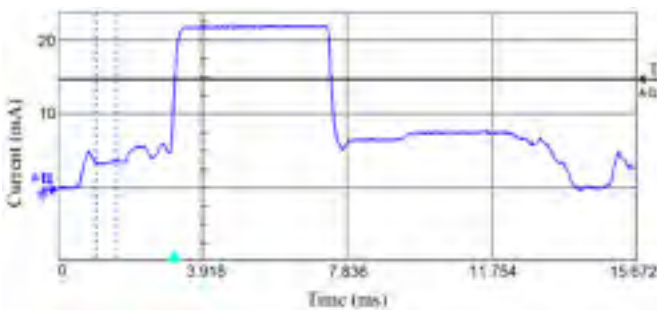


Fig. 4. Pre-processing current consumption after wake-up of the CC1310 wireless MCU while executing the TI 15.4 stack.

TABLE II. CURRENT CONSUMPTION OF TWO DIFFERENT MCUS AT 26MHz

Parameter	MCU	While loop	Fibonacci	CoreMark
Current [mA]	SAMD21	2.2	2.27	2.44
	STM32L4	1.7	2.42	2.9

clocking schemes and available instructions. In fact, metrics such as the million instructions per second (MIPS) are only an approximation as to how a set of processors' performance would vary since different amounts of work can be done in one cycle for each processor. As the CoreMark benchmark became an industry standard, MCU manufacturers began advertising the CoreMark/MHz metric in their datasheets for a more realistic way of comparing performance. In the proposed approach, this metric is judged to be accurate enough to estimate the time it takes different MCUs to process the same workload at different operating frequencies. Moreover, a realistic baseline value of this processing period is determined and listed in Table I. This value is obtained by analyzing the current consumption waveform of a reference CC1310 MCU illustrated in Fig. 4. The latter depicts four main activity periods, namely processing, transmission, standby, and reception. The processing time is estimated using eq. (2).

III. SIMULATION RESULTS

To demonstrate how the proposed approach estimates the energy consumption contribution of the different components and how it can serve as a guideline for choosing the right COTS components, four different combinations of transceivers and MCUs are studied, as depicted in Fig. 5. Two different Sub-GHz transceivers are selected, notably the S2LP from STMicroelectronics (STM) [13] and AX5243 from ON Semiconductor [14] and two different MCUs, notably the STM32L433 Cortex M4 from STM [11] and the SAMD21 Cortex M0+ from Atmel [12]. The same BME280 digital humidity, pressure and temperature sensor from BOSCH [15] is used in all of the four nodes. As illustrated, the energy consumption can vary considerably from one combination to another and according to the model, the cortex M4 MCU is more energy efficient than the cortex M0+ MCU. This is due to the fact that the latter is slower and consumes approximately the

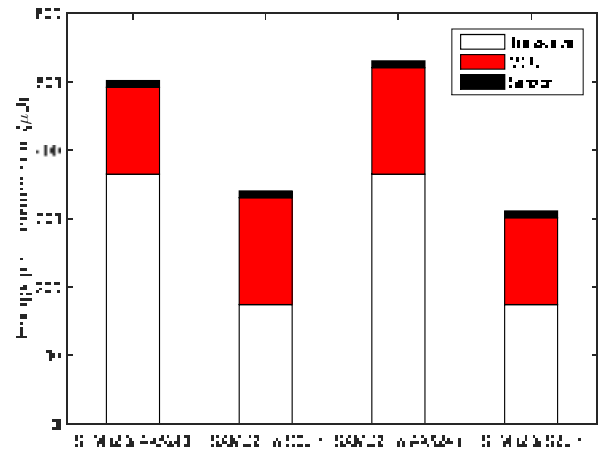


Fig. 5. Energy consumption contribution of each component in two separate nodes (both MCUs are running at 32MHz).

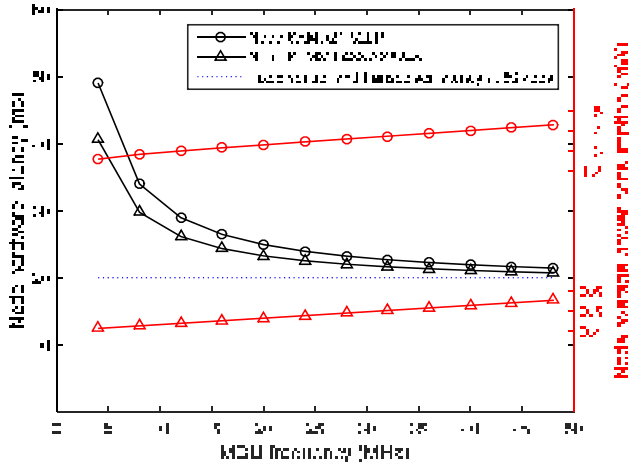


Fig. 6. End-to-end hardware latency and average overall power consumption at a 1 Hz sampling rate for different clock speeds of two separate nodes.

same amount of current. Moreover, as expected from a wireless sensor node, Fig. 5 shows that the overall energy consumption is dominated by the transceiver's energy.

In order to estimate the system's latency, Fig. 6 plots the variation of t_{meas} on two nodes for different MCU clock frequencies. The STM32L433 Cortex M4 is faster than the SAMD21 Cortex M0+ which reduces the system's latency especially at a low clock speed. As the latter increases, the latency becomes more and more dominated by the sensor's and transceiver's latencies. In addition to high current consumption in active mode, the SAMD21 MCU draws more current in low power mode. Consequently and according to Fig. 6, the average power consumption of the STM node is around three times less than that of the SAMD21 node.

Since choosing the optimal components will depend on the application specifications, Fig. 7 shows that the average power consumption of a sensor node across different sampling periods varies. The same MCU and sensor are used, and both transceivers are transmitting and receiving at the same data rate. The sensor is assumed to generate a payload of 48 bits for each measurement of the three physical quantities without oversampling or filtering. As illustrated in Fig. 7, the S2LP is

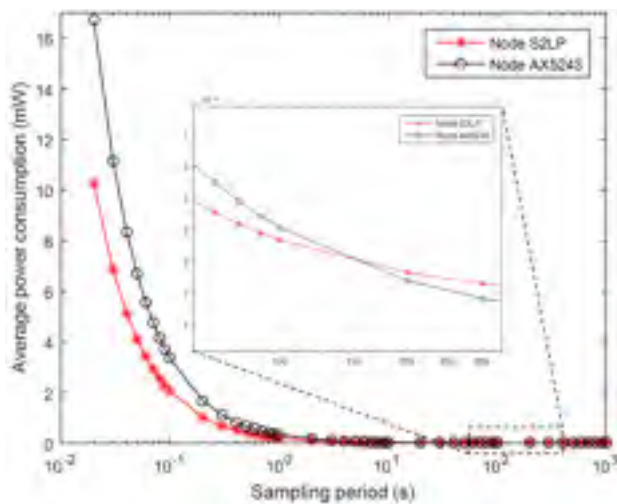


Fig. 7. Impact of the sampling period on the overall power consumption.

more power efficient at very high sampling rates. However, the AX5243 is better at a sampling period of above 2 minutes and a half, approximately, outlining the importance of considering the sampling period when selecting components.

IV. CONCLUSION

In this paper, a model-based approach for building an ultra-low power WSN along with a high-level of abstraction modeling framework based on Simulink/StateFlow were presented. The models capture the energy consumption contributions of each of the studied components.

The approach can lead to a fast and effective method of designing wireless sensing systems for low-power operation. Through simulations, the impact of using different configurations on power consumption is determined. As a result, designers can make the optimal COTS components selection for a given application when the approach is applied.

ACKNOWLEDGEMENT

The authors would like to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Microsystems Strategic Alliance of Québec (ReSMiQ).

REFERENCES

- [1] S. Orcioni, M. Giammarini, C. Scavongelli, G. B. Vece, and M. Conti, "Energy estimation in SystemC with Powersim," *Integration, the VLSI Journal*, vol. 55, pp. 118-128, 9/2016.
- [2] S. Jäger, T. Jungebloud, R. Maschotta, and A. Zimmermann, "Model-Based QoS Evaluation and Validation for Embedded Wireless Sensor Networks," *IEEE Systems Journal*, vol. 10, no. 2, pp. 592-603, 2016.
- [3] MEMSIC, "TELOSB MOTE PLATFORM," data sheet, Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/telesh_datasheet.pdf.
- [4] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, 2005, p. 48: IEEE Press.
- [5] X. Vilajosana, Q. Wang, F. Chraïm, T. Watteyne, T. Chang, and K. S. J. Pister, "A Realistic Energy Consumption Model for TSCH Networks," *IEEE Sensors Journal*, vol. 14, no. 2, pp. 482-489, 2014.
- [6] T. Watteyne et al., "OpenWSN: a standards - based low - power wireless development environment," *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 5, pp. 480-493, 2012.
- [7] A. M. Mehta and K. S. Pister, "WARPWING: A complete open source control platform for miniature robots," in *Intelligent Robots and Systems (IROS)*, 2010 IEEE/RSJ International Conference on, 2010, pp. 5169-5174.
- [8] EEMBC. CoreMark: An EEMBC Benchmark. Available: <https://www.eembc.org/coremark/about.php>
- [9] T. Instruments, "CC1310 SimpleLink Ultra-Low-Power Sub-1 GHz Wireless MCU," ed, 2016.
- [10] T. Instruments. (2016). TI 15.4-Stack Wiki. Available: http://processors.wiki.ti.com/index.php/TI_15.4-Stack_Wiki
- [11] STMicroelectronics, "Ultra-low-power ARM Cortex M4 32-bit MCU and FPU," ed, 2017, p. 224.
- [12] Microchip, "32-bit ARM-Based Microcontrollers SAM D21E / SAM D21G / SAM D21J," ed, 2017.
- [13] STMicroelectronics, "S2-LP Ultra-low power, high performance, sub-1GHz transceiver," ed, 2016, p. 91.
- [14] O. Semiconductor, "AX5243 Advanced High Performance ASK and FSK Narrow-band Transceiver for 27 - 1050 MHz Range," ed, 2016.
- [15] BOSCH, "BME280 Combined humidity and pressure sensor," ed, 2016.

ANNEX II

C CODE: BME280 SENSOR DATA COMMUNICATION

TX side

```
while(1)
{
    /* Initialize the BME Sensor */
    bme280_set_power_mode(BME280_FORCED_MODE);

    /* Create packet with incrementing sequence number and random payload */
    packet[0] = (uint8_t)(seqNumber >> 8);
    packet[1] = (uint8_t)(seqNumber++);
    bme280_read_pressure_temperature_humidity(&g_u32ActualPress,
    &g_s32ActualTemp, &g_u32ActualHumidity);

    packet[2] = (uint8_t)(g_u32ActualPress >> 24);
    packet[3] = (uint8_t)(g_u32ActualPress >> 16);
    packet[4] = (uint8_t)(g_u32ActualPress >> 8);
    packet[5] = (uint8_t)(g_u32ActualPress);

    packet[6] = (uint8_t)(g_s32ActualTemp >> 24);
    packet[7] = (uint8_t)(g_s32ActualTemp >> 16);
    packet[8] = (uint8_t)(g_s32ActualTemp >> 8);
    packet[9] = (uint8_t)(g_s32ActualTemp);

    packet[10] = (uint8_t)(g_u32ActualHumidity >> 24);
    packet[11] = (uint8_t)(g_u32ActualHumidity >> 16);
    packet[12] = (uint8_t)(g_u32ActualHumidity >> 8);
    packet[13] = (uint8_t)(g_u32ActualHumidity);

    //Display_print3(display, 0, 0, "%u , %u , %u \n", i1, i2, i3);

    /* Send packet */
    RF_EventMask terminationReason = RF_runCmd(rfHandle,
    (RF_Op*)&RF_cmdPropTx,
    RF_PriorityNormal, NULL, 0);

    /* Power down the radio */
    RF_yield(rfHandle);
    usleep(PACKET_INTERVAL);
}
```

RX side

```

void callback(RF_Handle h, RF_CmdHandle ch, RF_EventMask e)
{
    if (e & RF_EventRxEntryDone)
    {
        /* Toggle pin to indicate RX */
        PIN_setOutputValue(ledPinHandle, Board_PIN_LED2,
                          !PIN_getOutputValue(Board_PIN_LED2));

        /* Get current unhandled data entry */
        currentDataEntry = RFQueue_getDataEntry();

        /* Handle the packet data, located at &currentDataEntry->data:
         * - Length is the first byte with the current configuration
         * - Data starts from the second byte */
        packetLength      = *(uint8_t*)&currentDataEntry->data;
        packetDataPointer = (uint8_t*)&currentDataEntry->data + 1;

        /* Copy the payload + the status byte to the packet variable */
        memcpy(packet, packetDataPointer, (packetLength + 1));
        RFQueue_nextEntry();
    }

    g_u32ActualPress=((uint32_t)packet[2] << 24) | ((uint32_t)packet[3] << 16) |
    ((uint32_t)packet[4] << 8) | ((uint32_t)packet[5]);g_s32ActualTemp =
    ((uint32_t)packet[6] << 24) | ((uint32_t)packet[7] << 16) | ((uint32_t)packet[8]
    << 8) | ((uint32_t)packet[9]); g_u32ActualHumity=((uint32_t)packet[10] << 24) |
    ((uint32_t)packet[11] << 16) | ((uint32_t)packet[12] << 8) |
    (uint32_t)packet[13]);

    Display_print3(display, 0, 0, "%u KPa(Pressure), %u DegC(Temp), %u
    %%RH(Humidity)\n", g_u32ActualPress/1000, g_s32ActualTemp/100,
    g_u32ActualHumity/1000);
}

```

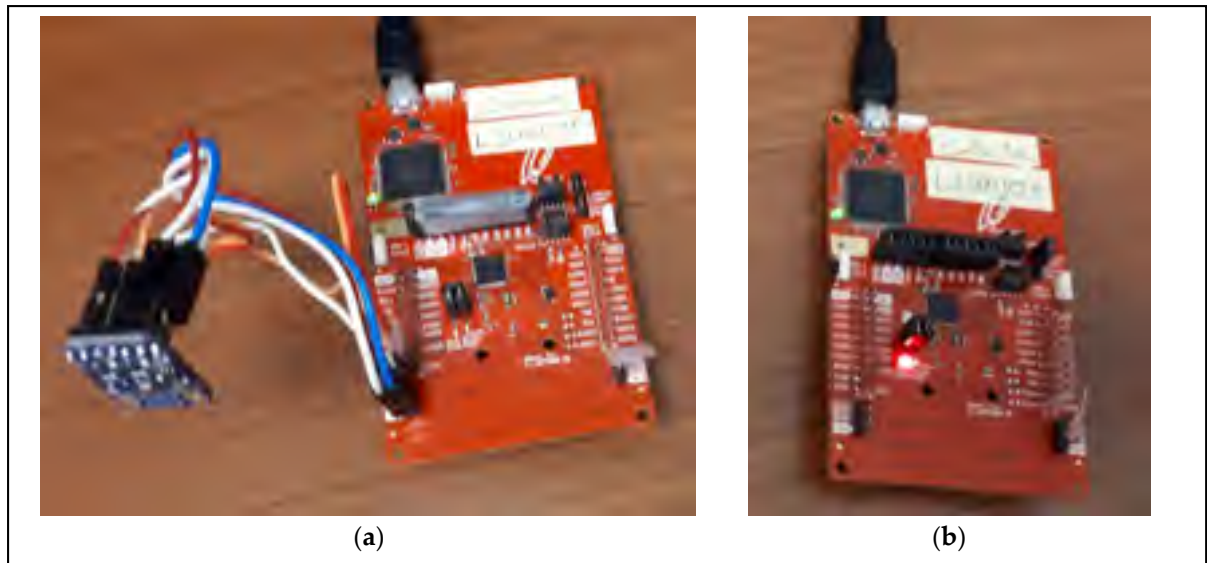


Figure-A II-1 Hardware used for point-to-point communication of BME 280 sensor data, (a) is the transmitter, and (b) is the receiver

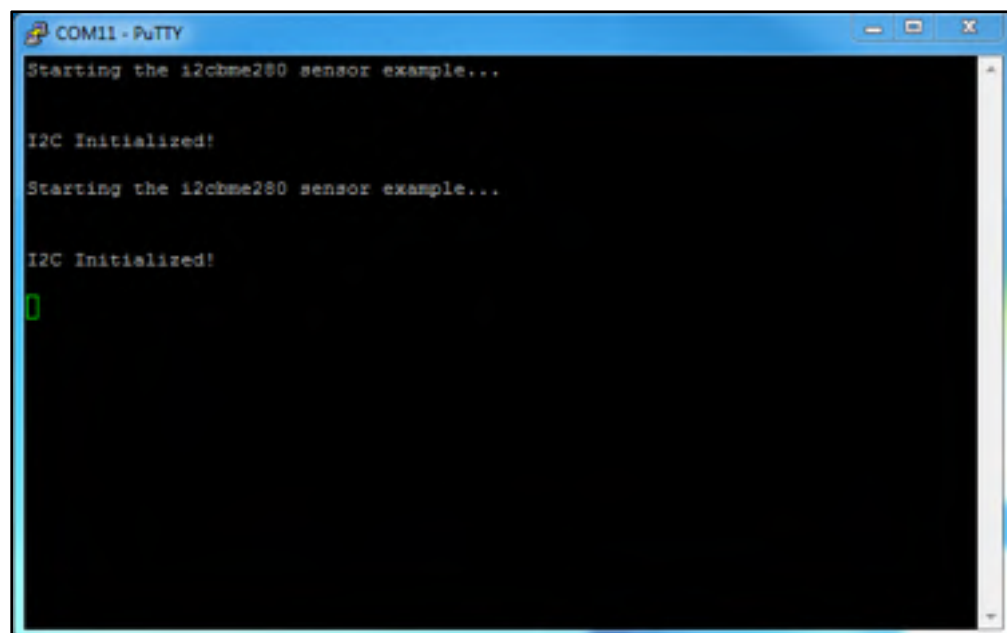
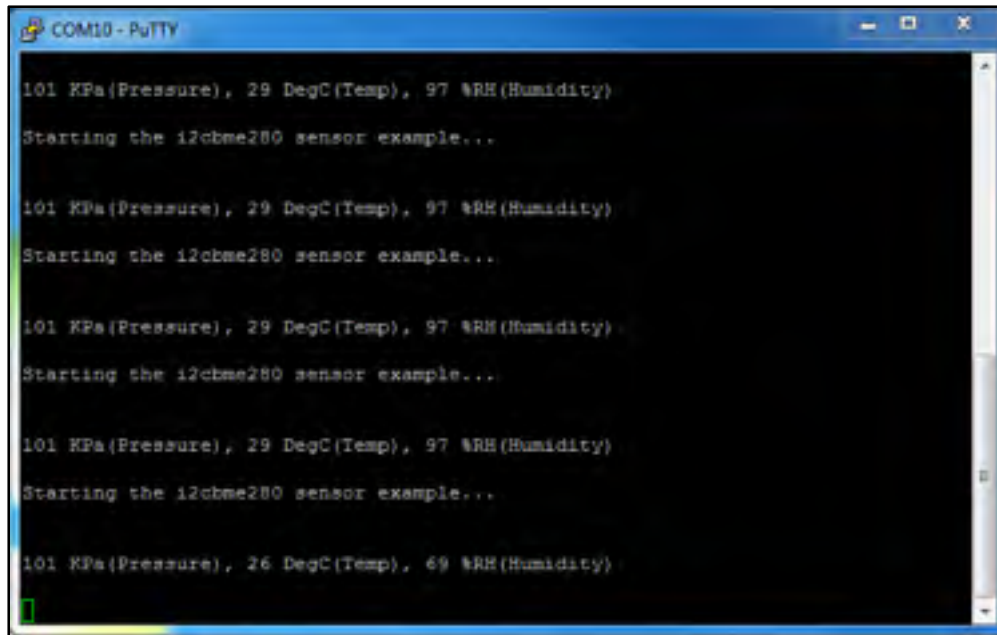


Figure-A II-2 Screenshot of the transmitter node output



A screenshot of a PuTTY terminal window titled "COM10 - PuTTY". The window has a blue title bar and standard window controls (minimize, maximize, close). The terminal area is black with white text. The output consists of a repeating sequence of two lines: "101 KPa(Pressure), 29 DegC(Temp), 97 %RH(Humidity)" followed by "Starting the i2cbme280 sensor example...". This sequence is repeated four times. The final line of output is "101 KPa(Pressure), 26 DegC(Temp), 69 %RH(Humidity)", which is followed by a green cursor. A vertical scrollbar is visible on the right side of the terminal window.

```
COM10 - PuTTY
101 KPa(Pressure), 29 DegC(Temp), 97 %RH(Humidity)
Starting the i2cbme280 sensor example...
101 KPa(Pressure), 29 DegC(Temp), 97 %RH(Humidity)
Starting the i2cbme280 sensor example...
101 KPa(Pressure), 29 DegC(Temp), 97 %RH(Humidity)
Starting the i2cbme280 sensor example...
101 KPa(Pressure), 29 DegC(Temp), 97 %RH(Humidity)
Starting the i2cbme280 sensor example...
101 KPa(Pressure), 26 DegC(Temp), 69 %RH(Humidity)
```

Figure-A II-3 Screenshot of the receiver node output

ANNEX III

SENSOR NODE MODEL USING STATEFLOW



Figure-A III-1 BME280 sensor model

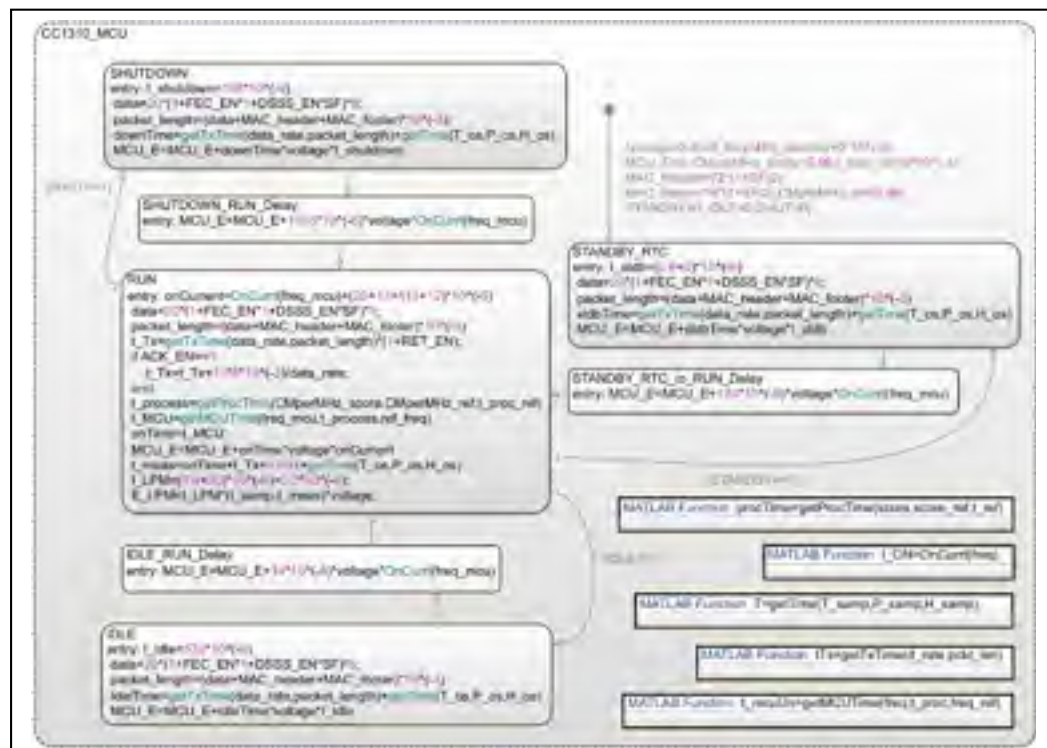


Figure-A III-2 CC1310 MCU model

Figure-A III-4 Simulink/Stateflow modeling framework

ANNEX IV

NOISE DENSITY MEASUREMENT PROCEDURE

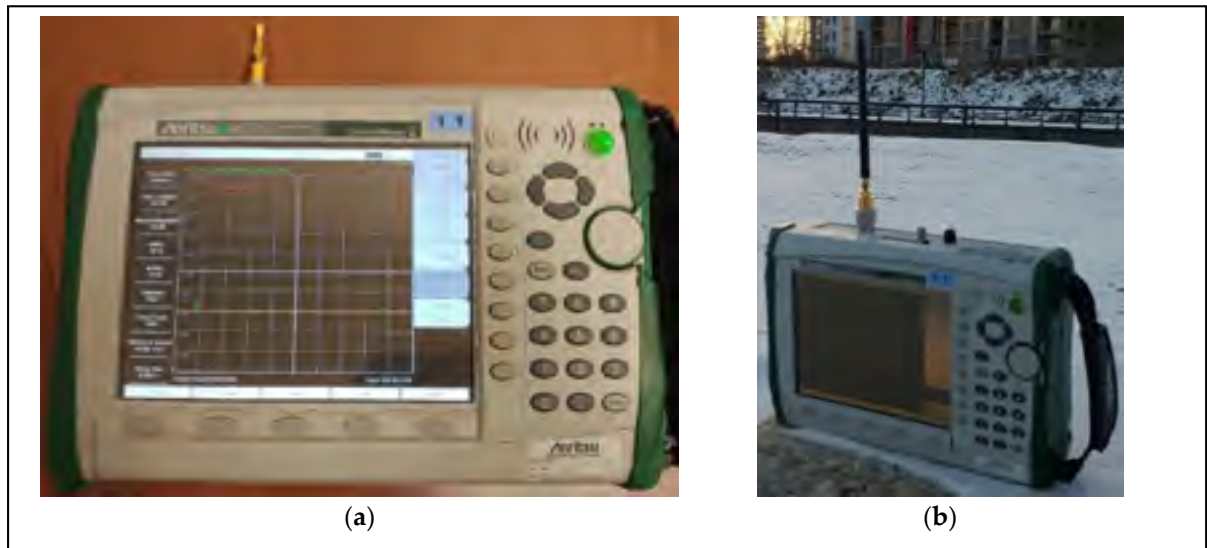


Figure-A IV-1 (a) Spectrum analyzer noise figure and (b) ambient noise measurement setups

Even when using a modern spectrum analyzer, a good measurement of the ambient noise requires following the steps below:

- The Displayed Average Noise Level (DANL) of the machine is measured with a 10Hz resolution bandwidth (RBW), a 50 Ohm termination and an internal attenuation set to 0 dB in order to determine its noise floor;
- Use the marker noise functionality and check if it lowers the DANL by 2.51 dB because of log power averaging;
- Measurement of noise density in the deployment site by doing at least ten sweeps and using the same RBW (10 Hz). Moreover, the averaging and marker noise functionalities are used;
- Subtract the machine's measured noise floor (11 dB in this case) and add the 2.51 dB if it was found to be subtracted by the marker noise functionality of the machine in order to get a good measurement of the ambient noise density.

BIBLIOGRAPHY

- Abreu, R., Berardinelli, G., Jacobsen, T., Pedersen, K., & Mogensen, P. (2018). A Blind Retransmission Scheme for Ultra-Reliable and Low Latency Communications. Dans *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)* (pp. 1-5). doi: 10.1109/VTCSpring.2018.8417721
- Ahmad, Z., Ahmad, I., Love, D. J., & Smida, B. (2018). Analysis of Two-Unicast Network-Coded Hybrid-ARQ With Unreliable Feedback. *IEEE Transactions on Vehicular Technology*, 67(11), 10871-10885. doi: 10.1109/TVT.2018.2869911
- Alameh, A. H., Gratuze, M., Elsayed, M. Y., & Nabki, F. (2018). Effects of Proof Mass Geometry on Piezoelectric Vibration Energy Harvesters. *Sensors (Basel, Switzerland)*, 18(5), 1584. doi: 10.3390/s18051584
- Angelopoulos, G., Paidimarri, A., Chandrakasan, A. P., & Médard, M. (2013). Experimental study of the interplay of channel and network coding in low power sensor applications. Dans *2013 IEEE International Conference on Communications (ICC)* (pp. 5126-5130). doi: 10.1109/ICC.2013.6655396
- Anwar, A. K., & Lavagno, L. (2010). MEOW: Model-based design of an energy-optimized protocol stack for wireless sensor networks. Dans *IEEE Local Computer Network Conference* (pp. 590-597). doi: 10.1109/LCN.2010.5735778
- ASSOCIATION, I. S. (2015). *IEEE Standard for Low-Rate Wireless Networks*. IEEE.
- Bacco, M., Berton, A., Gotta, A., & Caviglione, L. (2018). IEEE 802.15.4 Air-Ground UAV Communications in Smart Farming Scenarios. *IEEE Communications Letters*, 22(9), 1910-1913. doi: 10.1109/LCOMM.2018.2855211
- Bader, S., Ma, X., & Oelmann, B. (2014). On the Modeling of Solar-Powered Wireless Sensor Nodes. *Journal of Sensor and Actuator Networks*, 3(3). doi: 10.3390/jsan3030207
- Balázs-Hegedűs, J. (2019). Tehran, Toronto and San Francisco connected.
- Bartolomeu, P., Alam, M., Ferreira, J., & Fonseca, J. (2016). Survey on low power real-time wireless MAC protocols. *Journal of Network and Computer Applications*, 75, 293-316.
- Bdiri, S., Brini, O., & Derbel, F. (2018). A Self-duty-cycled Digital Baseband for Energy-enhanced Wake-up Receivers. Dans *SENSORNETS 2018 - 7th International Conference on Sensor Networks* (pp. 15-18). doi: 10.5220/0006512600150018

- Biswas, S., Roy, S. D., & Chandra, A. (2018). Cross-layer energy model for beacon-enabled 802.15.4 networks. *Journal of Ambient Intelligence and Humanized Computing*. doi: 10.1007/s12652-018-0923-z
- Blanco-Novoa, O., Fernández-Caramés, T., Fraga-Lamas, P., & Castedo, L. (2018). A cost-effective IoT system for monitoring Indoor radon gas concentration. *Sensors*, 18(7), 2198.
- BOSCH. (2016). BME280 Combined humidity and pressure sensor *Sensor modes*.
- Cai, Y., Wang, W., Qian, W., Xing, J., Tao, K., Yin, J., . . . Chen, H. (2019). FPGA Investigation on Error-Flare Performance of a Concatenated Staircase and Hamming FEC Code for 400G Inter-Data Center Interconnect. *Journal of Lightwave Technology*, 37(1), 188-195. doi: 10.1109/JLT.2018.2881924
- Chai, R., & Zhang, Y. (2015). A Practical Supercapacitor Model for Power Management in Wireless Sensor Nodes. *IEEE Transactions on Power Electronics*, 30(12), 6720-6730. doi: 10.1109/TPEL.2014.2387113
- Cid-Fuentes, R. G., Cabellos-Aparicio, A., & Alarcón, E. (2014). Energy Buffer Dimensioning Through Energy-Erlangs in Spatio-Temporal-Correlated Energy-Harvesting-Enabled Wireless Sensor Networks. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 4(3), 301-312. doi: 10.1109/JETCAS.2014.2337194
- Costa, G. D., Damasceno, A., & Silva, I. (2019). CitySpeed: A Crowdsensing-Based Integrated Platform for General-Purpose Monitoring of Vehicular Speeds in Smart Cities. *Smart Cities*, 2(1). doi: 10.3390/smartcities2010004
- Daneels, G., Municio, E., Van de Velde, B., Ergeerts, G., Weyn, M., Latré, S., & Famaey, J. (2018). Accurate Energy Consumption Modeling of IEEE 802.15. 4e TSCH Using Dual-Band OpenMote Hardware. *Sensors*, 18(2), 437.
- Dobslaw, F., Zhang, T., & Gidlund, M. (2016). QoS-Aware Cross-Layer Configuration for Industrial Wireless Sensor Networks. *IEEE Transactions on Industrial Informatics*, 12(5), 1679-1691. doi: 10.1109/TII.2016.2576964
- É, M., Maman, M., Guizzetti, R., & Duda, A. (2017). Comparison of the Device Lifetime in Wireless Networks for the Internet of Things. *IEEE Access*, 5, 7097-7114. doi: 10.1109/ACCESS.2017.2688279
- Ferry, N., Ducloyer, S., Julien, N., & Jutel, D. (2011). Power/Energy Estimator for Designing WSN Nodes with Ambient Energy Harvesting Feature. *EURASIP Journal on Embedded Systems*, 2011(1), 242386. doi: 10.1155/2011/242386

- Fricke, J. C., & Hoeher, P. A. (2009). Reliability-based retransmission criteria for hybrid ARQ. *IEEE Transactions on Communications*, 57(8), 2181-2184. doi: 10.1109/TCOMM.2009.08.070454
- Group, B. S. I. (2016). Bluetooth Core Specification v 5.0 *Specification of the Bluetooth System* (Vol. Volume 0, pp. 2822).
- Gungor, V. C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., & Hancke, G. P. (2013). A Survey on Smart Grid Potential Applications and Communication Requirements. *IEEE Transactions on Industrial Informatics*, 9(1), 28-42. doi: 10.1109/TII.2012.2218253
- Hamming, R. W. (1950). Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2), 147-160. doi: 10.1002/j.1538-7305.1950.tb00463.x
- Harada, H., Mizutani, K., Fujiwara, J., Mochizuki, K., Obata, K., & Okumura, R. (2017). IEEE 802.15. 4g based Wi-SUN communication systems. *IEICE Transactions on Communications*, 100(7), 1032-1043.
- Hedayati, K., Rubin, I., & Behzad, A. (2010). Integrated Power Controlled Rate Adaptation and Medium Access Control in Wireless Mesh Networks. *IEEE Transactions on Wireless Communications*, 9(7), 2362-2370. doi: 10.1109/TWC.2010.07.091616
- Herlich, M., Maier, C., & Dorfinger, P. (2018). A Black Box Measurement Method for Reliability of Wireless Communication. Dans *Proceedings of the Workshop on Experimentation and Measurements in 5G* (pp. 1-6). ACM.
- Hoel, R. (2007). FEC Implementation *Design Note DN504*: Texas Instruments.
- Hu, X., Wang, K., Liu, X., Sun, Y., Li, P., & Guo, S. (2018). Energy Management for EV Charging in Software-Defined Green Vehicle-to-Grid Network. *IEEE Communications Magazine*, 56(5), 156-163. doi: 10.1109/MCOM.2018.1700858
- Instruments, T. (2016). CC1310 SimpleLink Ultra-Low-Power Sub-1 GHz Wireless MCU.
- Islam, M. R. (2010). Error correction codes in wireless sensor network: An energy aware approach. *International Journal of Computer and Information Engineering*, 4(1), 59-64.
- Issariyakul, T., & Hossain, E. (2006). Performance modeling and analysis of a class of ARQ protocols in multi-hop wireless networks. *IEEE Transactions on Wireless Communications*, 5(12).
- Joakim Lindh, Christin Lee, & Hernes, M. (2017). Measuring Bluetooth Low Energy Power Consumption: Texas Instruments.

- Khosravirad, S. R., & Viswanathan, H. (2017). Analysis of feedback error in Automatic Repeat reQuest. *arXiv preprint arXiv:1710.00649*.
- Kitchin, R. (2014). The real-time city? Big data and smart urbanism. *GeoJournal*, 79(1), 1-14. doi: 10.1007/s10708-013-9516-8
- Klaina, H., Vazquez Alejos, A., Aghzout, O., & Falcone, F. (2018). Narrowband Characterization of Near-Ground Radio Channel for Wireless Sensors Networks at 5G-IoT Bands. *Sensors*, 18(8), 2428.
- Kong, P. (2016). Wireless Neighborhood Area Networks With QoS Support for Demand Response in Smart Grid. *IEEE Transactions on Smart Grid*, 7(4), 1913-1923. doi: 10.1109/TSG.2015.2421991
- Kong, P., Liu, C., & Jiang, J. (2017). Cost-Efficient Placement of Communication Connections for Transmission Line Monitoring. *IEEE Transactions on Industrial Electronics*, 64(5), 4058-4067. doi: 10.1109/TIE.2016.2644604
- Konstantopoulos, C., Koutroulis, E., Mitianoudis, N., & Bletsas, A. (2016). Converting a Plant to a Battery and Wireless Sensor with Scatter Radio and Ultra-Low Cost. *IEEE Transactions on Instrumentation and Measurement*, 65(2), 388-398. doi: 10.1109/TIM.2015.2495718
- Laurent, J., Julien, N., Senn, E., & Martin, E. (2004). *Functional Level Power Analysis: An Efficient Approach for Modeling the Power Consumption of Complex Processors* présentée à Proceedings of the conference on Design, automation and test in Europe - Volume 1.
- Leck, R. (2006). Results of Ambient RF Environment and Noise Floor Measurements Taken in the U.S. in 2004 and 2005. In W. M. ORGANIZATION (Éd.). GENEVA: COMMISSION FOR BASIC SYSTEMS STEERING GROUP ON RADIO FREQUENCY COORDINATION.
- Ledwaba, L. P. I., Hancke, G. P., Venter, H. S., & Isaac, S. J. (2018). Performance Costs of Software Cryptography in Securing New-Generation Internet of Energy Endpoint Devices. *IEEE Access*, 6, 9303-9323. doi: 10.1109/ACCESS.2018.2793301
- Lee, K., Kim, S., Kim, J., & Choi, S. (2018). DRaMa: Device-Specific Repetition-Aided Multiple Access for Ultra-Reliable and Low-Latency Communication. Dans *2018 IEEE International Conference on Communications (ICC)* (pp. 1-6). doi: 10.1109/ICC.2018.8422166
- Leech, C., Raykov, Y. P., Ozer, E., & Merrett, G. V. (2017). Real-time room occupancy estimation with Bayesian machine learning using a single PIR sensor and

- microcontroller. Dans *2017 IEEE Sensors Applications Symposium (SAS)* (pp. 1-6). doi: 10.1109/SAS.2017.7894091
- Libelium. (2012). New Wifi module for Waspote to connect directly to Cloud Servers and iPhone/Android platforms.
- Limited, A. (2013). CoreMark Benchmarking for ARM Cortex Processors *Application Note* 350 (pp. 15).
- Makki, B., Amat, A. G. i., & Eriksson, T. (2014). On Noisy ARQ in Block-Fading Channels. *IEEE Transactions on Vehicular Technology*, 63(2), 731-746. doi: 10.1109/TVT.2013.2276371
- Malak, D., Medard, M., & Yeh, E. M. (2018). ARQ with Cumulative Feedback to Compensate for Burst Errors. *arXiv preprint arXiv:1808.03153*.
- Mathworks. (1997–2018). Stateflow User's Guide *R2018b*.
- Mhatre, V. P., & Rosenberg, C. P. (2006). The Impact of Link Layer Model on the Capacity of a Random Ad hoc Network. Dans *2006 IEEE International Symposium on Information Theory* (pp. 1688-1692). doi: 10.1109/ISIT.2006.261642
- Mhlana, K., Mafumo, M., Sotenga, P., Phate, T., & Olwal, T. (2015). Design and simulation of a bluetooth communication system. Dans *2015 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)* (pp. 7-12). doi: 10.1109/ETNCC.2015.7184800
- Michelusi, N., & Mitra, U. (2015). Cross-Layer Design of Distributed Sensing-Estimation With Quality Feedback; Part II: Myopic Schemes. *IEEE Transactions on Signal Processing*, 63(5), 1244-1258. doi: 10.1109/TSP.2014.2388440
- Min, A. W., Wang, R., Tsai, J., Ergin, M. A., & Tai, T.-Y. C. (2012). *Improving energy efficiency for mobile platforms by exploiting low-power sleep states* présentée à Proceedings of the 9th conference on Computing Frontiers, Cagliari, Italy. doi: 10.1145/2212908.2212928
- Oliveira, L., Rodrigues, J. J., Kozlov, S. A., Rabêlo, R. A., & Albuquerque, V. H. C. d. (2019). MAC Layer Protocols for Internet of Things: A Survey. *Future Internet*, 11(1), 16.
- Orcioni, S., Giammarini, M., Scavongelli, C., Vece, G. B., & Conti, M. (2016). Energy estimation in SystemC with Powersim. *Integration, the VLSI Journal*, 55, 118-128.
- Oussama, B., Dominic, D., & Frederic, N. (2018). A Model-Based Approach for the Design of Ultra-Low Power Wireless Sensor Nodes. Dans *NEWCAS 2018 - 16th IEEE International NEWCAS conference* (pp. 248-251).

- Patil, M., & Biradar, R. C. (2017). Dynamic error control scheme based on channel characteristics in wireless sensor networks. Dans *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (pp. 736-741). doi: 10.1109/RTEICT.2017.8256694
- Phillips, C., Sicker, D., & Grunwald, D. (2013). A Survey of Wireless Path Loss Prediction and Coverage Mapping Methods. *IEEE Communications Surveys & Tutorials*, 15(1), 255-270. doi: 10.1109/SURV.2012.022412.00172
- Pocovi, G., Kolding, T., Lauridsen, M., Mogensen, R., Markmller, C., & Jess-Williams, R. (2018). Measurement Framework for Assessing Reliable Real-Time Capabilities of Wireless Networks. *IEEE Communications Magazine*, 56(12), 156-163. doi: 10.1109/MCOM.2018.1800159
- Rodrigues, M. L., Montez, C., Budke, G., Vasques, F., & Portugal, P. (2017). Estimating the Lifetime of Wireless Sensor Network Nodes through the Use of Embedded Analytical Battery Models. *Journal of Sensor and Actuator Networks*, 6(2). doi: 10.3390/jsan6020008
- Rosas, F., Souza, R. D., Pellenz, M. E., Oberli, C., Brante, G., Verhelst, M., & Pollin, S. (2016). Optimizing the Code Rate of Energy-Constrained Wireless Communications With HARQ. *IEEE Transactions on Wireless Communications*, 15(1), 191-205. doi: 10.1109/TWC.2015.2469289
- Sadok, B., Oussama, B., & Faouzi, D. (2018). Digital Back-end Based on a Low-power Listening Protocol for Wake-Up Receivers. *Sensors & Transducers*, 224(8), 22-27.
- Sansoda, B., Khattiya, W., & Choomchuay, S. (2013). Performance comparison of convolutional codes for UWB-WBAN applications. Dans *2013 IEEE Region 10 Humanitarian Technology Conference* (pp. 336-339). doi: 10.1109/R10-HTC.2013.6669066
- Sarvi, B., Rabiee, H. R., & Mizanian, K. (2017). An adaptive cross-layer error control protocol for wireless multimedia sensor networks. *Ad Hoc Networks*, 56, 173-185.
- Selvakumar, D., Nanda, K., & Pasupuleti, H. B. (2015). Wireless sensor device hardware architecture — Design and analysis for high availability. Dans *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)* (pp. 1-5). doi: 10.1109/NTMS.2015.7266533
- Serrano, W. (2018). Digital Systems in Smart City and Infrastructure: Digital as a Service. *Smart Cities*, 1(1). doi: 10.3390/smartcities1010008

- Shariatmadari, H., Li, Z., Iraj, S., Uusitalo, M. A., & Jäntti, R. (2017). Control channel enhancements for ultra-reliable low-latency communications. Dans *2017 IEEE International Conference on Communications Workshops (ICC Workshops)* (pp. 504-509). doi: 10.1109/ICCW.2017.7962708
- Sheikh, A. U. H., & Parsons, J. D. (1983). The frequency dependence of urban man-made radio noise. *Radio and Electronic Engineer*, 53(3), 92-98. doi: 10.1049/ree.1983.0022
- Singh, D., Aliu, O. G., & Kretschmer, M. (2018). LoRa WanEvaluation for IoT Communications. Dans *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 163-171). doi: 10.1109/ICACCI.2018.8554713
- Sodhi, Z., Flatt, J., & Jean-Noé. (2018). Getting to the open smart city (pp. 6,10): Future Cities.
- Sommer, P., Maret, Y., & Dzung, D. (2018). Low-Power Wide-Area Networks for Industrial Sensing Applications. Dans *2018 IEEE International Conference on Industrial Internet (ICII)* (pp. 23-32). doi: 10.1109/ICII.2018.00011
- Srbínovska, M., Dimcev, V., & Gavrovski, C. (2017). Energy consumption estimation of wireless sensor networks in greenhouse crop production. Dans *IEEE EUROCON 2017 -17th International Conference on Smart Technologies* (pp. 870-875). doi: 10.1109/EUROCON.2017.8011235
- Sun, W., Yuan, X., Wang, J., Li, Q., Chen, L., & Mu, D. (2018). End-to-End Data Delivery Reliability Model for Estimating and Optimizing the Link Quality of Industrial WSNs. *IEEE Transactions on Automation Science and Engineering*, 15(3), 1127-1137. doi: 10.1109/TASE.2017.2739342
- Swaminathan, R., & MadhuKumar, A. S. (2016). Joint recognition of error correcting codes and interleaver parameters in a robust environment. Dans *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)* (pp. 1-6). doi: 10.1109/PIMRC.2016.7794694
- Tan, Y. K., & Panda, S. K. (2011). Energy Harvesting From Hybrid Indoor Ambient Light and Thermal Energy Sources for Enhanced Performance of Wireless Sensor Nodes. *IEEE Transactions on Industrial Electronics*, 58(9), 4424-4435. doi: 10.1109/TIE.2010.2102321
- Toledo, S., Orchan, Y., Shohami, D., Charter, M., & Nathan, R. (2018). Physical-Layer Protocols for Lightweight Wildlife Tags with Internet-of-Things Transceivers.
- Veiga, A. A., & Abbas, J. C. (2019). Proposal and Application of Bluetooth Mesh Profile for Smart Cities' Services. *Smart Cities*, 2(1). doi: 10.3390/smartcities2010001

- Vilajosana, X., Tuset-Peiro, P., Vazquez-Gallego, F., Alonso-Zarate, J., & Alonso, L. (2014). Standardized Low-Power Wireless Communication Technologies for Distributed Sensing Applications. *Sensors*, 14(2), 2663.
- Vilajosana, X., Wang, Q., Chraim, F., Watteyne, T., Chang, T., & Pister, K. S. J. (2014). A Realistic Energy Consumption Model for TSCH Networks. *IEEE Sensors Journal*, 14(2), 482-489. doi: 10.1109/JSEN.2013.2285411
- Virk, K., Hansen, K., & Madsen, J. (2005). System-level Modeling of Wireless Integrated Sensor Networks. Dans *2005 International Symposium on System-on-Chip* (pp. 179-182). doi: 10.1109/ISSOC.2005.1595672
- Vračar, L., Prijić, A., Nešić, D., Dević, S., & Prijić, Z. (2016). Photovoltaic Energy Harvesting Wireless Sensor Node for Telemetry Applications Optimized for Low Illumination Levels. *Electronics*, 5(2), 26.
- Vuran, M. C., & Akyildiz, I. F. (2009). Error Control in Wireless Sensor Networks: A Cross Layer Analysis. *IEEE/ACM Transactions on Networking*, 17(4), 1186-1199. doi: 10.1109/TNET.2008.2009971
- Wang, D., & Zhang, J. (2017). The Influence of a Particular Error Pattern on the Error Correction Performance of RS+Convolutional Concatenated Code. Dans *2017 International Conference on Computer Systems, Electronics and Control (ICCSEC)* (pp. 358-361). doi: 10.1109/ICCSEC.2017.8446698
- Wu, F., Rüdiger, C., & Yuce, M. (2017). Real-Time Performance of a Self-Powered Environmental IoT Sensor Network System. *Sensors*, 17(2), 282.
- Wu, P., & Jindal, N. (2011). Coding versus ARQ in Fading Channels: How Reliable Should the PHY Be? *IEEE Transactions on Communications*, 59(12), 3363-3374. doi: 10.1109/TCOMM.2011.102011.100152
- Xu, L. D., He, W., & Li, S. (2014). Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233-2243. doi: 10.1109/TII.2014.2300753
- Zhao, X., Guo, J., Chou, C. T., Misra, A., & Jha, S. K. (2015). High-Throughput Reliable Multicast in Multi-Hop Wireless Mesh Networks. *IEEE Transactions on Mobile Computing*, 14(4), 728-741. doi: 10.1109/TMC.2014.2333731
- Zorzi, M., Rao, R. R., & Milstein, L. B. (1997). ARQ error control for fading mobile radio channels. *IEEE Transactions on Vehicular Technology*, 46(2), 445-455. doi: 10.1109/25.580783