

CONTRIBUTIONS TO THE EFFICIENT USE OF WIRELESS SENSOR NETWORKS IN DISTURBED ENVIRONMENTS

Teză destinată obținerii
titlului științific de doctor inginer
la
Universitatea "Politehnica" din Timișoara
în domeniul INGINERIE ELECTRONICĂ
ȘI TELECOMUNICAȚII
de către

Ing. Cosmin Cîrstea

Conducător științific: prof.univ.dr.ing. Aurel Gontean
Referenți științifici: prof.univ.dr.ing. Teodor Petrescu
prof.univ.dr.ing. Dan Pitică
prof.univ.dr.ing. Marius Oteșteanu

Ziua susținerii tezei: 28.02.2013

Seriile Teze de doctorat ale UPT sunt:

- | | |
|---|--|
| 1. Automatică | 9. Inginerie Mecanică |
| 2. Chimie | 10. Știința Calculatoarelor |
| 3. Energetică | 11. Știința și Ingineria Materialelor |
| 4. Ingineria Chimică | 12. Ingineria sistemelor |
| 5. Inginerie Civilă | 13. Inginerie energetică |
| 6. Inginerie Electrică | 14. Calculatoare și tehnologia informației |
| 7. Inginerie Electronică și Telecomunicații | 15. Ingineria materialelor |
| 8. Inginerie Industrială | |

Universitatea „Politehnica” din Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2013

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității „Politehnica” din Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Acknowledgements

This work was partially supported by the strategic grant POSDRU/88/1.5/S/50783, Project ID50783 (2009), co-financed by the European Social Fund – Investing in People, within the Sectoral Operational Programme Human Resources Development 2007-2013.

This paper is primarily addressed to those interested in the field of wireless sensor networks, specifically in modelling of harvested power as well as in routing and scheduling protocols for networks with static and mobile nodes.

The work could not be completed without the support and care of my family and friends to whom I would like to thank.

I would like to express my sincere gratitude towards Prof. Dr. Ing. Aurel Gontean for his implication, support and ideas that led to the completion of this work.

I would also like to thank Dr. Ing. Teodor Petrita for his implication, ideas and suggestions that led to the elaboration of Chapter 2 and to SI. Dr. Ing. Cristian Vasar and Conf. Dr. Ing. Catalin Căleanu for the support they have given me in the development of the thesis.

I also address special thanks to my friend and colleague Dr. Ing. Mihai Cernăianu for all those interesting discussions, advices and undelayed help.

Special consideration goes to my friend Ing. Roxana Davidescu for the help and advices she has given me for the work described in Chapter 5.

Timișoara, February 2013

Cosmin Cîrstea

***To my family
and friends,***

Cîrstea, Cosmin

Contributions to the efficient use of wireless sensor networks in disturbed environments

Teze de doctorat ale UPT, Seria 7, Nr. 61, Editura Politehnica, 2013, 129 pagini, 62 figuri, 14 tabele.

ISSN: 1842-7014

ISBN: 978-606-554-630-1

Cuvinte cheie: wireless sensor networks, energy harvesting model, routing and scheduling protocols, simulation environment.

Abstract,

This thesis is focused on improving the functionality of wireless sensor networks (WSNs) through efficient protocols with the purpose of extending the overall lifetime of network nodes while maintaining a high quality of service. The numerous applications in which WSNs are used some of which require placement in harsh environments where human access is seldom possible combined with the limited power capabilities of sensor nodes have determined researchers to develop various methods for improving energy consumption. The work presented in this thesis addresses energy efficiency at the node and network level for networks with static and mobile nodes by proposing algorithms for hierarchical organization, optimum routing, path determination and scheduling of tasks. It also places the foundations of a simulator for WSNs and provides insight into energy harvesting solutions by proposing a model for determining the capabilities of radio frequency energy harvesting systems.

Contents

Chapter 1 Wireless sensor networks: applications and challenges	11
1.1 Goals of the work.....	11
1.2 Applications for WSNs.....	11
1.3 Architectures and topologies for WSNs.....	12
1.3.1 Architectures for WSNs	13
1.3.2 Network topologies	15
1.3.3 Protocol stack for WSNs	18
1.4 Classification of protocols for WSNs.....	19
1.4.1 Medium access protocols	19
1.4.2 Routing protocols	23
1.4.3 Transport protocols.....	27
1.4.4 Time synchronization protocols	27
1.5 Energy harvesting solutions for WSNs.....	28
1.5.1 Photovoltaic energy harvesters	28
1.5.2 Piezoelectric energy harvesters	28
1.5.3 Thermoelectric energy harvesters.....	29
1.5.4 Radio frequency energy harvesters.....	29
1.6 Conclusions.....	31
Chapter 2 Performance analysis and modelling of a RF energy harvester	32
2.1 Introduction	32
2.2 Inductive current measurement system	32
2.2.1 Experimental setup used for performance analysis.....	34
2.2.2 Validation of the ICMS functionality	35
2.3 Modelling power transfer.....	37
2.3.1 Modelling antenna pattern	39
2.3.2 Modelling ground reflections	39
2.4 Proposed model validation	41
2.5 Conclusions.....	46
2.6 Contributions.....	47
Chapter 3 A graphical user interface for simulating WSNs	48
3.1 Introduction	48
3.2 Description of the proposed visualization tool for WSNs	49
3.2.1 Network and simulation dependent parameters	50
3.2.2 Functionality of the graphical window.....	55

3.3	Conclusions.....	56
3.4	Contributions.....	57
Chapter 4	Efficient clustering techniques for static WSNs	58
4.1	Introduction	58
4.2	A new method for cluster head election	59
4.2.1	Proposed algorithm.....	60
4.2.2	Simulation analysis of the proposed CH election method	62
4.3	A new method for electing attending CHs.....	67
4.3.1	Description of the experimental setup	68
4.3.2	Analysis of the proposed method for election of attending CHs	70
4.4	Conclusions.....	75
4.5	Contributions.....	76
Chapter 5	Routing and scheduling techniques for mobile WSNs	77
5.1	Introduction	77
5.2	A GA based routing approach for mobile WSNs	78
5.2.1	Stages of the genetic algorithm.....	79
5.2.2	Simulation environment and network parameters	84
5.2.3	Simulation results.....	85
5.3	A reinforcement learning strategy for task scheduling.....	89
5.3.1	Scheduling of tasks using Q-Learning.....	91
5.3.2	Description of the simulation scenario	97
5.3.3	Analysis of the simulation results.....	99
5.4	Conclusions.....	104
5.5	Contributions.....	106
Chapter 6	Conclusions and contributions.....	107
6.1	Thesis overview and conclusions	107
6.2	Summary of contributions	110
	Bibliography.....	111
	Appendix A	121
	Appendix B	122
	Appendix C	124
	Appendix D	126
	Appendix E.....	127

Index of figures

Figure 1.1 Single-hop architecture for WSN with 4 nodes and 1 base station	13
Figure 1.2 Multi-hop architecture for WSN with 4 nodes and 1 base station	14
Figure 1.3 Star network topology	16
Figure 1.4 Mesh network topology with four sensor nodes and one base station ...	17
Figure 1.5 Hybrid star-mesh network topology	17
Figure 1.6 Protocol stack for WSNs [1]	18
Figure 1.7 S-MAC frame [15]	21
Figure 1.8 P2110 Powerharvester Receiver IC block diagram [5].....	30
Figure 2.1 a. ICMS simplified block diagram; b. ICMS circuit and casing [56]	33
Figure 2.2 Block diagram of the measurement setup [56]	35
Figure 2.3 INA output voltage; b. Calculated slope values [56]	36
Figure 2.4 a. Current over 5k Ω resistor; b. Current over 100 μ F capacitor [56]	37
Figure 2.5 Field regions of an antenna [61].....	38
Figure 2.6 Two ray ground model	40
Figure 2.7 Reflection coefficient modulus for marble according to equation 2.6	40
Figure 2.8 a. Hallway measurement setup; b. Parking lot measurement setup.....	42
Figure 2.9 In line measurement – average harvested power vs. distance	43
Figure 2.10 a. Harvested power analysis setup; b. Spectral analysis setup.....	43
Figure 2.11 Harvested vs. received power levels.....	44
Figure 2.12 Received vs. 2 ray vs. 1 ray power levels.....	45
Figure 2.13 Indoor harvested vs. received vs. modelled power, in line on maximum radiation direction.....	45
Figure 2.14 RF-DC converter efficiency vs. RF_{IN} (dBm) [50]	46
Figure 3.1 Proposed visualization tool for modeling WSN behavior	49
Figure 3.2 Network properties tab.....	51
Figure 3.3 Node properties tab	52
Figure 3.4 New node type	52
Figure 3.5 Node hardware consumption pop-up	53
Figure 3.6 Statistics tab.....	54
Figure 3.7 Statistics about all the actions executed by node 1	56

8 Index of figures

Figure 4.1 Half nodes die (proposed algorithm) vs. last node dies (IMSD) [84]	62
Figure 4.2 Number of packets sent for different network sizes (proposed vs. IMSD) [84]	64
Figure 4.3 Number of packets sent for different number of CHs (proposed vs. IMSD) [84]	65
Figure 4.4 Number of packets sent for different packet sizes (proposed vs. IMSD) [84]	66
Figure 4.5 Measurement location (several node positions with white) [91]	69
Figure 4.6 Packet loss between each node and the base station [91]	69
Figure 4.7 Comparison between the number of alive nodes [91]	72
Figure 4.8 Comparison of network throughput for different OA sizes [91]	72
Figure 4.9 Energy dissipated by each node due to unsuccessful packet delivery [91]	73
Figure 4.10 Average energy lost due to unsuccessful packet delivery for different OA sizes [91]	74
Figure 5.1 Block diagram of the genetic algorithm implementation	79
Figure 5.2 Representation of a path and associated chromosome	80
Figure 5.3 Example of one-point crossover	82
Figure 5.4 Example of the mutation procedure	83
Figure 5.5 Example of the repair function	83
Figure 5.6 Simulation scenario for a mobile WSN with 50 nodes placed in an OA of 300x300 meters	84
Figure 5.7 Comparison of convergence times between GA and Dijkstra`s	86
Figure 5.8 Comparison between Genetic Algorithm and Dijkstra shortest path lengths	87
Figure 5.9 Comparison between how network nodes deplete energy for different weights of the objective function	89
Figure 5.10 Block diagram of the Q-Learning algorithm	96
Figure 5.11 Representation of the simulation scenario	98
Figure 5.12 Actions executed by node 1	99
Figure 5.13 Actions executed by node 2	100
Figure 5.14 Actions executed by node 3	101
Figure 5.15 Actions executed by node 4	102
Figure 5.16 Action failure rate [%]	104

Index of tables

Table 4-1 LEACH [26] Radio characteristic parameters	58
Table 4-2 Message delivery increase [%] with network size compared to IMSD	63
Table 4-3 Message delivery increase [%] for different number of CHs compared to IMSD.....	65
Table 4-4 Message delivery increase [%] for different packet sizes compared to IMSD.....	66
Table 4-5 Detected packet loss variation with distance [91]	70
Table 4-6 Comparison between network lifetime for the two algorithms	71
Table 4-7 Increase in packet throughput of the proposed scheme compared to LEACH	73
Table 4-8 Average energy dissipated due to unsuccessful packet delivery for different OA sizes.....	74
Table 5-1 Expected prices for agent actions	94
Table 5-2 Values for Q-Learning factors	95
Table 5-3 Energy consumption of IRIS node components [92]	97
Table 5-4 Number of actions executed for nodes 1, 2 and 3	103
Table 5-5 Energy consumption/action in accordance with node and network parameters	103
Table 5-6 Energy efficiency/exhaustive action [%]	103

Acronyms

WSN	Wireless Sensor Network
QoS	Quality of Service
RF	Radio Frequency
RFID	Radio Frequency Identification
EIRP	Effective Isotropic Radiated Power
BS	Base Station
CH	Cluster Head
LEACH	Low Energy Adaptive Clustering Hierarchy
CSMA	Carrier Sense Multiple Access
TDMA	Time Division Multiple Access
CDMA	Code Division Multiple Access
RTS	Request To Send
CTS	Clear To Send
ICMS	Inductive Current Measurement System
DAQ	Data Acquisition
NI	National Instruments
VI	Virtual Instrument
HPBW	Half Power Beam-Width
EIRP	Effective Radiated Isotropic Power
(I)MSD	(Improved) Minimum Separation Distance
OA	Observation Area
F(/H/L)ND	FIRST(/HALF/LAST) Node Dies
LQ(I)	Link Quality (Indicator)
GA	Genetic Algorithm
MDP	Markov Decision Process
RL	Reinforcement Learning
MDC	Mobile Data Collector

Chapter 1

Wireless sensor networks: applications and challenges

This chapter represents a short introduction into wireless sensor networks applications, architectures, protocols used for efficient energy consumption and communication as well as energy harvesting solutions. A review of existing network architectures and protocols is presented with the purpose of identifying key features of WSNs where further research and innovative solutions are required.

1.1 Goals of the work

The primary goals of this thesis are to provide contributions to WSNs in terms of protocols developed for efficient use of node energy resources considering optimum communication and scheduling techniques. We have investigated currently developed MAC layer protocols for both static and mobile WSNs with the purpose of highlighting existing drawbacks and providing solutions which will increase the efficiency of the network in terms QoS (extended lifetime, packet throughput, task selection and optimum scheduling).

Energy harvesting solutions have also been investigated and we have shown that the process of harvesting RF energy can be successfully modeled and simulated to provide insight into the amount of energy which can be available using this technique. Also the design of a low input power battery charging circuit has been proposed.

We have also began the development of a WSN simulator for both static and mobile nodes capable of allowing the user to define simulation scenarios at will and which can provide useful statistic information.

1.2 Applications for WSNs

Advancements and developments in electronics, sensor technologies and communications that have marked the past decade have influenced the evolution of new attractive research directions such as WSN.

A WSN made up of a large number (tens or even hundreds) of sensor nodes organized in an ad hoc fashion and deployed in an area of interest with the purpose of performing sensing or actuating tasks. Features such as small size, low power and the ability of wireless communication make WSN the ideal solution for numerous applications such as remote environmental monitoring, medical healthcare monitoring, military surveillance, etc. [1] Using WSN in such distinct

applications has determined the development of a considerable number of networks, from the point of view of design and operation, as there are applications. However, given the fact that these networks are mostly deployed in field where human access is rarely possible, several constraints have to be taken into consideration such as the nodes' limited resources (memory, computational ability, energy, bandwidth, coverage, reliability, communication security, operation mode for extended lifetime etc.) which pose severe constraints in terms of QoS. A description of several application fields of WSNs is briefly presented in the following paragraphs.

Environmental observation – refers to a plurality of uses beneficial both for the general population such as an early warning system but also for the research community, in the form of long term data collection [2]. These features make applications such as volcanic studies and eruption warning systems, meteorological observation, fire detection, earthquake, flood and tsunami warning of great importance in the context of developing countries.

Disaster prevention – such WSNs are deployed in potentially hazardous locations where human intervention is frequent with the purpose of increasing workplace safety such as underground mines (monitoring deteriorating grounds, toxic gases), refineries (monitoring gases and personel in case of fire) etc.

Agricultural management – in the context of a continuously growing number of population worldwide, the necessity of satisfying food demands is slowly becoming a problem especially in arid regions where people suffer from hunger because precipitation is scarce and unpredictable [3]. In order to prepare correct strategies for effectively using available resources, information on several environmental parameters such as soil moisture, precipitation frequency etc., are needed.

Structural health monitoring – refers to detecting, localizing, estimating the extent of damage and predicting the remainder of life time for a given structure [4]. Because of the many advantages that they provide such as low production and maintenance costs, reduced size, ease of deployment, wide area of coverage, etc., WSN currently provide the best solution for such applications [5].

Military surveillance – due to the fact that they can be easily deployed on the battlefield, WSNs are used in several military applications such as monitoring militant activities in remote areas, identification and movement of enemy forces and analysis of their progress [6].

1.3 Architectures and topologies for WSNs

As previously mentioned, a WSN consists of a significant number of sensing nodes (*motes*) deployed in a region of interest and one or more base stations (*sinks*) located close to or inside the sensing region. Base stations (BS) issue commands and gathers data from sensor nodes, acts upon the received data and forwards it to a server or gateway outside the network for interpretation. Sensing nodes acquire information from the environment and forward it either directly to the BS if it is within range or to neighboring nodes which in turn forward the information until it reaches the BS.

1.3.1 Architectures for WSNs

WSNs architectures are highly dependent on the application they are intended for and on the environment they are distributed in. According to [1] WSNs can be classified into five different categories from the point of view of communication, mobility, placement, configuration and node type:

I. Communication method

Depending on the size of the observation area, network topology and environment characteristics, the communication method can be classified in two categories: single-hop and multi-hop.

- *Single hop communication* is used when a node is in the vicinity of a BS and can communicate directly with it. Single-hop communication is more susceptible to communication errors and packet loss and can also be very energy consuming for long distance communication [7]. Figure 1.1 presents the single-hop architecture with four nodes and a base station.

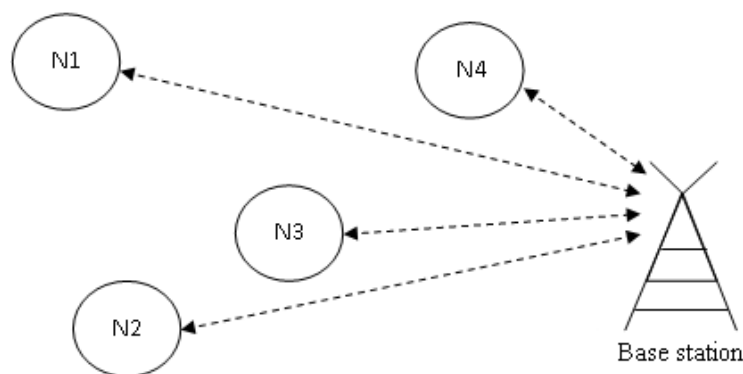


Figure 1.1 Single-hop architecture for WSN with 4 nodes and 1 base station

- *Multi-hop communication* is used when a node is located at a greater distance from the BS than the transmitting range. In this case several approaches are available.
 - Flooding, where a node broadcasts messages to all neighbors which in turn forward them to their neighbors until they reach the BS. This method is highly redundant as the same message reaches the base station from several paths. This high redundancy also translates in high communication costs as the same message is transported via multiple paths until it reaches the base station.
 - Such issues can be avoided through several approaches such as allocating an optimal path for message delivery or adaptively creating clusters and electing cluster heads (CHs) to serve as “*in field*” BS that

communicate with each other or transmit the message directly to the sink. Such approaches are described in detail in subchapter 1.5. Figure 1.2 shows a representation of a multi-hop WSN with four nodes and one base station.

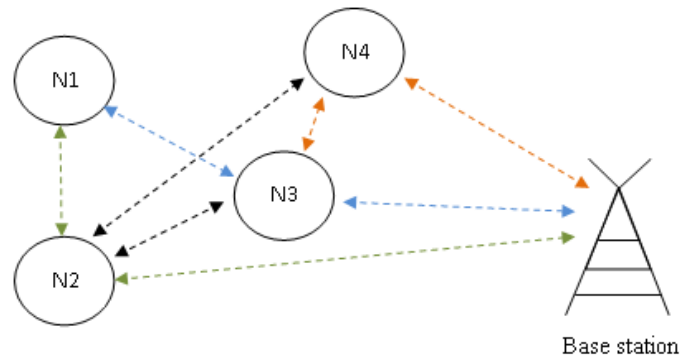


Figure 1.2 Multi-hop architecture for WSN with 4 nodes and 1 base station

However opinions are split in the research community when it comes to choosing between single-hop and multi-hop communication.

Fedor and Collier [8] have proven theoretically and experimentally that it is more energy efficient to use multi-hop communication. However their experiments were limited to the situation when nodes are aligned and the intermediate node is located at the middle of the distance between the source node and the destination node.

On the other hand, the authors of [9] have proven through mathematical modelling and experimental tests that for using a transceiver which has a reception power much higher than the maximum transmitting power, single-hop transmission is more effective.

We can conclude that choosing the correct architectural approach between single and multi-hop networks is highly dependent on the nature of the application.

II. Mobility for WSNs

From the mobility point of view of nodes, WSNs can be static or mobile. The most frequently met WSNs are static ones. Such networks are easier to deploy, control and maintain. On the other hand, there are several applications that require mobile sensor nodes to accomplish the task of sensing and actuating. An example of such a network is a wireless biosensor network using autonomously controlled animals [10]. This type of network uses a wireless transceiver attached to a rat. By sending different impulses to the brain of the rat it can be controlled to go left or right. Attaching a miniature camera to the rat and sending it into different disaster areas in search of trapped humans combined with the ability to control the rat's motion turns it into a biosensor. Compared to static networks the design of mobile networks must take into consideration the mobility effect which increases the complexity of implementation.

III. Placement for WSNs

When considering the deployment of nodes WSNs can be characterized into deterministic and non-deterministic networks. In deterministic networks sensor nodes are placed according to a plan and their positions are fixed and do not change once deployed. This type of sensor network is seldom used as few applications allow for a previously built deployment plan. In most situations non deterministic networks are used, as planning is difficult or sometimes impossible due to the harsh and hostile deployment environments. Non deterministic networks are more scalable and flexible but also require higher control complexity which translates into increased power consumption [1].

IV. Configuration for WSNs

From the point of view of configuration, WSNs can be divided into non self-configurable and self-configurable. Non self-configurable networks have nodes that are not able to organize themselves into a network and for that reason they rely on a central controller that issues individual commands for each node, collects data and instructs the node what to do. This type of network is only suitable for small scale applications. In contrast, self-configurable networks have nodes that possess the ability to autonomously organize and maintain their connectivity as well as collaborate to accomplish a given task. A network with self-configurability is suitable for large scale applications as it allows the accomplishment of complicated sensing tasks.

V. Node type

WSNs can be classified into homogeneous and heterogeneous networks depending on the type of nodes used. Homogeneous networks are made up of sensor nodes that have the same capabilities in terms of storage, energy consumption and computational power. On the other hand, heterogeneous networks have sensor nodes that are equipped with more processing and communication capabilities than most other nodes. These nodes are used to perform more difficult tasks with the purpose of improving the overall energy consumption and prolonging the lifetime of the network.

1.3.2 Network topologies

There are a number of different network topologies for radio communications out of which only a number apply to WSN such as star, mesh and hybrid star-mesh networks. Each of these topologies comes with a series of advantages and disadvantages and several compromises have to be made when selecting the proper topology. A brief description of these network topologies is outlined below.

I. Star Networks

A star network is a communication topology in which remote nodes can only transceive messages to and from the BS and not between themselves. Figure 1.3 depicts a star network topology with four sensor nodes and one BS.

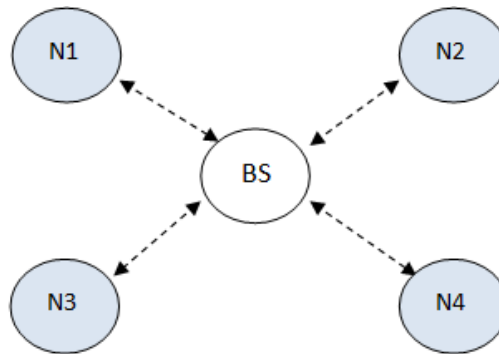


Figure 1.3 Star network topology

Using such a topology presents several advantages, such as architectural simplicity, low communication latency and the ability to keep the node's power consumption to a minimum. The disadvantage is that the BS must be within communication range with all the nodes of the network which makes them highly dependent on the BS. In case one of the nodes fails, the whole network is not affected, but when the central node fails, the whole network stops functioning.

II. Mesh networks

A mesh network allows all nodes within transmission range to communicate with each other, thus forming a multi-hop network. This type of network has the advantage of reduced redundancy and scalability. The network can be easily extended by adding more nodes which can in turn forward the information to the desired location. In case of node failure adjacent nodes can still communicate with each other using other nodes to forward the message to the desired location. The disadvantage of such a network topology is higher power consumption for the nodes that are used for multi hopping (message forwarding). In addition, as the number of communication hops increases, so does the time necessary for message delivery.

Figure 1.4 presents a mesh network topology with four sensor nodes and one base station.

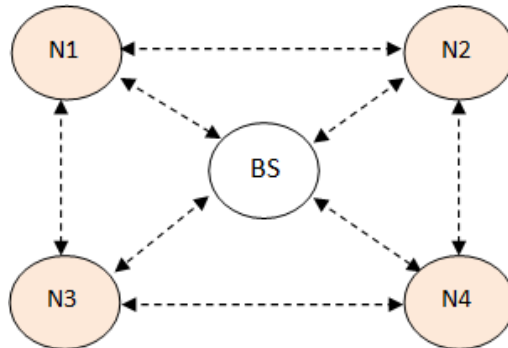


Figure 1.4 Mesh network topology with four sensor nodes and one base station

III. Hybrid star-mesh networks

The hybrid star-mesh network is made up of a combination of star and mesh networks used together to obtain a highly efficient communications topology while maintaining the ability to keep power consumption at a minimum. Figure 1.5 presents the hybrid star-mesh network topology.

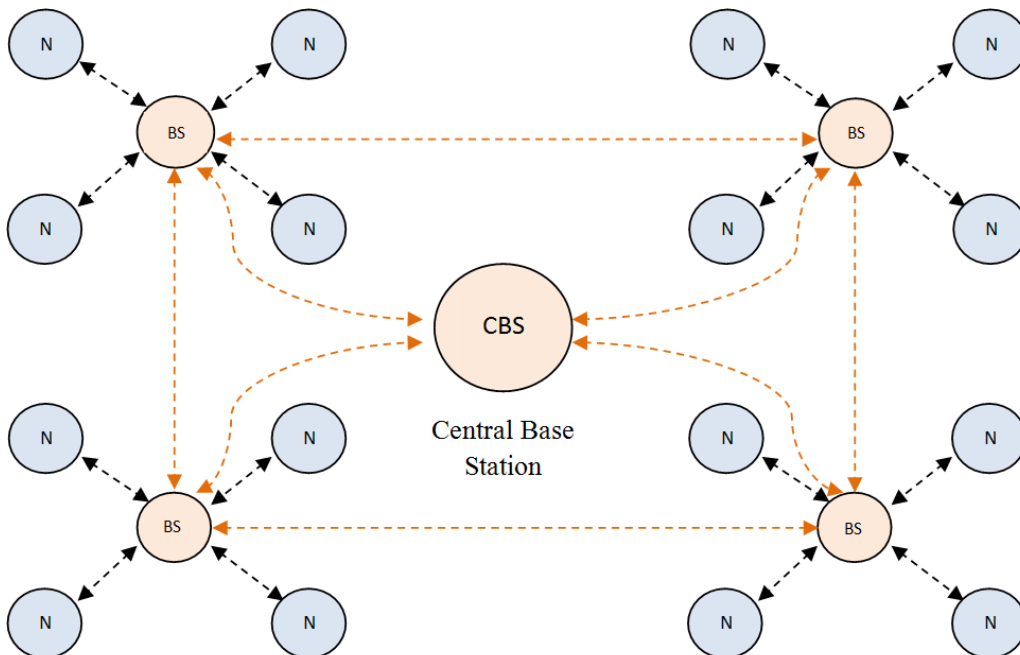


Figure 1.5 Hybrid star-mesh network topology

As can be seen from Figure 1.5, the lowest nodes in the topology are not enabled with multi-hopping capabilities (nodes marked with *N* in the above figure) and this

allows for reduced power consumption at this level. All other nodes have this capability enabling them to forward messages from the low power nodes to the other nodes in the network. Generally, the nodes with multi-hop capabilities have higher power consumption because they have to receive and transmit messages from adjacent nodes to other nodes with multi-hop capabilities.

1.3.3 Protocol stack for WSNs

When considering the protocol stack for WSNs depicted in Figure 1.6 one can observe that there are similarities with the Open System Interconnect model used for local area networks.

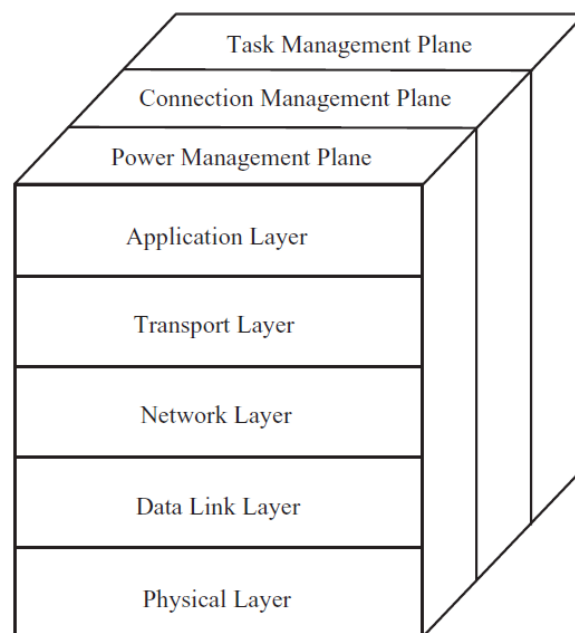


Figure 1.6 Protocol stack for WSNs [1]

The protocol stack for WSN consists of five layers according to [1]:

- *Application layer* – contains a variety of application protocols to suite various sensor network applications
- *Transport layer* – responsible for reliable data delivery
- *Network layer* – responsible for routing the data.
- *Data link layer* – responsible for data stream multiplexing, data frame transmission and reception, medium access and error control.
- *Physical layer* – responsible for physical transmission and reception over a communication medium.

On the other hand the protocol stack can be divided into a group of management planes across each layer according to [11]. These layers are:

- *Power management plane* – responsible for managing the power level of a sensor node for sensing, processing and transmitting/receiving. For example at the MAC layer, the node can turn off its transceiver when there is nothing to transmit or receive, thus saving power.
- *Connection management plane* – responsible with the connectivity within the network when the network is firstly deployed, or in case a node fails and data must be rerouted.
- *Task management plane* – responsible for task distribution among sensors nodes in a network in order to improve energy consumption and prolong network lifetime.

1.4 Classification of protocols for WSNs

Efficient energy consumption is one of the main challenges for WSN due to the numerous application fields that use such networks. As the nature of the environment seldom allows human intervention combined with the fact that sensor nodes are battery powered pose severe constraints on WSNs in terms of energy consumption. That is why numerous protocols have been developed that try to optimize various parameters of the sensor network with the purpose of optimizing energy consumption. It is important to mention that protocols which address communication issues are based on the IEEE 802.15.4/ZigBee [12 13] wireless communication standards. The next paragraphs of this chapter present a classification of energy saving protocols currently used for WSNs with focus on those categories which are of interest for this work.

1.4.1 Medium access protocols

Medium access control (MAC) protocols are one of the most important for WSN efficient communication as a lot of energy is consumed in several faulty situations [14] such as:

- *Collisions* – can occur when two or more sensors try to send data at the same time over the same transmission medium. All packets that cause collisions have to be discarded and retransmitted which causes an increase in energy consumption.
- *Overhearing* – happens when a node receives packets that are designated to other nodes.
- *Idle listening* – refers to the case when packet transmission/reception is not synchronized and a node listens to a transmission medium to receive possible traffic designated to it.
- *Over emitting* – is the case when a node tries to communicate with another that is not ready to receive. The first node continuously transmits a message until the other node receives it.
- *Packet overhead* – refers to the situation when a bigger number of packets are used to make a data transmission than normally required.

The design of a good MAC protocol should take into consideration the following aspects:

- *Energy efficiency* – refers to the energy consumed per unit of successful communication. A MAC protocol must be energy efficient in order to prolong the lifetime of the network.
- *Scalability* – should be able to be used in networks where the number of nodes is not previously defined.
- *Adaptability* – should be able to adapt to unexpected changes in the network topology such as failures due to limited node lifetime, node addition altered connectivity, etc.
- *Latency* – refers to the communication delay between the time a packet has been sent and the time it is received. For some applications latency is not a critical factor but on the other hand real time monitoring of an event has very strict latency requirements.

MAC protocols can be divided into three categories: contention-based, contention-free and hybrid protocols [14].

I. Contention-based protocols

In WSNs that use contention based protocols sensor nodes use probabilistic media access control protocols such as Carrier Sense Multiple Access (CSMA) in which a node verifies the absence of traffic on the shared transmission medium before transmitting. "Carrier sense" refers to the fact that a transmitter uses feedback from the receiver that detects a carrier wave before transmitting. If a carrier is sensed the node waits for the transmission medium to be free and then sends its own messages. "Multiple access" describes the fact that multiple nodes use the same medium (bandwidth) to send and receive data. Contention-based protocols also use variations of CSMA such as CSMA with Collision Detection (CSMA/CD) in which when a collision is detected transmission is stopped thus avoiding the possibility of a second collision. Another improvement to CSMA is CSMA with Collision Avoidance (CSMA/CA) where if the transmission medium is sensed as busy the transmitting node backs off for a random time after which the process is repeated. Numerous contention-based protocols for WSNs have been developed out of which we would like to point out some of the most representing such as Sensor-MAC (S-MAC) [15,16], Dynamic duty cycle S-MAC (DS-MAC) [17] or the Mobility aware S-MAC (MS-MAC) [18].

For example the **S-MAC** protocol proposed by Ye et al. [15, 16] that has the primary goal of reducing energy consumption while maintaining scalability and collision avoidance. Such performances are obtained through periodic listen and sleep, coordinated synchronization and message passing. A complete cycle is called a frame. Each frame is made up of a listen period reserved for communication with other nodes, followed by a sleep period during which the node goes into idle mode if it has no data to transceive, or remains awake if does as shown in Figure 1.7.

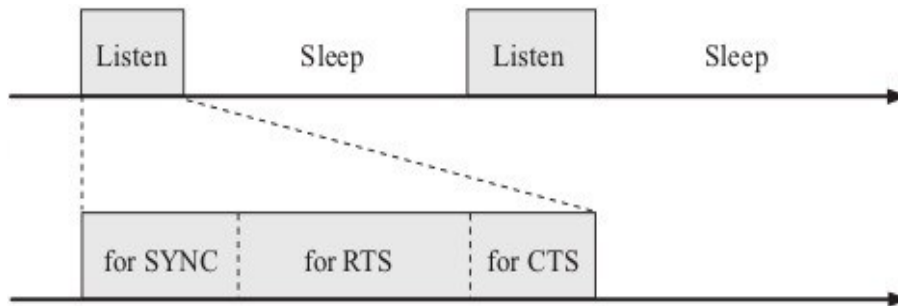


Figure 1.7 S-MAC frame [15]

The listen period is divided into three smaller intervals SYNC, RTS and CTS. The duration of the listen period is dependent on MAC layer parameters such as radio bandwidth and contention period.

In S-MAC all nodes are free to choose their own listen and sleep schedules but in order to avoid overhead neighboring nodes coordinate their sleep/listen schedules. To establish such coordination a node exchanges its schedule by periodically broadcasting a SYNC packet to all neighboring nodes. S-MAC also allows different schedules to enable multi-hop operation of the network. In order to avoid timing errors due to clock drift, a relative timestamp is used and listen periods are made longer than the clock drift. To maintain synchronization and avoid long term clock drifts a node will update its schedule and broadcasts it periodically in a SYNC packet that contains the address of the sender and the next sleep time.

To avoid collision S-MAC uses both virtual and physical carrier sensing. In virtual carrier sensing each data packet contains a duration field that indicates the duration of the transmission. When a node wants to send data it first checks for the availability of the transmission medium. If the medium is unavailable it will receive a data packet designated to another node and read the duration of the transmission field. This field is recorded internally in a variable called network allocation vector (NAV), and sets a timer for it. When the timer reaches zero, the node knows that the medium is free and tries to transmit its message. In physical carrier sensing a node that wishes to transmit performs carrier sensing. If the medium is busy the node will go to sleep for an arbitrary amount of time after which it will check for the availability of the medium again. This procedure repeats until the medium is free and the node is able to send its data. To avoid overhearing S-MAC puts a node to sleep after it receives an RTS or CTS packet that is not designated to it. A node can then wake up when the NAV value becomes zero.

S-MAC also introduces a transmission mechanism called message passing. As data packets can be very big, if one byte is wrongly transmitted the whole packet has to be resent. To avoid such problems S-MAC segments a data packet into small fragments and sends them in a burst. Only one RTS and CTS are used to reserve the medium. If an ACK is not received for a fragment that fragment is sent again.

S-MAC is more energy efficient than IEEE 802.11 but due to fixed sleep time/awake periods some portion of the bandwidth is always unusable and this makes the transmission delay higher. The main disadvantage of S-MAC is high message delivery latency as S-MAC sacrifices latency for energy saving [16].

The **DS-MAC** protocol proposed by Lin et al. [17] achieves a good tradeoff between energy consumption and latency without incurring much overhead. All nodes assume the functionality of S-MAC but each node also keeps account of their energy consumption level and average latency. Based on these two factors nodes dynamically adapt the duty cycle so that if the sleep period becomes intolerable, the listen period is doubled and more time is allocated for transeiving messages. Thus the high latency problem is solved but not the high traffic load one. Comparing to S-MAC nodes must keep account of their average latency and energy consumption, which requires additional storage and processing.

The **MS-MAC** [18] focuses on WSNs that are used in mobile applications such as monitoring patients' health conditions, wearable bio-sensors or soldiers equipped with sensors in battle field. MS-MAC is practically the same as S-MAC when the nodes are stationary. Mobility is addressed based on a change in the received SYNC signal level from a neighboring node which signifies that either the node in question or neighboring ones are moving. The level of change in the received signal is an indicator of the mobiles speed. Instead of storing only schedule information SYNC messages in MS-MAC also store information about the estimated speed of a neighbor. If there is more than one mobile node the SYNC message includes the maximum estimated speed among all mobile neighboring nodes. This information is used by nodes to create an active zone around a mobile node when it moves from one cluster to another, so that the mobile node can create connections with new neighbors. Using mobility estimation makes MS-MAC a highly efficient protocol for stationary nodes while maintaining a certain level of performance for mobile networks as well.

II. Contention-free protocols

Contention-free protocols use channel sharing access schemes such as Time Division Multiple Access (TDMA) which allows nodes to share the same transmission channel by dividing the access time into time slots. All nodes that share the same channel are aware of the time division scheme and each node can only tranceive during a pre-allocated time period. The following paragraphs describe a couple of representative contention-free based protocols proposed for WSNs.

The Traffic-Adaptive Medium Access (**TRAMA**) [19] is a TDMA based protocol that tries to provide collision free channel access for WSNs. Energy efficiency is obtained by introducing transmission schedules to avoid packet collision and switching nodes into idle mode when there are no data packets intended for them. By using a transmitter election algorithm that promotes channel reuse as a function of competing traffic around any given source or receiver throughput and fairness are achieved. TRAMA employs adaptive distributed election scheme that selects receiver nodes based on a schedule announced by transmitting nodes. Nodes using this protocol exchange their two hop neighborhood information and the transmission schedule which specifies which nodes are receivers in a given interval and then select the nodes that should transmit and receive during a period of time.

The Self-Organizing Medium Access Control protocol (**SMACS**) [20] enables node discovery and schedule establishment without the need for any local or global coordinator nodes. All nodes can turn on/off their transceiver and tune the carrier frequency to different bands. A channel is assigned to a link as soon as the link is discovered, therefore by the time all nodes hear from neighbors they will have formed a connected network where at least one multi-hop path is available between

any two nodes. Each node also has a TDMA frame in which it schedules different time slots to communicate with its neighbors. During one time slot a node can only communicate with one neighbor. To reduce collision probability each link operates on a different frequency chosen randomly from the possible assigned frequency band. Using such scheduling reduces the energy consumption of a node in the detriment of high latency due to the low utilization of the bandwidth. For example, if a node wants to transmit data packets to a neighboring node it can only do this in the timeslot allocated to it even though there may be unused frequency channels at that time.

VI. Hybrid protocols

Hybrid protocols combine the characteristics of both contention-based and contention-free protocols with the purpose of obtaining energy efficient WSNs. A couple of such protocols are briefly described in the following sections.

Zebra-MAC (**Z-MAC**) [21] combines the use of TDMA and CSMA to improve node energy consumption. The main feature of Z-MAC is the adaptability to dynamic contention levels in the network. For reduced contention the protocol adopts a CSMA functionality that allows high channel utilization at low latency. Under high contention, the protocol adopts a TDMA functionality which can achieve high channel utilization and reduce collisions among two hop neighbors at low energy costs. By combining CSMA and TDMA, Z-MAC obtains high performances when it comes to synchronization errors, slot assignment failures and dynamic topology changes [8].

Funneling-MAC is a hybrid protocol [22] that uses both TDMA and CSMA/CD for addressing the funneling effect [23] which occurs in multi-hop networks when data is transmitted in a multi hop fashion from sensing nodes towards one or more sinks.

The funneling effect is characterized by increased traffic intensity in the vicinity of the sink that results in packet congestion, collisions, delays and significant energy consumption [23]. Funneling MAC uses CSMA/CD throughout the entire spread of the network and TDMA only in the funneling region to provide additional scheduling opportunities for nodes closer to the sink. Using Funneling MAC reduces the influence of the funneling effect and improves throughput and energy consumption.

1.4.2 Routing protocols

Adequate routing techniques contribute to the reduction of energy consumption, latency, data throughput which provides increased QoS. The varieties of applications in which WSNs are used have determined the development of a numerous routing protocols which can be classified into four different categories [24]:

- Location aided protocols
- Data aggregation protocols
- Mobility based protocols
- Heterogeneous protocols

This section provides a detailed description of different classes of routing protocols for WSNs and provides examples of some of the most representative protocols for each category.

I. Location aided protocols

Location aided protocols determine energy efficient communication paths between nodes using information obtained from global positioning systems (GPS) or from localization algorithms [24]. The most accurate solution but costly solution in terms of both energy and financial expenses is equipping all sensor nodes with a GPS receiver. On the other hand localization algorithms sacrifice accuracy at the price of reduced energy consumptions and production costs.

The Location based Energy-Aware Reliable routing protocol (**LEAR**) [25] is a cluster based protocol provides efficient data routing paths based on the geographical location of sensors in the network. Location information obtained from devices such as GPS. Each node publishes its position to all neighbors and each node constructs a routing table based on the distances to its neighbors. When a node wants to send messages it checks the routing table and sends them to the neighbor which has the shortest distance to its location. The shortest distance is calculated by comparing the Euclidian distances from all neighboring nodes. In order to attain energy efficiency, LEAR uses an Enhanced Greedy Algorithms for routing packages from source to destination. Cluster heads are not previously elected and any node that has the minimal distance to the source can be selected as the next hop destination. In simulation LEAR has proven to outperform other reference protocols such as LEACH [26] and efficiently extend network lifetime and packet delivery.

II. Data aggregation protocols

This category of protocols has been developed to optimize communication, energy consumption and increase QoS for dense or mobile sensor networks that collect similar data or monitor the same event. Networks in this situation suffer from increased redundancy and high energy consumption as similar measurements are acquired and transceived throughout the network. In order to address these problems adequate data aggregation and compression techniques must be adapted.

An example of such a protocol is the Low-Energy Adaptive Clustering Hierarchy (LEACH) [26] which is a dynamic clustering protocol that tries to obtain even energy consumption among sensor nodes through random rotation of local cluster heads. Each node in the network can take the role of CH at some point in time thus distributing work load among all sensor nodes. The CH election method is dependent on several parameters which will be further described. The operation of this protocol is divided into rounds (application dependent period of time) and each such round begins with a cluster organization set-up phase preceded by a steady-state phase where data transfer occurs. The set-up phase occurs as follows:

The advertisement phase – each node individually decides if it becomes a CH considering a suggested percentage (P) of CHs previously determined as well as based on the number of times the node has been CH so far. Randomly choosing a number between 0 and 1 the node (n) can advertise itself as CH if this number is less than a threshold $T(n)$ calculated as follows [26]:

$$T(n) = \begin{cases} \frac{P}{1 - P \cdot (r \bmod \frac{1}{P})}, & \text{if } n \in G \\ 0, & \text{otherwise} \end{cases} \quad (1.1)$$

where r is the current round and G is the set of nodes that have not been CHs in the last $\frac{1}{p}$ rounds.

After each round, the probability of the remaining nodes to be elected as CHs must be increased since there are fewer nodes eligible for this task.

Cluster set-up phase – after the CH have been elected, neighboring nodes are informed so that they can elect the CH to attend to. Each node decides the corresponding CH to attend to, based on the received signal strength (which is actually a measure of distance) of the cluster head advertisement message.

Inter cluster communication – after the clusters have been defined, each CH creates and broadcasts a TDMA schedule which informs each attending nodes when to communicate. Together with the schedule a CDMA code is broadcast so that interference with neighboring clusters can be avoided.

For simulation purposes the authors of LEACH have chosen the first order radio model where the radio dissipates $E_{elec}=50$ nJ/bit to run the transceiver circuit and $e_{amp}=100$ pJ/bit/m² for the transmit amplifier with an initial energy level of 0.5 J available for all network nodes. Also a path loss coefficient, $\alpha=2$, has been considered for each communication. Thus the energy needed to transmit a k bit log message over distance d can be calculated using the following formulas [26]:

$$E_{Tx}(k,d)=E_{Tx-elec}(k)+ E_{Tx-amp}(k,d)$$

$$E_{Tx}(k,d)= E_{elec} *k+e_{amp} *k*d^2 \quad (1.2)$$

The energy required to receive a message is:

$$E_{Rx}(k)=E_{Rx-elec}(k)=E_{elec} *k \quad (1.3)$$

An important factor that must considered is that the simulations performed Heinzelman et al. [26] are restricted to considering just the energy required for communication and processing but not the energy that is dissipated through sensing, logging, actuation or the transient energy required by the MCU or the transceiver to switch between different states.

Data transmission phase – begins after the clusters have been created and the TDMA schedule has been broadcast to all nodes. During this phase all nodes transmit sensed data to elected CHs. After a certain time determined a priori the next round begins and the protocol continues from the advertisement phase again.

The LEACH protocol proposed by Heinzelman et al. [26] is one of the pioneer routing protocols for WSNs. Numerous routing protocols have derived from LEACH that address issues such as data aggregation and correlation [27], heterogeneity [28] or mobility [29] to mention just a few.

III. Mobility based protocols

Mobility based protocols consider the mobile characteristics of sensor nodes when electing the best route for transporting a message from a node to a sink. Some networks have nodes which are able to move and which might go out of range or new nodes that might come within communication range. This important aspect must be taken into consideration when designing an energy efficient protocol.

An example of a protocol for WSNs that considers mobility is the Energy Aware Geographic Routing Protocol (EAGRP) proposed by Elrahim et al. [30]. This protocol takes into consideration the distances between nodes as well as the energy level of each node before choosing the correct path to send a message. This is done by calculating the average distance of all neighboring nodes of a "sender" node and their energy levels. Using this information the node with the maximum energy and which is located at a distance equal to or less than the average distance to all neighboring nodes is selected as the next destination. Mobility is addressed at the node level and before a message is sent each node also checks if it is still in the same neighborhood. If it has moved greater than a distance referred to as "flooding distance" it will send out a flood with the new position coordinates and determine the coordinates of the new quadrant it is in. The selection of the adequate neighbor with the corresponding energy level and smallest distance is made and this node will receive the packet. The procedure is repeated until the packet reaches its destination. In simulation EARGP has proven more efficient in terms of energy consumption, packet delivery, throughput and delay [30] compared to several routing protocols used for WSNs and MANETs such as Dynamic Source Routing (DSR) [31], Ad hoc On-demand Distance Vector Routing (AODV) [32] and Greedy Perimeter Stateless Routing (GPSR) [33].

IV. Heterogeneous protocols

Heterogeneous WSNs are made up of nodes with different hardware characteristics that perform specific tasks, sometimes others than the majority of nodes in the network. Such nodes may be submitted to intense or limited functionalities as they can dispose of more or less energy than other nodes. The unique characteristics of individual sensor nodes must be taken into consideration when designing energy efficient routing protocols for WSNs.

To address heterogeneity issues for WSNs, Han has proposed LEACH-HPR [28], an energy efficient CH election protocol that uses a minimum spanning tree algorithm to construct efficient inter cluster routing. The heterogeneous network considered in LEACH-HPR is made up of three types of nodes, A (least energy), B (medium energy) and C (most energy). The network is organized into clusters governed by CHs that gathers data from attending nodes, processes it and then forwards it to the BS if it is in range or to the closest CH otherwise. For electing the CHS each node assigns an internal timer with a value inverse proportional to the energy level it has left. While the timer counts down, each node listens to the transmission medium for any advertisements from neighboring nodes. If no message is received until the timer counts down the node assigns itself as the CH by transmitting a message to all neighbors. The message contains the nodes' ID, the amount of energy left and a header identifier that distinguishes the message as an announcement message.

Each node determines the cluster it belongs to by selecting the cluster head that has the most energy and which is closer to it based on the received signal strength of the announcement message. After a node decides to join a cluster it broadcasts a join-request message to the CH.

Each CH also elects an assistant from the remaining nodes of the cluster which performs data fusing and assigns tasks to other nodes within the cluster. The assistant is chosen to be the cluster node with the most remaining energy. The strategy of using this collaboration of CH and CH assistant greatly improves energy consumption and prolongs the network lifetime [28].

To address the problem of CHs that cannot communicate directly to the BS multi hop routing is used. In order to determine the optimum path from a CH to the BS Prim's algorithm [34] is used, which finds a minimum spanning tree for a connected weighted graph.

Simulation performed by the author have shown that by using an efficient CH election method combined with an improved Prim algorithm LEACH-HPR significantly balances and reduces energy consumption thus prolonging the lifetime of the network.

1.4.3 Transport protocols

Transport protocols play an important role in extending the lifetime of a WSN as they run over the network layer and address issues of flow congestion, orderly transmission, loss recovery, fairness and timing (QoS) [1]. Limited bandwidth can affect the performance of WSNs due to bit errors, congestion and packet loss. These issues have to be addressed by transfer protocols in order to improve the performance metrics of WSNs.

Congestion has a direct impact on energy efficiency as it can cause buffer overflow, queuing delay and increased packet loss. Congestion must be efficiently controlled or mitigated and this can be achieved by:

- *Congestion detection*
- *Congestion notification*
- *Congestion mitigation and avoidance*

Congestion and bit errors can cause packet loss which leads to a decrease in reliability and QoS which further decreases energy efficiency. Other causes of packet loss are node malfunction, incorrect routing or node energy depletion. Packet loss can be efficiently improved by using loss detection and notification or retransmission algorithms.

An example of a reference transfer protocol for WSNs is PSFQ (Pump Slowly Fetch Quickly) [35], a reliable and robust downstream transport protocol which uses NACK based loss detection and notification and local retransmission for applications that require packet 100% reliability.

1.4.4 Time synchronization protocols

Many applications require that a common view of time exist among all or some of the nodes that comprise a WSN. Time synchronization aims to provide a common timescale for local clocks of nodes in a network in order to avoid drifts that may occur due to clock imperfections.

Time synchronization protocols for WSNs address the problem of time drifts with the purpose of resolving several issues that may occur due to faulty synchronization such as:

- *Correlation* when performing a complex task such as data fusion where data collected from different nodes is aggregated to obtain a meaningful result.

- *Power saving* schemes imply the use of coordinated awake/sleep schedules for nodes that inform them when they should transmit or when there may be data available for them to receive. Precise timing coordination offers the possibility of fair transmission medium access and collision avoidance thus resulting in reduced energy consumption.

In order to address these issues a considerable number a time synchronization protocols have been developed most of them based on a TDMA scheme such as, Lightweight Tree-Based Synchronization (LTS) [36], or using biologically inspired algorithms such as DESYNC [37] or FFTS (Fast Fault-tolerant Time Synchronization) [38] that achieves guaranteed fast time synchronization level for all non-faulty nodes in a network.

1.5 Energy harvesting solutions for WSNs

Taking into consideration the numerous applications in which WSNs are used, many of which make human access difficult or unfeasible in terms of costs, any solutions that offer such networks extended working possibilities are of great importance and that is why harvesting ambient energy from the environment can prove very useful.

Energy harvesting is the process through which energy available from external sources such as solar, thermal or wind, to mention just a few, is captured and converted into electrical energy which is stored in electronic devices such as capacitors, super-capacitors or batteries for later use in powering or offering alternative power resources for small autonomous wireless devices.

In the following paragraphs we will provide insight into several harvestable energy sources currently available with focus on RF harvesting.

1.5.1 Photovoltaic energy harvesters

This method is used for converting the energy of solar radiation into electrical power using semiconductors that possess the photovoltaic effect [39]. Electrical power is generated using solar panels composed of solar cells which are made up of photovoltaic material. Several implementations of solar energy harvesting nodes have been developed such as *Prometheus* [40], *Everlast* [41] or *Helimote* [42]. Applications for solar powered WSNs include *ZebraNet* [43] which is a mobile wireless sensor network with sparse network coverage in which nodes are equipped with GPS sensors used to track the migration patterns of zebras. Another example of such solar power wireless network is *SHiMer* [44] used for sensing and actuation purposes in monitoring structural health.

1.5.2 Piezoelectric energy harvesters

Sensor nodes equipped with piezoelectric harvesting capabilities convert mechanical strain into electric power. The source of the strain can come from different sources such as seismic vibrations, acoustic noise and even human motion. An example of

nodes that use piezoelectric energy harvesting are the one proposed by Paradiso and Feldmeier [45] which describe a compact piezoelectric pushbutton which is able to transmit wirelessly in a region of 12 to 30 meters from a single button push without the need for any other source of energy.

1.5.3 Thermoelectric energy harvesters

Thermoelectric converters are based on the Seebeck effect discovered in 1821 by Thomas Johann Seebeck which proved that the presence of a thermal gradient between two distinctive conductors produces a voltage drop. This is due to the fact that a temperature gradient in a conductive material produces heat flow which in turn triggers the diffusion of charge carriers. This effect has led to the development of thermoelectric generators (TEGs) which convert heat into electricity. An example of WSNs that use scavenged thermoelectric energy is the ECT 310 DC/DC converter [46] proposed by EnOcean [47] which combined with the STM 300/312 energy harvesting wireless modules allows the implementation of sensors or actuators which are powered solely by heat.

1.5.4 Radio frequency energy harvesters

Harvesting RF energy is the process by which ambient energy from radio transmitters is captured by a receiving antenna and converted into electrical power. It is practically the same principle as that used in passive RFID devices which are powered when subject to a RF signal of a certain frequency. In order to make use of the RF signal energy a conversion circuit is needed which can extract enough DC power from the received electromagnetic waves to power the micro-sensor device. As most RF energy harvesting systems operate in far field it is crucial for the converter to be able to extract energy at a very low power density of the received signal since the power density of the signal drops off with the square of the distance between the transmitter and receiver according to the Friis transmission equation for free space [48]:

$$P_r = P_t \cdot G_t \cdot G_r \cdot \left(\frac{\lambda}{4\pi R} \right)^2 \quad (1.4)$$

where P_r is the power available at the receiving antenna, P_t is the power at the output of the transmitting antenna, G_r and G_t are the gains of the transmitting/receiving antennas, λ is the wavelength and R is the distance between the two antennas.

In order to get a better understanding of the capabilities of such RF-DC converters we will further describe two examples of such devices insisting on one which we have used for modelling and simulation.

Le et. al. [49] have developed an RF-DC conversion circuit which can operate at a distance of maximum 44 meters with a received signal power as low as 5.5 μ W (-22.6 dBm) from a 4W (EIRP) radiation source. According to the authors the system can provide a voltage of 1V DC and 0.3 μ A at a distance of 15 meters [49].

Other examples of energy harvesting systems are the ones developed by Powercast which are 915 MHz based devices that convert RF energy received from a

transmitter into effective DC power using the P2110 [50] RF-DC converter. The energy harvested can be used to power a sensor node or to recharge a battery.

The Powercast P2110-EVAL-01 Energy Harvesting Kit for Wireless Sensors [51] consists of the following:

- One 3W EIRP 915 MHz transmitter (TX91501) with an 8.1 dBi gain integrated antenna [52].
- Two dipole (omnidirectional, 1.0 dBi gain) and two patch (directional 6.41 dBi gain) antennas.
- Two P2110 energy harvesting boards which convert the RF signal using the P2110 RF integrated circuit and store it in a capacitor/super-capacitor.
- Two wireless sensor boards that can be attached to the P2110 PCBs which provide temperature, humidity and light information through a 2.4 GHz radio module.
- One access point development platform, which communicates with the wireless sensor nodes, and interfaces with a PC for data output.

The key feature of this energy harvesting kit is the P2110 915MHz Powerharvester IC [50], the block diagram of which can be seen in Figure 1.8.

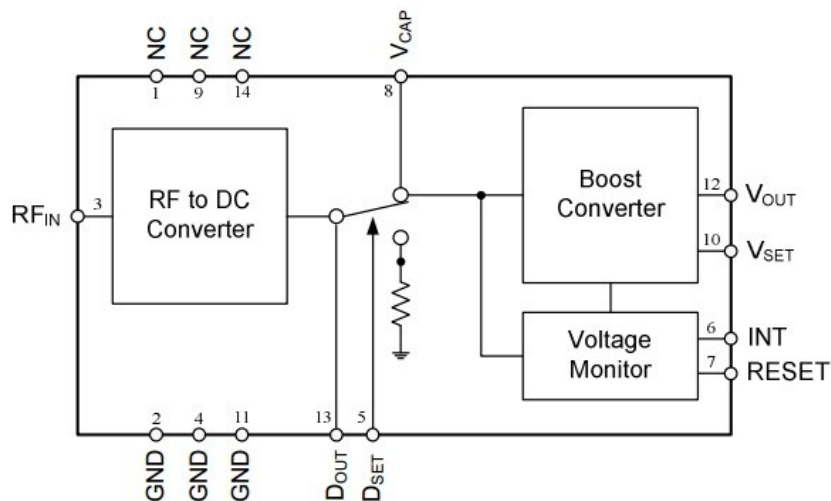


Figure 1.8 P2110 Powerharvester Receiver IC block diagram [5]

When a voltage threshold of 1.2V is reached on the capacitor (V_{CAP} output), enough energy is available to power the wireless sensor node for data acquisition and transmission or to enable the boost converter which can provide a regulated voltage output of 3.3 V which can be boosted up to 5.25 V enabling the circuit to be used for other applications which require higher voltages. According to the specifications [50], the input impedance of the circuit is internally matched to 50 Ω . The RF-DC converter can operate down to -11.5 dBm input signal power and can provide a maximum output current of up to 50 mA [50].

1.6 Conclusions

This chapter is meant as an introduction into the world of wireless sensor networks, but more importantly to provide insight on the numerous challenges and research opportunities that arise from designing and using such a technology.

As we have pointed out throughout this chapter qualities such as robustness, ease of deployment, low production cost and reduced maintenance have made WSNs very attractive for a variety of monitoring and actuation applications.

However promising using WSNs may seem, limited energy capabilities pose severe constraints on such networks and that is why many researchers have dedicated their time and effort to propose energy efficient algorithms and protocols to address issues such as packet throughput, congestion, routing, synchronization and scheduling with the purpose of enhancing QoS and improving energy consumption.

Several of these issues represent the backbone of this thesis and they will be further elaborated in following chapters which will also present advancements obtained in terms of increased packet throughput, efficient clustering, routing and scheduling all with the goal of providing increased quality of service. Our research has also materialized in a paper [53] which presents a classification, comparison and analysis of state of the art protocols and provides solutions for dealing with WSN regarded issues.

As energy consumption is of such high importance, we have also made a brief introduction into energy harvesting solutions currently used for WSNs. We have focused on RF energy harvesting since we have analyzed the previously described Powercast system in order to determine whether prior knowledge about the energy which can be harvested can be ascertained when designing such devices.

Chapter 2

Performance analysis and modelling of a RF energy harvester

2.1 Introduction

As previously mentioned in Chapter 1 there are several methods of energy harvesting popularly used for WSNs such as photovoltaic, piezoelectric, thermoelectric and harvesting power from propagating RF signals which is actually one of the most popular energy harvesting methods for passively powered devices [54].

RF powered devices are used in various applications such as structural health monitoring, telemetry systems and even modern biomedical implants to reduce the chances of infection and replace the need for batteries [55]. All these applications however require the presence of a power conversion circuit that can extract DC power, from electromagnetic waves, which the sensor node can then use or store.

In the remainder of this chapter we will present the performance analysis of a RF harvesting device, the Powercast P2110-EVAL-01 Energy Harvesting Kit for Wireless Sensors [51] previously described in Chapter 1. We will present the design of a current measurement system used for determining the output power capabilities of the RF energy harvester as well as a methodology for modelling and simulating the harvesting capabilities of the P2110 harvester which can be adopted for any similar devices. The experimental setup used is also described as well as the experiments performed in field that have been conducted to validate the proposed model.

2.2 Inductive current measurement system

In order to determine the performance of the Powercast P2110-EVAL-01 system in terms of harvested power, we have devised and built a current measuring system based on an inductive method (which we will from now on refer to as ICMS) which permits us to interpret the current flowing at the output of the energy harvester when connecting loads of any types. The proposed device cannot work as a stand-alone application as it lacks the presence of a data acquisition system which can translate the measured values into effective current. Since the maximum rated current at the output of the system according to the producer [50] is of 50 mA the proposed device must be able to measure currents with a resolution in the μA range. Figure 2.1 a. presents a schematic drawing of the device while Figure 2.1 b. depicts the actual circuit and casing as described in our research [56].

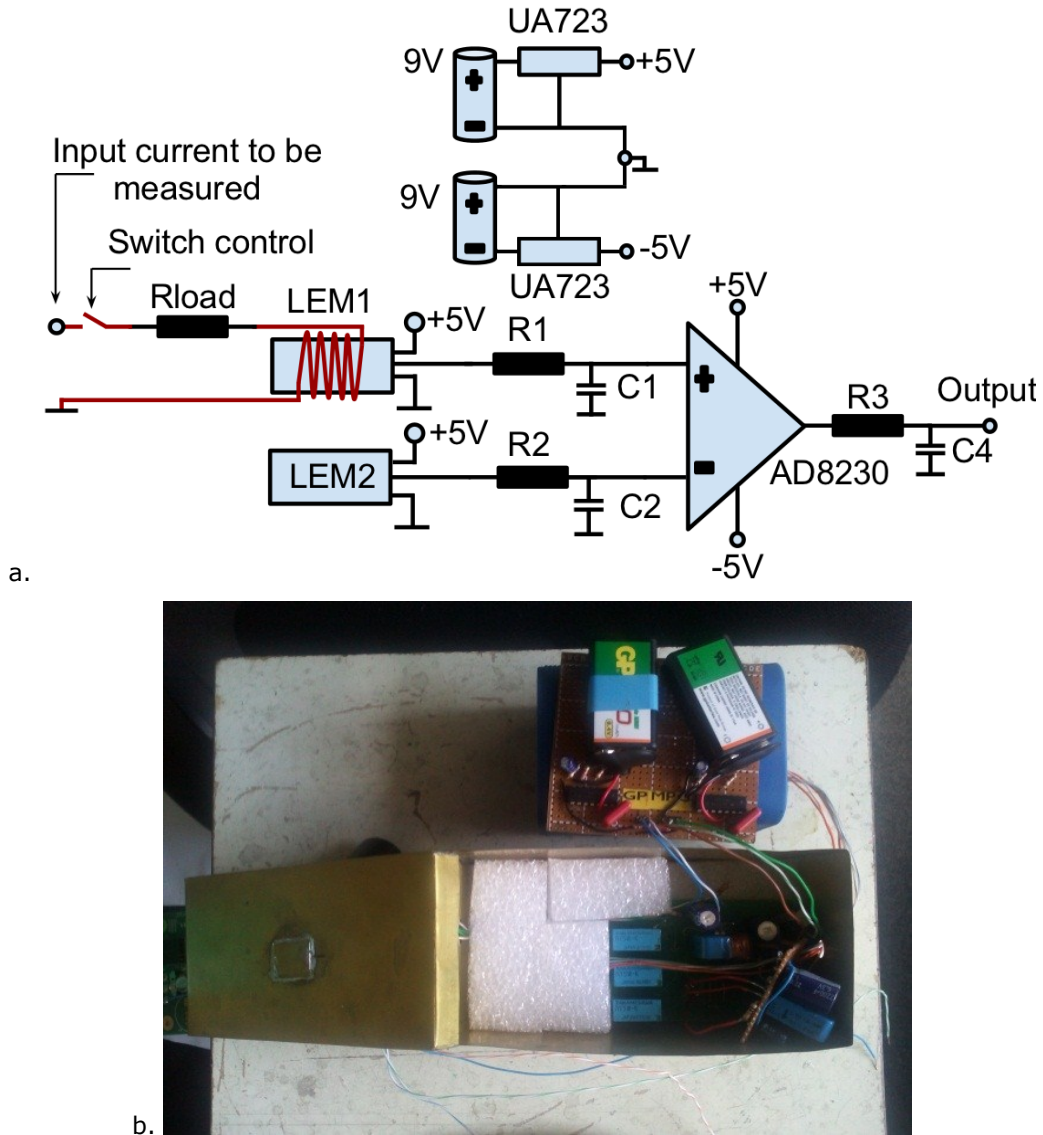


Figure 2.1 a. ICMS simplified block diagram; b. ICMS circuit and casing [56]

As can be observed from the block diagram of the device (Figure 2.1 a.) the main components of the ICMS are the two inductive current transducers of type LTS 15 NP [57] (LEM1 and LEM2). These transducers are designed for high current measurement in the order of Amperes and as stand-alone they cannot be used for the current application requirements. That is why the latter of the two (LEM2) is used as reference while the other (LEM1) is used for measuring. To be able to measure currents down to the μA the effective signal is fed through a 0.5 mm cooper wire through the upper transducer. The wire is swirled around it 110 times

thus amplifying the electromagnetic field and allowing for the precise measurement of low currents down to the desired scale required by the application.

Because no sudden variations of the input signal are expected and also to eliminate possible occurring interferences the outputs of both transducers are low pass filtered using two 1Hz passive filters. The filtered signal is then brought at the inputs of the AD8230 rail-to-rail precision instrumentation amplifier which according to the specifications of the producer [58] has a maximum offset voltage of 10 μV at $\pm 5\text{V}$ operation voltage and low output voltage drift with temperature of 50 $\text{nV}/^\circ\text{C}$. For the same reasons previously mentioned the output of the AD8230 has also been low pass filtered with a 1Hz passive filter before it is fed into a National Instruments USB 6251 DAQ system [59].

To be able to connect loads of various types (in our case resistive loads of 5 and 10 $\text{k}\Omega$ and a capacitive load of 100 μF) a relay which is controlled by NI DAQ has been placed at the 3.3 V output of the P2110 energy harvesting board.

The ICMS is powered through a 5V DC regulated power source obtained from two 9V batteries using two $\mu\text{A}723$ CN [60] voltage regulators (chosen due to their excellent thermal behavior which is important as the whole circuit is encased in a brass casing).

In order to avoid electromagnetic interference from external source which may affect the measurements and provide false readings the entire system has been encapsulated in a brass casing as can be seen in Figure 2.1 b.

2.2.1 Experimental setup used for performance analysis

In order to be able to interpret the power available at the output of the P2110 board and to compensate for fabrication differences between the two transducers an automated calibration program was implemented using the NI LabView 2009 development environment (Appendix A). Initially a 5 $\text{k}\Omega$ resistor was connected as load to test the capabilities of the measurement system which was later replaced with one of 10 $\text{k}\Omega$.

To perform testing and calibration of the system a VI that controls an Agilent N6700B profile modular power system mainframe DC power source equipped with a 35V N6744B DC Power Module was created. With the help of the VI we prescribe voltages ranging from 0 to 32V DC with a step of 100 mV which allows us to measure current values from 0 to 3.2 mA with a step of 10 μA . The VI also controls the DAQ and reads from it the output voltage of the ICMS which together with the prescribed currents have been written to a .csv file and which will later be used as a compensation table. The VI also calculates and stores the slope between two consecutively prescribed points. With these measurements we have been able to perform self-calibration of the ICMS during in field measurements. A schematic representation of the measurement setup is available in Figure 2.2.

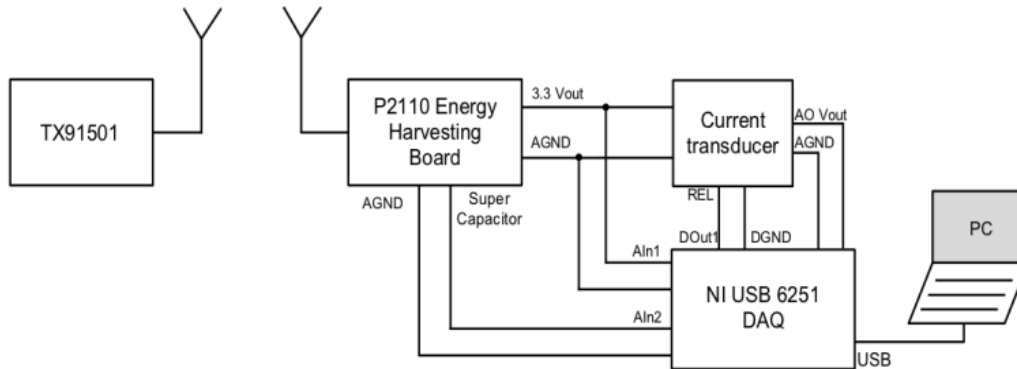


Figure 2.2 Block diagram of the measurement setup [56]

As can be observed from Figure 2.2, the output of the energy harvester is connected both directly to an analog input of the NI DAQ board analog inputs for voltage measurement and also at the input of the ICMS via a resistive load of 5 k Ω can be for current measurement.

As the energy harvested by the RF-DC converter is stored on the printed circuit board in a super capacitor which in turn powers the boost converter when the voltage reaches 1.2 V we have also measured the voltage drop over the super capacitor using the NI DAQ system.

A different VI was developed for the control of the DAQ. This VI also measures all previously mentioned voltage drops which are stored in files together with the acquisition time and performs self-calibration of the ICMS by setting the 0V level (to compensate for manufacturing drifts between the two transducers as well as INA drifts due to the consumption of batteries). Internal filtering of the signals using a 2nd order Butterworth filter with a cutoff frequency of 1Hz is also performed by the VI. After self-calibration the system is ready for use and the current read is interpreted using the previously created compensation table and using interpolation between points.

2.2.2 Validation of the ICMS functionality

The response of the ICMS during calibration when voltages are prescribed from the Agilent N6700B DC power source is available in Figures 2.3 a. while Figure 2.3 b. depicts the resulting slope values.

For visibility reasons Figure 2.3 only shows the system response and slopes for measurements from 0 to 100 μ A.

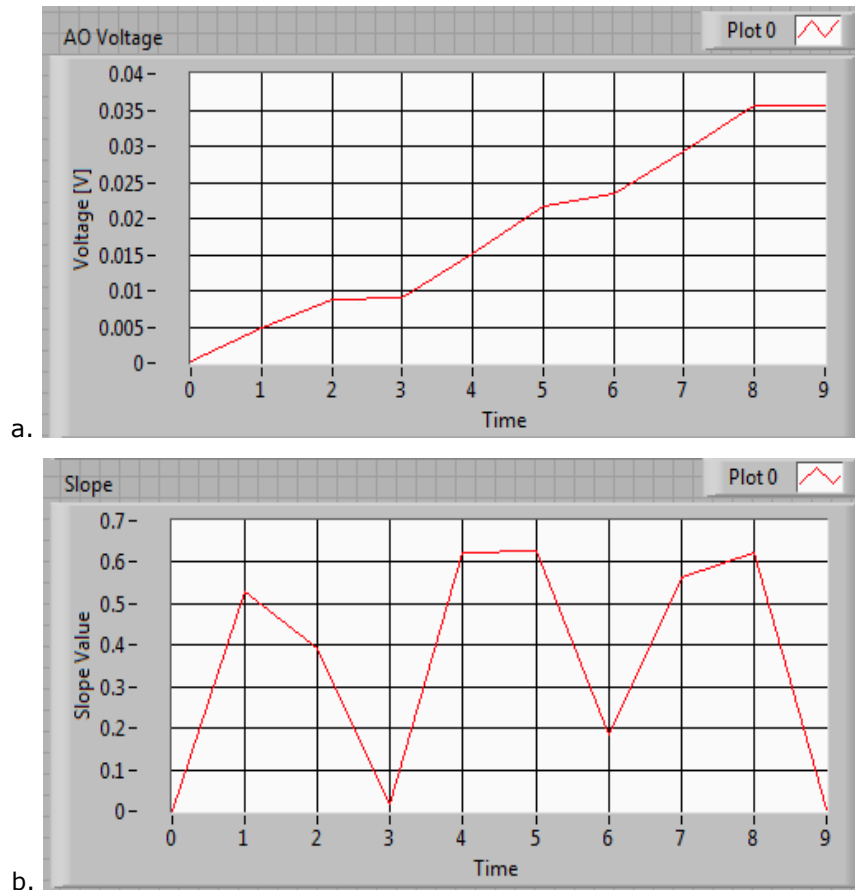


Figure 2.3 INA output voltage; b. Calculated slope values [56]

An obvious characteristic of the system, visible from Figure 2.3 a., is its lack of linearity in response to the prescribed voltages and that is why the compensation table plays important role in converting voltages into currents. Another important feature visible from the same figure is that the system output values measured for $100\ \mu\text{A}$ are below $40\ \text{mV}$ which allowed us to set the DAQ system at its highest resolution of 12 bits/sample (available for measurements below 0.5V) for accurate measurements of low voltages.

For testing purposes we have connected two different loads at the output of the P2110 system, a resistive $5\text{k}\Omega$ load and a capacitive $100\ \mu\text{F}$ load and the results of the measurement are visible in Figure 2.4 a. and b.

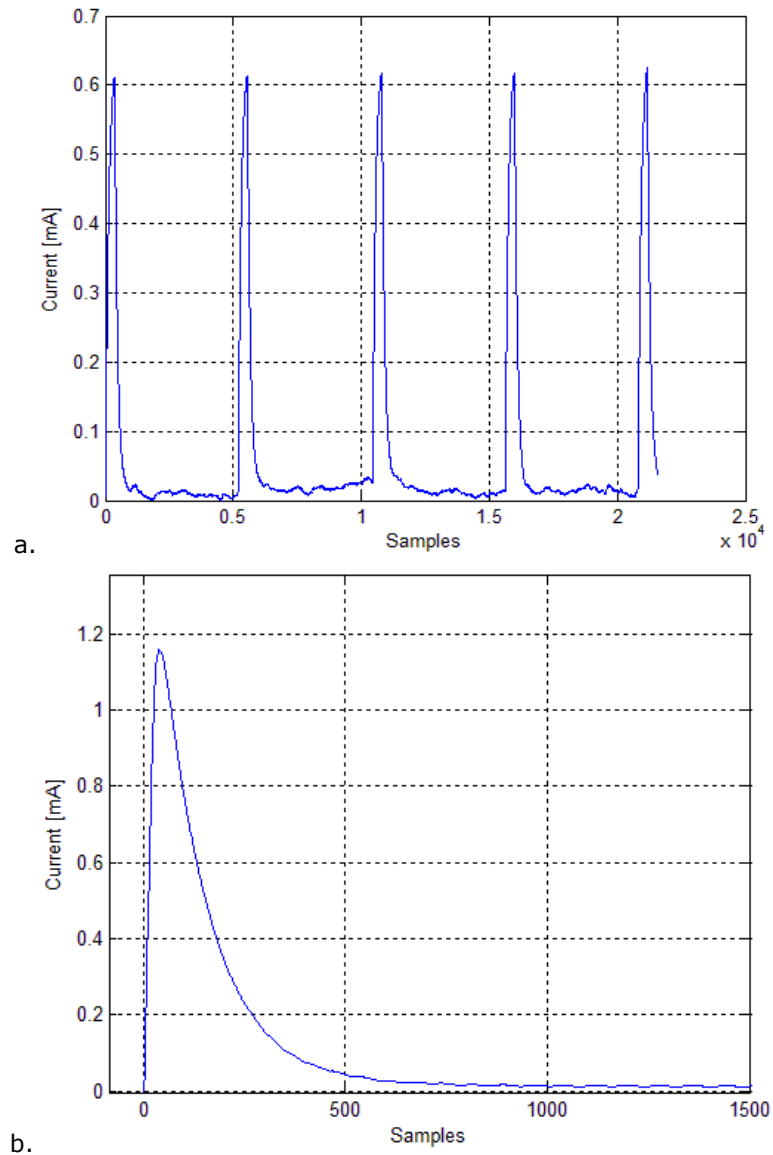


Figure 2.4 a. Current over 5kΩ resistor; b. Current over 100 μF capacitor [56]

2.3 Modelling power transfer

According to literature studies on antennas and electromagnetic fields that surround them [61], three field regions around an antenna can be identified, the reactive near-field, radiating near-field (Fresnel) and the far-field (Fraunhofer) regions. A representation of the fields surrounding an antenna can be seen in Figure 2.5.

38 2. Performance analysis and modelling of a RF energy harvester

The *reactive near-field* region is commonly known to exist at a distance $R < 0.62 \cdot \sqrt{D^2/\lambda}$ where D is the largest dimension of the antenna (in our case the diagonal $D=233.65$ mm according to the producer datasheet [52]) and λ is the wavelength of the radio wave (327.64mm for 915MHz). In this region immediately surrounding the antenna reactive field components predominate. In this region the relationship between the strengths of the electric and magnetic fields are difficult to predict as either of them can dominate at some point in time [61].

The *radiating near-field (Fresnel)* region is bounded between $R \geq 0.62 \cdot \sqrt{D^2/\lambda}$ and $R < 2 \cdot D^2/\lambda$ and in this region the radiation fields predominate and the angular field distribution is dependent on the distance from the antenna. In this region the relationship between the strengths of electric and magnetic fields are still complex to predict [61].

The *far-field (Fraunhofer)* region usually begins from $R > 2 \cdot D^2/\lambda$ and the ratio between the electric and magnetic field strengths becomes constant and the angular field distribution is independent of the distance from the antenna [61].

Since in near-field direct coupling between the transmitting and receiving antennas is present we are interested in determining the covering range of the power harvester in far-field. As the formula which determines the boundary from which the far field region begins $R > 2 \cdot D^2/\lambda$ is applicable when the largest dimension of the radiating antenna ($D=233.65$ mm) is larger than the wavelength ($\lambda=327.64$ mm for 915MHz) we can safely consider that the boundary for the far-field region begins from 3λ according to [62].

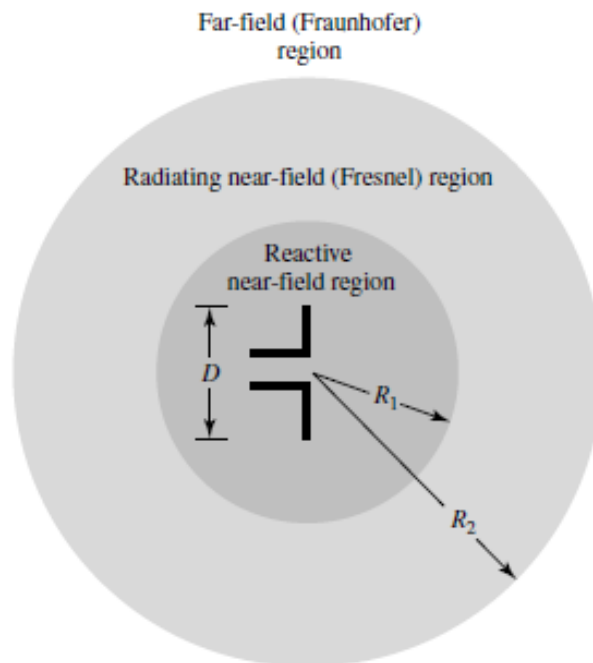


Figure 2.5 Field regions of an antenna [61]

2.3.1 Modelling antenna pattern

According to [63] the gain of an antenna placed in origin of the standard right handed spherical coordinates system (more information is available in Appendix B) can be calculated using the following formula:

$$G(\theta, \varphi) = G_0 \cdot \sin^{2p} \theta \cdot \cos^{2q} \frac{\varphi}{2} \quad (2.1)$$

where θ is the vertical plane angle (or elevation) of the antenna, φ is the angle with the horizontal plane (or azimuth) and G_0 is the adimensional antenna gain:

$$G_0 = \frac{U}{P_{elec}/4\pi} \quad (2.2)$$

where U is the radiation intensity (power radiated per solid angle), and P_{elec} is the electrical power received by the antenna from the transmitter.

Considering that according to the producer the transmitter module has the HPBW of 60° [52] in both planes with vertical polarization, the exponents $p=2.41$ and $q=10$ can be calculated from the HPBW condition [63] (Appendix C).

The expected gain of the transmitting antenna can be calculated using the following formula according to [63, 64]:

$$g_0 = 45.1 - 10 \cdot \log_{10}(\text{HPBW}_{\text{horiz}} \cdot \text{HPBW}_{\text{vert}}) = 9.5 \text{ dBi} \quad (2.3)$$

where HPBWs in horizontal and vertical cuts are in degrees. However, the value for the antenna gain according to the producer is $g_0 = 8.1 \text{ dBi}$ [52], which is smaller due to dielectric (PCB) losses, and which we will use in our calculations. The reception antenna is also a dipole, with a gain of $g_d = 1 \text{ dBi}$ [52] (non-logarithmic value of $G_d = 1.26$). There is however no direct indication of the transmitter power in the specifications of the producer but rather of EIRP ($P_{eir} = 3W$). Thus, the power density pattern (not considering floor reflections) for the transmitter would be [65]:

$$S(r, \theta, \varphi) = \frac{P_{eir}}{4 \cdot \pi \cdot r^2} \cdot \sin^{4.82} \theta \cdot \cos^{20} \frac{\varphi}{2} \quad (2.4)$$

where r is the distance from the transmitting antenna. Considering that both receiving and transmitting antennas are placed at the same height ($\theta = \pi/2$) and oriented towards each other, the expected received power would be, in mW :

$$P_{rx}(r, \theta, \varphi) = \frac{P_{eir}}{4 \cdot \pi \cdot r^2} \cdot \sin^{4.82} \frac{\pi}{2} \cdot \cos^{20} \frac{\varphi}{2} \cdot \frac{\lambda^2}{4 \cdot \pi} \cdot G_d \cong \frac{2.04}{r^2} \cdot \cos^{20} \frac{\varphi}{2} \cdot G_d \quad (2.5)$$

2.3.2 Modelling ground reflections

In order to obtain a high accuracy model of the energy harvesting capabilities of the system under analysis, reflections from the surrounding environment must also be taken into consideration. Since experimental measurements have been made in an enclosed environment as well as in open space, the influence of reflections from the ground as well as from walls and ceiling should be also included in the model. Due to the fact that the most important reflection is the one coming from the ground and because reflections from walls and ceiling have a small influence on the propagation of the signal we have chosen to use only the one ray (direct) and two ray (including

ground reflections) models. As experimental measurements have been performed on marble and asphalt surfaces we have chosen the electric permittivity of marble to be $\epsilon_r = 8.6$ in the GHz range and $\epsilon_r = 10 - j3$ for asphalt [66-68]. Figure 2.6 presents the two ray ground reflection model for transmitter and receiver at the same height from the ground (as in our experimental measurements).

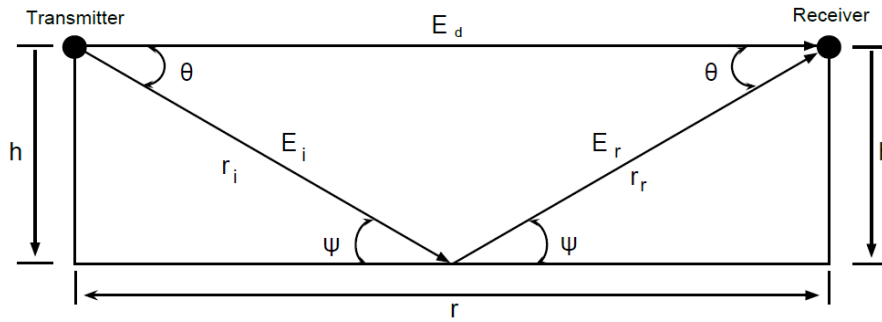


Figure 2.6 Two ray ground model

where, E is the magnitude of the electric field (d – direct ray, i – incidence ray and r – reflected ray), h is the distance from the ground for both transmitter and receiver, θ is the angle between the direct ray vector and the incidence ray vector and ψ is the incidence angle between the second ray and the ground plane ($\psi = \pi/2 - \theta$). When using the two ray ground model we must also take into consideration the vertical polarization reflection coefficient which we can determine using formula 2.6 according to [62]:

$$\rho_V = \frac{\epsilon_r \sin \psi - \sqrt{\epsilon_r - \cos^2 \psi}}{\epsilon_r \sin \psi + \sqrt{\epsilon_r - \cos^2 \psi}} \tag{2.6}$$

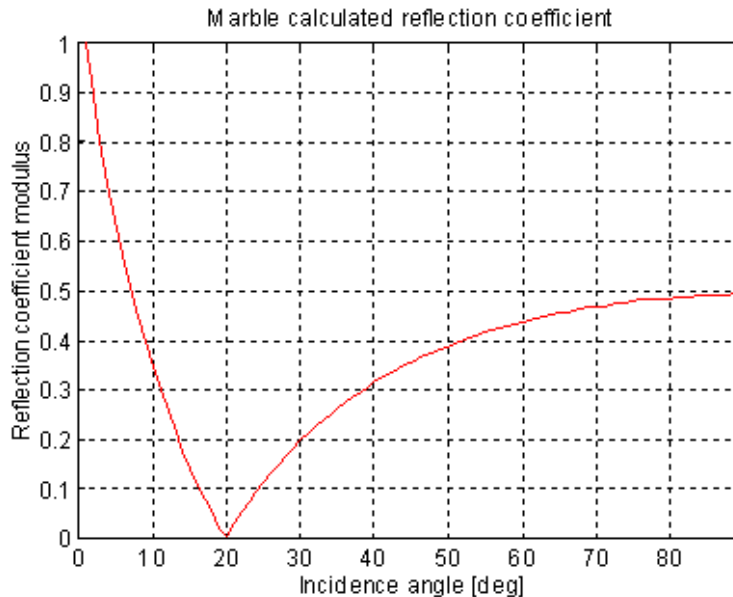


Figure 2.7 Reflection coefficient modulus for marble according to equation 2.6

We can observe from Figure 2.7 that we cannot neglect the reflection coefficient as it can influence the signal with 3dB over 20 degrees incidence, and even more for lower angles. For long distances (differences between the direct and reflected ray lengths is smaller than 10%) the received power can be calculated using the following formula according to [69]:

$$P_r = P_{\text{eir}} \cdot G_d \cdot \left(\frac{\lambda}{4 \cdot n \cdot r} \right)^2 \cdot |1 + \rho_V \cdot \exp(j\Delta\phi)|^2 \quad (2.7)$$

where $\Delta\phi$ in this case is not the azimuth but the phase difference between the direct and reflected rays and G_d is the receiving antenna (dipole) gain.

An important aspect which is visible from Figure 2.6 is that when using the two ray model we should expect a difference in time of arrival and phase between the direct and reflected rays. The difference in phase is very important because it has either constructive (the two signals add for a difference in phase smaller than $\lambda/2$) or destructive (the two signals subtract for a difference in phase greater than $\lambda/2$). The differences in amplitude between the direct and reflected rays are due to the fact that there is a significant difference between the lengths of the two rays (greater than 5% from the length of the direct ray). Since in our case short distances are considered we must also take into account the attenuations of the signal. Therefore the received power can be determined using the following formula:

$$P_r = P_{\text{eir}} \cdot G_d \cdot \left(\frac{\lambda}{4 \cdot n \cdot r} \right)^2 \cdot \left| 1 + \rho_V(\psi) \cdot \cos^{p+1.37+1}(\psi) \cdot \exp\left(j \cdot \frac{2 \cdot n}{\lambda} \cdot r \cdot \frac{1 - \cos\psi}{\cos\psi}\right) \right|^2 \quad (2.8)$$

where: $\psi = \arctg(h/r)$, h is the height at which the receiving antenna is placed and r is the distance between receiving and transmitting antennas (Figure 2.6). The $\cos^{p+1.37+1}(\psi)$ term accounts for the difference in magnitude between the two signals. Exponents $p = 2.41$ and 1.37 (Appendix C) account for the HPBW of the transmitting/receiving antennas while exponent 1 is due to the fact that the ratio of distances between the direct and reflected ray is $\cos(\psi)$ (from Figure 2.6 we can observe that $\cos(\psi) = \frac{r_r}{r}$). The term $r \cdot \frac{1 - \cos(\psi)}{\cos(\psi)}$ represents the path difference between the direct and reflected rays which multiplied by $\frac{2 \cdot n}{\lambda}$ accounts for the phase difference considering the wavelength of the signal.

It is generally known in literature that the dipole (receiving) antenna has a $HPBW_{\text{ver}}$ of 78° [69] and so we can determine its gain using the following formula [71]:

$$G_{\text{dip}}(\theta, \phi) = G_d \cdot \sin^{2 \times 1.37}(\theta) \quad (2.9)$$

Formula (2.8) only considers ground reflection, the strongest in our setup, however for a complete analytic model other four weaker reflections should also be included: from the ceiling, from front wall and from the two side walls.

2.4 Proposed model validation

In order to determine whether the proposed model can be used to efficiently estimate the harvesting capabilities of a RF energy harvester, several in field measurements were needed for comparison. In our case we have performed measurements in two different environments: a hallway (Figure 2.8 a) and a parking lot (Figure 2.8 b) as well as in an enclosed environment (Figure 2.10).

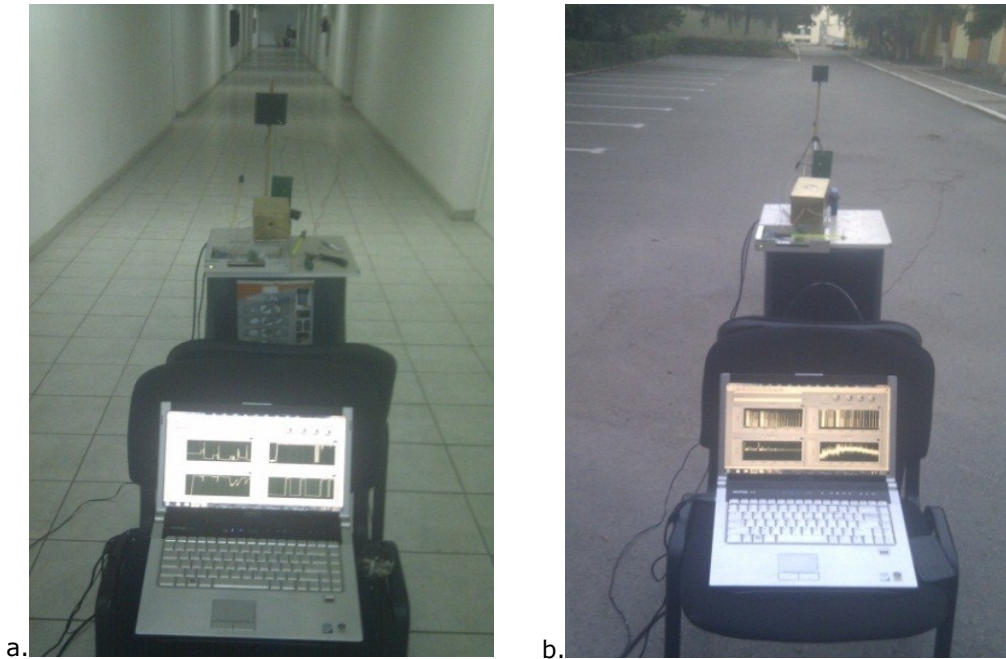


Figure 2.8 a. Hallway measurement setup; b. Parking lot measurement setup

The experimental setup (visible in Figure 2.8) consists of a laptop PC running a NI LabView application, described in section 2.2, which measures the voltage at the output of the P2110 energy harvester as well as the current from the ICMS. We have connected a resistive load of $5\text{ k}\Omega$ at the output of the energy harvester. The value for the resistive load was chosen in such a way because the amount of energy available at the output should be enough to power a WSN node (a microcontroller board equipped with sensors and a 2.4 GHz wireless transmitter).

The transmitter and receiving antennas have been placed at the same height of 1.5m from the ground. The transmitter was initially placed at 2m in front of the receiver and was later moved in a line further from it with a step of 0.5m until the P2110 circuit was no longer able to harvest any energy. This behavior was noticed at a distance of approximately 10 meters for the hallway and 14 meters for the parking lot. The measurement results are available in Figure 2.9.

As expected and in accordance with the Friis transmission equation [48] the average output power decreases with the square of the distance between the transmitter and receiver. An important aspect which can be observed is that, as expected, the environment plays an important role in the energy harvesting capabilities because a difference of up to 8 dBm exists between the amounts of energy the system can harvest in the two different environments. This difference is most likely due to the effects of reflection and absorption from walls, ceiling and floor in the hallway. After a distance of 10 meters however the energy harvester is unable to perform in the hallway while the experiments performed in open space show a behavior closer to

our expectations due to the fact that the environment has little effect on the receiving signal strength and only ground reflections intervene.

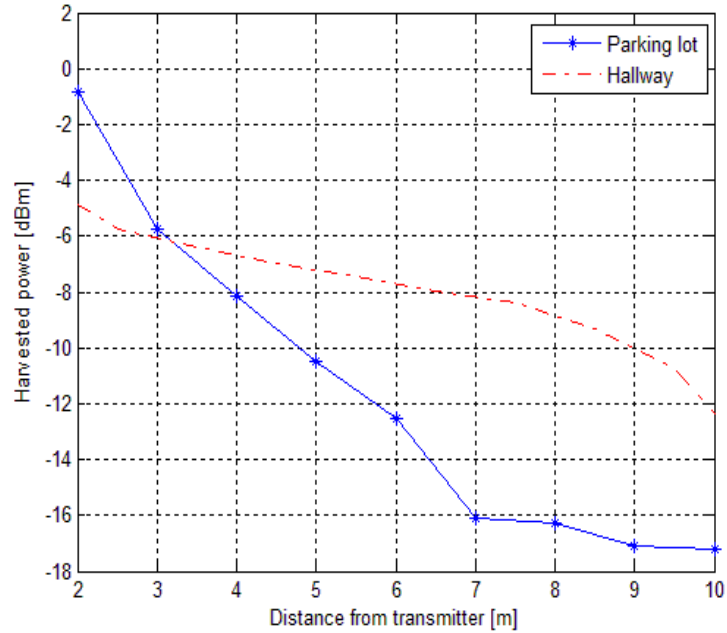


Figure 2.9 In line measurement – average harvested power vs. distance

Since in line measurements reflect only the harvesting capabilities on the main lobe of radiation, an analysis was necessary to determine what the harvested energy would be when the transmitter and receiver were not directly facing each other. That is why we have analyzed the amount of harvested power in an enclosed environment (a room) as can be seen in Figure. 2.10 a and 2.10 b.

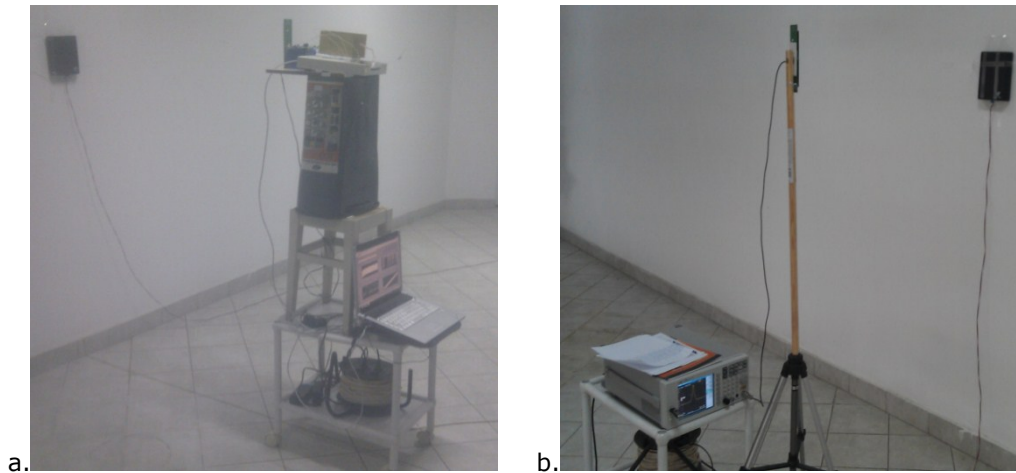


Figure 2.10 a. Harvested power analysis setup; b. Spectral analysis setup

44 2. Performance analysis and modelling of a RF energy harvester

In this case two analyses were performed. One using the ICMS and the experimental setup, described in section 2.2.1, to determine the amount of harvested energy (Figure 2.10 a) and another analysis where the P2110 circuit was removed and the receiving dipole antenna was connected at the input of an Agilent N9320B RF Spectrum Analyzer (Figure 2.10 b) to determine the available energy at the input of the RF-DC converter. In each of the scenarios analysis the transmitter and receivers were placed distance of 1.5 meters from the ground.

In order to create a map of the harvested and received energy, the transmitter was fixed on a wall while the experimental setup (respectively the spectrum analyzer) was placed on a mobile table. The tiles on the floor were used as distance markers and measurements were performed by moving the table in a matrix at each intersection of the tile beginning from 1 meter from the transmitter. The distance between two intersecting tiles was of 42.5 cm. As a result we have obtained a map with the available power at each point in the matrix. The maximum front distance we have measured was of 4.85 m and the maximum lateral distance was of 3.4 m.

Measurement results were stored in ".csv" files and later used to create a power map available in Figure 2.11, which compares the amounts of harvested and received energy levels.

We have also simulated the proposed model from section 2.3 using the Matlab simulation environment in order to determine the amount of expected power which can be obtained using both the one and two ray propagation models. The results are available in Figure 2.12 where they are compared with the ones obtained through in field analysis.

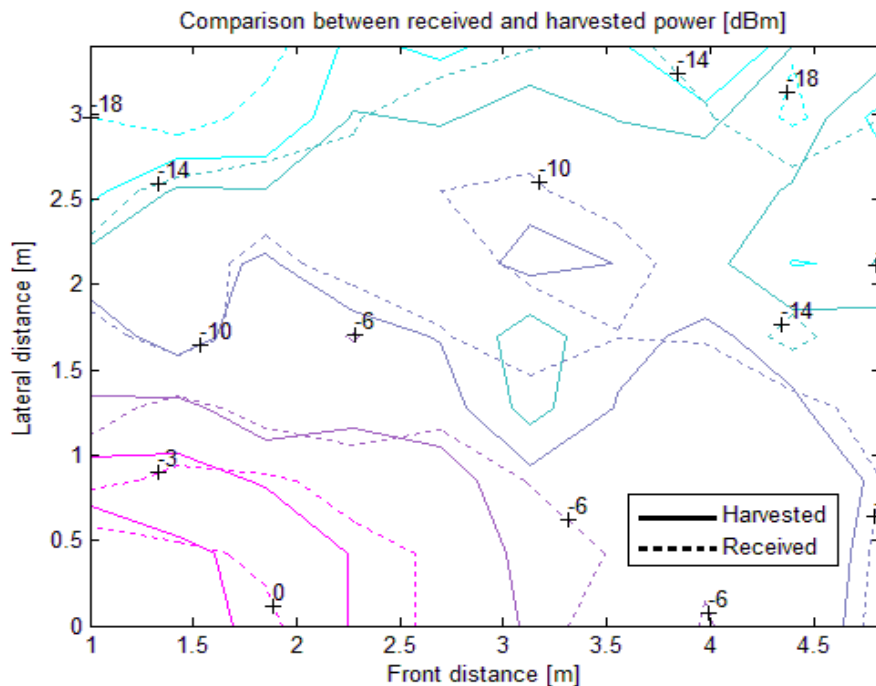


Figure 2.11 Harvested vs. received power levels

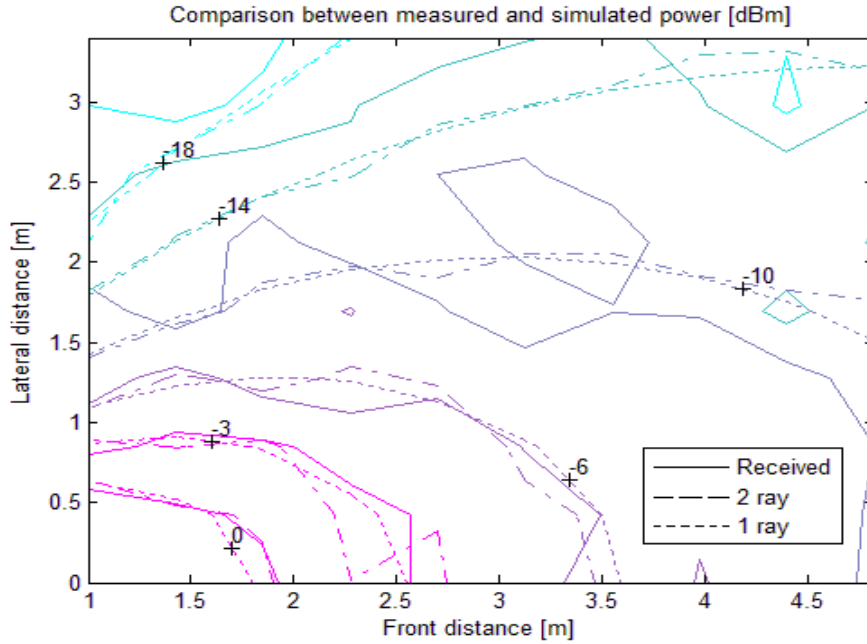


Figure 2.12 Received vs. 2 ray vs. 1 ray power levels

One can observe from the energy map (Figure 2.12) that there are little differences between the two simulation models (one and two rays). However the two ray model is the closest to reality and in terms of error we are situated in the 5 dBm maximum error range mainly at large distances and a 2.1 dBm standard deviation.

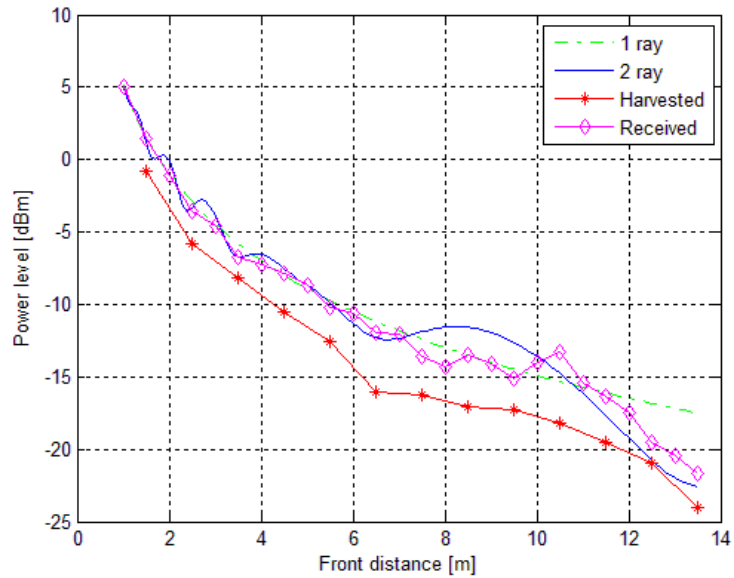


Figure 2.13 Indoor harvested vs. received vs. modelled power, in line on maximum radiation direction

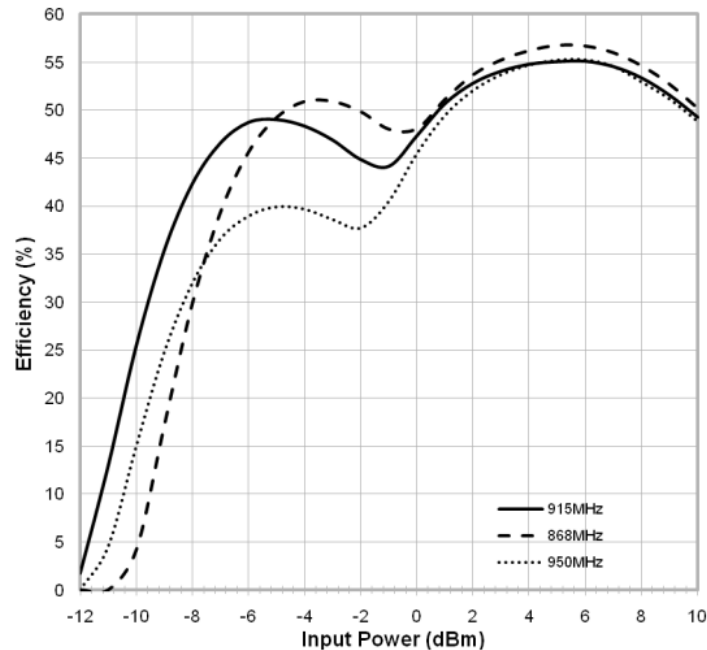


Figure 2.14 RF-DC converter efficiency vs. RF_{IN} (dBm) [50]

Figure 2.13 depicts the power levels for harvested and received energy and as expected there is a difference between the two because most likely due to the efficiency of the power harvester converter which can go as low as 1% or less for input levels below -12dBm according to the producer [50] (Figure 2.12 b). Since there is little insight provided by the manufacturer about the input circuit, it is difficult to obtain an accurate prediction for the harvested power at low levels. We can see from Figure 2.14 that the harvested power is lower than the received power due to the efficiency of the RF-DC and Boost converters circuit. During our experiments we have noticed that the measured efficiency of the circuit far exceeds the manufacturers' specifications because at -20 dBm input power the efficiency of the circuit is still over 20% and not below 1% as specified in the datasheet. This behavior is most likely due to other unexpected RF fields existent in the environment in which the analysis was performed.

2.5 Conclusions

In this chapter we have described a method for analyzing the performance of RF energy harvesters. We have based our analysis on the Powercast P2110 harvester. We have provided a detailed description of how the analysis was performed in terms of hardware devices and mathematical background.

In order to properly analyze the proposed device we have developed a measurement setup consisting of an inductive current measurement system which is capable of measuring currents down to the μA range and National Instruments USB

6251 data acquisition system which was used together with an interface developed in the NI LabView environment used for logging and analysis of the acquired parameters.

Using wave and antenna theory we have proposed a mathematical model which can be used to simulate the performances of the RF harvester. The model is based on the one (direct) and two (with ground reflections) ray models. Using beam-shaped antenna patterns we have managed to predict the received power level within 2dBm dispersion in the 5 meters range of the radiation beam. Experiments performed in two different environments (indoor and outdoor) have shown differences in terms of harvested power for the two environments which come from variations of the received signal due to reflections. We have also noticed that below -10dBm input power level it becomes very difficult to predict harvested power as the operating limit (given in the datasheet of the device) is -11.5dBm. From this we have been able to conclude that the system under evaluation has 5 to 6 meters range in the main lobe of radiation.

2.6 Contributions

- 1. Design and development of an experimental setup** for analyzing the performance of radio frequency energy harvesters
- 2. Practical development of a low current measurement system**
 - Circuit design proposal and practical implementation
 - Functionality validation through in field testing
 - NI LabView programming of calibration and measurement software
- 3. NI LabView based programs for control, data acquisition, interpretation and logging** for the experimental setup
- 4. Mathematical model using the one and two ray reflection models for simulating** the harvesting capabilities of RF energy harvesters
- 5. Simulation and analysis of the proposed mathematical model in the Matlab environment**
- 6. In field performance analysis of the Powercast P2110-EVAL-01 RF energy harvester**
- 7. Validation of the proposed model through comparison of experimental results with the proposed mathematical simulated model**

Chapter 3

A graphical user interface for simulating WSNs

3.1 Introduction

In this chapter we describe a graphical user interface developed with the purpose of providing a stable and easy to use simulation environment for the work which will be presented Chapters 4 and 5. The work described in this chapter also represents the foundation for a network simulator that can be used to easily model the behavior of WSNs.

There are currently several network simulators available for download over the internet which can be used to model the behavior of WSNs, each with their own advantages and disadvantages. We will briefly enumerate several of the most frequently used simulators and provide a brief description in accordance with survey studies available in literature [73, 74]:

- **NS-2** [75] – is one of the first and most used simulators as it was developed in 1989 targeting LAN networks but has since been extended for wireless support. It is based on the C++ programming language and due to its considerable existence it provides numerous protocols publicly available.
- **TOSSIM** [76] – is a discrete event simulator developed at UC Berkeley with specific target to TinyOS applications such as the MICA Mote. It is based on the nesC language (a dialect of C) and provides high accuracy of the running application
- **GloMoSim** [77] – provides a scalable environment for simulating both large wired and wireless networks it is based on a C language programming library called PARSEC (Parallel Simulation Environment for Complex Systems). It is however restricted to existing protocols for IP based networks.
- **Castalia** [78] – Is a C++ based simulator which provides parametric definition of the network and can be used to simulate wide application specific network platforms.
- **(J)Prowler** [79] – are event driven WSN simulators which provide both deterministic and probabilistic simulation modes. They are developed in the (Java) Matlab environments but like TOSSIM they are based on the protocols provided by TinyOS.

All these simulators also dispose of a visualization tool which can be used to define network characteristics. Our purpose is also to provide visual characteristics to the proposed methods addressing WSNs as an alternative simulation environment which can be easily adapted and modified to suite user requirements.

3.2 Description of the proposed visualization tool for WSNs

Since most of our simulations have been performed in the Matlab environment, the visualization tool has also been developed in the same environment. A graphical representation of the main window of the application is available in Figure 3.1.

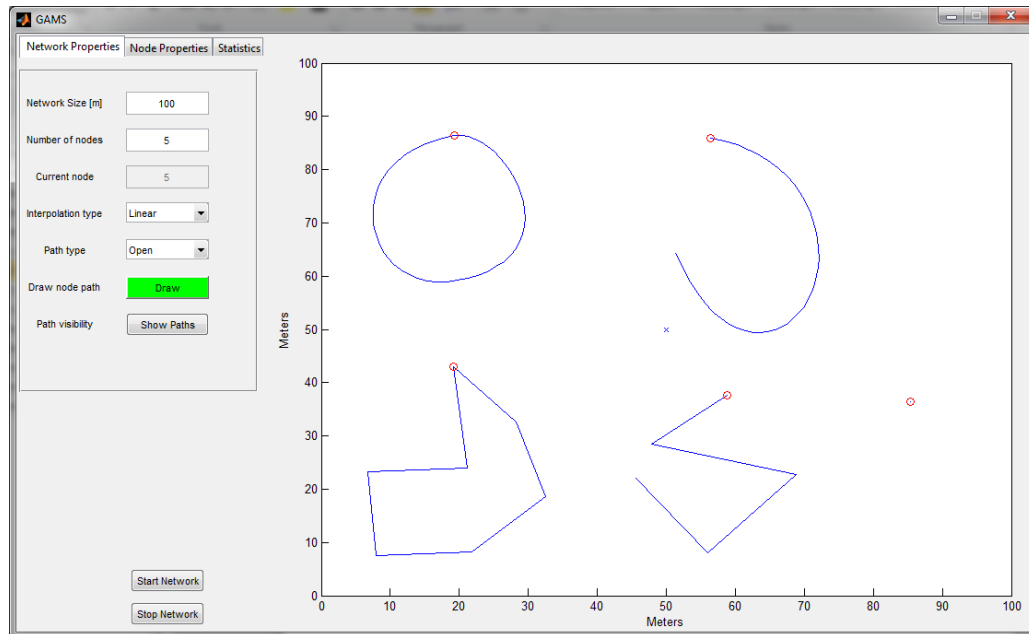


Figure 3.1 Proposed visualization tool for modeling WSN behavior

The primary features of the main window can be divided in two categories:

- Network and simulation dependent parameters (available on the left side of visualization tool).
- A graphical window that depicts the motion of sensor nodes but which can also be used to illustrate various statistical measurements of the simulation scenario.

Before we begin explaining the functionalities of the above categories we will explain the usage of the two buttons that are available on the lower left side of the tool:

- **Start Button** – By pressing this button after the network and node properties have been defined the network will begin to function dependent on the characteristics of the simulated network (if only static nodes are used then the protocols described in Chapter 4 are used; if only nodes with mobile characteristics are defined then the work presented in Chapter 5 will be used).
- **Stop Button** – Pressing this button will simply close the visualization tool.

3.2.1 Network and simulation dependent parameters

The main features of the graphical user interface that we have developed are represented by the three tabs available on the left side of the simulator which allow the user to define various network and node parameters. The functionality of the three tabs is as follows.

The **Network Properties** (visible in detail in Figure 3.2) tab allows the user to set various parameters such as:

- **Network size** which is expressed in meters and represents the size of the observation area. It must be noted that for the developed simulator the OA is of square shape (initially set to 100x100 meters).
- The **Number of nodes** in the network can be easily modified by writing the desired value in the corresponding field (initially set to 1).
- After these two parameters have been modified, the user can then begin to place nodes in the OA by pressing the **Draw** button.
- If the user wishes to simulate a network with mobile nodes he will have to define various points of a node path. The simulator then updates the path correspondingly in accordance with the **Interpolation type** chosen by the user which can be either Linear or Spline.
- Another parameter available for mobile nodes is that of **Path type**. Here the user can chose whether the path take by the node is closed or open. Examples of paths open and closed with linear or spline interpolation are available in Figure 3.1. The upper two paths have been defined using spline interpolation while the lower using linear. Also the paths on the left side have been chosen as closed while the ones on the right as closed.
- The **Show Paths** button allows the user to either hide or show the paths that each sensor node moves on once the simulation has started.

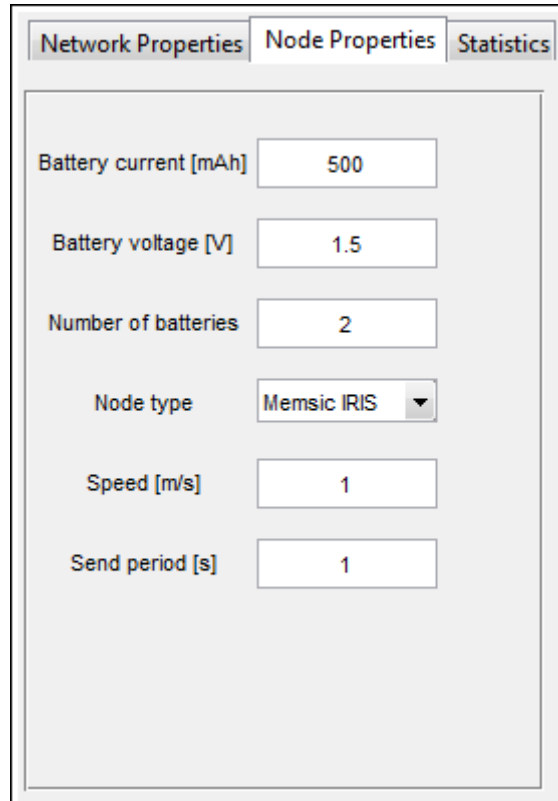
Network Properties	Node Properties	Statistics
Network Size [m]	100	
Number of nodes	5	
Current node	5	
Interpolation type	Linear	
Path type	Open	
Draw node path	Draw	
Path visibility	Show Paths	

Figure 3.2 Network properties tab

After having defined the observation area characteristics the user can define individual node parameters available in the **Node Properties** tab available in Figure 3.3. It must be noted that a modifications of the **Interpolation type** and of the **Path type** as well as of any parameters available in the **Node Properties** tab will affect the node in cause which is currently being drawn.

The parameters available in the **Node Properties** tab are as follows:

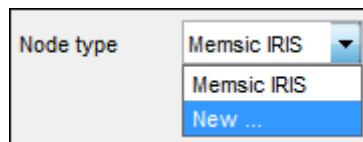
- The current available in the battery that powers the sensor node.
- The voltage of each battery.
- The number of batteries used by the node.
- The speed at which the node travels in case mobility is desired.
- The period at which the node must send or act upon the environment.
- The hardware characteristics of the node.



Property	Value
Battery current [mAh]	500
Battery voltage [V]	1.5
Number of batteries	2
Node type	Memsic IRIS
Speed [m/s]	1
Send period [s]	1

Figure 3.3 Node properties tab

When considering hardware consumption, the user can select to use either the Memsic node energy consumption model (available in Table 5-3) or can select to define their own model by clicking on the “New ...” option provided by the **Node type** field as can be observed from Figure 3.4.



Node type
Memsic IRIS
New ...

Figure 3.4 New node type

By clicking on the “**New ...**” option a popup will appear like the one visible in Figure 3.5 in which the user can define the hardware characteristics of the new node type to be used in the network.

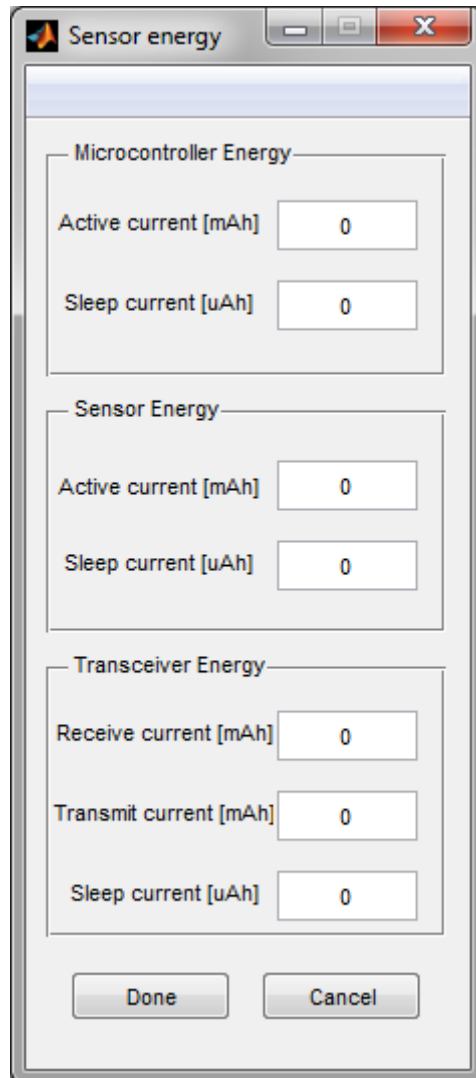


Figure 3.5 Node hardware consumption pop-up

It must be noted that by defining a new node type all nodes from that point on will use those characteristics unless a new model is defined or the IRIS node characteristics are selected again. This feature allows the simulation to perform for WSNs that have nodes with heterogeneous characteristics.

The **Statistics** tab is designed strictly for analyzing the performance of the WSN under simulation as it provides access to various statistic information regarding the overall behavior of the network and that of individual nodes. A detailed representation of the **Statistics** tab is available in Figure 3.6.

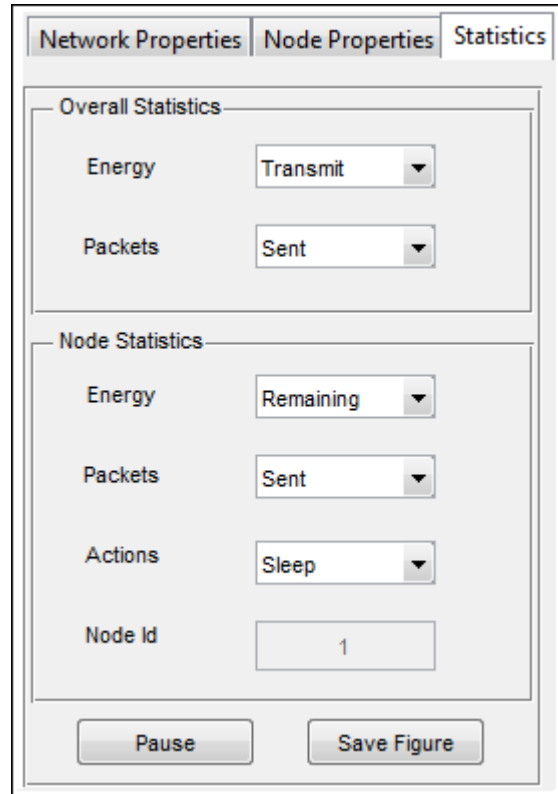


Figure 3.6 Statistics tab

As previously mentioned the **Statistics** tab is comprised of two fields:

- **Overall Statistics** – provides information regarding the overall behavior of the simulation scenario in terms of:
 - **Energy** used for:
 - Transmitting packets
 - Receiving packets
 - Forwarding packets
 - **Packets** throughout the network:
 - Sent
 - Failed to receive
 - Sent directly to BS
- **Node Statistics** – provides information regarding any of the nodes present in the simulation in terms of:
 - **Energy**:
 - Remaining
 - Used to send packets
 - Used to receive packets

- **Packets:**
 - Sent
 - From others
 - Sent directly to the BS
 - Failed to receive
- **Actions** performed (as specified in Chapter 4.3):
 - Sleep
 - Route
 - Acquire
 - Acquire and route
 - All actions

By inputting a number in the **Node Id** field the user can select which node he wishes the simulator to show statistics about.

All statistical information that the user wishes to see will be visible in the graphical window present on the right side of the simulator.

There are two buttons also available:

- **Pause** – allows the user to pause the simulation in order to perform analysis of the statistical information.
- **Save Figure** – allows the user to save the image from the graphical window in a “.bmp” format. Upon pressing this button the simulation environment will also generate a “.csv” file of the statistical data being plotted with the following file name “**StatisticsName_YY_MM_DD-MM-SS.csv**”. Where **StatisticsName** represents the name of the statistical data being saved (e.g. Node2EnergyRemainig) and **YY_MM_DD-MM-SS** are indicators of the date and time at which the file was saved (e.g. 13_01_14-12-00).

3.2.2 Functionality of the graphical window

The graphical window visible on the right side of the simulation tool has several purposes. The first and most important purpose is that it allows the user to define the network topology. After having modified the network parameters as desired (from the **Network Properties** and **Node Properties** tabs) the user can then define the position of each node within the observation area by clicking on the graphical window. For example, if the user wants to define a static node he will do this simply by left clicking on the graphical window at the desired position and then right clicking to release the node. If the user wants to draw a mobile node he will do this by left clicking on the window at different positions throughout the observation area. When he has finished a right click will release the node and the simulator will update the node path defined by the points which the user has inputted on the window based on the desired characteristics of the path (chosen **Interpolation type** between points and **Path type**: closed or open). Examples of the five types of nodes accepted by the simulator are visible in the graphical window shown in Figure 3.1.

The second important functionality provided by the graphical window is that of showing statistical information as selected by the user according to the description of the **Statistics** tab previously described. An example of how the graphical window looks like when selecting to view the statistical information about all the actions executed by node 1 at a random period in time is available in Figure 3.7.

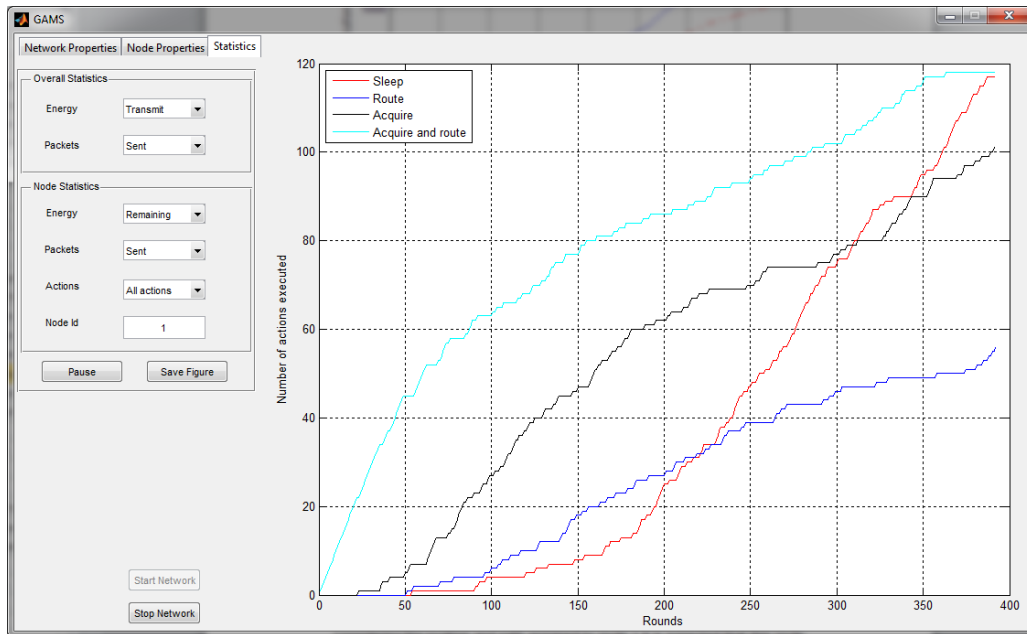


Figure 3.7 Statistics about all the actions executed by node 1

3.3 Conclusions

In this chapter we have presented a visualization tool which can be used to model the behavior of WSNs and we have described the individual characteristics of the features available by using the tool. The visualization tool gives the user freedom to define specific characteristics of the simulated network such as size, number of nodes as well as node hardware characteristics. The ability to configure individual parameters of the network and of sensor nodes is a useful tool which allows the simulation of both homogeneous and heterogeneous WSNs. The aspect of giving the user the possibility to define specific hardware characteristics of nodes is of great importance as it allows for a more precise model of the network behavior.

Another quality of the developed simulation environment is that it supports both static and mobile nodes. The topology of the nodes within the observation area must however be defined by the user and we have yet to implement a load mechanism which would allow for simulation of a previously defined scenario rather than restricting the user to define its own.

The visualization tool also provides another distinctive feature in the sense is that it allows the user to monitor statistic information about the evolution of the simulated

scenario regarding both overall and individual node statistics. The simulation can also be paused for a more thorough analysis of statistical data which can also be saved in files of .bmp format for later use.

The main purpose of this application is that of providing the user easy access to the work described in Chapters 4 and 5 but also to provide a modular simulation environment which can be easily adapted to any protocol defined by the user. We intend to make the work available online for free access to all those interested in the field of WSNs.

3.4 Contributions

- **Development of a graphical user interface for simulating the behavior of WSNs** which provides several distinctive features:
 - Allows the simulation of **both static and mobile** sensor networks
 - Makes use of an **existing node energy consumption model** which can be easily changed to suite application needs
 - **High degree of heterogeneity** considering individual node characteristics (both hardware and mobility wise)
 - Uses a **real path loss model** obtained through environmental measurements
 - **Statistical analysis** of various network parameters
- Can be easily **modified** to suite the **development of new protocols**

Chapter 4

Efficient clustering techniques for static WSNs

4.1 Introduction

As previously stated in Chapter 1, one of the methods used to obtain efficient energy consumption at both the node and network level is through clustering. By organizing the network into clusters of nodes and appointing local cluster heads that govern communication, impose local scheduling and perform data aggregation, correlation and fusion the overall QoS of the network can be significantly increased.

The basic idea behind clustering is to appoint certain nodes within the network specific tasks in order to distribute the overall workload of the network with the purpose of extending the lifetime of nodes.

One of the pioneer protocols developed for WSNs that addresses clustering is the Low Energy Adaptive Clustering Hierarchy protocol proposed by Heinzelman et al [26] and described in detail in Chapter 1. Briefly, LEACH is a round (period of time defined a priori) based protocol in which all nodes can play the role of CH at some point in time by means of comparison between a randomly generated number and a threshold value calculated based on a suggested percentage of CHs. Once a node has been appointed the task of CH it cannot be reelected for a pre-defined period of time for this task.

A major advantage of such a clustering strategy is that the overall workload of the network is uniformly distributed among all network nodes which translate in extended lifetime capabilities.

There are however several issues when considering this technique, such as:

- The parameters used by the authors of LEACH in their simulations are limited to transceiver electronics (Table 4-1) but do not consider other sources of energy consumption such as the microcontroller, data acquisition, logging etc. which means that they do not reflect the reality of a WSN node. All nodes were considered to have 0.5 J of available initial energy.

Table 4-1 LEACH [26] Radio characteristic parameters

Operation	Energy dissipated
Transmitter electronics	50 nJ/bit
Receiver electronics	50 nJ/bit
Transmit amplifier	100 pJ/bit/m ²

- Randomly electing CHs as indicated in LEACH will lead to areas overcrowded with CHs while leaving other areas unattended. A solution to this downside was later proposed by Hansen et al. [80] (MSD) and improved by Chalak et al. [81] (IMSD). Both solutions propose that a minimum separation distance (MSD) should exist between the CHs. Their simulations were based on a centralized variant of LEACH (LEACH-C [82]) which assumes that the BS governs the election of CHs and can communicate to all nodes within the network. Also their simulation parameters were restricted to a 100x100 meter network with 20/50/100/200/350/500 number of nodes. Through their simulations the authors claim to have obtained an 80% increase in the number of packets that reach the BS.
- Attending CHs are elected by nodes using the highest RSSI (closeness) criterion which is known to be a bad indicator of link quality [83].

In this chapter we propose and describe a new method for electing CHs which considers among other issues a minimum distance between CHs. We also propose a method for non-cluster head nodes to elect attending CHs. Both methods have been validated through simulations in the Matlab environment and the results are also presented. Since comparison was made with the LEACH based IMSD protocol the same energy model as used by the authors of LEACH was used.

4.2 A new method for cluster head election

Before we begin to explain the proposed CH election method we present a brief introduction into what a centralized control approach for WSNs implies.

In a centralized approach, such as in LEACH-C, cluster head election is performed by the BS which informs all nodes about the elected CHs for the current round. In return all nodes inform the BS about their position and energy status each round, information which the BS uses to determine nodes eligible for becoming CHs. Only nodes with energy level above the network average are eligible for being elected as CHs for that round.

We consider that using a centralized approach can have several disadvantages, such as:

- All nodes must be within communication range with the BS
- If not all nodes can directly communicate with the BS their energy and position information must be forwarded through CHs which induces further strain on these nodes
- Sending position and energy information each round increases communication overhead and latency
- If nodes are added to the network outside the communication range with the BS, scalability becomes a problem

Having in view all these issues we consider that a local approach may be best suited because the energy consuming task of sending messages at the beginning of each round can be replaced with local computations which reduce overhead and latency.

Also a local approach allow for scalability in situations where the network is spread on larger areas that extend over the communication range of nodes with the BS.

As opposed to LEACH-C, the MSD and IMSD enhancements randomly elect CHs considering that there is a minimum (a priori known) distance between them. If the desired number of CHs cannot be obtained, random nodes are chosen from the remaining eligible ones (MSD), or the improved version where the minimum separation distance is reduced and the algorithm is run all over again until the desired number of CHs is obtained (IMSD). After election the BS informs the nodes about their new status and clusters are formed. Until the next round when the process is repeated the network performs the same as LEACH.

We have performed simulations in the Matlab environment of the two enhancements to LEACH-C, MSD and IMSD according to the authors' specifications and we have noticed the following:

- The desired number of CHs is constantly maintained
- In IMSD the CHs are more evenly spread throughout the OA
- Due to the selection algorithm of the CHs all network nodes have equal possibilities of becoming CHs
- The vast majority of nodes deplete their energy at about the same time

Following our simulations of the LEACH, LEACH-C, MSD and IMSD protocols we have come to the conclusion that simply rotating the task of CHs between sensor nodes may not necessarily be the best approach and that CHs should also be elected on other criterions such as the number of neighboring nodes in the area defined by the separation distance (or communication range) and also considering the expected payload of the packet to be sent with respect to the maximum allowed.

4.2.1 Proposed algorithm

In order to determine what the impact would be in considering other criterions in the election of CHs we have developed an extension to LEACH which we have entitled Adaptive Separation Distance and Load Distribution (LEACH-ASDLD) [84] which is further described. The proposed algorithm, being based on LEACH, is also round based and its communication is structured on 3 layers:

- **Layer 1** represents a **neighborhood reconnaissance** procedure
- **Layer 2** is reserved for **sensing and data gathering**
- **Layer 3** is for to inter **cluster head and CH to BS communication**

For our simulations of the proposed algorithm we have made the following assumptions about the network model [84]:

- The network consists of 100 nodes randomly deployed.
- The size of the OA was varied from 50x50m to 200x200m with a step of 25x25m. We assume that some knowledge of the observation area size is previously known (same concept as in MSD/IMSD).
- All nodes have the same battery power as well as the same hardware and software architecture and their communication distance is of 100 meters.

- Energy consumption constraints are those of LEACH (Table 4.1).
- Environmental noise and communication errors are not considered.
- Network nodes are synchronized (e.g. using an RT Clock).

Based on these assumptions we will next provide a more detailed description about the proposed algorithm which can be divided into 4 stages [84]:

Neighborhood reconnaissance – during this stage nodes inform their neighbors about their position (if a location device exists) or simply send a dummy message and neighbors can calculate the distance to the sender node using the RSSI indicator. Communication is performed in a TDMA fashion and only during the first round (briefly upon network deployment) after which this time window is reserved for advertisements (performed by newly added nodes which inform about their presence or by nodes which have depleted their energy which inform about their future absence from the network).

Cluster set-up phase – since the size of the observation area is generally known the separation distance will actually define the number of clusters that will be formed each round. For a desired number of N clusters the separation distance can be calculated using the following formula [84]:

$$SD = \sqrt{\frac{L^2}{N}} \quad (3.1)$$

where the OA is considered a square of size $L \times L$, and SD is the separation distance resulting from fitting N squares into the OA.

Since each node also knows its sensor hardware and what it is measuring it will of course know the size of the packet to be sent and can calculate the ratio between the expected packet size and the maximum allowed by the IEEE Standard 802.15.4 [12] which is of 127 bytes. With this information and the one obtained from stages 1 and 2 all nodes can calculate a threshold value using the following formula [84]:

$$Th = \left(\frac{E_r}{E_0}\right) \left(1 - \frac{N_{SD}}{N_R}\right) \left(1 - \frac{P_{crt}}{P_{max}}\right) \quad (3.2)$$

where E_r is the remaining energy, E_0 is the initial energy, N_{SD} represents the number of nodes in the separation distance, N_R the number of nodes in the communication range, P_{crt} is the current payload and P_{max} is the maximum payload.

In order to determine the CHs, each round all nodes will calculate the threshold value using formula 3.2 and set a timer according to the value. The node with the smallest timer value will advertise itself as CH, as long as there are no other advertised CHs closer than the separation distance. This procedure is performed until the entire OA is covered and there are no eligible nodes left.

By electing CHs in this manner the opposite behavior of LEACH will happen in the sense that even though all nodes are eligible for becoming CHs, nodes with a larger number of neighbors in the SD have the highest probability. This actually means that, unlike LEACH which uniformly distributes the task of CH among all nodes to evenly distribute energy consumption, in our case, nodes in key places will be more stressed as they will perform the task of CH more often.

Stages 3 and 4, **Inter cluster communication** and **The steady-state (data transmission)** phases are performed as indicated by the LEACH algorithm (Chapter 1).

4.2.2 Simulation analysis of the proposed CH election method

In order to determine the impact of the proposed algorithm on the network parameters such as lifetime and message delivery, we have performed simulations in the Matlab environment which we will further describe.

I. Network lifetime

To determine the impact of the proposed algorithm in terms of network lifetime and compare them to IMSD we have performed simulations on randomly distributed networks of 100 nodes over observation areas with different sizes ranging from 50x50 meters to 200x200 meters with a variation of 25x25 meters. The results of our simulations can be observed in Figure 4.1.

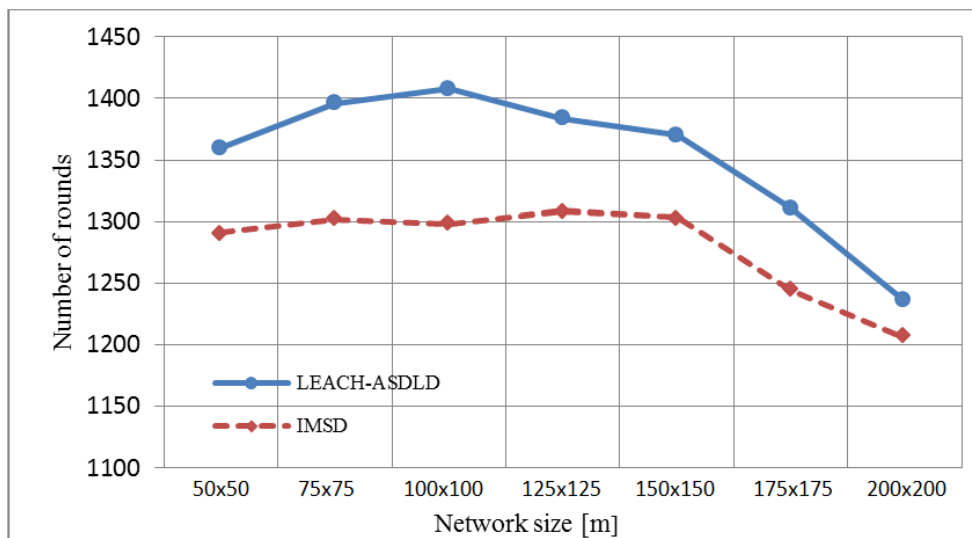


Figure 4.1 Half nodes die (proposed algorithm) vs. last node dies (IMSD) [84]

When comparing several algorithms in terms of network lifetime there are several metrics generally used in literature when addressing WSNs and these are [26]:

- *First Node Dies (FND)* metric compares algorithms considering the time at which the first node has remained without energy.
- *Half Node Dies (HND)* metric compares algorithms based on the time at which half of the nodes have remained without energy.
- *Last Node Dies (LND)* metric compares algorithms having in view the time at which all nodes have depleted their energy resources. Among the three, this metric is actually the most used in literature studies.

When comparing the proposed algorithm with IMSD from the point of view of the FND metric, our proposed solution is outperformed by IMSD, behavior which is actually expected, because as previously mentioned, our solution stresses nodes in key places (where there is a higher density of nodes in the SD). If we however consider the HND metric (visible in Figure 4.1), when half of the nodes deplete their energy using the proposed algorithm while in using IMSD all nodes in the network have already consumed their energy resources. This actually makes comparing with the LND metric futile. From these results we can draw the conclusion that electing CHs in the proposed manner provides extended monitoring capabilities of the WSN over the OA.

There is however a peculiar behavior which can be observed from Figure 4.1 and which needs further explanations. Contrary to initial expectations, with the increase of network OA one would expect that the HND metric would decrease. Behavior noticed only after the OA size exceeds 100 meters. This behavior is related with the size of the OA and with the network model, specifically with the communication distance of nodes which is of 100 meters. This actually means that more messages are sent directly to the BS instead of to locally elected CHs (which are thus relieved of the energy consuming task of receiving messages and performing data aggregation and fusion). This behavior ceases to exist as the observation area size exceeds the communication distance.

II. Message delivery

The impact of the proposed algorithm on the overall message delivery throughout the network was tested also through simulation in several different scenarios which will be further described.

a) Different network sizes

To determine how the proposed algorithm would influence the overall message delivery we have performed simulations on randomly deployed WSNs over square areas of different sizes ranging from 50x50 meters to 200x200 meters with a variation of 25x25 meters. The size of the transmitted packet for all nodes was considered fixed (200bits) and each node sent a total of 20 packets each round. The number of CHs per round was also fixed and set to 10. The results of the performed simulations are available in Table 4-2 and can also be seen in Figure 4.2.

Table 4-2 Message delivery increase [%] with network size compared to IMSD

<i>Size of the operation area [meters]</i>	<i>Message delivery increase [%]</i>
50x50	15.68
75x75	14.27
100x100	12.50
125x125	9.02
150x150	7.22
175x175	4.14
200x200	3.99

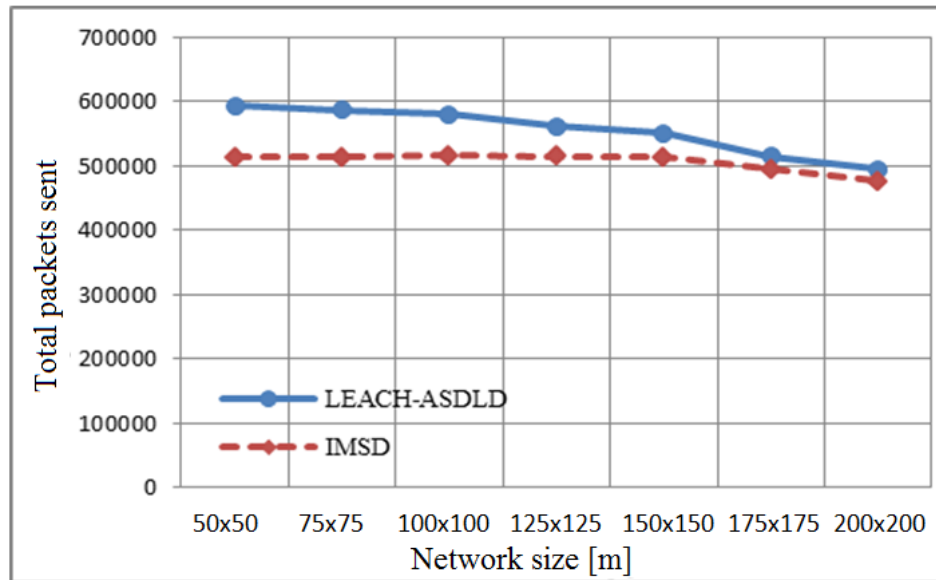


Figure 4.2 Number of packets sent for different network sizes (proposed vs. IMSD) [84]

Throughout our simulations we have observed that there is an increase in overall packets sent throughout then network depending on the size of the OA. This increase ranges from 15% for the 50x50 meters network and goes down to 4% for the 200x200 meters network. This behavior is actually expected because there should be a tight correlation between the number of nodes in the network, the size of the OA and the communication range. If this correlation is not considered, increasing the OA will lead to the point where the WSN exceeds the saturation point and is no longer able to perform communication because the distances between nodes become too large with respect to the number of nodes in the network and the communication distance.

b) Different number of CHs

An important topic in the field of hierarchical WSNs is electing the right number of CHs. This aspect is in tight correlation with the desired application, network spread pattern and the number of nodes distributed in the observation area. In this paper we have not addressed the issue of determining the optimum number of CHs but we have performed simulations with different number of CHs in order to determine the impact on the overall packet delivery throughout the network for the proposed algorithm. We have considered in our simulations a network consisting of 100 nodes randomly spread over an OA of 100x100 meters size and the number of CHs was varied between 3 and 10. The results can be seen in Figure 4.3.

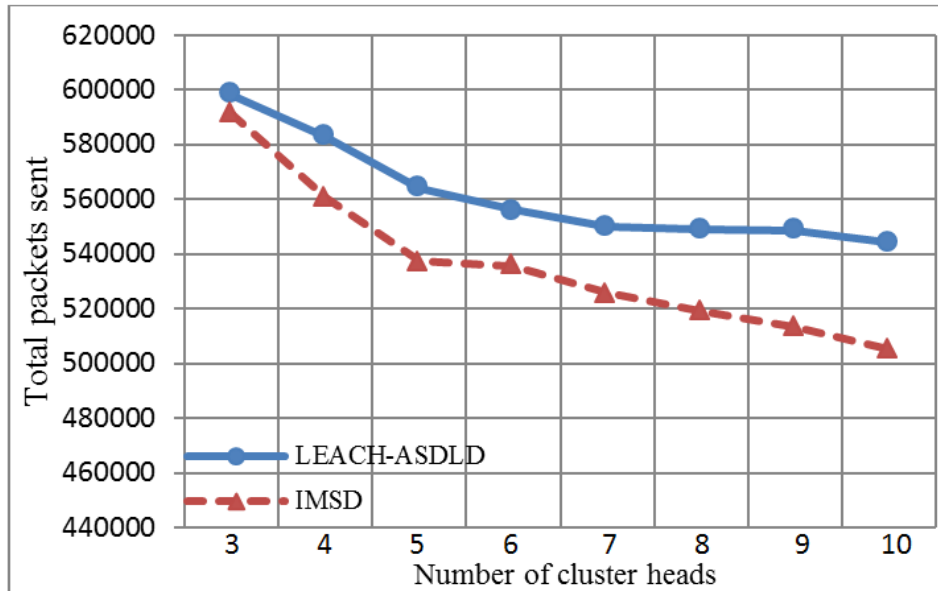


Figure 4.3 Number of packets sent for different number of CHs (proposed vs. IMSD) [84]

As can be seen from Figure 4.3 both algorithms have the highest number of delivered packets when the number of CHs is 3, where our proposed algorithm shows only a small improvement of 2%. This behavior is actually expected because the target of both algorithms is to provide network optimizations through clustering. If the number of CHs tends to 0 or to the number of nodes in the network the two will perform the same. A numerical representation in percentage of the differences between the two algorithms in terms of overall packets sent per number of CHs is also presented in Table 4-3.

Table 4-3 Message delivery increase [%] for different number of CHs compared to IMSD

Number of CHs	Message delivery increase [%]
3	2.01
4	3.79
5	4.72
6	3.70
7	4.52
8	5.48
9	6.41
10	7.88

As the number of CHs is increased the percentage of received messages by using the proposed algorithm also increases with a maximum of 7.8% for 10 CHs.

c) Different packet sizes

Since the proposed algorithm also considers the size of the packet to be sent, we have performed simulations which take into account the presence of an event in the network which would trigger an increase in the amount of communicated data. To simulate the event we have spread the OA into 4 quadrants (same as in the 2 D cartesian coordinate system) and for a specified number of rounds (we have chosen 300), the size of the packet will be increased from 200 bits/packet with a specified percentage (25, 50, 75 and 100) in quadrant 2 and then in quadrant 4. The size of the OA was considered of 100x100 meters with 100 nodes randomly spread throughout the area. When comparing the two algorithms we have noticed that when increasing the packet size, due to the election method of electing CHs of the IMSD the area in which the event takes place is left without CH (because nodes in those areas consume more energy/packet sent) which is not the case in our proposed algorithm. We have also noticed through simulations that by reducing the SD in areas where an event takes place there is an increase in the overall packet transmission. Through experimental tryouts we have obtained the best results by reducing the SD for the nodes in the area where the event takes place by 7 [%] / 50 bit extra / packet. The results are displayed in Figure 4.5.

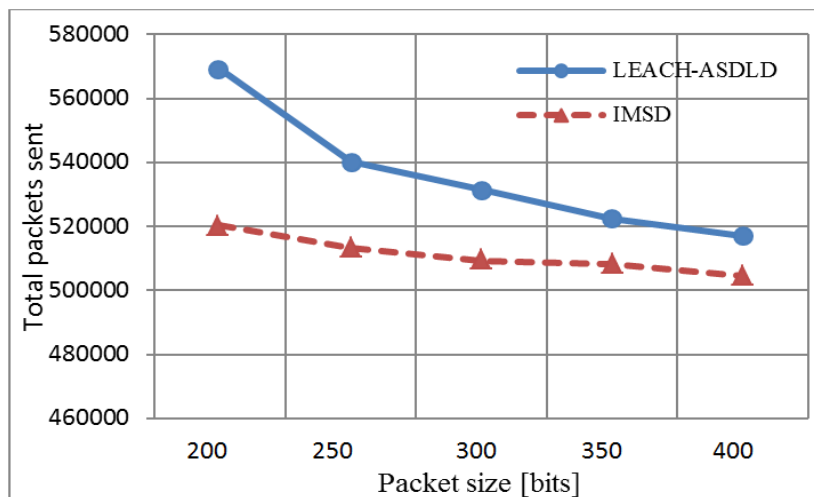


Figure 4.4 Number of packets sent for different packet sizes (proposed vs. IMSD) [84]

Table 4-4 Message delivery increase [%] for different packet sizes compared to IMSD

Packet size [bits]	Message delivery increase [%]
200	8.61
250	5.01
300	4.15
350	3.66
400	2.75

Our results show an increase in packet delivery that ranges from 8% for the 250 bits packet size to 3% for the 400 bits packet size when comparing the two algorithms (detailed results are available in Table 4-4).

4.3 A new method for electing attending CHs

There are usually several metrics used in routing protocols to assess the link quality between two communicating nodes which we will briefly describe:

- **Assessing the RSSI** [85] – is the most used method for WSNs as it can be easily implemented for nodes with limited power and it induces very little communication overhead.
- **Per-hop round trip time (RTT)** [86] – is used for measuring the round trip delay of packets between neighboring nodes. This metric however does not provide information about the lossy characteristics of a link but it helps reduce delay and channel contention.
- **Per-hop Packet Pair Delay (PktPair)** [87] – is usually applied in wired networks to measure the delay between a pair of back-to-back probe packets to a neighboring node. It is a useful technique to assess link quality in terms of retransmissions and bandwidth availability.
- **Expected Transmission Count (ETX)** [88] – estimates the number of retransmissions needed to send packets by measuring the loss rate between a pair of nodes. To calculate the loss rate, each node sends a packet every second that contains the count of packets received from each neighboring node in the last ten seconds and thus all nodes that receive this message can estimate the link loss.

In hierarchical routing protocols CHs are elected by nodes that wish to attend to a cluster usually using the closeness criterion. The distance to a CH is either known a priori or determined in field either by being broadcast by the CH (nodes can broadcast their position if they are equipped with a position detection device such as a GPS) or it can be calculated as a function of the RSSI as in LEACH [26].

Whatever the chosen methods neither of them provide a good link quality indicator [83] meaning that no prior knowledge is available about the characteristics of the link in terms of possible communication faults. This fact is due to the tight correlation between the lossy features of a link and the characteristics of the environment in which the network is used.

Throughout literature there exist several studies which have been performed by researchers that evaluate the lossy characteristics of links with the purpose of creating a link model which can be used for estimating the features of a link. An example of such a research is that performed by Ganesan et al. [89] which have deployed 180 nodes in an unobstructed parking lot with the purpose of determining the loss and asymmetry of packet reception at the link and MAC layers. Another example of a more comprehensive study is that performed by Zhao and Govindan [90] on a test bed of 60 linearly deployed in a hallway of a building, a state park and in an unobstructed parking lot. Their analysis of the environmental data has shown a variation of packet loss with distance and with time indicating the presence

of what the authors refer to as a "grey area" [90] (an interval of packet loss variation with distance and time).

All these studies were performed with the purpose of determining whether LQ can be somehow anticipated. Even though the main purpose of the studies was not accomplished an important feature can be noticed, specifically from the latter, more extensive of the studies mentioned, which is the fact that for a given link the packet loss varies constantly in an interval (most likely tightly correlated with environmental characteristics in which the experiments have been performed).

In order to determine whether a new approach can be adopted which would allow for a more efficient CH election we propose that during the initial reconnaissance procedure explained in the previous subchapter, each node must also send a predefined number of packets (we have considered 1000) in a TDMA fashion. The packets should contain an index (from 1 to 1000 in this case) and they should also be the size of the expected packet to be transmitted throughout the network lifetime.

Using this approach, at the end of the reconnaissance procedure each node will be able to calculate the percentage of failed received packets from neighbors which can then be used as a LQI in electing the appropriate CH.

To test the proposed scheme we have performed in field measurements to obtain a packet loss model for a certain environment. We have used the results in our simulations to determine the impact of the proposed scheme on several network parameters such as lifetime, overall failed received packets and energy dissipated through unsuccessful packet delivery.

In the following subchapters we describe the experimental setup used for in field measurements as well as the simulations performed and the impact of the proposed CH election method. Since we have performed our simulations on the LEACH protocol we have entitled our CH election method LEACH-PLC (or LEACH with Packet Loss Consideration [91])

4.3.1 Description of the experimental setup

Several tests have been performed using the MEMSIC IRIS [92] WSN with the purpose of obtaining a packet loss model based on a realistic situation. The MEMSIC sensor network used consists of 10 IRIS nodes and each node is equipped with a 7.37 MHz Atmega 1281 processor with 128 kB flash, 4 kB EEPROM and 8KB RAM, a low power O-QFSK 2400 GHz radio transceiver (AT86RF230) with 100 meters maximum communication distance and sensors for temperature, humidity, barometric pressure, acceleration and ambient light sensing. Also, each node is driven by the TinyOS [93] operating system which also supports error correction and detection.

Our tests have been performed in an open environment as can be seen from Figure 4.5 by placing the BS in a fixed position and the 10 nodes in a straight line 10 meters apart from each other (the sensor nodes are too small to be visible in Figure 4.5 but the position of three of them was marked with red dots).



Figure 4.5 Measurement location (several node positions with white) [91]

Each node was programmed to communicate sensor data of approximately 200 bits/packet every 10 seconds to the base station. The experiment was stopped when 1000 packets were received from each node. Under the same conditions the experiment was repeated two more times in the same environment with the nodes placed at the same positions. Using the MEMSIC MoteView (Appendix D) data analysis program we were able to analyze the health packets sent by each node and determine the packet loss for each link to the BS. Results are shown in Figure 4.6.

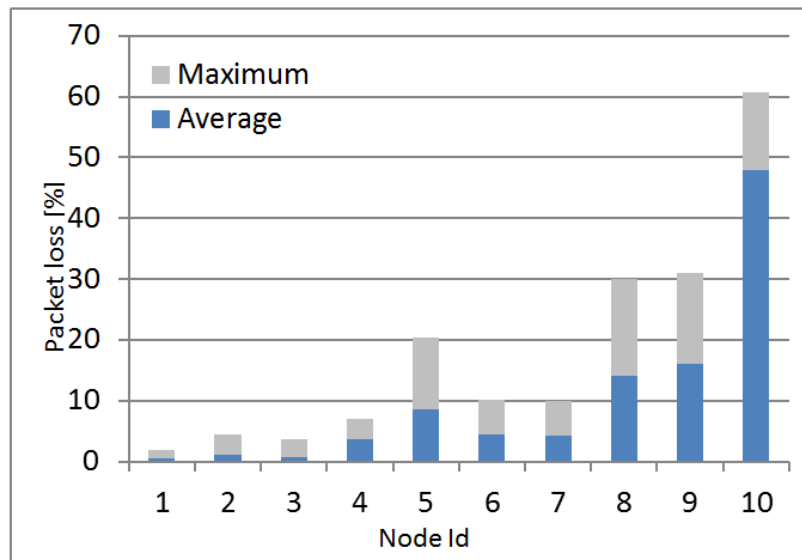


Figure 4.6 Packet loss between each node and the base station [91]

In accordance with the experiments described by the authors of [89] and [90] we have also noticed that packet loss varies for each link between a given interval (referred to as a "grey zone" by Zhao and Govindan [90]). This variation is specific to the environment and increases with distance in the sense that nodes closer to the BS experience packet loss variation between 0 and 2% while the last node between 40 and 60%. The results were validated by experiments performed in the following two days which showed a minor discrepancy of only 1% from the results obtained in the first day.

The resulting loss model was stored in the form of a look up table and used in the simulations of LEACH [26] to determine the impact of choosing attending CHs based on the link quality obtained through the reconnaissance procedure. The simulation results are presented in the following subchapter.

4.3.2 Analysis of the proposed method for election of attending CHs

In order to determine the impact of electing attending CHs based on the link quality indicator we have considered that all nodes within the environment are subject to the same packet loss as detected through our experimental results. The results of our measurements can be seen in Table 4-5.

Table 4-5 Detected packet loss variation with distance [91]

Distance[m]	Min[%]	Max[%]	Average[%]
10	0	2	0.57
20	0	4.59	1.19
30	0	3.75	0.72
40	1.38	7.14	3.73
50	2.09	20.37	8.60
60	2.54	10.18	4.52
70	1.86	10	4.29
80	2.77	30	14.13
90	5.90	30.95	16.08
100	23.33	60.74	47.89

All nodes within the network were set to elect attending CHs based on the smallest average packet loss and not based on the RSSI metric. In terms of packets sent we have considered that all nodes must send during each round a total of 20 packets, each 200 bits long.

The measured packet loss variation was also considered in our simulations to account for failed received packets and to provide a more realistic simulation environment. Simulations were performed in the Matlab environment on networks with 100 nodes placed in observation areas of dimensions ranging from 50x50 meters to 150x150 meters with a variation step of 25x25 meters. We have analyzed

the results of using the proposed election method and compared them with those of LEACH in terms of network lifetime, packet throughput and energy dissipated due to unsuccessful packet delivery, results which we will further present.

I. Network lifetime

From our simulation results we have observed that using the proposed method for electing the attending CHs results in an increase in network lifetime that ranges from 40% for the 50x50 network size to 17% for the 150x150 meters network considering the LND metric (most often used in literature). The results can also be seen in Table 4-6 for all 3 metrics.

Table 4-6 Comparison between network lifetime for the two algorithms

Network size	Metric	LEACH	LEACH-PLC	Increase [%]
	FND	732	150	0
50x50m	HND	1247	1960	57.2
	LND	1620	2270	40.1
	FND	812	183	0
75x75m	HND	1260	1739	38.0
	LND	1680	2180	29.8
	FND	749	137	0
100x100m	HND	1217	1418	16.5
	LND	1548	2103	35.9
	FND	820	98	0
125x125m	HND	1380	1491	8.0
	LND	1377	1787	29.8
	FND	750	91	0
150x150m	HND	1100	1044	0
	LND	1289	1510	17.1

It can be observed from Table 4-6 that using the link quality indicator as a metric for electing the attending CHs does not have a significant impact on how nodes remain without energy using the FND metric (issue also discussed in the previous subchapter). It can also be observed that the two solutions tend to the same behavior in terms of how nodes remain without energy as the size of the OA increases. This comes to consolidate the conclusion drawn in the previous subchapter that there should be a tight correlation between the size of the OA, the number of nodes and the communication distance.

A graphical comparative representation of how nodes remain without energy using the two solutions is available in Figure 4.7 (for the 100x100 meter observation area size).

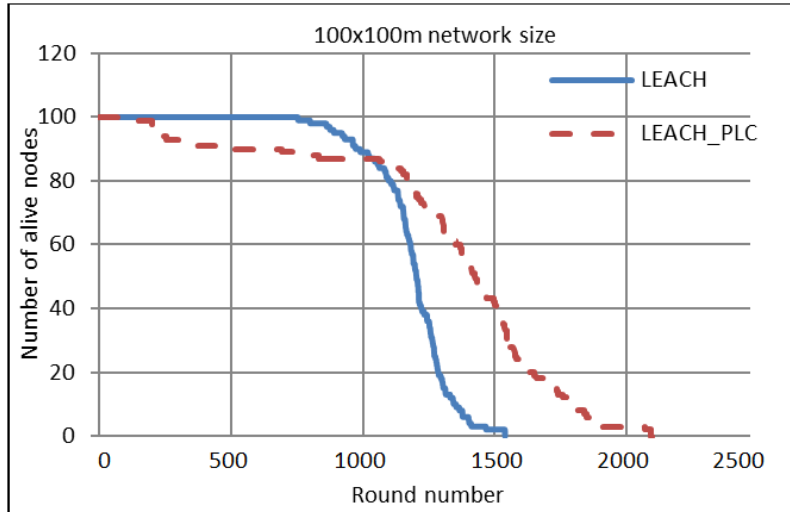


Figure 4.7 Comparison between the number of alive nodes [91]

II. Network throughput

The significance of the proposed method to elect attending CHs can be also determined by analyzing network throughput (by throughput we refer to the average rate of successful packets delivered which will be represented in packets per round). The results of our simulations can be observed in Figure 4.8 which compares the overall network throughput between LEACH and the proposed method [91].

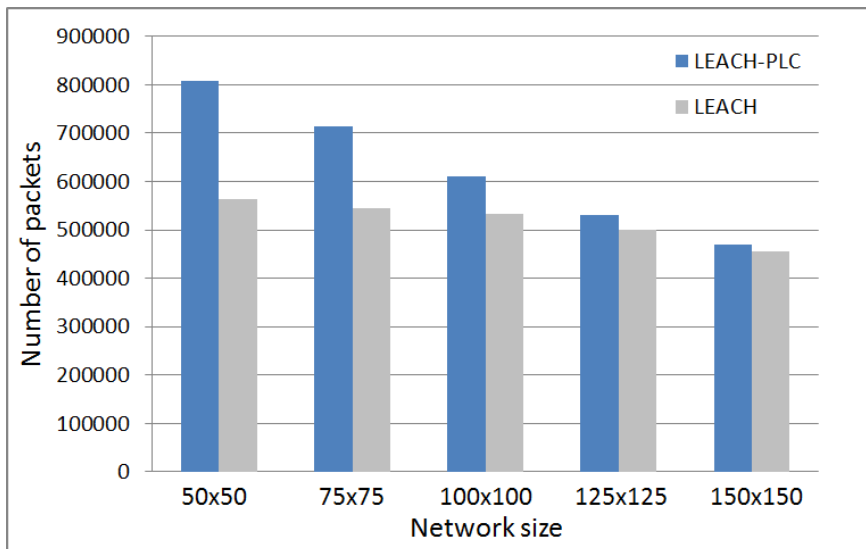


Figure 4.8 Comparison of network throughput for different OA sizes [91]

As can be observed from Figure 4.8 the throughput of the proposed election scheme is considerably larger than that of LEACH due to the fact that there are fewer packets lost each round. Results in terms of packet increase between the two algorithms under test are also available in Table 4-7.

Table 4-7 Increase in packet throughput of the proposed scheme compared to LEACH

Network size [m]	Increase in throughput [%]
50x50	43.20
75x75	30.69
100x100	14.48
125x125	6.47
150x150	3.06

It can be seen that electing the attending CHs considering the link quality calculated as proposed can significantly increase the overall packet throughput of the network. The greatest increase detected was for the smallest network size and it is of 43 %. As expected increasing the size of the OA will reduce the differences between the two algorithms.

I. Energy wasted due to packet loss

Another aspect which we have considered of great importance and worth studying was the impact of the proposed election scheme on the energy dissipated due to unsuccessful packet delivery both at the network and node levels. Figure 4.9 depicts a comparison between LEACH and the proposed method in terms of energy dissipated by individual nodes for the 100x100 meters network size.

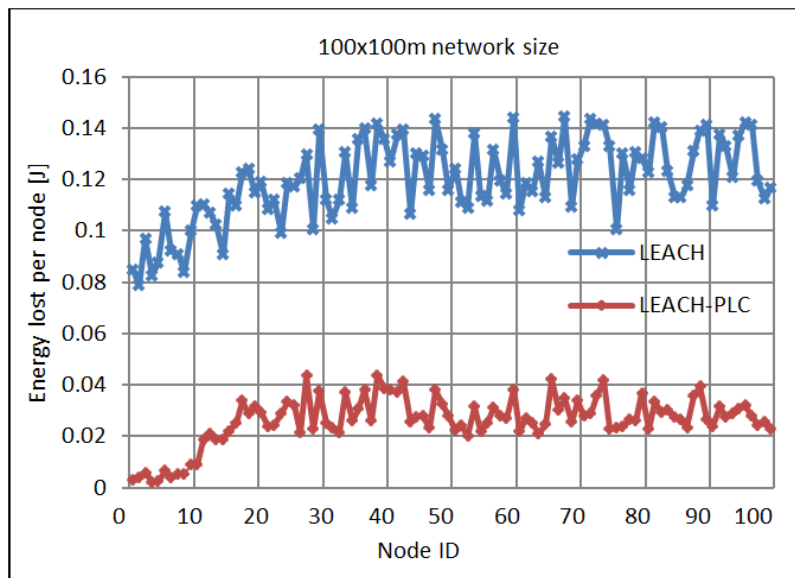


Figure 4.9 Energy dissipated by each node due to unsuccessful packet delivery [91]

It can be observed from Figure 4.9 that using the proposed method individual nodes will waste approximately three times less energy as compared to LEACH. We have also calculated the overall energy dissipated throughout the network for each size of the OA. A graphical representation of the average energy dissipated each round is shown Figure 4.10.

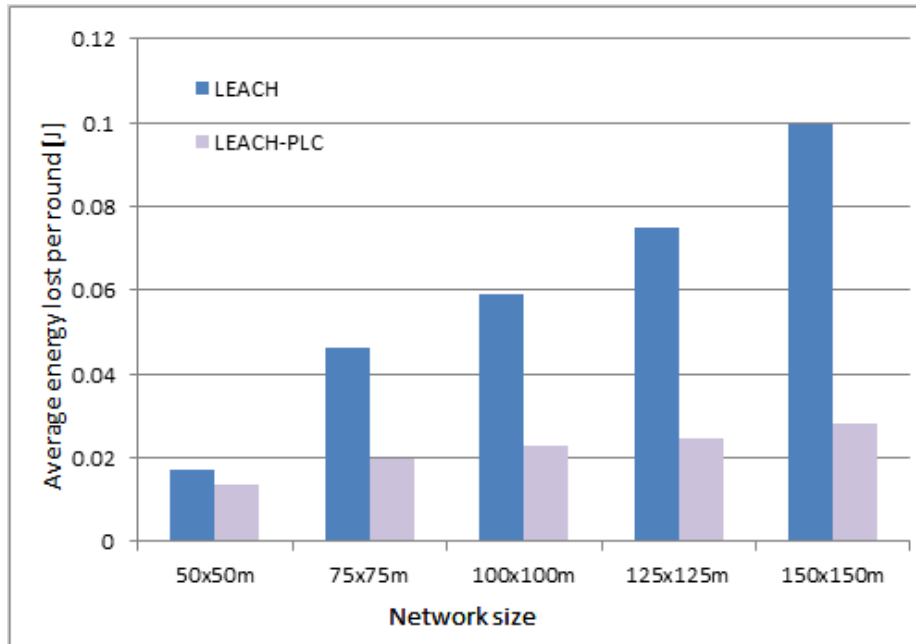


Figure 4.10 Average energy lost due to unsuccessful packet delivery for different OA sizes [91]

Figure 4.10 shows that significant improvements can be obtained using the proposed election method. Another important aspect which can be observed is that the energy dissipated by using LEACH can reach up to 30% of a nodes' initial energy resources (which are of 0.5 J as specified in the network model [26]). The numerical results are available in Table 4-8 where each value is expressed in *Joule*.

Table 4-8 Average energy dissipated due to unsuccessful packet delivery for different OA sizes

Network size	LEACH [J]	LEACH-PLC [J]
50x50	0.0171	0.0138
75x75	0.0463	0.0198
100x100	0.0491	0.0227
125x125	0.0748	0.0247
150x150	0.0997	0.0283

It can also be noted that using PLC the overall energy dissipated through unsuccessful packet delivery varies little with the increase in OA size and reaches a maximum of 8% for the 150x150 meters network.

4.4 Conclusions

In this chapter we have addressed issues of energy efficient clustering for static WSNs randomly distributed over square observation areas of different sizes. We have specifically focused our attention of two issues of great importance which represent the basis of many research studies: efficient cluster head election and adequate intra cluster communication.

When considering efficient CH election we have proposed, described and simulated a new method that is based on several important network parameters such as the number of neighbors in the area surrounding the CH, the expected packet size to be delivered and the separation distance between CHs for a uniform distribution throughout the extent of the OA

We have tested the proposed method through Matlab simulations and we have presented the beneficial impact of electing CHs as proposed by comparison with another similar approach used in literature. Our simulations have shown increase in network lifetime and message delivery under for various network parameters.

The second research direction addressed in this chapter describes a new method that nodes within the network can implement for choosing the best CH to attend. The proposed method will however induce delay and overhead but unlike other methods these downsides are restricted to the initial moment when the network begins to function.

This method is based on results obtained from experiments performed in field with the purpose of determining a path loss model for a certain environment. The experimental results have validated other researches performed by several authors and have also been used in simulations to determine the impact of electing attending CHs based a link quality indicator. They have also served as a path loss model which has been used throughout our simulations and which all nodes have been subject to.

Our simulation results have shown significant improvements in terms of network lifetime, packet throughput and energy dissipated due to packet loss.

Both techniques presented in this chapter have been thoroughly explained but have however only been validated through simulation and an implementation on a real WSN still needs to be performed.

4.5 Contributions

1. **Proposal of a new method for electing cluster heads** based on:
 - a. **The number of nodes in an area surrounding the CH**
 - b. **Estimated packet size**
 - c. **Adaptive separation distance between CH**
2. **Validation of the proposed election method** through Matlab simulations
3. **Comparison between the proposed method and IMSD**, an improvement to a known clustering protocol (LEACH)
4. **Analysis of comparison results** which have proved the **effectiveness** of the proposed method in terms of:
 - a. **Network lifetime**
 - b. **Packet throughput when varying several network parameters (such as the number of CHs, the size of the observation area and the packet size)**
5. **Proposal of a new method for non-CH nodes to elect attending CHs based on the link quality indicator**. We have also proposed an approach to determine this network parameter which is highly dependent on environmental characteristics.
6. **In field experiments** using the MEMSIC IRIS WSN for link quality analysis. The resulting measurements have been used to create a packet **loss model for simulating** the wireless sensor network in an environment.
7. **Matlab simulations** for validation of the proposed election scheme
8. **Comparison of simulation results with the LEACH protocol** CH election method. **Validation of the proposed scheme** by pointing out the **improvements** obtained in terms of:
 - a. **Network lifetime**
 - b. **Message throughput**
 - c. **Energy dissipated through packet loss**

Chapter 5

Routing and scheduling techniques for mobile WSNs

5.1 Introduction

This chapter is reserved for routing and scheduling issues related to mobile WSNs.

Most of the studies existent in literature have focused their attention on WSNs in which nodes are static or have some degree of limited mobility (as those presented in Chapter 1). Usually when considering mobility most researches are focused on networks with static nodes in which mobile elements (also referred to as mobile data collectors or mobile sinks) are used for data gathering [94-95]. Studies purely focused on networks with mobile nodes are those developed for ad hoc MANETS [30-32]. When referring to static or partially static WSNs communication paths can easily be determined using search algorithms that minimize a certain cost function such as the distance between nodes or the number of hops given that an interconnection graph between nodes is available (determining the best of the two is actually another topic of great interest which will also be briefly addressed in this chapter). Examples of such algorithms are Bellman-Ford, breadth-first or Dijkstra's algorithm [96]. These algorithms are able to solve the shortest path problem in polynomial time for networks with fixed infrastructures but according to [97, 98] they are not suited for networks with rapidly changing topologies such as mobile WSNs. A possible solution for determining the optimum communication path for mobile WSNs could emerge from using evolutionary algorithms such as neural networks (NNs) or genetic algorithms (GA) [98]. Implementing such algorithms on microcontroller based devices such as WSN nodes can prove quite difficult when considering computational time, however this downside can be overcome when distributed on platforms such as FPGAs [99].

In this chapter we present a detailed description of how we have implemented a GA for determining the optimum communication path for mobile WSNs with nodes that move on arbitrary defined paths with different speeds. The paths taken by network nodes are considered pseudo-random because they are randomly defined by the user but they become repetitive when the node reaches the end of a path (if the path is closed the end of the path is also the beginning and the node will continue to move on that path with the same speed, but if the path is open the node will travel backward until it reaches the start point).

The second research direction addressed in this chapter is that of scheduling operations (acquire/send/receive/forward/sleep) for mobile WSNs nodes with the purpose of providing efficient energy consumption while maintaining high QoS.

Throughout literature there are several studies which address scheduling techniques for WSNs such as S-MAC [15, 16], DS-MAC [17] or MS-MAC [18] which are described in Chapter 1.

These protocols strive to support good scalability and collision avoidance through fixed duty cycles which reduce idle listening [15-18]. The latter of these is obtained by adopting periodic listen and sleep schedules and maintaining synchronization between nodes. Listening and sending however, are operations tightly correlated to whether or not there is actually any information to be passed among sensor nodes. More specifically adopting communication schedules becomes redundant if there are no acquisition schedules which provide data to communicate. Also, depending on the nature of the application, network nodes may only serve a specific purpose in time (for example, some nodes may mostly be used as routers while others perform acquisition without routing – marginal nodes).

To address the issue of scheduling and interdependence between tasks we propose a reward based approach which uses reinforcement learning (RL) to teach network nodes what tasks to perform. This method is described in detail in the second part of this chapter.

5.2 A GA based routing approach for mobile WSNs

When considering mobile WSNs, path optimization can be formulated as finding the minimum of a specific cost function from a source (S) to a destination (D) node at each period of time that the node is programmed to send. This translates in a combinatorial optimization problem found in many designs and fields of research [100] which has been successfully solved using evolutionary algorithms such as genetic algorithms (GAs) and neural networks (NNs) [101].

When considering the topology of a mobile WSN at discrete points in time (t), the network can be represented by the undirected weighted graph:

$$G(t)=[N(t),A_{i,j}(t),C_{i,j}(t)] \quad (4.1)$$

where $N(t)$ represents the number of nodes with energy levels above 0 at time t , $A_{i,j}(t)$ represents the set of connections (edges) between nodes i and j at time t and $C_{i,j}(t)$ is the cost value of each connection $A_{i,j}(t)$ at time t .

Each link between two nodes i and j can be classified using the $I_{i,j}(t)$ indicator [99], where:

$$I_{i,j}(t)=\begin{cases} 1, & \text{if a link from } i \text{ to } j \text{ exists at time "t"} \\ 0, & \text{otherwise} \end{cases} \quad (4.2)$$

Hence we can formulate the best routing path issue as an optimization of the following objective function:

$$f_i(t)=w_1 \left\| \sum_{\substack{j=S \\ j \neq i}}^D I_{i,j}(t)C_{i,j}(t) \right\| + w_2 \left\| \sum_{\substack{j=S \\ j \neq i}}^D I_{i,j}(t) \right\| \quad (4.3)$$

where $i \in \{1, N(t)\}$, w_1 is the weight considered for shortest path and w_2 is the weight used for the number of hops, with the constraint that $w_1 + w_2 = 1$. Both shortest path

and number of hops function members have been normalized so that they will have the same influence on the overall value of the fitness function.

In the following paragraphs we will present a description of how the genetic algorithm was implemented and configured to determine the optimum communication path. We will also describe each individual stage the algorithm undergoes. A schematic block diagram of the GA functionality is presented in Figure 5.1.

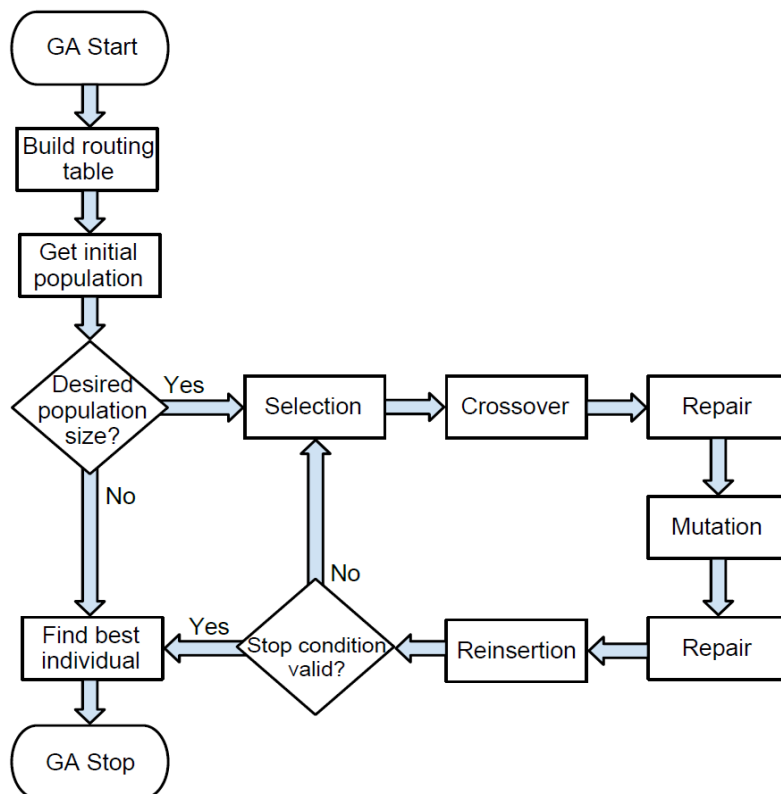


Figure 5.1 Block diagram of the genetic algorithm implementation

5.2.1 Stages of the genetic algorithm

As can be seen from the block diagram represented in Figure 5.1 the GA begins by building the routing table and obtaining the initial population. The first stage is not actually part of the genetic algorithm representation as known in literature but we have included it here because the algorithm cannot be applied if the procedure is not executed.

Building the routing table is a task reserved for real time routing protocols such as OFSF [102], DSDV [103] or DSR [104] which can provide the topological information needed rapidly. The other stages of the GA implementation are described in the following paragraphs.

I. Population initialization

It is important to mention that in this chapter we refer to population as being a number of predefined paths from a source to a destination node. Obtaining the initial population is an issue of high importance when implementing GAs. There are generally two types of issues that can be discussed when initializing the population [105], size and initialization procedure, both of which play an important role as will be further detailed.

Electing the appropriate population size is a matter of optimizing the convergence time of the GA [106]. This is actually a direction of intense research in the field of genetic algorithms but it is not the target of the current paper. It is however imperative to mention that the size of the initial population must be carefully chosen because electing too little individuals may determine the algorithm to converge to local minima of the objective function, which is not desired.

When considering population initialization there are generally two methods: heuristic and random initialization. In heuristic initialization the fitness of the population is already low [105] and the algorithm tends to converge to local minima solutions due to the lack of population diversity. To avoid this possible situation we have generated the initial population using random backtracking.

II. Genetic representation of paths

In order to relate to the genetic representation from now on in this section we will refer to a path as a chromosome. Chromosomes in the population can be of fixed or variable sizes and they are made of genes. A gene is a genetic representation of a node and it is identified by the nodes' index. The places of a gene in the chromosome are referred to as a locus (pl. loci). The first and last loci are reserved for the source and destination nodes. Figure 5.2 illustrates the genetic representation of a path with its associated genes.

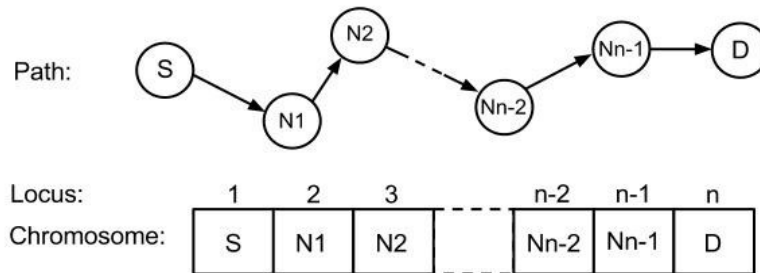


Figure 5.2 Representation of a path and associated chromosome

The length of a chromosome can be of any size but it normally should not exceed the number of genes in the network $N(t)$ as this is an indicator that a loop has been formed somewhere along the path which must be excluded. Compared to fixed length, variable length chromosomes are desired and will be used in our implementation of the GA as they provide increased diversity of the selected population.

In order to optimize the desired objective function the chromosomes elected in the initial population must undergo a series of genetic operations such as selection, crossover, mutation and reinsertion which will be further described.

III. Selection of individuals

One of the most important stages of a genetic algorithm is selecting the individuals from the initial population and inserting them into the mating pool. These individuals will later be used as basis for creating new chromosomes which will form the next generation.

Selection schemes can be classified into proportionate and ordinal-based [107], both of which depend strongly on the selection pressure which is the degree to which better individuals are favored. The higher the selection pressure the more the fitter individuals are favored for election.

Proportionate selection schemes are more sensitive to selection pressure as they elect chromosomes based on their fitness value relative to the fitness of other chromosomes in the population. Examples of such selection schemes include roulette wheel, stochastic remainder and stochastic universal selections [107].

In ordinal based selection schemes, chromosomes are ranked according to their fitness values and selection is performed based upon the ranks of individuals within the population. Examples of such schemes are (μ, λ) , linear ranking and tournament selections [107].

For our implementation of the genetic algorithm we have chosen to use the ordinal based tournament selection scheme [108]. An important parameter of tournament selection is the tournament size which gives the number of individuals elected from the initial population which will undergo tournament selection. Another important parameter of the genetic algorithm is the mating pool size which is actually a representation of how many times the tournament selection will be performed. Considering x to be the tournament size there are x individuals in selected which will concur for a place in the mating pool. Individuals are elected randomly from the available population and only the fittest individual of the x elected will go on to be used in the next stages of the genetic algorithm. The selection procedure is repeated until the desired number of individuals given by the mating pool is obtained. In order to maintain diversity and avoid finding local minima solutions individuals once elected to participate in the mating pool can no longer be elected for to take part in the tournament selection.

Maintaining only the best individuals resulting from the genetic algorithm stages is referred to as an elitist procedure, term which we will also use in our description.

IV. The crossover stage

After the selection procedure, individuals in the mating pool undergo the crossover stage of the genetic algorithm. In this stage genes are interchanged with the purpose of obtaining new, fitter individuals. The crossover stage can be an n -point procedure [109], meaning that n genes can be interchanged, where $n < ChromosomeLength - 2$ (-2 because we do not want to change the source and destination genes). In our implementation of the genetic algorithm we have chosen the one-point crossover scheme which will be briefly described.

From the mating pool two random chromosomes are elected. For the first chromosome a random integer number is generated, greater than 1 but smaller than the chromosome size (to avoid electing the source and destination genes for crossover). The selected gene index from the first chromosome is then searched for in the second one and if it is found the paths from that gene locus to the destination stage is performed.

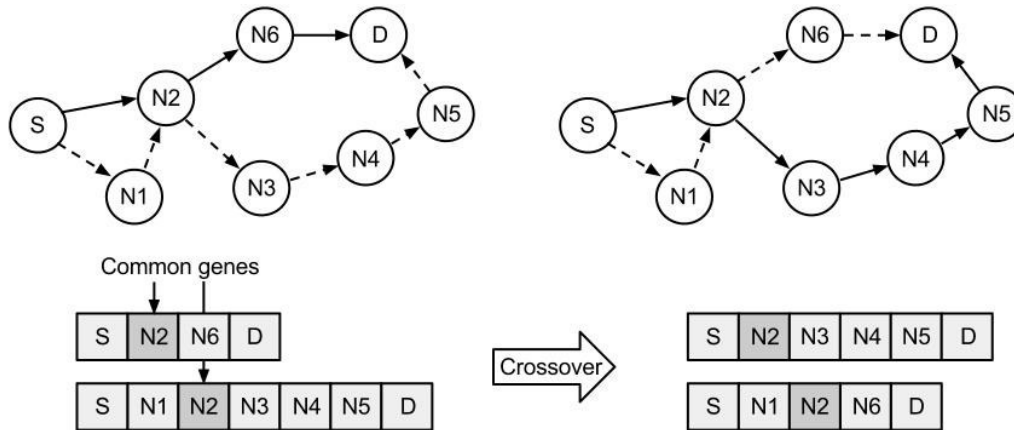


Figure 5.3 Example of one-point crossover

After crossover is performed the newly formed chromosomes are compared fitness wise with their parents and the best individuals are kept in the mating pool (elitist procedure). It is very important to observe that by performing crossover the newly formed chromosomes may have loops which need to be eliminated. This task is performed by the repair function which will be later described.

V. The mutation stage

The chromosomes in the new mating pool which have been subject to the crossover stage will afterward undergo mutation. In this stage of the GA for each individual in the mating pool a random gene is elected. If an alternate path from that gene to the destination node exists it will replace the current path. The fitness of the newly created chromosome will be compared that of its parent and the best of the two will be inserted in the new mating pool, meaning that this procedure is also elitist. A schematic description of how mutation is performed is presented in Figure 5.4.

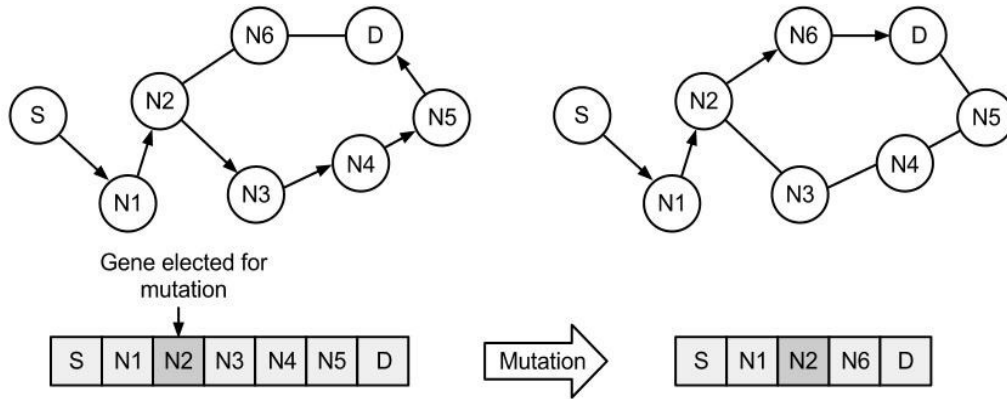


Figure 5.4 Example of the mutation procedure

Performing mutation can also lead to candidates that have loops and therefore the resulting mutated chromosomes must be subject to the repair function.

VI. The repair function

As previously stated the task of the repair function is to eliminate loops that may appear after the crossover and mutation stages. The implementation of this function is quite simple but imperative. A schematic representation of how the repair function works is depicted in Figure 5.5.

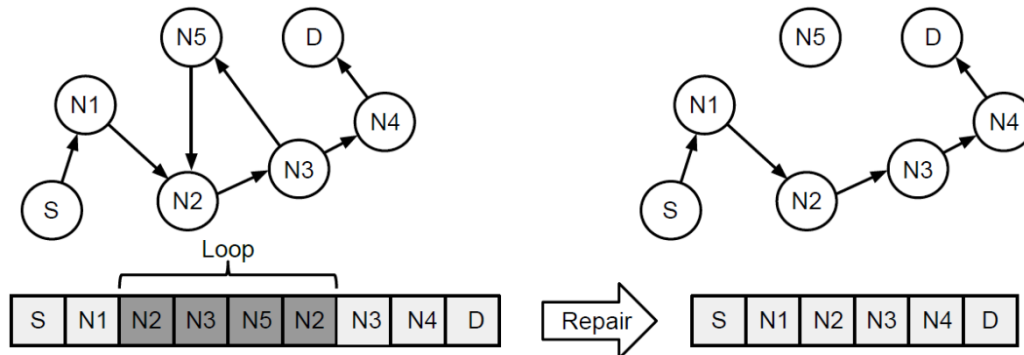


Figure 5.5 Example of the repair function

For each individual chromosome in the mating pool, beginning from the first gene after the source and ending with the one before destination, genes are compared. If two identical ones are found, a loop exists within the chromosome and the repair function will act upon the chromosome by replacing the path between the two genes with the one from the gene with the greatest loci to the destination, as can be seen in Figure 5.5.

VII. Reinsertion of newly formed individuals

After the individuals in that form the initial population have undergone all the stages described above they will be subject to the reinsertion stage. During this stage the chromosomes in the mating pool are compared with those that form the initial population. This comparison is performed of course fitness wise and only the best chromosomes are kept which will represent the new initial population which will undergo the stages of the GA previously described if the stop condition has not been reached.

VIII. Stop condition

The algorithm is considered to have converged to the best solution if a pre-defined number of iterations have passed or if no chromosome with a better fitness has been found for a specified number of consecutive runs.

5.2.2 Simulation environment and network parameters

To be able to assess the performance of the proposed genetic algorithm approach for determining the optimum communication path between sensor nodes with mobile characteristics we have performed a series of Matlab simulation on networks comprising of 10/20/30/40/50 nodes with various degrees of mobility. An example of a simulated scenario with 50 nodes is available in Figure 5.6.

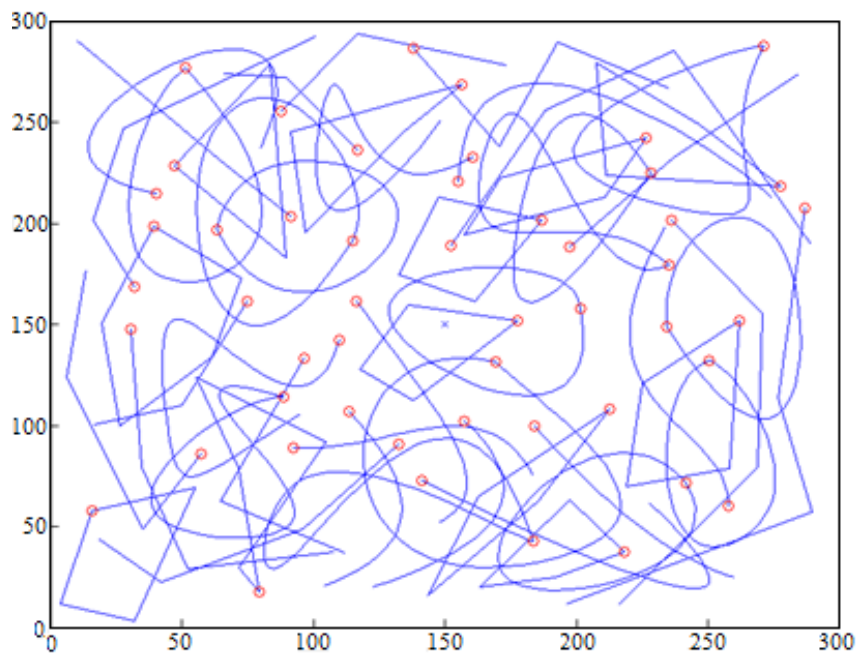


Figure 5.6 Simulation scenario for a mobile WSN with 50 nodes placed in an OA of 300x300 meters

In Figure 5.6 nodes are represented with a circle and the paths that nodes travel on are represented by a line. The destination node (or base station) is represented by an X in the center of the observation area. Each node was appointed the task of sending packets through routes provided by the GA each communication round. Several other considerations on network parameters and the execution of the GA have been made which will be further described.

Chosen network parameters:

- OA size: 300x300 meters
- Number of nodes: 10/20/30/40/50
- Random user defined paths for each node
- Node parameters:
 - User given speeds (1-5 meters/second) and time intervals for communication (1 second)
 - Initial energy: 0.5 J
 - Transmit and receive energy: 50nJ/bit
 - Packet size: 2000 bits
 - Maximum communication distance between nodes: 100 meters
- The network is considered noise and error free

GA considerations:

- Heuristically we have considered that the GA should only execute if the initial population is of at least 20% the number of nodes in the network (for example, if there are 50 nodes in the network there should be at least 10 available paths for the GA to execute)
- The selection, crossover and mutation stages are elitist (only the fittest individuals are retained for future reproduction)
- In order to determine what the compromise should be when electing the optimum path as the shortest or the one with the smallest number of hops we have varied the weights of the objective function as follows:
 - $w1 = 0, w2 = 1$ – just shortest path
 - $w1 = 0.5, w2 = 0.5$ – shortest path and number of hops have the same influence on the objective function
 - $w1 = 0, w2 = 1$ – just number of hops
- As stop condition we have considered that the GA has converged for each node after 100 executions or if no better solution has been obtained in the last 10 consecutive runs

5.2.3 Simulation results

Using the above considerations we have performed Matlab simulations to evaluate the performance of the proposed GA approach in terms of convergence time, path quality but also to see if we can determine what the compromise should be when electing between the shortest path and the one with the least number of hops.

I. GA convergence time

When implementing the genetic algorithm the convergence time for obtaining the best solution of the objective function is highly dependent on several parameters such as the initial population size, the degree of diversity of individuals in the initial population, the selection pressure, crossover and mutation attempts and most importantly the number of iterations specified by the stop condition. These however represent fundamental research directions for genetic algorithms and are outside the scope of this work.

In order to test the quality of the solutions provided by the proposed implementation of the GA we have compared the convergence times of the solutions with those of Dijkstra's algorithm for finding the shortest path. The results of the comparison are presented in Figure 5.7.

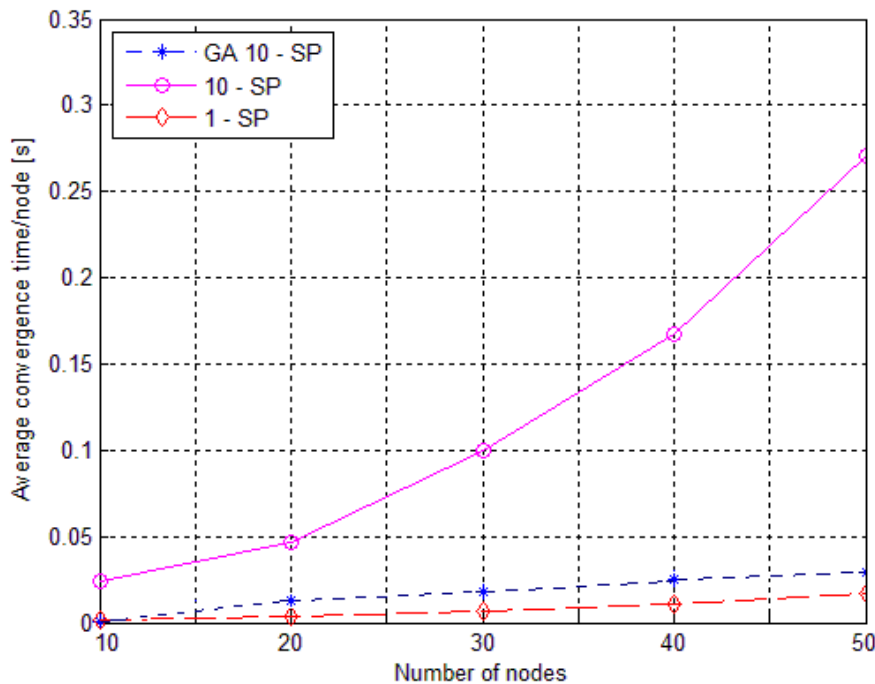


Figure 5.7 Comparison of convergence times between GA and Dijkstra's

When comparing the convergence time of the two algorithms two important issues arise. We can observe from Figure 5.7 that the convergence time for the GA to determine the best path is greater than that of Dijkstra's algorithm. This behavior comes as no surprise when considering that in literature Dijkstra's algorithm is known to be one of the fastest single source shortest path algorithms [110]. However when looking at the actual times needed to find the best path we can see that even for the 50 nodes network the GA requires approximately 27 ms/node to find the path, which means that the algorithm can be used for real time applications.

A second important aspect which needs to be mentioned is that using the GA approach for determining the best path will result in the generation of more than one path between the source and destination nodes (specifically the number of individuals in initial population). Moreover the paths generate by the GA will be highly divergent due to the crossover and mutation stages. This aspect is of great importance due to the high path failure probability of mobile ad hoc WSNs [104] and thus if a path fails the source node will have additional paths through which it can send data to the destination node. For this reason we have also calculated the time required for Dijkstra's algorithm to find k best paths (in our case 10) as those provided by the GA. When we analyze this situation we can see that using the genetic algorithm approach should be preferred as it is faster. For example the average time required by the GA to find 10 best paths is approximately 10 times less than that required by a k -SP Dijkstra's based shortest path algorithm.

II. Quality of the path provided by the GA

A very important evaluation parameter is that of quality of the generate paths by the genetic algorithm. In order to determine path quality we have compared the fitness of the resulting paths with those provided by Dijkstra's algorithm at random moments in time. Throughout our simulations we have observed that there are several situations in which the generated paths are not of high quality. The worst case scenario for the 50 node network which we have encountered throughout our simulations is depicted in Figure 5.8.

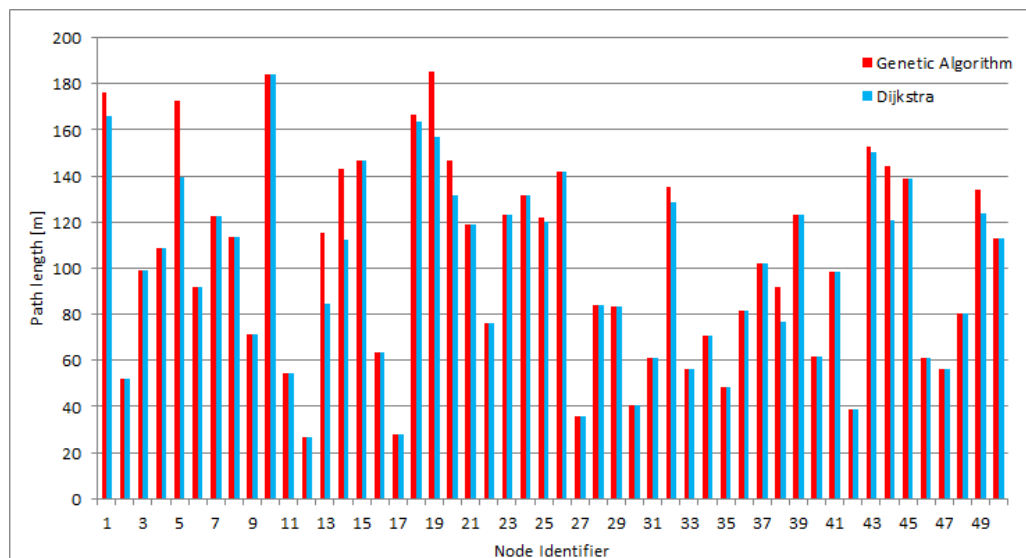


Figure 5.8 Comparison between Genetic Algorithm and Dijkstra shortest path lengths

By analyzing the data represented in the above figure we have noticed that the proposed GA approach will converge to global minimum solutions of the objective function in 74% of the cases. For the rest the error ranges between 1.6% and 19% (for node index 5 in Figure 5.8).

This behavior is most likely either due to the heuristic approach taken to determine the number of individuals in the initial population or because these individuals are not diverse enough which leads to finding local minima solutions rather than global ones. This issue can be however addressed by implementing a population sizing equation which will provide better results such as the one proposed by Ahn et al. [98].

III. Shortest path vs. number of hops

Since we have designed the objective function of the GA in such a way as to optimize more than one path parameter (length or number of hops) we have decided to address another topic of intensive research in the field of WSNs, that of finding a compromise between the shortest path and the smallest number of hops between the source and destination nodes.

According to the Friis transmission equation for free space [48] (eq. 1.4), the power available at the receiver is inverse proportional with the square of the distance between the source and destination nodes. Considering this characteristic one would be tempted to say that electing communication paths with nodes closer to one another would be the best way to send a packet. On the other hand, reducing the distance between nodes in a path will result in increasing the number of forwarding nodes which would consume energy for receiving and then transmitting the message along the path. In literature opinions are mixed and while some researchers [8, 112] argue that reducing the number of hops is more efficient others consider that minimizing the communication distance has a greater impact on the overall energy consumption of the network [7, 113].

Due to the weighted definition of the objective function of the genetic algorithm, the impact of choosing the communication path based on either the shortest path or the lowest number of hops (or a combination of the two) can be easily obtained by modifying the weights of the function.

A necessary condition for approaching the above issue is that there should be a considerable amount of nodes placed in the observation area. If this criterion is not satisfied the shortest path may be confused with the one with the lowest number of hops. For this reason we have chosen to perform our simulations on the network with the greatest number of nodes, the one depicted in Figure 5.6. The weights of the objective function have been modified according to the second consideration presented in the description of the GA parameters.

We have compared the results for the three simulation scenarios in terms of network lifetime and the results are available in Figure 5.9.

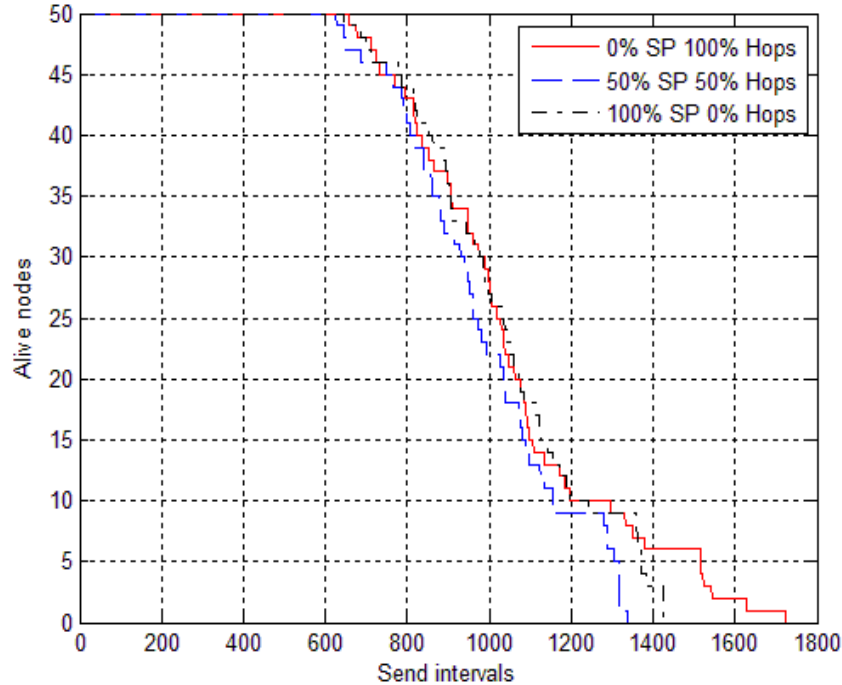


Figure 5.9 Comparison between how network nodes deplete energy for different weights of the objective function

As can be observed from Figure 5.9 there are very small differences between electing the optimum path as the one with the lowest number of hops when compared to the shortest path, or a combination of the two, if we consider the *FND* or the *HND* metrics.

There is however a difference when considering the *LND* metric in the sense that choosing the optimum path as the one with the least number of hops can provide an increase with up to 21% when compared with the shortest path. Our results come to strengthen the split opinions regarding which is the best method for electing the optimum communication path between a source and a destination node.

To conclude our simulation results regarding the best metric for choosing the optimum path we consider that this metric should be dependent on several factors such as the nature of the application, network topology, environment characteristics, size of the observation area, number of nodes in the network, etc..

5.3 A reinforcement learning strategy for task scheduling

In this section of the chapter we will address an important optimization issue for the optimum functionality of mobile WSNs, that of task scheduling.

Throughout literature there are few researches dedicated to scheduling of tasks for WSNs with mobile nodes. Most of these researches are actually focused on networks

with static nodes that have mobile data collectors (MDCs), which are actually a generic name given to mobile base stations [94-95]. The focus of these studies is that of creating efficient schedules for MDCs in order to provide full observation area coverage with the purpose of increasing the number of packets received while avoiding data loss due to buffer overflow [115].

A reference study for scheduling of operations for static WSNs is the S-MAC [15, 16] protocol and its counterpart for networks with mobile nodes MS-MAC [18], both of which are described in Chapter 1.

Briefly, S-MAC [15, 16] reduces the energy consumption of sensor nodes by implementing periodic listen and sleep schedules. It obtains this by dividing the communication frame into a listen period for communication with other nodes and a sleep period during which the node enters the idle mode. The schematic representation of the S-MAC frame is available in Figure 1.7.

The listen period is also divided into three smaller intervals:

- *SYNC* – used for nodes to synchronize schedules
- *RTS* – reserved for sending requests to transmit if data is available
- *CTS* – reserved for receiving data packets from other nodes

Schedule synchronization is maintained through periodic broadcast of *SYNC* messages which contain the address of the sender and the next sleep interval.

The MS-MAC protocols [18] developed for mobile WSNs implements scheduling of operations practically the same as S-MAC with the difference that it considers mobility of nodes by assessing the RSSI from neighboring nodes. In this case *SYNC* packets also contain the maximum estimated speed among all mobile neighboring nodes which is used to maintain connections with old nodes or establish new ones.

There are however several issues which need to be highlighted when using such an approach as above described.

First of all both solutions propose a fixed length frame of communication but neither address the issue of whether or not there are any packets to forward or to send. This means that there is actually no scheduling considered for data acquisition. This aspect is of great importance due to the following two issues:

- Sensors on a node consume a considerable amount of energy which can be equivalent to that of the microcontroller at full capacity of operation (e.g. the case of the IRIS MEMSIC node [92])
- The primary purpose of WSN node is that of measuring some environmental parameter. As most of the applications in which WSNs are used designed for measuring some slow varying phenomenon (such as temperature, gas concentration or the presence of an event, etc.) acquiring information about the environment at fixed intervals (fixed duty cycle) of time may provide redundant measurements.

Another possible drawback which results from using this type of scheduling is that some nodes within the network may never be used as forwarding nodes (e.g. marginal nodes) and therefore it is senseless for them to turn on their receiver during each frame.

A third possible issue which can be identified is that using fixed time frames implies that there should be a very accurate synchronization between the nodes not only for when the frame begins but also for each of the three periods within the time frame.

Considering the above downsides we argue that a different approach can and should be used to obtain scheduling of tasks for sensor nodes. We propose a scheduling scheme based on a reinforcement learning algorithm used to determine the optimal actions that a node should take given the different spatial states through which it passes.

In the remainder of this chapter we will describe the implementation of the proposed scheduling approach as well as present the results obtained through simulations in terms of the election of tasks for different environmental triggers of data acquisition as well as in terms of QoS.

5.3.1 Scheduling of tasks using Q-Learning

Considering the scenario described in section 5.2 of this chapter where mobile nodes move on pseudo-random paths with predefined speeds and at given points in time are supposed to perform some specific task, we can relate this behavior to that of a Markov Decision Process (more on MDPs is available in Appendix E).

As MDPs can be solved through solutions provided by linear or dynamic programming techniques, we have chosen a dynamic approach for addressing the scheduling problem, specifically Reinforcement Learning.

Because RL refers to a vast area of machine learning algorithms concerned with what actions an agent should take in order to maximize a notion of cumulative reward [115], the appropriate methodology to address the scheduling issue should be chosen based on the specifics of the problem at hand. There are basically three classes of methods for solving RL problems, each with its advantages and disadvantages, which we will briefly describe [115]:

- Dynamic programming methods – refer to algorithms that are used to compute optimal policies given a perfect model of the environment. The accuracy of the model required for the implementation of these algorithms translates in a complex and often hard to implement mathematical model which is not suitable for the computational resources of wireless sensor nodes.
- Monte Carlo methods – the application of these methods does not require a complete model of the environment but only a sequence of states, actions and rewards resulting from the interaction with the environment. In other words, these types of methods can be used to obtain optimal behavior without prior knowledge about the dynamics of the environment. They are however not suited for step-by-step incremental computation, but more for continuous or episode-by-episode computations.
- Temporal difference methods – represent a combination of both dynamic programming and Monte Carlo methods in the sense that they do not require prior knowledge of the environment dynamics (as in Monte Carlo) and they are able to provide estimates about the optimum policy to be used

without waiting for the final outcome of the transition between states (as in dynamic programming).

For these reasons and considering the above described assumptions about the mobile WSN we have chosen to address the issue of optimum scheduling using a temporal difference reinforcement learning method known in literature as Q-Learning [116], which we will further describe.

The main idea behind choosing this technique for scheduling is that Q-Learning is actually easy to implement, requires reduced computational resources and most importantly does not require a model of the environment in which the network operates. Considering these qualities we can safely say that the algorithm is ideal for being implemented on sensor nodes with constrained energy and computational resources such as those used in WSNs.

When addressing the issue of scheduling of tasks for WSNs, two possible approaches can be identified. A global approach in which all sensor nodes collaborate in the election of tasks or a local approach where individual nodes elect the appropriate tasks without caring for the overall behavior of the network.

A global approach is usually very difficult to implement as it requires either constant communication between nodes or the presence of a governing entity which dictates individuals what tasks to execute. Using a global approach may induce latency and communication overhead due to extensive message passing between nodes and the central governor. On the other hand, a local approach will provide independence for sensor nodes in electing tasks based on individual constraints as well as allow for a pro-active and real-time adaptation to different environmental and communication scenarios. An important aspect which must be considered when implementing individual scheduling is that of avoiding a selfish behavior which can be adopted by self-governing nodes which can lead to what is known in literature as the Tragedy of the Commons [117]. This phenomenon refers to the situation when individuals act independently and rationally in their self-interest without caring for the overall utility of their actions with respect to others. This behavior can and should be avoided by designing constrained utility functions which should influence nodes in their selection of tasks.

In order to adapt to the requirements of implementing the Q-Learning algorithm for task scheduling of nodes, several adaptations of the simulated scenario are necessary and will be used in the remainder of this chapter.

The mapping of reinforcement learning elements to our problem is as follows:

- **Agent:** each sensor node within the network corresponds to an agent in RL.
- **Environment:** is the world surrounding the agent with which it interacts.
- **Action:** Referring to the application context of the network, by action we refer to the set of tasks that a node can perform during each time period. Actions represent the main element of the RL algorithm as they are subject to scheduling. Examples of possible actions are: sleep, acquire from sensor, transmit, receive, forward, actuate, aggregate etc.

- **States:** In our implementation of Q-Learning we refer to states as the number of possible spatial positions through which each individual agent goes through and for which it is supposed to take a certain action.
- **Policy:** An agent's policy provides a correlation between each individual state and the action which should be taken for that state.
- **Reward function:** Is a mapping (typically a real number) of agent policy to a reward obtained for executing that policy. The agent's goal is to maximize the total reward obtained over time for a certain policy, which will actually represent the best state-action pair that the agent should execute.

Q-Learning as known in literature [116] is a reinforcement learning, model-free, technique based on learning an action-value function which represents the utility of taking an action in a given state. The problem can be modeled as that of an agent found in one of many states ($s_t \in S$) of an environment. For each state s_t , the agent must perform an action ($a_t \in A$) in order to transit to the next state (s_{t+1}). The state-action pair is referred to as a policy $\pi(s_t, a_t)$. Each execution of a policy will provide the agent with a reward based on the successfulness of the policy execution. The goal of the agent is thus to maximize the total reward by learning which action is optimal for each state. Q-Learning uses a look-up table $Q(s, t)$ to store the actual rewards obtained from executing a certain policy at some point in time. Initially the look-up table has all values set to zero. The number of lines of the look-up table is given by the number of possible state the agent can go through while the number of columns is given by the number of actions available. Our definition of the actions performed by an agent and the reward gained after execution are as follows:

- **Sleep** – Microcontroller, radio module and sensor board in sleep mode.
Reward function: $expectedPrice - energyUsed$
- **Route** – The agent simply plays the role of a forwarding node because it has no data to send. Microcontroller and radio module are turned on.
*Reward function: $expectedPrice * noOfMsgRx - energyUsed$*
- **Sample** – The microcontroller is powered and so is the sensor module as measurements are acquired. If the measurements are valid, then the radio module will be powered to send acquired data to the BS.
*Reward function: $expectedPrice * noOfMsgTx - energyUsed$*
- **RouteSample** – Microcontroller, radio and sensor modules are powered. The agent acts as both a router and performs environmental measurements.
*Reward function: $expectedPrice * noOfMsgRx * noOfMsgTx - energyUsed$*

The chosen expected price of each action is available in Table 5.1.

Table 5-1 Expected prices for agent actions

Action	Expected price
Sleep	2
Route	20
Sample	25
RouteSample	30

The value entries of the look-up table for each policy execution are calculated with incremental step updates as given by the following formula [116]:

$$Q(s_t, a_t) = Q(s_t, a_t) + a_t(s_t, a_t) \cdot [R_{t+1} + \gamma \cdot \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (4.1)$$

Which is equivalent to:

$$Q(s_t, a_t) = (1 - a_t(s_t, a_t)) \cdot Q(s_t, a_t) + a_t(s_t, a_t) \cdot [R_{t+1} + \gamma \cdot \max Q(s_{t+1}, a_{t+1})] \quad (4.2)$$

where, $Q(s_t, a_t)$ is the old value of the look-up table, $a_t(s_t, a_t)$ is the learning rate, γ is the discount factor and R_{t+1} is the reward obtained by performing policy $\pi(s_t, a_t)$.

The learning rate $a_t(s_t, a_t) \in (0, 1]$ controls the impact of the reward obtained by the executed policy over previous executions of the same policy. Specifically, setting the learning rate to 0 will make the agent not learn anything from the results of the policy execution while setting it to 1 will make the agent act selfishly and consider only the most recent information.

The discount factor $\gamma \in (0, 1]$ determines the impact of future rewards of the same policy on the current reward. For example, reducing the discount factor to 0 will make the agent act opportunistic and discard future rewards, while setting it to 1 will make the agent strive for a long term high reward.

Throughout the algorithm operation, actions are selected either based on past experiences (exploitation) or using exploration (trial of new actions) but a compromise must be however maintained between the two [116]. Exploration constrains the agent to try out new random actions while exploitation makes use of the already explored policies to maximize the overall agent reward. Most RL based algorithms use exploration with a certain probability ϵ which can be a constant value (usually ranging between 0.1 and 0.5) or can be heuristically chosen by starting with a high value and gradually decreasing. It can be observed that using a high exploration probability, the algorithm will not converge to an optimum state-action pair, while a low exploration probability may determine the convergence to local optimum solutions.

The values used for the three important factors used in implementing Q-Learning are available in Table 5.2.

Table 5-2 Values for Q-Learning factors

Factor	Value	Status
Learning rate (α)	[0.5,1]	Variable
Discount factor (γ)	0.5	Fixed
Exploration rate (ϵ)	[0.3,1]	Variable

Unlike typical implementations of Q-Learning we have chosen variable learning and exploration rates for reasons further described.

In its original implementation Q-Learning will converge to an optimal policy which is actually not desired given the dynamic behavior of typical applications for WSNs.

As the task of the scheduling algorithm is to find the best policy from the energy consumption perspective while maintaining QoS it is most likely that the algorithm will find the sleep action as the best because it will always return a positive reward upon execution. To avoid this we have introduced the notion of event.

An event happens when an action other than sleep is successfully executed for a given state in which the action of sleep has the best Q-table value. In this case the agent should act selfishly ($a_t(s_t, a_t) = 1$) as there is a possibility that the same policy will also be successful the next time the agent passes through that state. The effects of an event (in the sense previously mentioned) on the learning rate ($a_t(s_t, a_t) = 1$) are valid only once, after which it is reset to $a_t(s_t, a_t) = 0.5$.

Using a fixed exploration rate will also give other policies the chance to be executed, however if this threshold is not well chosen (e.g. it is too big) the algorithm may chose sleep as the best action if it has been most successful.

For this reason we have considered that the exploration rate should be the greatest success probability among all actions (except sleep) for a given state, but should not go below 0.3 ($\epsilon \geq 0.3$). Because WSNs may monitor some slow varying parameter, it is necessary for the network to behave accordingly when it occurs, without failure to measure or failed received packets (due to inappropriate election of actions). To address this issue, in our implementation of Q-Learning, the previously mentioned event will trigger a reset in the success rate of a policy (setting it to 1). The policy success will gradually decrease if the selected action is unsuccessful ($\epsilon = 0.3$).

A schematic representation of how the algorithm works is available in Figure 5.10.

In order to validate the impact of the proposed scheduling scheme we have performed Matlab simulations on a network scenario described in the following section.

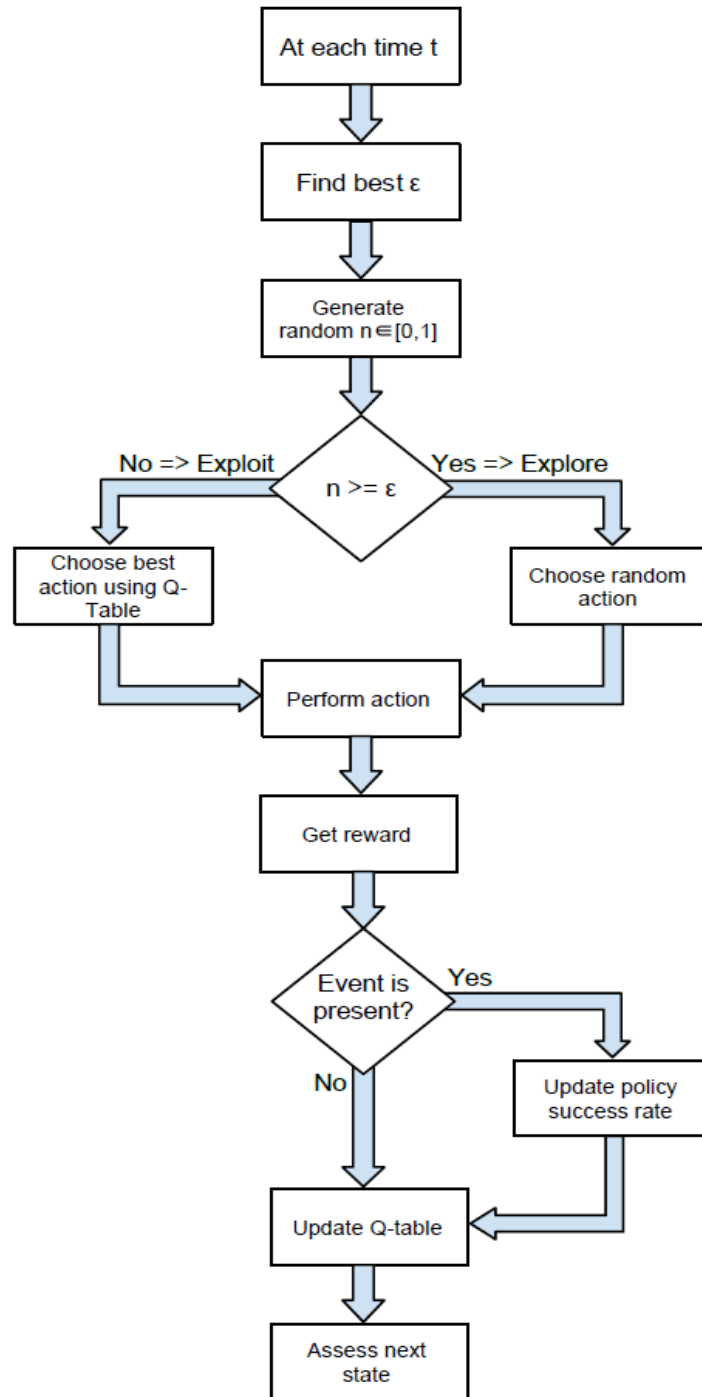


Figure 5.10 Block diagram of the Q-Learning algorithm

5.3.2 Description of the simulation scenario

For the simulation results to be closer to reality, an energy model of a node existent on the market should be taken into consideration. To address this issue we have performed our simulations based on the energy characteristics of the MEMSIC IRIS node [92] which are presented in Table 5-3.

Table 5-3 Energy consumption of IRIS node components [92]

Resource	Operation	Energy [J]
Available Energy	N/A	8640
Microcontroller	Active	32.40
	Sleep	0.43u
Radio	Receive	43.20
	Transmit	64.80
	Sleep	0.10u
Sensor	Active	27
	Sleep	0.20u

The available energy was calculated considering that each sensor node was equipped with two 1.5V, 800 mAh accumulators.

The following considerations were made on network parameters:

- OA size: 300x300 meters
- Number of nodes: 4
- Random user defined paths for each node
- Node parameters:
 - User given speeds (10-14 meters/second) and time interval for communication (2 seconds)
 - Packet size: 56 bytes (16 bytes message preamble + 40 bytes of data according to the MEMSIC specifications [92])
 - Transmitter operation rate: 38400 baud
 - Maximum communication distance between nodes: 100 meters
- The network is considered noise and error free

A graphical representation of the simulation scenario having the above characteristics is available in Figure 5.11.

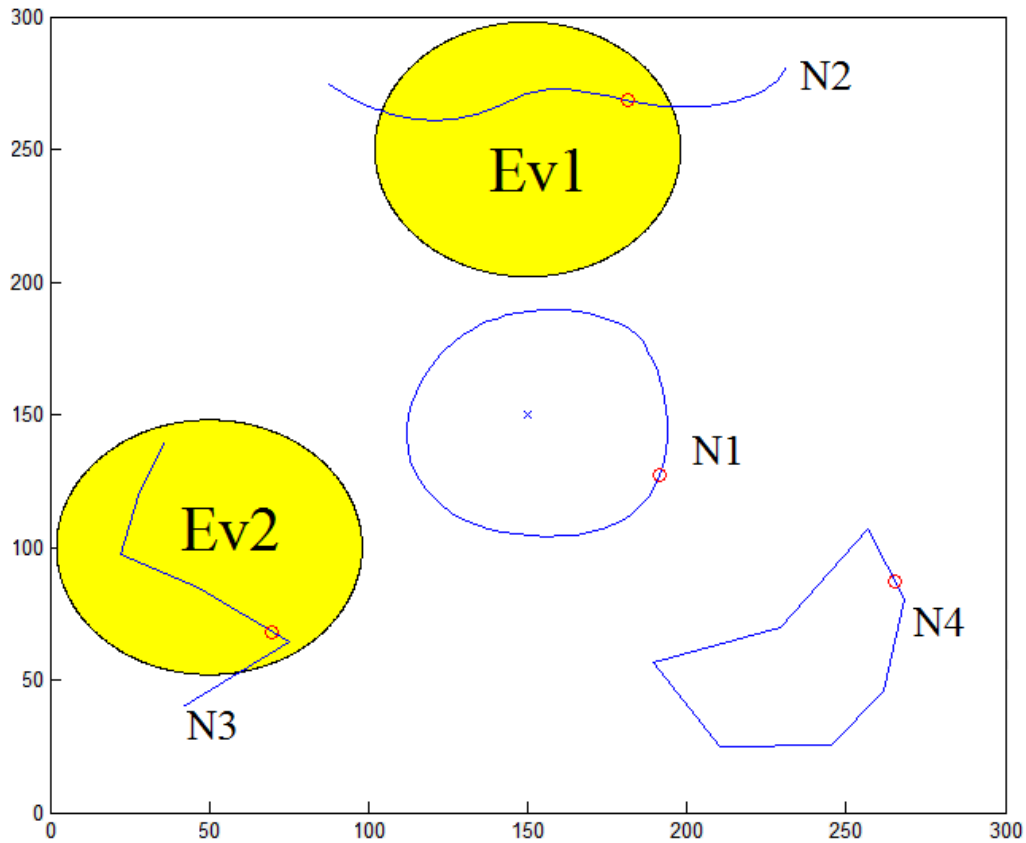


Figure 5.11 Representation of the simulation scenario

Each node position was carefully chosen to determine whether the proposed implementation of Q-Learning behaves as expected, specifically:

- Node 1 is the only one in range of the BS, meaning that its primary task will be that of routing packets.
- Nodes 2, 3 and 4 should perform measurements if an event is present in the OA (here we refer to event as a variation of the measured environmental parameters).

To test the behavior of nodes 2, 3 and 4 we have simulated a scenario in which events take place in the environment that should trigger sensor acquisition and data transmission. Only nodes 2 and 3 are actually subject to such events which are represented in Figure 5.11 by the two circled areas (entitled Ev1 and Ev2). The two events have similar existence characteristics in the sense that both are valid for 1500 rounds but event 1 was triggered at round 1000 while event 2 at round 1500.

5.3.3 Analysis of the simulation results

In this section we will analyze the results of our simulations for the network scenario described above in terms of both the successfulness of the proposed scheduling algorithm and its influence on the overall QoS.

Before we begin the analysis of the simulation results we must also point out that an environmental event is considered to be obsolete for a given state if no variation is detected for 100 consecutive transitions through that state. Taking into account this consideration it is expected that at the beginning of the simulation scenario all nodes must perform some measurement of environmental parameters.

I. Operation scheduling for each individual node

As nodes pass through strategically chosen spatial states and are subject to different environmental stimuli their behavior must be independently addressed to determine the effectiveness of the proposed scheduling algorithm.

We will begin our analysis with node 1 (N1 in Figure 5.11). The overall number of action executions can be observed from Figure 5.12.

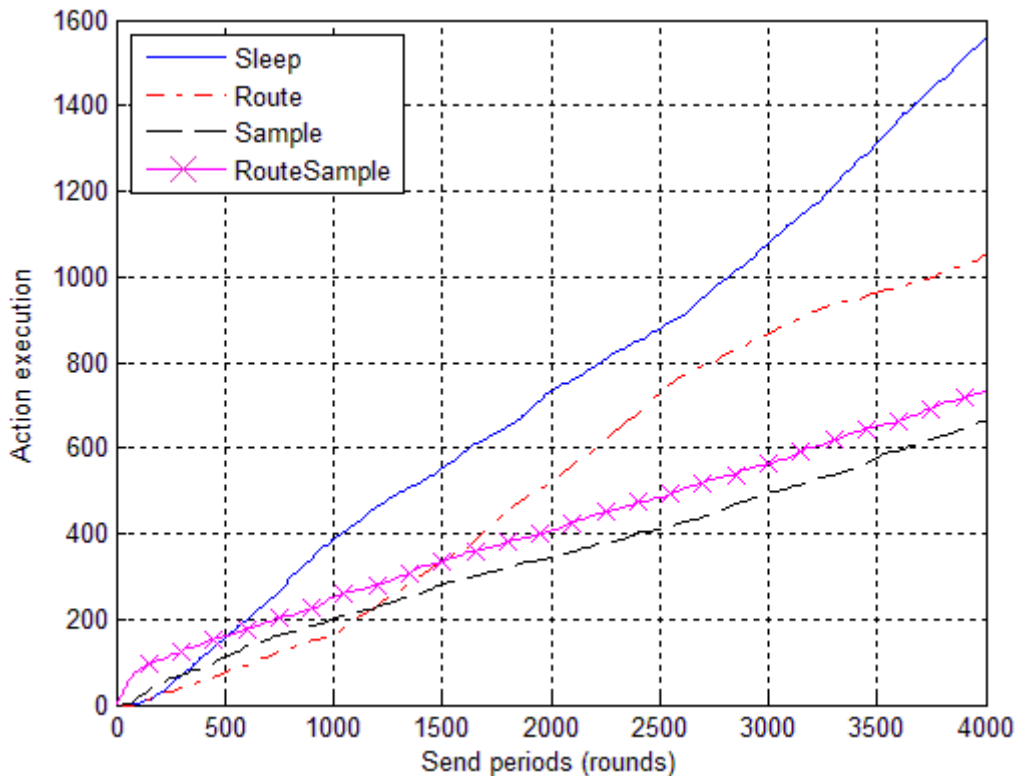


Figure 5.12 Actions executed by node 1

Considering the position and path assigned to node 1 it is expected that this node will act mostly as a router. However, a closer look at the first 500 rounds will reveal that the node behaves as router but also acquires environmental measurements. We can also observe that after some time the node will begin to enter idle mode. This is due to the fact that by now all nodes in the network should have sensed that the measured parameter is stable and therefore no messages are transmitted throughout the network.

A change is triggered by the first event (EV1) at round 1000 which should wake up node 2 (N2) into acquisition and transmission mode. This in turn should set node 1 (N1) to act as a router during the states in which it is within communication distance of node 2. One can observe from Figure 5.12 that node 1 behaves as expected since the number of routing actions increases shortly after 1000 rounds. When the second event (EV2) for node 3 (N3) is triggered at round 1500 the number of routing actions increases further as there are actually two nodes now sending.

Node 1 however still performs a number of sleep actions for some states. This is also expected as node 4 (N4) does not send any packets and given the chosen network topology node 1 is within communication distance with nodes 2 and 3 in approximately 60% of the states.

By comparison with node 1, due to its route in the observation area, node 2 has a different behavior as can be noted from Figure 5.13.

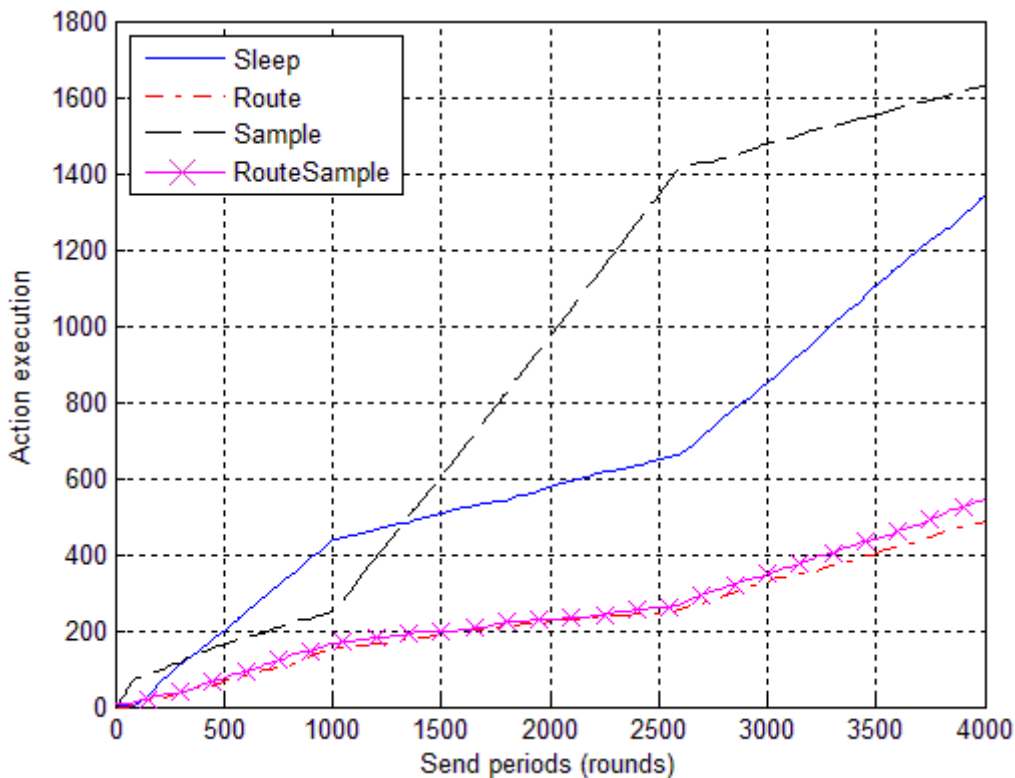


Figure 5.13 Actions executed by node 2

The first difference which can be observed is in the beginning of the network lifetime when the most prominent action executed is that of sensing the environment and transmitting messages to the BS (through node 1). After having sensed the environment the node also begins to go into idle mode but will still perform other actions to maintain the balance between exploration and exploitation previously mentioned. When event 1 is triggered (round 1000) we can observe that the node acts promptly and begins to perform measurements and transmit messages. Due to the extent of the area covered by event 1 which envelops approximately 60% of path taken by node 2 there is an abrupt increase in the number of times the *Sample* action is executed. This behavior ends with the disappearance of event 1, shortly after round 3000 when the sleep action prevails, as can be seen from Figure 5.14.

Since the behavior of node 3 which can be observed in Figure 5.15 is similar to that of node 2 we will only insist on the difference between the two (except that due to the time delay between the two events).

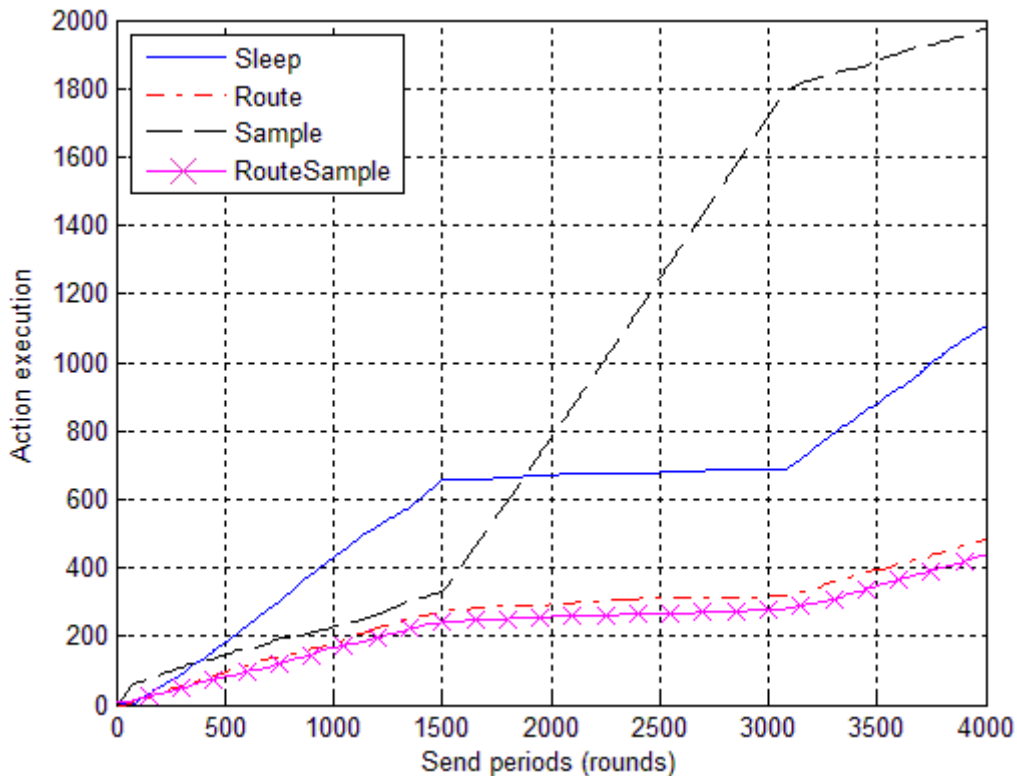


Figure 5.14 Actions executed by node 3

The difference between nodes 3 and 2 is firstly visible in Figure 5.11 from which we can observe that event 2 covers approximately 90% the number of states through which node 3 goes. This means that node 3 should execute considerably more *Sample* actions in detriment of *Sleep* which can also be seen from Figure 5.14 as well (the slopes of the *Sample* and *Sleep* actions are greater compared to those of node 2).

Node 4 was actually placed in the network to test for imperfections which may occur while implementing the algorithm as previously described. The overall number of actions executed by node 4 can be seen in Figure 5.15.

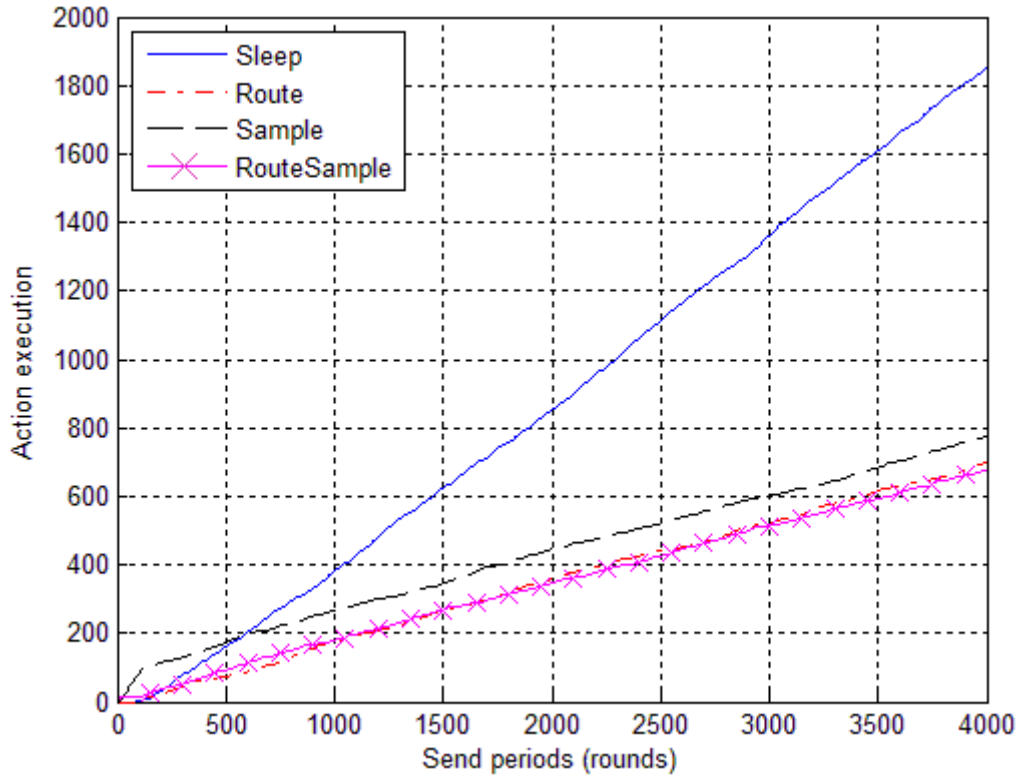


Figure 5.15 Actions executed by node 4

Since node 4 is positioned in such a way as never to be used as a router and because there is no event designated for it that would trigger the execution of the *Sleep* action more often than that given by the action selection method the node should spend most of its time in *Sleep* mode. This actually happens as can be seen from Figure 5.15. The main reason for the existence of this node is to provide proof that the algorithm performs as expected and that there are no sudden executions of other actions without any given reason. This is validated by the absence of spikes in any of the three curves which describe the number of action executions.

Now that we have established that the proposed algorithmic approach for task scheduling performs as expected we will next provide an insight on the impact of the algorithm on network lifetime and QoS.

II. Impact on network lifetime and QoS

Considering that the presence of node 4 in the above simulated scenario is strictly for testing possible misbehaviors of the scheduling algorithm and that its dominant action performed is that of *Sleep*, we will not analyze the energy consumption of this node nor the quality of service.

To get a better understanding about the impact of the proposed scheduling algorithm on the energy consumption of nodes 1, 2 and 3 we have presented in Table 5-4 the number of actions executed by each of the three nodes and their percentage from the total number of actions performed.

Table 5-4 Number of actions executed for nodes 1, 2 and 3

Node Index	Sleep	Route	Sample	RouteSample
1	1556 (39.80%)	1050 (26.25%)	662 (16.55%)	732 (18.30%)
2	1342 (33.55%)	485 (12.12%)	1630 (40.75%)	543 (13.57%)
3	1104 (27.60%)	482 (12.05%)	1975 (49.37%)	439 (10.97%)

Considering the network parameters described above (baud rate and round time) we have also calculated the energy required to perform each action, noting that the values for *Route*, *Sample* and *RouteSample* actions have been calculated for one packet sent or received. The results are available in Table 5-5.

Table 5-5 Energy consumption/action in accordance with node and network parameters

Action	Energy consumption
<i>Sleep</i>	0.162 [mJ]
<i>Route</i>	3.276 [J]
<i>Sample</i>	2.898 [J]
<i>RouteSample</i>	6.174 [J]

We have also calculated the energy consumption of all three nodes while exhaustively performing either of the *Route*, *Sample* or *RouteSample* actions and we have compared the results of both cases in order to determine the energy efficiency of the proposed algorithm in percent, results available in Table 5-6.

Table 5-6 Energy efficiency/exhaustive action [%]

Node Index	Route (%)	Sample (%)	RouteSample (%)
1	24.61	14.78	60.01
2	26.24	16.62	60.86
3	23.58	13.62	59.45

Given that in our simulation scenario we have positioned nodes in such a way as to perform specific tasks we can observe from Table 5-6 that considerable improvements can be achieved. For example, a reduction by 24.6% in energy consumption can be obtained by node 1 using the scheduling algorithm when compared to performing the *Route* action at all times. Also nodes 2 and 3 consume with 14 respectively 16 percent less energy compared to the situation when they would be continuously sampling the environment for changes. However it is important to mention that using the proposed algorithm, nodes actually learn through experimental tryouts what the best action is while in the case of using no scheduling the nodes would most likely be set to work at full capacity (using the

RouteSample action) by comparison to which our scheduling scheme behaves approximately 60% more efficiently.

The question posed however is that of how the algorithm performs in terms of QoS (failed measured events and failed received packets). Our analysis of the simulation results in terms of QoS is available in Figure 5.16.

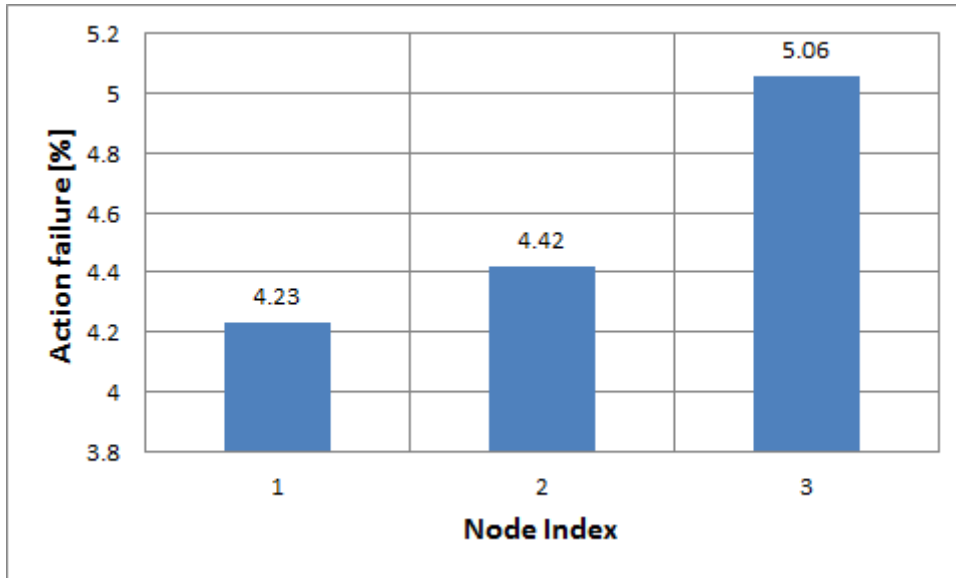


Figure 5.16 Action failure rate [%]

It can be observed from the above figure that the algorithm performs well and that the action failure probability is below 10%. For example, node 1 fails to forward only 4.23 of the packets designated for it while nodes 2 and 3 fail to measure the environment only 4.41 and 5.06 percent of the times an event is present in their range.

5.4 Conclusions

In this chapter we have addressed two issues of high importance for mobile WSNs that of finding optimum routing paths and that of scheduling tasks. We have provided a detailed description about the implementation of the two methods chosen to address the above issues as well as tested them to analyze their performance on various network parameters.

When considering optimum routing for WSNs with mobile nodes we have presented a genetic algorithm approach for finding the best communication paths in terms of either shortest path or the one with the smallest number of hops. We have simulated the proposed method using the Matlab environment in order to test its characteristics. Our simulations have shown that as expected the method has both advantages and disadvantages. Among the downsides that arise from using the propose approach we underline two, specifically high computational requirements

for implementation and path quality. When referring to the latter, our simulations have shown that the described method does not always converge to the optimum path criterion as specified by the objective function and in a worst case scenario only 76% of the optimum paths are found while for the rest the error ranges from 1.6% to 19%. When considering the advantages of the proposed algorithm several can be highlighted such as straightforward implementation, ease of objective function definition but most importantly the time required to find paths as well as the fact that a number of divergent paths result from using the algorithm. When considering convergence speed we have noticed that the maximum time required is of 27 ms/node for 10 paths which means that real time application is possible. The fact that there are more than one paths resulting from applying the proposed algorithm as well as their divergence degree due to the way the algorithm is implemented is a very important aspect as it gives sensor nodes the opportunity to change routes very easily in case of node failure.

The second research direction approached in this chapter is that of scheduling operation executions for sensor nodes with the purpose of increasing the efficiency of energy consumption while maintaining a high QoS. For these reasons we have chosen to use a dynamic programming technique called Q-Learning to address optimum scheduling of tasks. We have explained in detail the implementation of the algorithm as well as tested it through simulations in the Matlab programming environment.

The scheduling technique is easy to implement and use for WSNs with mobile nodes as it requires reduced computational resources and no prior knowledge of the environmental characteristics.

Unlike typical scheduling techniques which require communication between nodes the main feature of the proposed method is that nodes learn through experimental tryouts the best action to perform at different points in space and time.

The results of our simulations have proven that by using the proposed algorithm, sensor nodes are able to quickly adapt to environmental changes. We have noticed a 60% reduction in energy consumption when compared to exhaustively using all available hardware during the communication period. Since each node learns what action to perform during each of its states, the question posed is how fast the adaptation to environmental changes is performed and what the impact is on the QoS of the network. Our simulations show that implementing the learning algorithm with the specified parameter variations only a small number of events are overlooked (4 to 5%) which means that the algorithm also performs well in terms of the quality of service provided.

5.5 Contributions

- 1. Proposal of a genetic algorithm based method for electing the optimum communication path for mobile WSNs.**
- 2. Validation through simulations of the proposed method with the purpose of determining its applicability degree to mobile WSNs**
- 3. A reinforcement learning technique (Q-Learning) for scheduling of tasks for wireless networks with mobile nodes**
- 4. Proposal of algorithm parameter variation and the introduction of the notion of "event" to enhance the performance of the algorithm**
- 5. Validation of the proposed method through simulations and analysis in terms of network lifetime and QoS**

Chapter 6

Conclusions and contributions

The focus of this thesis is that of providing general optimization criteria regarding the efficient usage of wireless sensor networks which can be safely applied to most applications and environments. Three major research directions are targeted, radio frequency energy harvesting, efficient clustering and routing for networks with static nodes and route optimization and scheduling for networks with mobile nodes. The following sections present a brief overview of the theses as well as conclusions for each chapter, a summary of contributions and future research directions.

6.1 Thesis overview and conclusions

The first chapter entitled "**Wireless sensor networks: applications and challenges**" targets to introduce the user into the applications, architectures and protocols used for WSNs with the purpose of identifying possible research directions, some of which represent the basis of this thesis.

A thorough classification of protocols that address efficient energy consumption is presented with focus on routing protocols for networks with both static and mobile nodes. A detailed description of the most representative protocols is presented as they will serve as comparison directions in other chapters of the thesis.

As the main issue for WSNs which also represents the target for developed protocols is that of energy consumption, the introductory chapter also describes techniques used to harvest energy from various environmental sources concentrating on radio frequency energy harvesting. The Powercast P2110-EVAL-01 Energy Harvesting Kit is briefly described as it will serve as reference for comparing a proposed mathematical model, detailed in Chapter 2, in order to determine whether prior knowledge can be obtained about the amount of energy available from using this harvesting technique.

The second chapter of the thesis, "**Performance analysis and modelling of a RF energy harvester**", addresses the issue of radio frequency energy harvesting for wireless sensor networks. The purpose of this chapter is to provide a mathematical model that can easily be implemented in a simulation environment to obtain information about the capabilities of a given energy harvester. The chapter provides a detailed description about the equations that describe the mathematical model and also about the experimental setup which was devised in order to test the functionality of the proposed model.

Conclusions: As a result of the work described in the second chapter, an experimental setup proposal emerges which can be used to analyze the performance of radio frequency energy harvesters. The development of a low current measurement system is described which together with a data acquisition system and associated programming in the National Instruments LabView environment can be used for determining the harvesting characteristics of RF harvesters. The experimental setup was built with the purpose of validating a

mathematical model which based on the characteristics of the sending and receiving antennas can be used for simulating the behavior of such harvesters. Through in field measurements using the experimental setup we have validated the proposed model and we have shown that it can be safely used to predict the harvesting capabilities of RF energy harvesters within 2dBm dispersion.

Chapter three of the thesis "**A graphical user interface for simulating WSNs**", describes a graphical user interface developed with the purpose of giving any user the possibility to define unique topologies for simulating WSNs but also to test the behavior of the simulated network using the protocols devised in Chapters 4 and 5. Another distinctive feature of the developed simulation environment is that it can be easily adapted to suite the development of new protocols. It is however a work in progress and future efforts are necessary before it can be turned into a veritable wireless sensor network simulator.

Conclusions: The development for a graphical user interface which would help simulate the behavior of WSNs is necessary because it gives users the possibility to define network simulation scenarios with unique characteristics while providing access to all statistical information available without having to interpret purely mathematical results. The proposed simulation environment provides dynamic adaptation to various simulation scenarios and network topologies as properties such as observation area size, number of nodes, node hardware characteristics etc. can be easily modified to suite application needs. Another important feature is the availability to access distinct statistical information which can be viewed directly in the simulator or saved in files for later usage.

Chapter four, "**Efficient clustering techniques for static WSNs**", focuses on efficient routing techniques for WSNs meant to reduce the energy consumption while maintaining increased quality of service. This chapter addresses two research directions for hierarchical networks, that of efficient cluster head election and that of route optimization between network nodes of different hierarchical status.

The first direction targets to provide an energy efficient method for CH election with the purpose of obtaining even coverage of the observation area. The proposed method implies that cluster heads must be elected considering that a minimum separation distance exists between them but also taking note of the number of nodes in their vicinity and of the expected packet size to be delivered. Using the proposed method nodes in key places throughout the network will be constrained to perform the difficult task of cluster head in the detriment of others. Using this method has a positive impact on the overall network lifetime and also provides increased packet delivery.

Based on a thorough analysis of literature researches together with results obtained through experimental tryouts, the second research direction proposes a new method for network nodes to elect attending cluster heads. The method implies the usage of what is referred to as a "reconnaissance procedure" in which sensor nodes broadcast messages to neighbors in a TDMA fashion. At the end of this procedure nodes should be able to assess the link quality between themselves and all nodes within the communication distance. Based on this link indicator nodes will elect future communications paths rather than using other metrics such as RSSI. We have shown through simulations that networks using this method can benefit from reduced energy consumption and increased packet throughput.

Conclusions: The two directions presented in chapter three of this thesis provide energy efficient solutions for hierarchical organization and routing of wireless sensor networks with static nodes. Both proposals have been validated through numerous simulations in the Matlab environment. The simulations have addressed different scenarios in terms of size of the observation area, number of cluster heads, packet size etc. Analysis of the simulation results show that by implementing the proposed methods important improvements can be obtained in terms of network lifetime and packet delivery (various percentual increases depending on the size of the observation area as well as on the number of desired cluster heads). Also, the straightforwardness of the proposed methods makes them easy to implement and test on a real WSN.

The last chapter of this thesis entitled "**Routing and scheduling techniques for mobile WSNs**" aims at providing routing and scheduling techniques for wireless sensor networks with mobile nodes. We present the implementation of a genetic algorithm approach for efficient path selection. We provide a detailed description of how the method was implemented as well as a description of the simulation environment for which it was tested. The results of our simulations show that the method is prone to errors when considering the quality of the provided paths but this issue can be solved by modifying the parameters of the algorithm accordingly.

When considering task scheduling, we propose a method based on Markov decision process called Q-Learning. The learning algorithm was implemented by adding the notion of "event" and using own definitions of parameter variations. Through extensive simulations we have shown that the method performs well and can be easily used for training WSN nodes what actions to perform at given periods in time.

Conclusions: The research directions approached in this chapter are addressed to wireless sensor networks with mobile nodes and they are of great importance as they refer to issues such as routing and scheduling of tasks.

The routing method described in this chapter which is based on a genetic algorithm approach solves, with a reduced error rate the problem of choosing optimum paths given that an interconnection table between nodes exists. We have observed through simulations that the time required for the algorithm to converge to an optimum path is less than 27 ms/node which allows for usage on systems with real time response requirements. Another benefit from using this method is the fact that it provides more than one divergent path from a source to a destination node which allows for quick adaptation in case of node failure.

The scheduling method described in presents several advantages such as ease of implementation at reduced computational costs and efficient usage of hardware resources while maintaining high quality of service. The simulation results show that this method can be easily implemented and will prove effective using the proposed constrains on algorithm parameter variation and the consideration of the "event" as described and its impact on the functionality of the algorithm. We have shown that using the proposed scheduling scheme an increase in network lifetime with up to 60% can be obtained while maintaining the QoS high, with losses below 6%.

6.2 Summary of contributions

Among the contributions resulting from this thesis the author would like to point out seven of the most important:

- 1. A mathematical model for estimating the energy harvesting capabilities of radio frequency harvesters.**
- 2. An experimental setup which can be used to analyze the performance of radio frequency energy harvesters.**
- 3. The foundations of a robust network simulator for static and mobile wireless sensor networks developed in the Matlab environment.**
- 4. Proposal of a new method for static networks to electing cluster heads using a separation distance and considering network topology characteristics as well as the expected size of the delivered packets.**
- 5. Proposal of a new method for networks with static nodes to choose attending cluster heads based on information about the link quality obtained through the implementation of a reconnaissance procedure.**
- 6. A genetic algorithm based approach for generating divergent paths while maintaining the applicability constraint for real time applications.**
- 7. An adapted Markov decision process technique (Q-Learning) for opportunistic and efficient scheduling of node tasks while maintaining a high quality of service.**

Bibliography

- [1] J. Zheng, A. Jamalipour, "Wireless Sensor Networks A Networking Perspective", Institute of Electrical and Electronics Engineers (IEEE), John Wiley & Sons Inc., Hoboken, New Jersey, ISBN: 978-0-470-16763-2, 2009
- [2] G. Pottie, W. Kaiser, "Wireless integrated sensor networks (WINS)", Communications of the ACM, Vol. 5, 2000, pp. 51-58
- [3] J. Panchard, S. Rao, T. V. Prabhakar, H. S. Jamadagni, J.-P. Hubaux, "Common-sense net: Improved water management for resource-poor farmers via sensor networks", Proc. IEEE International Conference on Information and Communication Technologies and Development (ICTD 2006), 2006, pp. 22-33.
- [4] C. P. Fritzen, "Vibration-based Structural Health Monitoring – Concepts and Applications", Key Engineering Materials Vols. 293-294, 2005, pp. 3-20
- [5] S. Kim, "Wireless sensor networks for structural health monitoring", Master dissertation, University of California at Berkeley, 2005
- [6] M. Winkler, K. D. Tuchs, K. Hughes, G. Barclay, "Theoretical and practical aspects of military wireless sensor networks", Journal of Telecommunications and Information Technology (JTIT), Vol. 2, 2008, pp. 37-45
- [7] M. Haenggi, "Routing in ad hoc networks: A case for long hops", IEEE Communications Magazine, Vol. 43, Issue 10, 2004, pp. 93-101
- [8] S. Fedor, M. Collier, "On the problem of energy efficiency of multi-hop vs. one-hop routing in Wireless Sensor Networks", 21th International Conference on Advanced Information Networking and Applications Workshop (AINAW'07), 2007, pp. 380-385
- [9] U. M. Pesovic, J. J. Mohorko, K. Benkic, Z. F. Cucej, "Single-hop vs. Multi-hop – Energy Efficiency Analysis in Wireless Sensor Networks", 18 Telecommunications Forum (TELFOR'10), Belgrade, Serbia, 2010, pp. 471-474
- [10] Y. Li, S. S. Panwar, S. Mao, "A wireless biosensor network using autonomously controlled animals", IEEE Network, Vol. 20, No 3, May/June 2006, pp. 6-11
- [11] L. B. Ruiz, J. M. Nogueira, A. A. F. Loureira, "Sensor network management", SMART DUST: Sensor Network Applications, Architecture and Design, CRC Press, Boca Raton, FL, 2006
- [12] IEEE 802.15.4, <http://standards.ieee.org/getieee802/download/802.15.4-2011.pdf>, Last visited on 28.12.2012

-
- [13] Zigbee standard, <http://www.zigbee.org/Standards/Overview.aspx>, Last visited on 28.12.2012
- [14] I. Demirkol, C. Ersoy, F. Alagoz, "MAC protocols for wireless sensor networks: A survey", *IEEE Communications Magazine*, April 2006, pp. 151-121
- [15] W. Ye, J. Heidemann, D. Estrin, "An energy efficient MAC protocol for wireless sensor networks", *Proceedings of IEEE INFOCOM'02*, New York, 2002, pp. 1567-1576
- [16] W. Ye, J. Heidemann, D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks", *IEEE/ACM Transactions on Networking*, Vol. 12, No. 3, June 2004, pp. 493-506
- [17] P. Lin, C. Qiao, X. Wang, "Medium access control with a dynamic duty cycle for sensor networks", *IEEE Wireless Communications and Networking Conference (WCNC'04)*, Vol. 3, 2004, pp. 1534-1539
- [18] H. Pham, S. Jha, "An adaptive mobility-aware MAC protocol for sensor networks (MS-MAC)", *Proceedings of 2004 International Conference on Mobile Ad Hoc and Sensor Networks (MASS'04)*, Oct. 2004, pp. 558-560
- [19] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks", in *Proceedings of 1st ACM Conference on Embedded Network Sensor Systems (SenSys'03)*, Los Angeles, CA., Nov 2003, pp. 181-192
- [20] K. Sohrabi, J. Gao, V. Ailawadhi, G. J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network", *IEEE Personal Communications*, 2000, pp. 16-17
- [21] I. Rhee, A. Warrier, M. Aia, J. Min, "Z-MAC: a hybrid MAC for wireless sensor networks", in *IEEE/ACM Transactions on Communication, Networking & Broadcasting; Computer & Processing*, 2008, pp. 511-524
- [22] G-S. Ahn, E. Miluzzo, A. T. Campbell, S. G. Hong, F. Cuomo, "Funneling-MAC: A localized, sink-oriented MAC for boosting fidelity in sensor networks", in *Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys'06)*, Boulder, Colorado, Nov. 2006, pp. 236-306
- [23] C. Y. Wan, S. E. Eisenman, A. T. Campbell, J. Crowcroft, "Siphon: Overload traffic management using multi-radio virtual sinks", in *Proceedings of 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys'05)*, San Diego, CA., Nov 2005, pp. 116-129
- [24] N. Bulusu, J. Heidemann, D. Estrin, "GPS-less low cost outdoor localization for very small devices", *IEEE Personal Communication Magazine*, Vol. 7, No. 5, Oct. 2000, pp. 28-34
- [25] R. Alasem, A. Reda, M. Mansour, "Location Based Energy-Efficient Reliable Routing Protocol for Wireless Sensor Networks", *Recent Researches in Communications, Automation, Signal processing, Nanotechnology, Astronomy and Nuclear Physics*, WSEAS Press, Cambridge, UK, 2011, ISBN: 978-960-474-276-9

- [26] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, „Energy-efficient communication protocol for wireless microsensor networks“, Proceedings of the 33rd International Conference on System Sciences (HICSS '00), January 2000, pp. 1-10
- [27] W. Bo, H. Han-ying, F. Wen, “An Improved LEACH Protocol for Data Gathering and Aggregation in Wireless Sensor Networks“, International Conference on Computer and Electrical Engineering (ICCEE 2008), 2008, pp. 398-401
- [28] L. Han, „LEACH-HPR: An energy efficient routing algorithm for Heterogeneous WSN“, IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2010), 2010, pp. 507-511
- [29] G. K. Santhosh, C. P. Vinu, J. K. Poullose, “Mobility Metric Based LEACH-Mobile Protocol“, 16th International Conference on Advanced Computing and Communications (ADCOM 2008), 2008. pp. 248-253
- [30] A. G. A. Elrahim, H. A. Elsayed, S. H. Elramly, M. M. Ibrahim, “A new Routing Protocol for Mobility in Wireless Sensor Networks“, Online publication: Journal of Selected Areas in Telecommunications (JSAT), February, 2011
- [31] D. B. Johnson, D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks“, Mobile Computing, Kluwer Academic Publishers, 1996
- [32] C. Perkins, E. Belding-Royer. S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing“, RFC Editor, 2003
- [33] B. Karp, H. T. Hung, “GPRS: Greedy perimeter stateless routing for wireless networks“, Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom'00), 2000, pp. 243-254
- [34] R. C. Prim, “Shortest connection networks and some generalizations“, Bell System Technical Journal, 36, 1957, pp. 1389–1401
- [35] C. Y. Wan, A. T. Campbell, L. Krishnamurthy, “PSFQ: a reliable transport protocol for wireless sensor networks“, Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications, 2002, pp. 1-11
- [36] J. V. Greunen, J. Rabaey, “Lightweight time synchronization for sensor networks“, in Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications, (WSNA'03), San Diego, California, Sept. 2003, pp. 11-19
- [37] J. Degesys, I. Rose, A. Patel, R. Nagpal, “DESYNC: Self organizing desynchronization and TDMA on wireless sensor networks“, in Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN'07), Cambridge, MA, Apr. 2007, pp. 11-20
- [38] U-J. Jang, S-G. Lee, J-Y. Park, S-J. Yoo, “Fault-Tolerant WSN Time Synchronization“, Wireless Sensor Networks Journal, Vol. 2, No. 10, 2010, pp. 739-745

-
- [39] C. J. Chen, "Physics of Solar Energy", John Wiley and Sons, ISBN-10: 0470647809, 2011, pp. 22-25
- [40] X. Jiang, J. Polastre, D. Culler, "Perpetual Environmentally Powered Sensor Networks", In Fourth International Symposium on Information Processing in Sensor Networks (IPSN'05), April 2005, pp. 463-468
- [41] F. Simjee, P. H. Chou, "Everlast: Long-life, Supercapacitor-operated Wireless Sensor Node", In Proceedings of the 2006 International Symposium on Low Power Electronics and Design, 2006, pp. 197-202
- [42] A. Kansal, J. Hsu, S. Zahedi, M. B. Srivastava, "Power Management in Energy Harvesting Sensor Networks", ACM Transactions on Embedded Computing Systems (TECS), Vol. 6, Issue 4, Article 32, 2007
- [43] P. Zhang, C. M. Sadler, S. A. Lyon, M. Martonosi, "Hardware Design Experiences in ZebraNet", In Proceedings of the Second International Conference on Embedded Net-worked Sensor Systems, 2004, pp. 227-238
- [44] D. Musiani, K. Lin, T. S. Rosing, "An Active Sensing Platform for Wireless Structural Health Monitoring", In Proceedings of the Sixth ACM International Conference on Information Processing in Sensor Networks, 2007, pp. 390-399
- [45] J. A. Paradiso, M. Feldmeier, "A Compact, Wireless, Self-Powered Pushbutton Controller", In Proceedings of the 3rd International Conference on Ubiquitous Computing, 2001, pp. 299-304
- [46] A. Anders, "Temperature Difference as Energy Source", EnOcean's Perpetuum Magazine, Vol. 8, Issue 1, ISSN 1862-0698, 2011
- [47] EnOcean home page, <http://www.enocean.com/en/home/>, last visited on 02.01.2013
- [48] H. T. Friis, "A note on a simple transmission formula," Proc. IRE 34, 1946, pp. 254-256
- [49] T. Le, K. Mayaram, T. Fiez, "Efficient Far-Field Radio Frequency Energy Harvesting for Passively Powered Sensor Networks," IEEE Journal of Solid-State Circuits, Vol. 43, Issue 5, 2008, pp. 1287-1302
- [50] Powercast P2110 - 915 MHz RF Powerharvester Receiver, Product Datasheet Rev A, 2010, <http://www.powercastco.com/PDF/P2110-datasheet.pdf>, Last visited on 19.09.2012
- [51] Powercast P2110-EVAL-01 Energy Harvesting Kit for Wireless Sensors, User's Manual, 2010, <http://www.powercastco.com/PDF/P2110-EVAL-01-manual.pdf>, Last visited on 19.09.2012
- [52] Powercast TX91501 - 915 MHz Transmitter User's Manual. 2010 Rev A, <http://www.powercastco.com/PDF/TX91501-manual.pdf>, Last visited on 19.09.2012

- [53] **C. Cirstea**, "Energy Efficient Routing Protocols for Wireless Sensor Networks: A Survey", IEEE 17th International Symposium for Design and Technology in Electronic Packaging (SIITME'11), 2011, pp. 277-282
- [54] K. Finkenzeller, "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification", 2nd ed. Chichester, Sussex, U.K, Wiley, 2003, ISBN 0-470-84402-7
- [55] T. Lehmann, Y. Moghe, "On-chip active power rectifiers for biomedical applications", IEEE International Symposium on Circuits and Systems (ISCAS'05), Japan, 2005, Vol. 1, pp. 732-735
- [56] **C. Cirstea**, M. Cernaianu, A. Gontean, "An Inductive System for Measuring Microampere Currents", IEEE 18th International Symposium for Design and Technology in Electronic Packaging (SIITME'12), Alba Iulia, Romania, 2012, pp. 197 - 200
- [57] LTS-15 NP Current Transducer, Product Datasheet, <http://www.lem.com/docs/products/lts%2015-np.pdf>, Last visited on 19.09.2012
- [58] Analog Devices AD8230 16V Rail-to-Rail, Zero-Drift, Precision Instrumentation Amplifier, Product Datasheet, http://www.analog.com/static/imported-files/data_sheets/AD8230.pdf, Last visited on 19.09.2012.
- [59] National Instruments 6251 multifunction data acquisition system <http://sine.ni.com/nips/cds/view/p/lang/ro/nid/202802>, Last visited on 19.09.2012
- [60] Texas Instruments μ A723 CN Precision Voltage Regulators, Product Datasheet, <http://www.ti.com/lit/ds/symlink/ua723.pdf>, Last visited on 19.09.2012
- [61] C. A. Balanis, "Antenna Theory – Analysis and Design", Third Edition, John Wiley & Sons, Inc, ISBN: 0-471-66782-X, 2005
- [62] A. Ignea, "Compatibilitate Electromagnetica" (English title: "Electromagnetic Compatibility"), Editura de Vest, ISBN 978-973-36-0453-2, 2007, pp. 55
- [63] T. Petrița, "Approximation of antenna diagram for BTS antennas", in Proceedings of the 34th International Conference on Telecommunications and Signal Processing (TSP'11), Budapest, Hungary, 2011, pp. 257 - 260
- [64] T. Petrița, "Contributii la modelarea unor caracteristici ale antenelor", PhD Thesis, Editura Politehnica Timisoara, ISBN: 978-606-554-567-0, 2012
- [65] W.L. Stutzmann, "Estimating Directivity and Gain of Antennas", IEEE Antennas and Propagation Magazine, Vol. 40, No.4, 1998, pp. 7-11
- [66] R.P. Singh, M.P. Lal, T. "Laboratory measurement of dielectric constant and loss tangent of Indian rock samples", Annals of Geophysics, Vol. 33, No.1,1980, pp. 111-140

- [67] B. Zhang, Y. Zhong, H. Liu, F. Wang, "Experimental Research on Dielectric Constant Model for Asphalt Concrete Material", *Advanced Materials Research*, Vols. 250-253, 2011, pp. 2760-2764
- [68] J. Baker-Jarvis, M. D. Janezic, R. F. Riddle, R. T. Johnk, P. Kabos, C. Holloway, R.G. Geyer, C.A. Grosvenor, "Measuring the Permittivity and Permeability of Lossy Materials: Solids, Liquids, Metals, Building Materials, and Negative-Index Materials," NSIT technical note, 2005, pp. 142
- [69] A. Ignea, E. Marza, A. De Sabata, "Antene si Propagare" (English title: "Antennas and Propagation"), Editura de Vest, ISBN 973-36-0351-1, 2002
- [70] T. Macnamara, "Introduction to Antenna Placement and Installation", John Wiley and Sons, ISBN: 978-0-470-01981-8, 2010
- [71] T. Petrita, "Comparison of two approximation models for near-field of BTS antennas", in *Proceedings of the 35th International Conference on Telecommunications and Signal Processing (TSP'12)*, Prague, Czech Republic, 2012, pp. 330-334
- [72] ISO 31-11:1992 Standard: Quantities and Units -- Part 11: Mathematical signs and symbols for use in the physical sciences and technology
- [73] H. Sundani, H. Li, V. Devabhaktuni, M. Alam, P. Bhattacharya, "Wireless Sensor Network Simulators: A Survey and Comparisons", *International Journal Of Computer Networks*, Vol. 2, Issue 5, 2010
- [74] M. Jevtic, N. Zogovic, G. Dimic, "Evaluation of Wireless Sensor Network Simulators", 17th *Telecommunications Forum (TELFOR'09)*, Belgrade, Serbia, 2009, pp. 1303-1306
- [75] Ns-2 Simulator, <http://www.isi.edu/nsnam/ns/>, Last visited on 27.01.2013
- [76] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications", 1st *ACM Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, Nov. 2003, pp. 126-137
- [77] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: A library for parallel simulation of large-scale wireless networks", *SIGSIM Simulation Digest*, Vol. 28, No. 1, 1998, pp. 154-161
- [78] S. Park, A. Savvides, and M. B. Srivastava, "Simulating networks of wireless sensors," *Winter Simulation Conference*, Arlington, Virginia, Dec. 2001, pp. 1330-1338
- [79] G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi "Simulation-based optimization of communication protocols for large-scale wireless sensor networks," *IEEE Aerospace Conference Big Sky*, Mar. 2003, pp. 1339-1346
- [80] E. Hansen, J. Neander, M. Nolin, and M. Björkman, "Energy-efficient cluster formation for large sensor networks using a minimum separation distance", in *The Fifth Annual Mediterranean Ad Hoc Networking Workshop*, June 2006

- [81] A. R. Chalak, S. Misra, M. S. Obaidat, "A cluster-head selection algorithm for Wireless Sensor Networks", 17th IEEE International Conference on Electronic Circuits and Systems, ICECS'10, 2010, pp. 130-133
- [82] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "An application-specific protocol architecture for wireless sensor networks", in press: IEEE Transaction on Wireless Networking, October 2002, pp. 660-670
- [83] A. Vlavaianos, L. K. Law, I. Broustis, S. V. Krishnamurthy, M. Faloutsos, "Assessing Link Quality in IEEE 802.11 Wireless Networks: Which is the Right Metric?", IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'08), 2008, pp. 1-6
- [84] **C. Cirstea**, M. Cernaianu, A. Gontean, "A Cluster Head Election Method with Adaptive Separation Distance and Load Distribution for Wireless Sensor Networks", The 4th International Conference on Emerging Network Technologies (EMERGING'12), Barcelona, Spain, 2012, pp. 37-42
- [85] C. Savarese, J. M. Rabaey, J. Beutel, "Locationing in distributed ad-hoc wireless sensor networks", In Proceedings of the IEEE Signal Processing Society International Conference on Acoustics, Speech, and Signal Processing 2001 (ICASSP '01), Vol. 4, Salt Lake City, UT, May 2001, IEEE, New York, pp. 2037-2040
- [86] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for IEEE 802.11 wireless networks", First Annual International Conference on Broadband Networks, BroadNets'04, October 2004, pp. 344-354
- [87] S. Keshav, "A Control-theoretic approach to flow control", In Special Interest Group on Data Communication, SIGCOMM, September 1991, pp. 3-15
- [88] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "High-throughput path metric for multi-hop wireless routing", Annual International Conference on Mobile Computing and Networking, MOBICOM'03, September 2003, pp. 419-434
- [89] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks", In Technical Report UCLA/CSD-TR 02-0013, Computer Science Department, UCLA, July 2002
- [90] J. Zhao and R. Govindan, "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks", Proceedings of the 1st international Conference on Embedded Networked Sensor Systems (SenSys'03), 2003, pp. 1-13
- [91] **C. Cirstea**, M. Cernaianu, A. Gontean, "Packet Loss Analysis in Wireless Sensor Network Routing Protocols", In the 35th International Conference on Telecommunications and Signal Processing (TSP'12), Prague, Czech Republic, 2012, pp. 37-41
- [92] MEMSIC IRIS Node Datasheet: <http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html>, Last visited on 22.01.2013

-
- [93] Philip Levis, "TinyOS/nesC Programming Reference Manual", January 2006.
- [94] Y. Gu, D. Bozdogan, E. Ekici, F. Ozguner, C. G. Lee, "Partitioning based mobile element scheduling in wireless sensor networks", Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2005, pp. 386-395
- [95] S. Gao, H. Zhang, S. K. Das, "Efficient Data Collection in Wireless Sensor Networks with Path-Constrained Mobile Sinks", IEEE Transactions on Mobile Computing, Vol. 10, No. 4, 2011, pp. 592-608
- [96] W. Stallings, "High-Speed Networks TCP/IP and ATM Design Principles", Englewood Cliffs, New Jersey: Prentice Hall, ISBN-10: 0135259657, 1998
- [97] C. W. Ahn, R. S. Ramakrishna, C. G. Kang, I. C. Choi, "Shortest path routing algorithm using Hopfield neural network", Electronics Letters, Vol. 37, No. 19, 2001, pp. 1176-1178
- [98] C. W. Ahn, R. S. Ramakrishna, "A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations", IEEE Transactions on Evolutionary Computation, Vol. 6, No. 6, 2002, pp. 566-579
- [99] G. Tufte, P. C. Haddow, "Prototyping a GA pipeline for complete hardware evolution", in Proceedings of the 1st NASA/DoD Workshop on Evolvable Hardware, 1999, pp. 76-84
- [100] O. Roeva, "Real-World Applications of Genetic Algorithms", Publisher: InTech, ISBN 978-953-51-0146-8, 2012
- [101] J. Inagaki, M. Haseyama, H. Litajima, "A genetic algorithm for determining multiple routes and its applications", in Proceedings of the 1999 IEEE International Symposium on Circuits and Systems (ISCAS'99), Vol. 6, 1999, pp. 137-140
- [102] J. Moy, "Open shortest path first protocol", Standards Track Network Working Group RFC 1583, March 1994
- [103] C. E. Perkins, P. Bhagwat, "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers", Proceedings of the Conference on Communications Architectures, Protocols and Applications (SIGCOMM'94), 1994, pp. 234-244
- [104] S. Murthy, J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks", ACM Journal of Mobile Networks Applications, 1996, pp.183-197
- [105] X. Hue, "Genetic algorithms for optimization: Background and applications", Edinburgh Parallel Computing Centre, University of Edinburgh, Scotland, Ver. 1.0, Feb. 1997
- [106] P. A. Diaz-Gomez, D. F. Hougen: "Initial Population for Genetic Algorithms: A Metric Approach", Proceedings of the 2007 International Conference on Genetic and Evolutionary Methods, GEM'07, Las Vegas, Nevada, USA, June 25-28, pp. 43-49, 2007

- [107] B. L. Miller, D. E. Goldberg, "Genetic algorithms, selection schemes, and the varying effects of noise", *Journal of Evolutionary Computation*, MIT Press Cambridge, MA, USA, Vol. 4 Issue 2, 1996, pp. 113-131
- [108] B. L. Miller, D. E. Goldberg, "Genetic algorithms, tournament selection and the effects of noise", *Complex Systems*, Vol. 9, 1995, pp. 193-212
- [109] W. M. Spears, V. Anand, "A Study of Crossover Operators in Genetic Programming", *Proceedings of the 6th International Symposium on Methodologies for Intelligent Systems (ISMIS '91)*, 1991, pp. 409-418
- [110] F. B. Zhan, "Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures", *Journal of Geographic Information and Decision Analysis*, Vol.1, No.1, 1997, pp. 70-82
- [111] P. Papadimitratos, Z. J. Haas, E. G. Sirer, "Path Set Selection in Mobile Ad Hoc Networks", *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, 2002, pp. 1-11
- [112] M. Neugebauer, J. Ploennigs, K. Kabitzsch, "Evaluation of energy costs for single hop vs. multi hop with respect to topology parameters", *IEEE International Workshop on Factory Communication Systems*, 2006, pp. 175-182
- [113] M. Haenggi, "Twelve reasons not to route over many short hops", *IEEE 60th Vehicular Technology Conference (VTC'04)*, Vol.5, 2004, pp. 3130-3134
- [114] A. A. Somasundara, A. Ramamoorthy, and M. Srivastava, "Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines", *Proceedings of the 25th IEEE International Real-Time Systems Symposium, (RTSS'04)*, 2004, pp. 296-305
- [115] R. S. Sutton, A. G. Barto, "Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)", MIT Press, A Bradford Book, ISBN-10: 0262193981, March 1998
- [116] C. J. C. H. Watkins, "Learning from Delayed Rewards", PhD thesis, Cambridge University, Cambridge, England, 1989
- [117] G. Hardin, "The Tragedy of the Commons", *Science, New Series*, Vol. 162, No. 3859, 1968, pp. 1243-1248
- [118] R. Bellman, "A Markovian Decision Process", *Indiana University Mathematics Journal*, Vol. 6, No. 4, 1957, pp. 679-684
- [119] M. L. Puterman, "Markov Decision Processes Discrete Stochastic Dynamic Programming", John Wiley and Sons Publication, ISBN 0-471-72782-2, 2005

Published papers:

- [1] **C. Cirstea**, "Energy Efficient Routing Protocols for Wireless Sensor Networks: A Survey", SIITME2011, Timisoara, Romania, 2011. IEEE Indexed
- [2] **C. Cirstea**, M. Cernaianu, A. Gontean, "Packet Loss Analysis in Wireless Sensor Networks Routing Protocols", 35th International Conference on Telecommunications and Signal Processing (TSP 2012), Prague, Czech Republic, ISBN 978-1-4673-1116-8, 3-4 July 2012, pp. 37-41. ISI Indexed
- [3] **C. Cirstea**, M. Cernaianu, A. Gontean, "A Cluster Head Election Method with Adaptive Separation Distance and Load Distribution for Wireless Sensor Networks", The Fourth International Conference on Emerging Network Intelligence (EMERGING'12), ISBN: 978-1-61208-239-4, 23-28 September, Barcelona, Spain, 2012, pp. 37-42. ISI Conference
- [4] **C. Cirstea**, M. Cernaianu, A. Gontean, "An Inductive System for Measuring Microampere Currents", SIITME2012, Alba Iulia, Romania, 2012. IEEE Indexed
- [5] **C. Cirstea**, T. Petrita, V. Popescu, A. Gontean, "Performance Analysis and Modelling of a Radio Frequency Energy Harvester", Accepted for publication at the Journal of Advances in Electrical and Computing Engineering, ISI Indexed, IF 0.55
- [6] M. Cernaianu, **C. Cirstea**, A. Gontean, "Thermoelectrical Energy Harvesting System: Modelling, Simulation and Implementation", ISETC2012, Timisoara, Romania, 2012. IEEE Indexed
- [7] M. Cernaianu, **C. Cirstea**, D. Lascu, A. Gontean, "Battery Charger System Employing Average Current Control", ICPES 2012, Hong Kong. ISI Conference
- [8] M. Cernaianu, A. Cernaianu, **C. Cirstea**, A. Gontean, "Thermo Electrical Generator Improved Model", ICPES 2012, published in Lecture Notes in Information Technology, Vol. 13, Power and Energy Systems, Hong Kong, 2012. ISI Conference

Appendix A

LabView calibration and measurement program

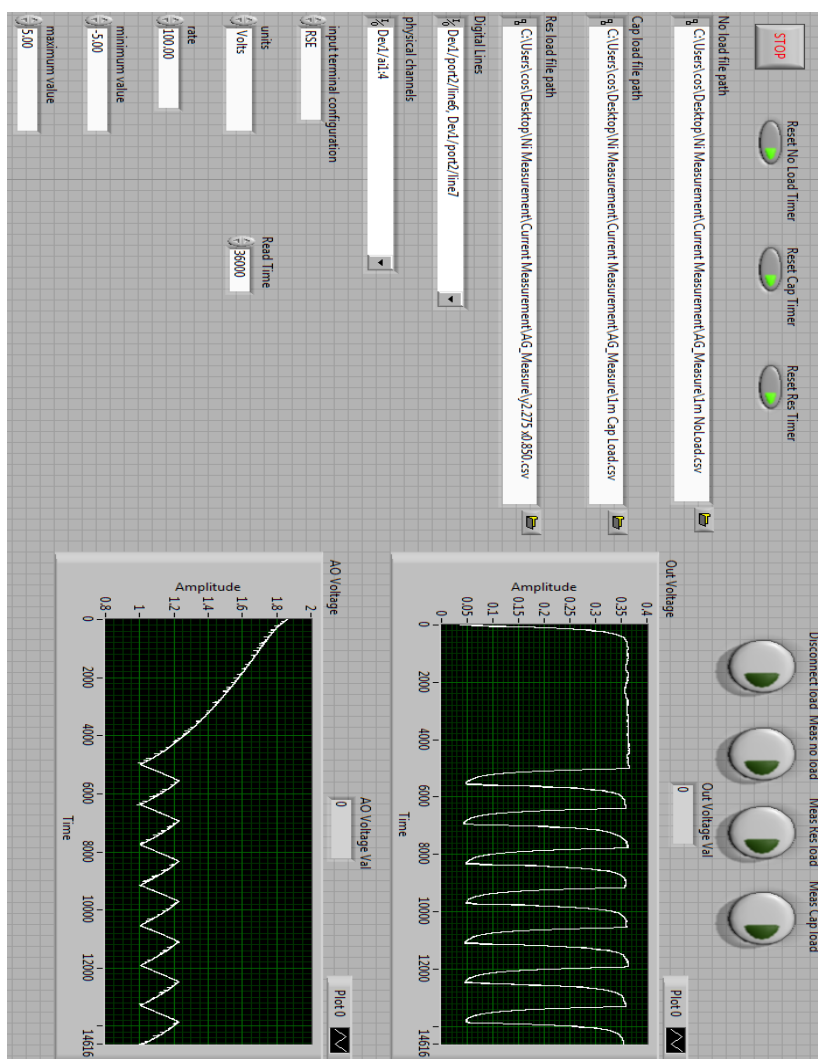


Figure A. 1. Front panel of the LabView calibration and measurement program

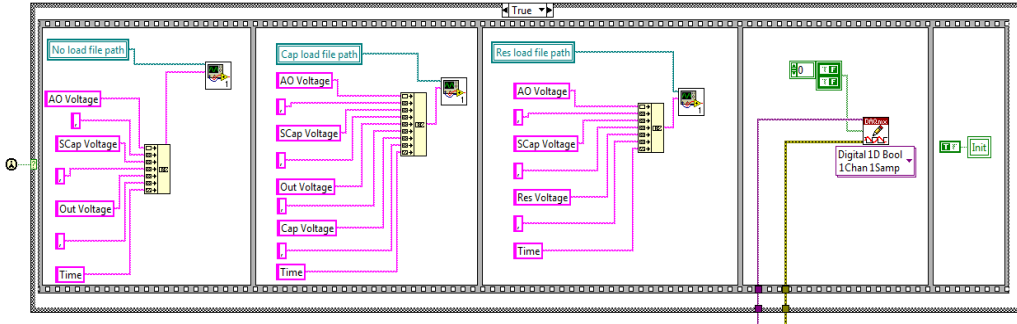


Figure A. 2. Simplified block diagram of the initialization procedure

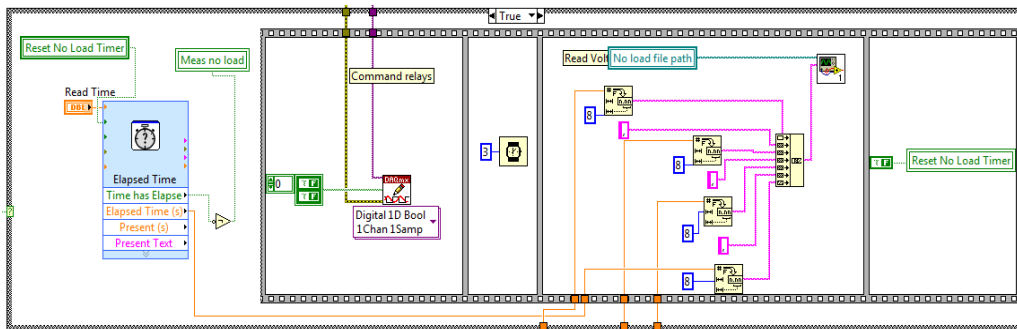


Figure A. 2. Simplified block diagram of the read procedure

Appendix B

Spherical coordinates system

As directional patterns of the antennas and Maxwells equations, which describe the interactions between the electric and magnetic fields, are better described using the spherical coordinates system (which we have used for our model presented in Chapter 2) we will briefly describe the transition between the cartesian and spherical coordinates systems. Figure B. 1 is a representation of the cartesian and spherical coordinates system.

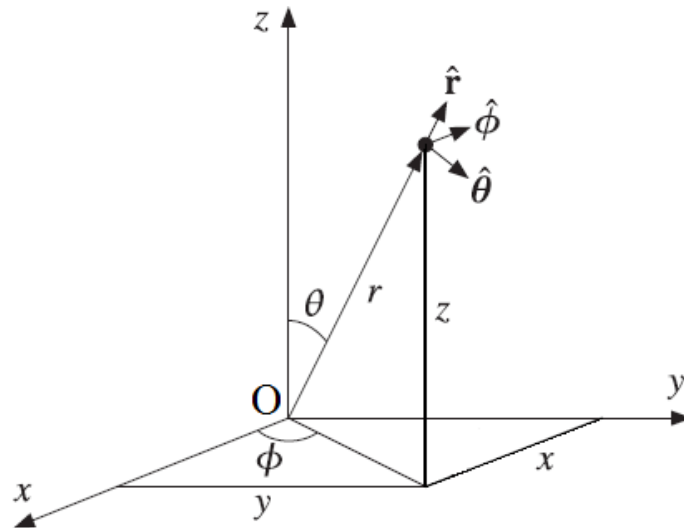


Figure B. 1. Cartesian and spherical coordinates system

The spherical coordinates are:

- The radius (or radial distance – r) which represents the Euclidian distance from the origin of the system to the point of interest

$$r \in [0, \infty)$$

- The inclination (or polar angle - θ) which is the distance between the Oz line and the line which unites the point of origin O with the point of interest

$$\theta \in [0, \pi)$$

- The azimuth (or azimuthal angle – ϕ) which represents the angle between Ox and the line which unites the point of origin O with the projection of the point of interest on the xOy plane

$$\phi \in [0, 2\pi)$$

In the case of Figure B. 1 the coordinate system is considered to be right handed meaning that once two of the coordinates have been chosen the third can be determined using the right hand rule. The coordinate system adopted in this paper is in conformity with the ISO-31-11 standard [72].

Appendix C

Gain approximation from HPBW angle

In antenna theory the HPBW angle is the separation in which the magnitude of the radiated power is decreased by half (- 3dB) from the peak of the main lobe. A representation is available in Figure C 1.

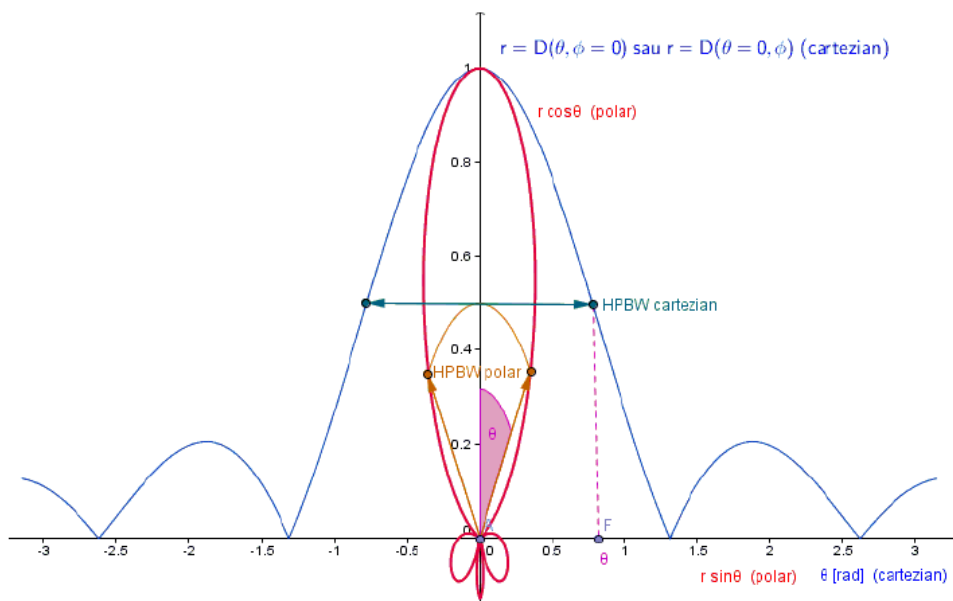


Figure C. 1. Antenna radiation pattern and HPBW (cartesian and polar) [64]

For dipole antennas with planar reflector (the case of the Powercast TX91501 transmitter [52]) the HPBW horizontal angle is between 60° and 90° .

The HPBW angle is very important because it can be used to estimate the antenna gain [64] which can be calculated using the following formula [64].

$$G = \frac{32383}{\text{HPBW}_{\text{vert}} \cdot \text{HPBW}_{\text{horiz}}} \quad (\text{C } 1)$$

where both HPBW angles are expressed in degrees.

By applying a logarithm to this formula we obtain formula 2.3:

$$g_0 = 45.1 - 10 \cdot \log_{10}(\text{HPBW}_{\text{horiz}} \cdot \text{HPBW}_{\text{vert}}) \quad (\text{C } 2)$$

When considering that the planar characteristics of the beam are independent we can approximate the directivity characteristic using [64]:

$$C(\theta, \varphi) = C_V(\theta) \cdot C_H(\varphi) \quad (\text{C } 3)$$

This means that the relationship between radiation and directivity can be written as follows:

$$F(\theta, \varphi) = C^2(\theta, \varphi) \quad (C 4)$$

Or more specifically:

$$F_V(\theta) = C^2(\theta) \quad (C 5)$$

$$F_H(\varphi) = C^2(\varphi) \quad (C 6)$$

In literature it is recommended [64] that the directional characteristics of antennas be approximated through pencil beam functions (the cosine function raised at a certain power). Therefore we can approximate the vertical and horizontal cuts as:

$$C_V(\theta) = \cos^p(\theta) \quad (C 7)$$

$$C_H(\varphi) = \cos^q\left(\frac{\varphi}{2}\right) \quad (C 8)$$

Since $\theta \in [0, \pi]$ and if the maximum radiation direction is in the xOy plane ($\theta = \frac{\pi}{2}$) then:

$$F_V(\theta) = \sin^{2p}(\theta) \quad (C 9)$$

Considering that $HPBW_{\text{vert}} = 60^\circ$ according to the manufacturer datasheet [52] we can determine the p coefficient from the HPBW condition as:

$$F_V\left(\frac{\pi}{2} \pm \frac{HPBW_{\text{vert}}}{2}\right) = \frac{1}{2} \Rightarrow \sin^{2p}\left(\frac{\pi}{2} - \frac{HPBW_{\text{vert}}}{2}\right) = \frac{1}{2} \quad (C 10)$$

By applying logarithm to the above equation we can determine the exponent p :

$$p = \frac{\log\left(\frac{1}{2}\right)}{2 \cdot \log\left(\sin\left(\frac{\pi}{2} - \frac{HPBW_{\text{vert}}}{2}\right)\right)} \cong 2.41 \quad (C 11)$$

The q coefficient can also be determined from the horizontal cut and considering the HPBW condition as follows:

$$F_H(\varphi) = \cos^{2q}\left(\frac{\varphi}{2}\right) = \frac{1}{2} \Rightarrow \cos^{2q}\left(\frac{HPBW_{\text{horiz}}}{4}\right) = \frac{1}{2} \quad (C 12)$$

By applying logarithm to the above equation the exponent q equals:

$$q = \frac{\log\left(\frac{1}{2}\right)}{2 \cdot \cos\left(\frac{HPBW_{\text{horiz}}}{4}\right)} \cong 10 \quad (C 13)$$

The same consideration from equation C 11 can be used to determine the 1.37 coefficient from 2.8 having in view that $HPBW = 78^\circ$.

Appendix D

The MEMSIC MoteView Graphical User Interface

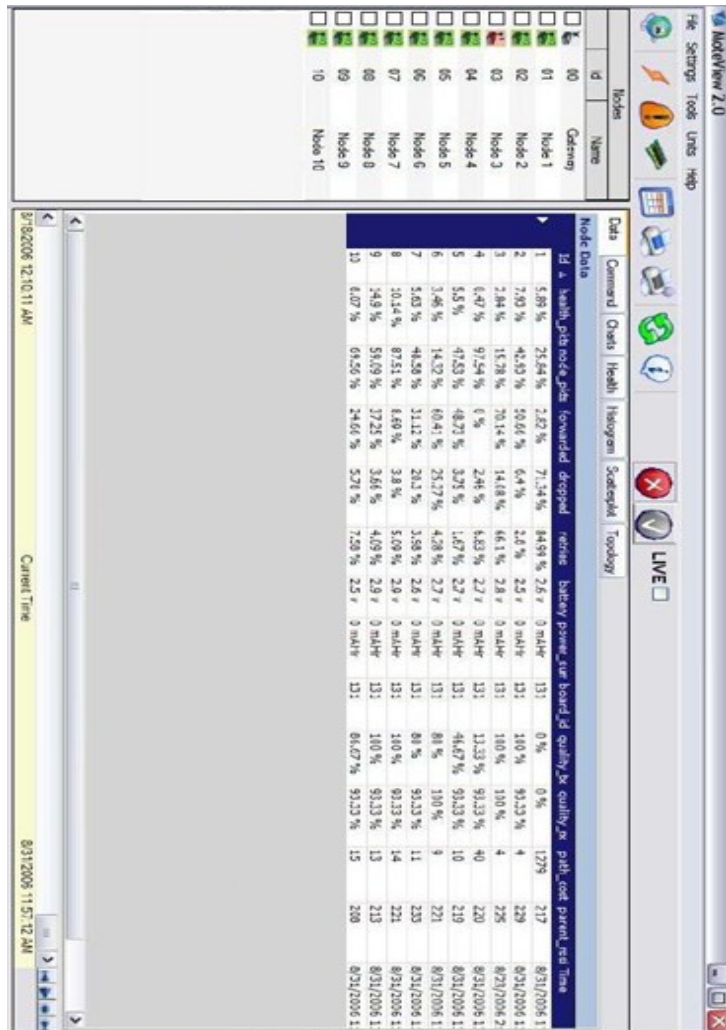


Figure D. 1. The MEMSIC MoteView program

Appendix E

Introduction to Markov decision processes

The sequential decision model which governs a Markov Decision Process was first described more than 50 years ago by Bellman [118]. The purpose was to lay the foundations of a mathematical theory that accounts for the relationship between the outcomes of present and future decisions in order to achieve good overall performance in time.

More exactly, given that an agent with the capability of taking decisions observes the states of a system under scrutiny at a specified point in time and chooses to take an action. The action has two outcomes:

- The agent receives a reward.
- The system moves to the next state according to a probability distribution determined by the action choice.

At the next point in time the agent may choose from a set of actions which one to perform. According to Puterman [119] the sequential decision problem can be modeled as in Figure E. 1.

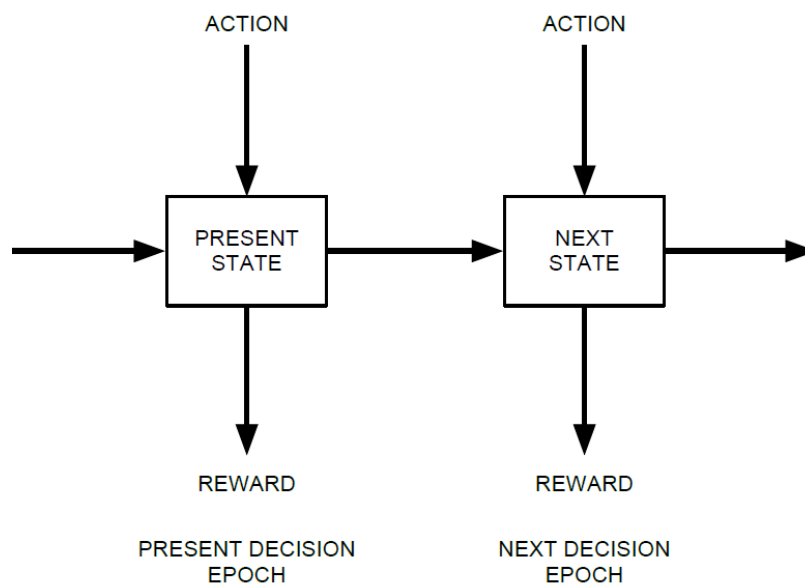


Figure E. 1. Representation of a sequential decision problem [119]

The problem can be split into five major elements [119]:

1. A set of decision epochs.
2. A set of system states.
3. A set of available actions.
4. The dependency of the reward on the state and the action.
5. The dependency of the transition probabilities on the state and the choice of action.

It is assumed that the elements above are known to the agent at the time of each decision. Considering the above assumptions the problem can be formulated as follows: for each decision epoch the agent chooses to perform one from a set of available ones in the current state occupied by the system at that time. For that action the agent receives a reward and the system moves on to the next state at the following decision epoch. The reward and transition probabilities depend on the state and the choice of action [119].

Model formulation

Decision epochs

The moments in time at which the decisions are made, are referred to as decision epochs ($\mathbf{T}=\{t_1, t_2 \dots t_N\}$). Depending on \mathbf{N} , the decision problem can be called a finite horizon ($\mathbf{N} \leq \infty$) problem or infinite horizon problem ($\mathbf{N} \rightarrow \infty$) [119].

States and Action Sets

At each decision epoch, the system is in a state \mathbf{s} ($\mathbf{s} \in \mathbf{S}$). At each state $\mathbf{s} \in \mathbf{S}$ the agent may choose action \mathbf{a} from the set of \mathbf{A}_s available actions. Sets \mathbf{S} and \mathbf{A}_s do not vary with \mathbf{t} and may each be either [119]:

1. Arbitrary finite sets
2. Arbitrary infinite sets
3. Compact subsets of finite dimensional Euclidian space, or
4. Non-empty Borel subsets of complete, separable metric spaces.

Rewards and transition probabilities

When action \mathbf{a} is chosen in state \mathbf{s} , at the \mathbf{t} decision epoch [119]:

1. The agent receives reward $r(\mathbf{s}, \mathbf{a})$.
2. The system state at time $\mathbf{t}+1$ is determined by the $p(\cdot | \mathbf{s}, \mathbf{a})$ probability distribution.

The reward can be considered income or cost if its value is positive or negative. If the reward depends on the state of the system at the next decision epoch, it will be expressed as $r(\mathbf{s}, \mathbf{a}, \mathbf{j})$, function that denotes the value when the system occupies the state \mathbf{j} at time $\mathbf{t}+1$. Its expected value at time \mathbf{t} may be evaluated by computing:

$$r(\mathbf{s}, \mathbf{a}) = \sum_{\mathbf{j} \in \mathbf{S}} r(\mathbf{s}, \mathbf{a}, \mathbf{j}) \cdot p(\mathbf{j} | \mathbf{s}, \mathbf{a}) \quad (\text{D } 1)$$

Where the non-negative function $p(j|s, a)$ represents the probability that the system will be in state $j \in \mathbf{S}$ at time $t+1$, when the agent chooses action a in state s at time t (it is also called the transition probability function).

It is important to note that many system transitions might happen between t and $t+1$. Usually, all the information necessary to make a decision at time t is summarized in $r(s, a)$ and $p(j|s, a)$, however, under some criteria, $r(s, a, j)$ must be used. It is usually assumed that:

$$\sum_{j \in \mathbf{S}} p(j|s, a) = 1 \quad (\text{D } 2)$$

In finite horizon Markov decision processes, no decision is taken at time t and the reward at this point is only a function of the state. This can be denoted by $r(s)$ and is sometimes referred to as a salvage value.

The collection of objects $\{\mathbf{T}, \mathbf{S}, \mathbf{A}_s, p(\cdot|s, a), r(s, a)\}$ represents a Markov decision process. The "Markov" qualifier is used because the transition probability and reward function depend on the past only through the current state of the system and the action taken by the agent in that state [119].